

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Pen-Based Interfaces for Intelligent Statics Tutoring Systems

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Computer Science

by

Levi Scott Lindsey

August 2013

Thesis Committee:

Dr. Thomas Stahovich, Chairperson
Dr. Frank Vahid
Dr. Victor Zordan

Copyright by
Levi Scott Lindsey
2013

The Thesis of Levi Scott Lindsey is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

First and foremost, I thank my adviser, Professor Stahovich, for invaluable guidance, advice, and inspiration. I thank all of my labmates—Reyna, Justin, Hank, Nick, Jim, Kevin, Matt, and Johann—for sharing both long hours in the lab and long lunch breaks out of it. I especially thank Reyna and Justin for providing an engineering perspective when we were envisioning pedagogically sound solutions to many of the system’s problems. I also thank Reyna and Justin for helping to break these systems and chase down bugs. I thank Hank for his contributions with the problem editor program, and for putting together the log-file submission system. I thank Reyna, Justin, and Hank for all of their help designing and encoding new problems. I thank Nick for his contributions with parsing text for the symbolic and expanded equation-entry stages. I also thank Nick for driving me and my bike home on so many days when I was too lazy to bike back uphill.

To my beautiful fiancée, Jacqueline Schwartzstein. As rewarding as remaining at UCR to finish a PhD would be, leaving to be with her will be so much more so.

ABSTRACT OF THE THESIS

Pen-Based Interfaces for Intelligent Statics Tutoring Systems

by

Levi Scott Lindsey

Master of Science, Graduate Program in Computer Science
University of California, Riverside, August 2013
Dr. Thomas Stahovich, Chairperson

Here we present two intelligent tutoring systems for statics, the sub-discipline of engineering mechanics concerned with the analysis of mechanical systems in equilibrium under the action of forces. These systems are pen-based: one runs on Windows tablet PCs and the other on LivescribeTM smartpens with specially designed paper worksheets. It is common for novice students to attempt to solve problems without understanding the fundamental concepts involved. For example, they may attempt to solve a new problem by adapting the solution to an example problem. This approach can lead to errors as novices often categorize problems on the basis of surface similarity rather than the structural—i.e., conceptual—similarity. Our new instructional model guides students in explicitly examining the structural elements that govern the solution. For example, before the student draws forces on a free-body diagram, the system requires the student to explicitly identify all *interaction points*, points at which other objects apply forces to the body. The student must then identify what kind of interaction occurs at each *interaction point* before representing them by force arrows. The system critiques the student’s work for each of these steps and

provides appropriate tutorial feedback. This instructional design has a number of benefits. It helps students to identify the structural elements that guide the solution process, which is important for problem-solving transfer. It also enables the system to accurately diagnose student errors. Because each step in the reasoning is explicitly recorded, the system can unambiguously determine the cause of an error and provide focused tutorial feedback. Also, the use of natural pen-based interfaces unburdens the student from extraneous cognitive load inherent in more traditional interfaces. We conducted two studies to evaluate these systems. The first included 43 students enrolled in Statics (ME 10) at UCR, while the second included 10 students enrolled in Introduction to Mechanical Engineering (ME 2). The results suggest that students find the systems to be useful for learning statics. However, the tablet-based system is more effective than the smartpen-based one, with the former leading to large and statistically significant learning gains in the second study.

Contents

List of Figures	xii
List of Tables	xv
1 Introduction	1
2 Literature Review	10
2.1 Automated Teaching and Intelligent Tutoring Systems	10
2.2 Pen-Based Computing Systems	13
2.3 Bridging the Gap: Pen-Based Tutoring Systems	14
3 Newton's Tablet: The Windows Tablet-PC System	16
3.1 The Problem-Solving Decomposition	17
3.2 The User Interface	27
3.2.1 The Graphical User Interface Layout	27
3.2.2 Student Input	30
3.2.3 Stroke Recognition	31
3.2.4 Beautification	41
3.3 Tutorial Feedback	43
4 Newton's Pen: The System for Livescribe Smartpens	47
4.1 The User Interface	49
4.1.1 The Worksheets	50
4.1.2 Providing Output	53
4.2 The Problem-Solving Decomposition	55
4.3 Challenges of Developing for the Livescribe Platform	60
4.3.1 Communicating Locations	60
4.3.2 Multi-Body Problems	62
4.3.3 Erroneous and Messy Ink	62
4.3.4 Programming Limitations	65
4.4 Tutorial Feedback	66

5 User Studies	68
5.1 Use in an Introduction to Statics Course	69
5.1.1 Experimental Setup	69
5.1.2 Student Performance	69
5.1.3 Attitudinal Survey	70
5.2 Supplementary User Study	77
5.2.1 Experimental Setup	77
5.2.2 Student Performance	80
5.2.3 Attitudinal Survey	83
6 Conclusion	87
A Newton's Tablet Stages and Modes	102
A.1 Newton's Tablet Stages	102
A.1.1 Pre-Problem Stage	105
A.1.2 Boundary Trace Stage	105
A.1.3 Point of Interaction Location Stage	107
A.1.4 Point of Interaction Classification Stage	108
A.1.5 Two-Force Member Identification Stage	110
A.1.6 Newton's-Third-Law Pair Identification Stage	111
A.1.7 Force Drawing Stage	112
A.1.8 Body Selection Stage	115
A.1.9 Equation Type Entry Stage	115
A.1.10 Positive Moment Direction Drawing Stage	117
A.1.11 Force Identification 1 Stage	118
A.1.12 Component Break Down Stage	119
A.1.13 Force Identification 2 Stage	120
A.1.14 Moment Arm Drawing Stage	121
A.1.15 Symbolic Term Entry Stage	122
A.1.16 Expanded Term Entry Stage	125
A.1.17 Equation Complete Stage	126
A.2 Newton's Tablet Modes	127
A.2.1 Drag Mode	127
A.2.2 Erase Mode	127
A.2.3 General Help Mode	128
A.2.4 Query Mode	130
A.2.5 Scratch Work Mode	131
A.2.6 Error Mode	131
B Newton's Pen Stages and Modes	133
B.1 Newton's Pen Stages	133
B.1.1 Boundary Identification Stage	138
B.1.2 Point of Interaction Identification Stage	140
B.1.3 Two-Force Member Identification Stage	142

B.1.4	Newton's-Third-Law Pair Identification Stage	144
B.1.5	Force Identification Stage	146
B.1.6	Equation Stages	148
B.2	Newton's Pen Modes	148
B.2.1	Erase Mode	148
B.2.2	Recent Messages Mode	149
B.2.3	Text Edit Mode	149
B.2.4	Error Mode	150
C	The Conceptual Errors	151
C.1	Boundary Trace Stage Errors	151
C.2	Point of Interaction Location Stage Errors	153
C.3	Point of Interaction Classification Stage Errors	154
C.4	Two-Force Member Identification Stage Errors	155
C.5	Newton's-Third-Law Pair Identification Stage Errors	155
C.6	Force Drawing Stage Errors	155
C.7	Body Selection Stage Errors	159
C.8	Equation Type Entry Stage Errors	159
C.9	Moment Direction Drawing Stage Errors	160
C.10	Force Identification 1 Stage Errors	160
C.11	Component Break Down Stage Errors	161
C.12	Force Identification 2 Stage Errors	163
C.13	Moment Arm Drawing Stage Errors	164
C.14	Symbolic Term Entry Stage Errors	165
C.15	Expanded Term Entry Stage Errors	170
D	The Problems	171
D.1	The Problem Encodings	171
D.1.1	The Newton's Tablet Problem Encodings	173
D.1.2	The Newton's Pen Problem Encodings	174
D.2	The Problem Set	174
D.2.1	Problem 0: The L-Beam	175
D.2.2	Problem 1: The Bolted Bar	176
D.2.3	Problem 2: The Load Binder	177
D.2.4	Problem 3: The Truck	178
D.2.5	Problem 4: The Cart	179
D.2.6	Problem 5: The Crane	180
D.2.7	Problem 6: The Un-Bolted Bar	181
D.2.8	Problem 7: The Toggle Clamp	182
D.2.9	Problem 8: The Wheel	183
D.2.10	Problem 9: The Can Crusher	184
E	Demonstrational Videos	185

F The Newton's Pen Documentation	186
F.1 User Manual	186
F.2 Quick Reference Guide	206

List of Figures

1.1 A Traditional Statics Problem Solution	3
1.2 The Livescribe Pen-and-Paper Interface	7
1.3 Pen Worksheets	8
3.1 Tracing Boundaries	18
3.2 Classifying Points of Interaction	19
3.3 Drawing Forces	21
3.4 Entering the Equation Type	21
3.5 Identifying Forces	23
3.6 Drawing Moment Arms	23
3.7 The Symbolic vs. Expanded Forms of an Equation	25
3.8 Entering the Expanded Equation	25
3.9 Newton's Tablet Graphical User Interface	28
3.10 A Correct Boundary Trace	32
3.11 Area-Based Trace Recognition	34
3.12 The Importance of Resampling for Matching Points	35
3.13 Searching Based on the Previous Match	37
3.14 Asymmetric Trace-Boundary Matches	39
3.15 Multi-Stroke Trace Merging	39
3.16 Beautification	42
3.17 An Error Help Message	44
4.1 The Livescribe Echo™ Smartpen	48
4.2 A Problem-Description Worksheet	51
4.3 A Free-Body Diagram Worksheet	52
4.4 The Pen Notification Icons	54
4.5 Identifying Boundaries	56
4.6 Identifying Points of Interaction	57
4.7 Identifying Forces	59
4.8 Discrepancy in Beliefs of Stroke Representations	63
5.1 The Pre- and Posttest Problem Description	79

5.2 User Study Pre- and Posttest Results (All Students)	81
5.3 User Study Pre- and Posttest Results (Pre-Processed)	82
A.1 State Diagram of the Free-Body Diagram Stages in Newton's Tablet	104
A.2 Boundary Trace Stage Screenshot	106
A.3 Point of Interaction Location Stage Screenshot	108
A.4 Point of Interaction Classification Stage Screenshot	109
A.5 Two-Force Member Identification Stage Screenshot	110
A.6 Newton's-Third-Law Pair Identification Stage Screenshot	112
A.7 Force Drawing Stage Screenshot	113
A.8 State Diagram of the Equilibrium-Equation Stages in Newton's Tablet	114
A.9 Equation Type Entry Stage Screenshot	116
A.10 Positive Moment Direction Drawing Stage Screenshot	118
A.11 Force Identification 1 Stage Screenshot	119
A.12 Component Break Down Stage Screenshot	120
A.13 Force Identification 2 Stage Screenshot	121
A.14 Moment Arm Drawing Stage Screenshot	122
A.15 Symbolic Term Entry Stage Screenshot	123
A.16 Expanded Term Entry Stage Screenshot	126
A.17 A General Help Mode Screenshot	129
A.18 A Query Box	130
A.19 An Error Mode Screenshot	132
B.1 Flowchart of the Boundary Trace Sub-Stage Logic	135
B.2 Flowchart of the Boundary Label Sub-Stage Logic	136
B.3 State Diagram of the Free-Body Diagram Stages in Newton's Pen	137
B.4 Boundary Identification Stage Example Work	139
B.5 Point of Interaction Identification Stage Example Work	141
B.6 Two-Force Member Identification Stage Example Work	143
B.7 Newton's-Third-Law Pair Identification Stage Example Work	145
B.8 Force Identification Stage Example Work	147
D.1 The Problem Editor	172
D.2 Problem 0: The L-Beam	175
D.3 Problem 1: The Bolted Bar	176
D.4 Problem 2: The Load Binder	177
D.5 Problem 3: The Truck	178
D.6 Problem 4: The Cart	179
D.7 Problem 5: The Crane	180
D.8 Problem 6: The Un-Bolted Bar	181
D.9 Problem 7: The Toggle Clamp	182
D.10 Problem 8: The Wheel	183
D.11 Problem 9: The Can Crusher	184
F.1 Newton's Pen User Manual: Page 1	187

F.2	Newton's Pen User Manual: Page 2	188
F.3	Newton's Pen User Manual: Page 3	189
F.4	Newton's Pen User Manual: Page 4	190
F.5	Newton's Pen User Manual: Page 5	191
F.6	Newton's Pen User Manual: Page 6	192
F.7	Newton's Pen User Manual: Page 7	193
F.8	Newton's Pen User Manual: Page 8	194
F.9	Newton's Pen User Manual: Page 9	195
F.10	Newton's Pen User Manual: Page 10	196
F.11	Newton's Pen User Manual: Page 11	197
F.12	Newton's Pen User Manual: Page 12	198
F.13	Newton's Pen User Manual: Page 13	199
F.14	Newton's Pen User Manual: Page 14	200
F.15	Newton's Pen User Manual: Page 15	201
F.16	Newton's Pen User Manual: Page 16	202
F.17	Newton's Pen User Manual: Page 17	203
F.18	Newton's Pen User Manual: Page 18	204
F.19	Newton's Pen User Manual: Page 19	205
F.20	Newton's Pen Quick Reference Guide	207

List of Tables

3.1	Menu Buttons for Newton's Tablet	29
4.1	Newton's Pen Paper Buttons	53
5.1	Newton's Pen Attitudinal Survey Results from ME 10 (Part 1)	71
5.2	Newton's Pen Attitudinal Survey Results from ME 10 (Part 2)	72
5.3	Newton's Pen Attitudinal Survey Results from ME 10 (Part 3)	73
5.4	Newton's Tablet Attitudinal Survey Results from ME 10 (Part 1)	74
5.5	Newton's Tablet Attitudinal Survey Results from ME 10 (Part 2)	75
5.6	Newton's Tablet Attitudinal Survey Results from ME 10 (Part 3)	76
5.7	User Study Pre- and Posttest Free-Body Diagram Scoring Rubric	79
5.8	User Study Pre- and Posttest Equation Scoring Rubric	80
5.9	User Study Pre- and Posttest Results (All Students)	81
5.10	User Study Pre- and Posttest Results (Pre-Processed)	81
5.11	User Study Pre- and Posttest Times	82
5.12	Newton's Tablet Attitudinal Survey Results from User Study (Part 1) . . .	84
5.13	Newton's Tablet Attitudinal Survey Results from User Study (Part 2) . . .	85
5.14	Newton's Tablet Attitudinal Survey Results from User Study (Part 3) . . .	86

Chapter 1

Introduction

Significant strides have been made in the field of intelligent tutoring systems as personal computers have become ubiquitous. These systems aim to facilitate the learning process for students. In some cases, the potential effectiveness of tutoring software has even been shown to rival that of human tutors [46, 67]. However, almost universally, intelligent tutorial systems are based upon WIMP (Windows, Icons, Menus, and Pointer) or keyboard interfaces. Such interfaces are far less intuitive than the traditional pen-and-paper interface used for problem solving in most domains, which can lead to extraneous cognitive load while students are struggling to grasp whatever concept is being taught [52]. Fortunately, pen-based computing technology has grown substantially in order to address the general need for natural user interfaces. The focus of this research has been to develop intelligent tutoring systems—built upon natural, pen-based interfaces—for the engineering sub-discipline of statics. We present two such systems here: Newton’s Tablet runs on Windows tablet PCs, and Newton’s Pen runs on Livescribe smartpens. Data sets have been collected from 43

students using these systems in an introductory statics course at the University of California, Riverside in the winter of 2013 and from 10 students using these systems in a supplementary user study.

These systems first guide the student through the process of selecting system boundaries and constructing free-body diagrams. Then they guide the student through the process of deriving equilibrium equations from these free-body diagrams. In order to better scaffold student learning, these systems break down the problem-solving processes into fine-grained sub-steps. This decomposition serves two goals: (1) it trains the student to explicitly evaluate each of the fundamental concepts involved in the problem-solving process; and (2) it enables these systems to better diagnose student errors and provide focused tutorial feedback.

Figure 1.1 shows a typical statics problem and the correct solution. In a traditional homework problem, written instructions prompt the student to solve for certain unknown forces, or other variables, given in the problem description. To solve the problem, the student must construct free-body diagrams and equilibrium equations. Doing so requires the student to analyze a number of attributes of the problem. For example, to construct the free-body diagrams, the student must identify the boundaries which need to be analyzed, identify the locations and types of all interactions these boundaries have with external forces, consider whether this system contains any two-force members or Newton's-third-law pairs, and then draw the actual arrows which model the external forces acting on the body. To derive equilibrium equations, the student must identify the relevant force components, and for moment equations, also identify the moment arms. In addition to requiring the

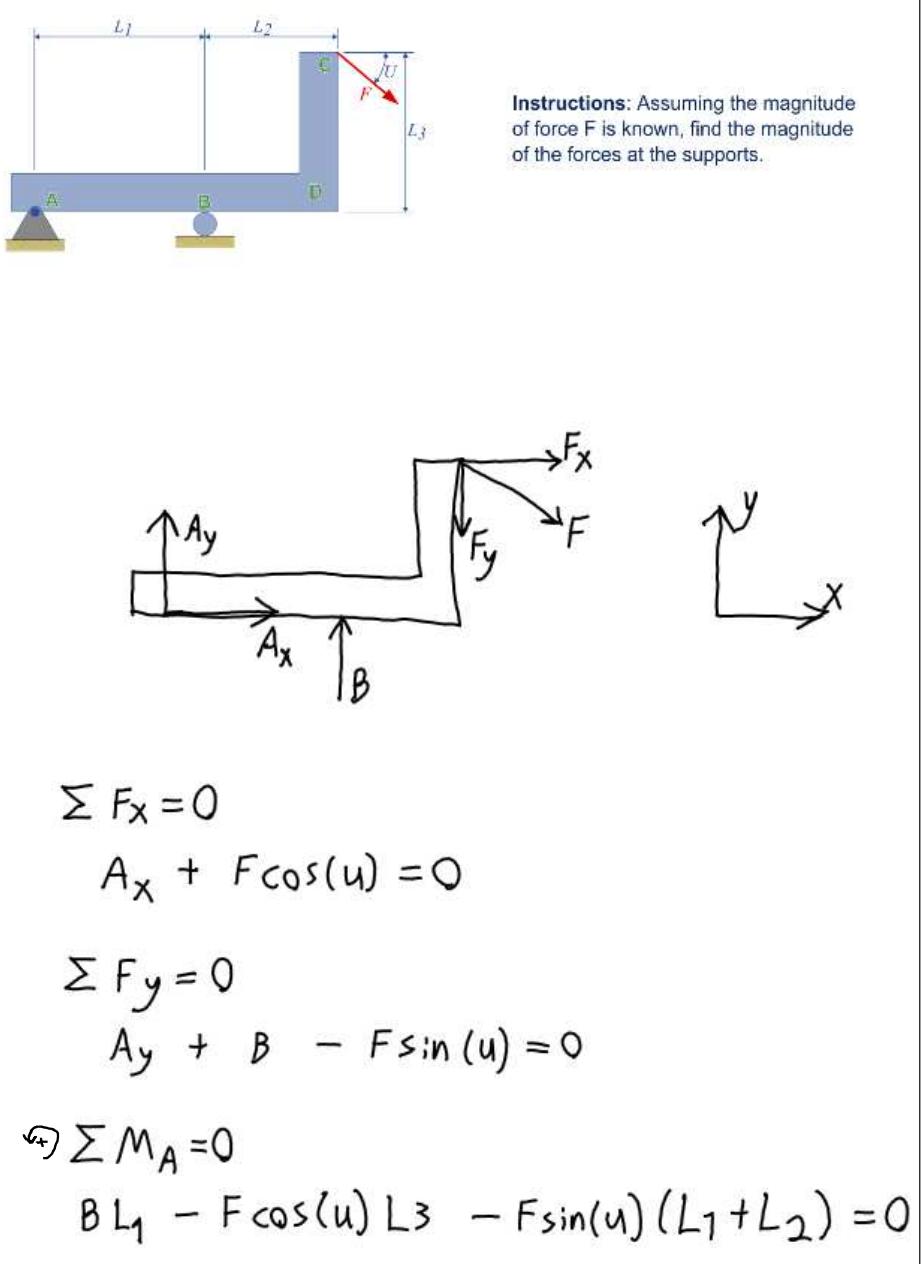


Figure 1.1: A Typical Solution to a Statics Problem . The problem statement, including an image and textual description, is given at the top of the page. A typical solution is written below this. The solution includes a free-body diagram and equilibrium equations.

application of statics principles, completing these tasks may also require geometric and algebraic calculations.

It is common for novice students to attempt to generate the various elements of a solution without identifying and understanding the fundamental concepts involved. For example, students often attempt to solve a new problem by adapting the solution from an example problem. This approach can lead to errors for novice students as they often categorize problems based on surface similarity rather than the structural—that is, conceptual—similarity. As a remedy, our tutoring systems employ a novel instructional model that guides students in examining the structural elements that govern the solution to a problem. With our systems, the student must make all of the reasoning steps in the solution process explicit, including those steps that are typically not recorded in a solution. In this sense, our instructional model is similar in spirit to elaborative interrogation [20]—an instructional technique in which students are asked to explain their own work.

To create a free-body diagram with our systems, the student begins by explicitly tracing the system boundary. This is intended to help the student differentiate between external forces that act on the system, and internal forces, which should not appear in the free-body diagram. The student then identifies all “interaction points”—points on the free body at which other objects apply forces. The student then identifies what kind of interaction occurs at each interaction point. For examples, the interactions can be from tension forces, weight forces, applied forces, and so on. Finally, the student must represent each of the interactions by force arrows. Our tutoring systems critique the student’s work for each of these steps and provides appropriate tutorial feedback.

This instructional design has a number of intended benefits. First, it helps students to identify the structural elements of the problem that guide the solution process. For example, it helps the student to understand that the types of interactions determine the forces on the free-body diagram. Research suggests that the ability to identify the structural elements of a problem is important for problem-solving transfer [45]. Second, this instructional model enables the system to accurately diagnose student errors. Because our systems explicitly record each step in the reasoning process, they can unambiguously determine the cause of an error. For example, if a student—working with pencil and paper—were to model a roller joint with two orthogonal force components, the student may have failed to identify the interaction as a roller or may have been confused about how to model a roller. However, because our systems separate these issues, they are able to provide precise and efficient feedback.

Our systems also employ a novel approach for guiding the student through the process of constructing equilibrium equations. Our systems help the student to focus on the conceptual reasoning by separating it from the mathematical analysis. The student begins by graphically identifying the force components and moment arms relevant to each equilibrium equation. The student assigns a symbolic label to the force components and moment arms. These labels are defined directly on the free-body diagram. Later, the student will compute their values from data given in the problem description. The student uses these labels to construct a *symbolic equation*. For example, a force component may be labeled “ F_x ” and a moment arm may be labeled as “ L ”. The system critiques the symbolic equation, and provides feedback. Once the student has derived the correct symbolic

equation, the student replaces the labels with mathematical expressions computed from the given data. In the current example, “Fx” may be computed using a trigonometric function, and “L” may be computed in terms of several dimensions and angles given in the problem.

Again, this approach has several intended advantages. It initially enables the student to focus on the statics concepts without the extra complications of geometric and algebraic calculations. Once the student has correctly applied the appropriate statics principles, the student can then focus on the calculations. We believe that this approach enhances problem-solving transfer. It also enables the system to accurately identify a student’s misconceptions and differentiate conceptual errors from math and geometry errors.

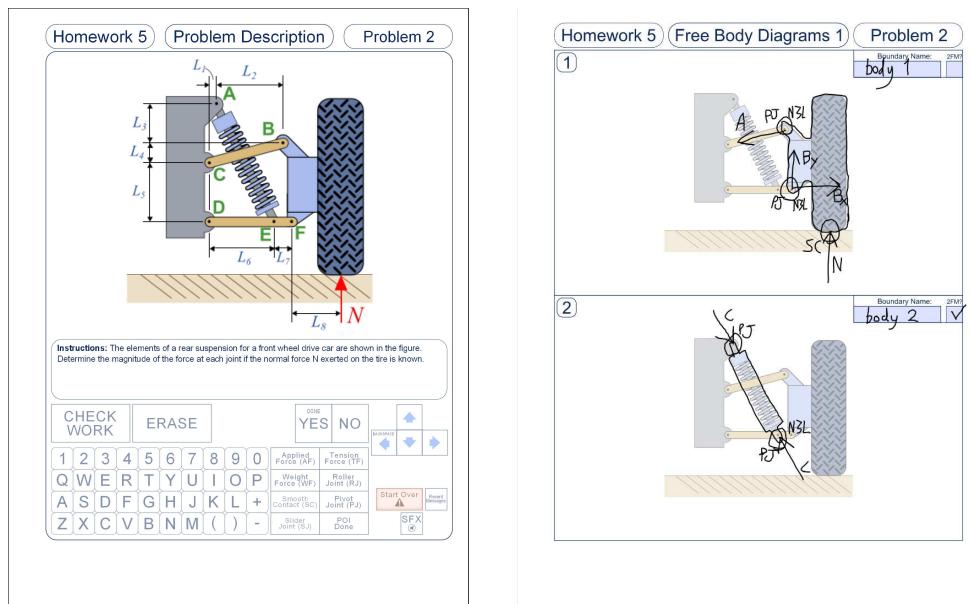
We developed Newton’s Tablet before Newton’s Pen, as development on the PC platform is easier than on the smartpen platform. Once we had implemented and evaluated our instructional model and algorithms on the PC, we adapted the system to run on the smartpen. This was a challenging task because of the limited computational power and novel interface capabilities of the smartpen.

When using Newton’s Tablet, the student draws on the tablet display with a stylus, mouse, or finger. Gestures are used for selecting and drawing objects. Text entry for labels and equations is performed using either a virtual or physical keyboard. This combination of gesture recognition and keyboard entry of text resulted in fluid interaction, which enables the student to focus on problem solving rather than the user interface. When critiquing the student’s work, the system highlights errors on the display and provides tutorial feedback in the form of text and graphics.

The smartpens work with special dot-patterned paper, and digitize the ink as



Figure 1.2: The Livescribe Pen-and-Paper Interface. The Livescribe smartpen has a camera at its tip that reads the dot pattern on the paper in order to create a digital copy of the ink. The smartpen also has an integrated speaker and a small display that can render scrolling text and images.



(a) Problem-Description Worksheet

(b) Free-Body Diagram Worksheet

Figure 1.3: Pen Worksheets.

time-stamped x/y-coordinates. Figure 1.2 shows the Livescribe pen-and-paper interface. Much of the user interface for Newton’s Pen is provided on specially designed worksheets. For example, Figure 1.3 shows the worksheets for a typical problem. As the student writes on the worksheets, the system interprets the digitized ink and evaluates the correctness of the work. The system provides several types of information to the student. These include: (1) directions for completing the current problem-solving stage; (2) messages informing the student about how each pen-stroke event was interpreted by the system; and (3) detailed messages describing problem-solving errors, which are presented when the student asks the system to evaluate the correctness of the work for the current stage. Newton’s Pen presents this output as brief text messages and images rendered on the display on the barrel of the pen, and short audio clips played from an integrated speaker. Designing this tutorial system to operate with the visually static ink-on-paper interface and with limited computing resources presented numerous challenges. For example, we had to minimize the memory footprint of the system because the smartpen has very limited memory.

Chapter 2 discusses related work in the fields of intelligent tutoring systems and pen-based computing interfaces. This is followed by in-depth descriptions of the two statics tutorial systems. Newton’s Tablet is described in Chapter 3 and Newton’s Pen is described in Chapter 4, with a focus on the differences from the tablet-PC version. Chapter 5 presents results from studies evaluating the educational effectiveness and usability of the systems. Finally, in Chapter 6, conclusions are presented.

Chapter 2

Literature Review

This chapter begins with a brief presentation of automated teaching techniques and intelligent tutoring systems (Section 2.1), then discusses research in the field of pen-based computing systems (Section 2.2), and finishes with a description of more-recent research on tutoring systems which use pen-based interfaces (Section 2.3).

2.1 Automated Teaching and Intelligent Tutoring Systems

There are many different types of systems for automated teaching—that is, teaching without any interaction from a human instructor. The most primitive of these systems simply convey information without presenting convenient means of monitored concept practice or assessment. Examples of this type of system include books, videos, and web sites. A slightly more effective version of these most basic systems might provide suggested practice problems or worked-out practice problems [61]. The fundamental problem with all of these systems is that they do not interact with the student.

An effective tutoring system must both describe a concept to the student and offer feedback about the student’s understanding of it. Many systems assess understanding through a series of questions. When the student provides an answer, the systems tell the student whether the answer is correct. These are called Computer Assisted Instructional (CAI) systems [60, 64]. The more useful instances of these systems might even explain where an error is and specifically why it is wrong. With the rise of the Internet and personal computers, automatic tutoring systems with varying degrees of sophistication have become pervasive. However, even systems such as these pale in comparison to a human tutor. The main flaw with these tutoring systems is that they are assessment oriented. That is, they start by telling the student what to do, and then afterward tell the student how well he/she did. They offer no guidance through the problem-solving process. In systems such as these, if the student does not know how to approach the given problem, then he/she has no choice but to interrupt the problem-solving process and search through other instructional material for the necessary guidance.

Wenger [69] suggests that “Learning is viewed as successive transitions between knowledge states, the purpose of teaching is accordingly to facilitate the student’s traversal of the space of knowledge states.” In essence, this is what makes a human tutor so effective. A human tutor can adapt to the student’s current solution path through a problem, and can dynamically offer suggestions on how to tackle each step. Furthermore, an effective human tutor is able to decompose the problem-solving process into a logical sequence of steps for the purpose of explicitly identifying the basic concepts involved in solving the problem. For an automated tutoring system to rival the abilities of a human tutor, it

must be able to effectively break down a problem into these more manageable sub-steps. Systems that adaptively guide a student through the sub-steps of a problem are known as Intelligent Tutoring Systems (ITSs). Decomposing a complex problem-solving task into logical sub-steps is consistent with the principles of Cognitive Load Theory (CLT) [53, 62], which suggests that all current cognitive tasks must be processed before meaningful learning can continue. Stated another way, the more one attempts to process in a short amount of time, the more difficult it will be to process.

Numerous ITSs have been developed for a wide variety of domains in the last 20 years [21, 57, 59, 58, 13, 14, 9, 63, 29, 50]. Previous research has also explored teaching tools specifically designed for constructing free-body diagrams and deriving their corresponding equilibrium equations. Roselli et al. [54] and Rosser and Valle [55] both developed web-based systems for statics, but they were not pen-based and provided little scaffolding. VanLehn et al. [68] developed a physics tutoring system which was intelligent, but it had a traditional user interface rather than a pen-based one. Also, the granularity of the sub-steps was fairly large, limiting the ability to provide focused tutorial feedback.

The vast majority of ITSs have been designed around WIMP or keyboard interfaces. Depending on the domain, such interfaces may deviate substantially from students' ordinary working environment, and therefore create an additional obstacle for the learning process. This work contributes to the body of research on ITSs by designing an intuitive natural user interface, which both emulates the conditions of traditional problem-solving and finely scaffolds student learning.

2.2 Pen-Based Computing Systems

Most educational computing systems rely on WIMP or keyboard based interfaces, and most of these systems are intended to assist with tasks which are traditionally performed with a pen and paper. Cognitive load theory dictates that users will perform better with systems which are less distracting to use, and a good deal of research supports this [52, 7, 49]. In particular, research by Oviatt et al. [52] showed that “as interfaces departed more from familiar work practice . . . , students would experience greater cognitive load such that performance would deteriorate in speed, attentional focus, meta-cognitive control, correctness of problem solutions, and memory.” This implies that an ITS which employs a natural user interface will produce increased learning gains compared to an ITS which does not.

Fortunately, a large volume of research has been conducted on sketch recognition and pen-based interfaces. Much of this research has been based upon developing tools for entering and evaluating mathematical equations [44, 37, 72, 56]. Developing tools for interpreting three-dimensional systems from two-dimensional sketches has also been a popular area of research [73, 30, 43]. Narayanaswamy [51] and Gennari et al. [25] have both developed pen-based tools for interpreting hand-drawn electric circuits. Alvarado and Davis [4] developed a tool for simulating simple, hand-drawn mechanical devices. Kara et al. [34] developed a tool for analyzing vibratory mechanical system sketches. Landay and Myers [35, 36] developed a tool for sketching graphical user interfaces—that is, layouts for standard WIMP interfaces. Hammond and Davis [28] developed a tool for drawing and evaluating Unified Modeling Language diagrams. Forbus et al. [23] developed a pen-based

tool for understanding military tactics. Kara and Stahovich [33] developed a tool for control system sketch analysis. Anderson et al. [6] developed a pen-based lecture tool which enables increased student involvement and facilitates the presentation of lecture material. Cheng [17] developed a web-based, pen-based tool which facilitates mathematical problem solving with peer review. Haciahmetoglu and Quek [27] developed a tool which enables sketch-based tutoring from a human tutor via web-based communication.

2.3 Bridging the Gap: Pen-Based Tutoring Systems

A relatively small body of research has been performed on systems that provide tutoring with pen-based interfaces. Lee and Kim [39] developed a pen-based mathematics ITS for solving simultaneous equations. Anthony et al. [8] developed a pen-based algebra ITS for equation problem solving. De Silva [18, 19] developed a pen-based ITS teaching Kirchhoff's laws for electric circuit analysis. Yin et al. [71, 22] developed a generic ITS applicable to virtually any sketch-based, spatial domain. However, this system requires the teacher to manually encode both the solution and all tutorial feedback logic for each problem, and the generic nature of the system means that it cannot be as effective as an ITS developed specifically for a single, specific domain.

A few pen-based tutoring systems have even been developed for the statics and physics domains. Valentine et al. [65, 10] developed an educational statics system, but it operates along the submit-and-compile paradigm, which neither helps a confused student to approach a problem nor adapts to a student's specific solution path. Cheema and LaViola [15, 16] developed a true ITS for entering and evaluating physics free-body dia-

grams in conjunction with equations in order to animate the diagrams. Lee et al. [41, 40] developed a pen-based statics ITS, which runs on the LeapFrog FLY pen system, which is a relatively primitive ancestor to the Livescribe smartpen. Lee [38, 32] developed a pen-based statics ITS, which runs on tablet PCs.

Unfortunately, all of these pen-based tutoring systems fail to effectively decompose the problem-solving process into small sub-steps to aid in scaffolding the student's learning. Doing so would allow a system both to form a better model of student cognition and to provide more-targeted feedback during the problem-solving process. The aim of this current research is to explore the potential efficacy of an ITS which is both pen-based and decomposes the problem-solving process into fundamental sub-steps.

Chapter 3

Newton’s Tablet: The Windows Tablet-PC System

We developed the tablet-PC version of this system first for two main reasons: the development tools for the Windows PC platform are much more powerful than those for Livescribe smartpen platform, and the standard PC program paradigm is much more intuitive for development. The tablet-PC version of the system, called Newton’s Tablet, was developed in C# using the Windows Presentation Foundation application programming interface. The system was then re-written—with significant modifications—to run on the Livescribe smartpen platform. This chapter describes the design of the tutorial system on the PC.

This chapter opens in Section 3.1 with an explanation of the problem-solving decomposition used in Newton’s Tablet and a discussion of its pedagogical significance. Section 3.2 provides a description of Newton’s Tablet user interface. Section 3.3 finishes

this chapter with a presentation of the details of tutorial feedback in Newton’s Tablet.

Appendix A enumerates and provides more detail in regards to the logical stages and functional modes used in Newton’s Tablet.

3.1 The Problem-Solving Decomposition

The instructional model developed with Newton’s Tablet decomposes the complex task of solving a statics problem into atomic steps. Each step requires the student to apply fundamental concepts. This process trains the student on how to approach the problem-solving process and what sorts of decisions need to be made.

The first step in the decomposed problem-solving process is to explicitly identify the boundary of each body in the system—something with novice students often struggle with. A correct boundary isolates the intended body from all other bodies. Common errors include completely redrawing all of the details from the original problem image or drawing overly abstract images, such as a “stick figure.” Neither of these cases adequately represents the intended system boundary. An accurate boundary helps the student differentiate between elements and forces which are internal to the body and those that are external—that is, inside and outside of the boundary, respectively. Newton’s Tablet requires the student to explicitly define the boundary of each intended body (Figure 3.1). The student does this by using the stylus to trace the boundary directly on top of the original problem image. The software then critiques the student’s trace and identifies typical mistakes such as cutting through a rigid body or including external supports. Once the student has identified the boundary of a valid system, Newton’s Tablet allows the student to drag it to an open area

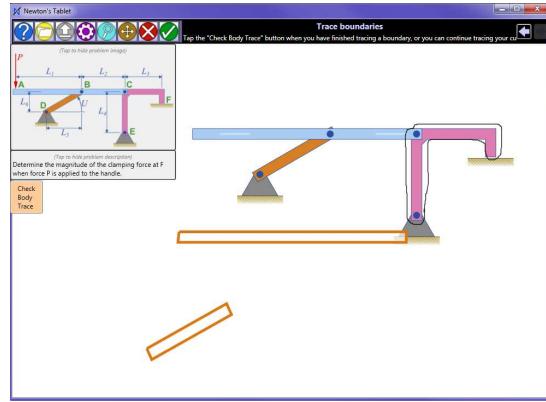


Figure 3.1: Tracing Boundaries.

in the workspace. This prevents the free-body diagrams of multiple bodies from overlapping while the student is constructing them.

After defining the boundaries of the system, the student needs to represent all of its external interactions as forces. Novice students often have difficulty identifying and correctly defining these forces. For example, students often select forces based on intuition or by analogy to other problems they have studied. As a remedy, the tutor requires the student to explicitly identify all of the relevant fundamental concepts to correctly model forces on a free-body diagram.

First, Newton's Tablet requires the student to identify all the locations at which the system interacts with other objects (Figure 3.2). The student does this by tapping the stylus at each point of interaction. Novice students often forget about forces such as gravity. Newton's Tablet then critiques the student's work, and requires the student to correct any errors in locating the interactions.

Next, the student must identify what type of interaction occurs at each of these

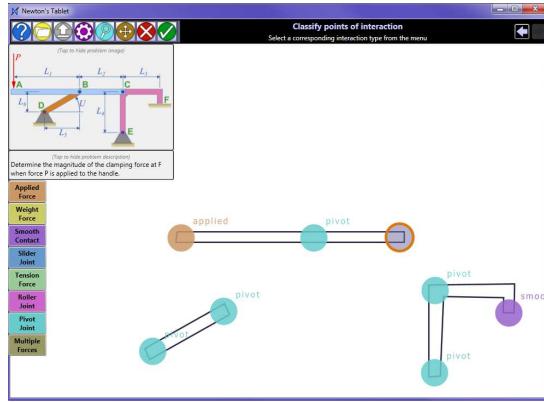


Figure 3.2: Classifying Points of Interaction.

locations (Figure 3.2). For example, the interaction could be produced by a roller joint, a pivot, a flexible element, and so on. It is possible for multiple interactions to occur at a single location. For example, a spring and pivot could be attached to the same location on a body. In such cases, the student must classify all interactions at that location. The student uses taps from the stylus to first select a point of interaction and then select its type. Again, Newton's Tablet critiques the student's work and requires the student to correct any errors in the interaction types before proceeding.

Identifying interaction types is crucial for correctly modeling forces. For example, a force from a pivot joint often acts in an unknown direction, and therefore must typically be modeled with two arrows, one for the x-component and another for the y-component. By contrast, the force from a roller joint always acts perpendicular to the surface upon which the roller rests, and thus is always represented by a single arrow normal to the surface.

At this point, Newton's Tablet prompts the student to identify any *two-force members* or *Newton's-third-law pairs* that may be present in the system. A two-force member is

a body that is subject to exactly two forces. For the body to be in equilibrium, those forces must be equal, opposite, and collinear. A third-law pair is a pair of interactions on two different free-bodies that are subject to Newton's third law. For such cases, the forces must be equal and opposite, but are typically not collinear. Identifying two-force members can simplify the analysis. For example, a pivot force ordinarily acts in an unknown direction, and therefore must be modeled with two orthogonal force components. However, if a pivot is on a two-force member, than the direction of the force is known: the line of action of the force passes through the two points at which forces are applied to the body. Novice students often have difficulty identifying two-force members and often fail to apply Newton's Third law correctly. With Newton's Tablet, the student use the stylus to identify bodies that are two-force members, or pairs of interactions that are subject to Newton's Third Law.

Finally, the student must represent each of the interactions with labeled force arrows. The previous steps are intended to guide the student through the process of identifying these forces. The student uses the stylus to draw a force arrow and the keyboard to enter a label for it (Figure 3.3). Newton's Tablet critiques the student's work and identifies mistakes.

After constructing the free-body diagrams, the student must derive equilibrium equations to solve for the unknowns. The process of deriving an equilibrium equation also involves synthesizing numerous fundamental concepts, and students must again be guided in using a systematic reasoning process.

First, the student must specify the system being analyzed. The student does this by tapping on the free-body diagram with the stylus. Then, Newton's Tablet requires the

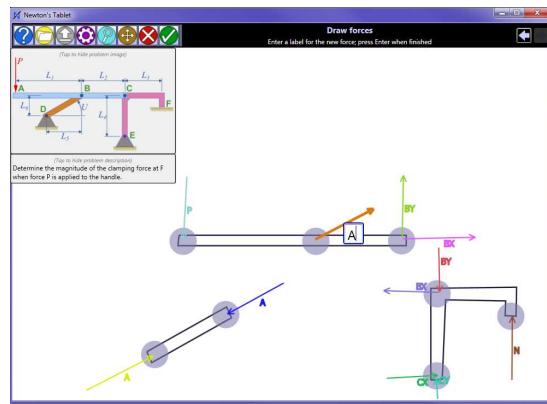


Figure 3.3: Drawing Forces.

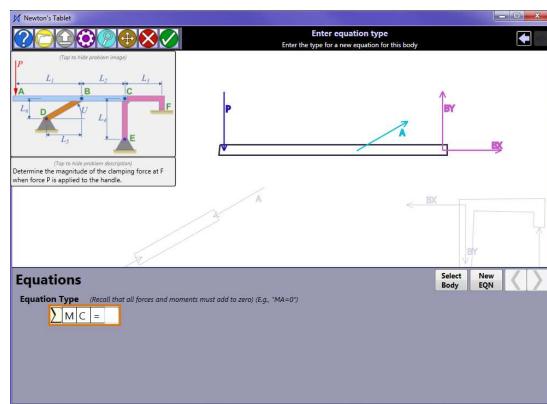


Figure 3.4: Entering the Equation Type.

student to identify which type of equilibrium equation is being considered (Figure 3.4). There are three possibilities: equilibrium in the x-direction, equilibrium in the y-direction, and moment equilibrium. The student uses the keyboard to enter the equation type. Newton's Tablet ensures that the equation type is valid.

Next, if the student is considering a moment equation, the student must identify whether moment will be considered positive in the clockwise or counter-clockwise direction. To specify this direction, the student uses the stylus to draw a curved arrow next to the equation type.

Newton's Tablet employs a novel approach for guiding the student through the process of constructing equilibrium equations. The system helps the student to focus on the conceptual reasoning by separating it from the mathematical analysis. The student begins by graphically identifying the force components that are relevant to the equilibrium equation. The student does this by simply tapping on the force arrows with the stylus (Figure 3.5). If any of these forces must be decomposed into x- and y-components, the student is prompted to do so. The student again uses the stylus and keyboard to draw and label the force component arrows. Here, the student assigns simple symbolic names, such as “AX” and “AY” to the component arrows, rather than deriving their magnitude from other known information. (The student will derive the magnitudes later.) The student then uses the stylus to identify which of these components are relevant to the current equilibrium equation.

If the student is considering a moment equation, then the student must also identify the moment arms for the forces. Here again, the student identifies the moment arms

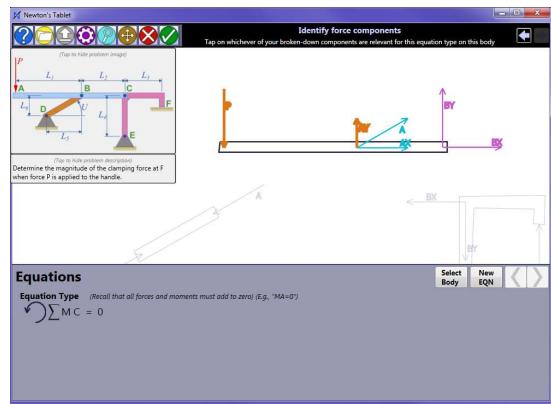


Figure 3.5: Identifying Forces.

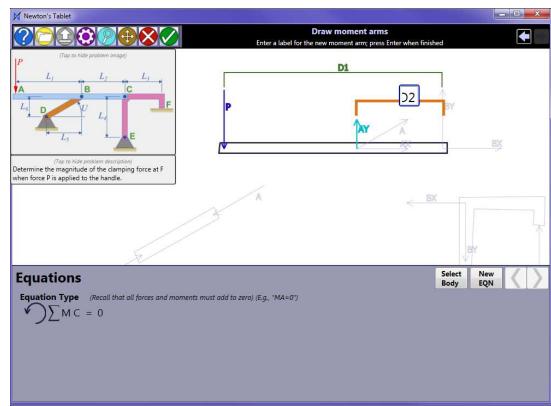


Figure 3.6: Drawing Moment Arms.

graphically by drawing a “bracket” gesture, such as those shown in Figure 3.6. The student uses a symbolic label, such as “D1” to represent the length of the moment arm. Later, the student will compute the lengths of the moment arms from information given in the problem.

Once the relevant forces and moment arms have been identified, the student must assemble the equilibrium equation. The system actually guides the student through constructing two forms of each equilibrium equation. First, the student constructs a *symbolic* equation using the symbolic labels assigned to the forces and moment arms. The student then expands the symbolic equation into an *expanded* equation in which the symbolic labels are replaced with values computed from information given in the problem statement.

Figure 3.7(d) shows an example of a symbolic equation, while part (e) of the figure shows the corresponding expanded equation. The symbolic equation contains only terms labeled directly on the free-body diagram. In the expanded equation, the student uses algebraic and geometric analysis to replace the symbolic terms with expressions computed from information given in the problem statement. This approach has several intended advantages. It initially enables the student to focus on the concepts without the extra complications of geometric and algebraic calculations. Once the student has correctly applied statics principles, then the student can focus on the calculations. We believe that this approach enhances problem-solving transfer. It also enables the system to accurately identify a student’s misconceptions and differentiate conceptual errors from math and geometry errors.

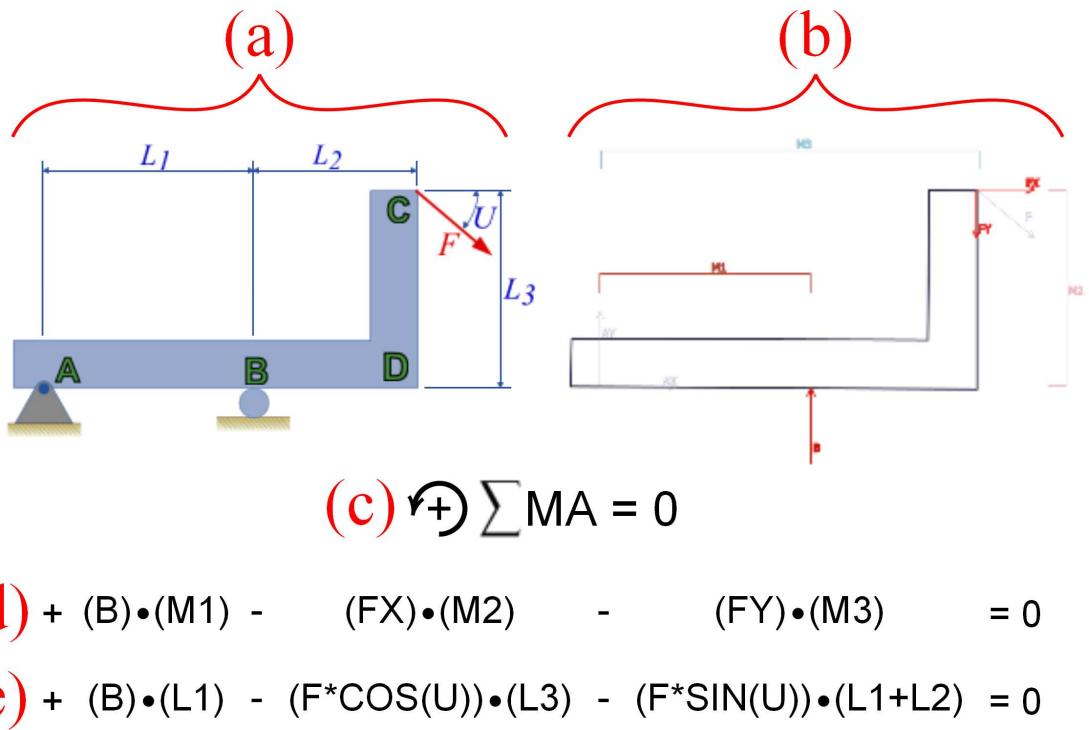


Figure 3.7: The Symbolic vs. Expanded Forms of an Equation. (a) The original problem description. (b) The student's free-body diagram. (c) The type of the equilibrium equation. (d) The symbolic form of the equation. (e) The expanded form of the equation.

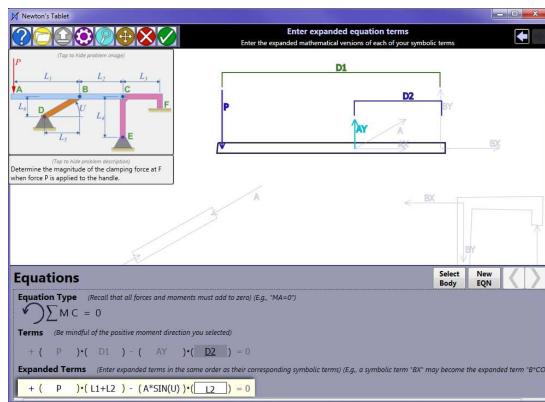


Figure 3.8: Entering the Expanded Equation.

Most problems require multiple equilibrium equations. The student repeats this process to create additional equations.

3.2 The User Interface

Section 3.2.1 presents an overview of the layout of Newton’s Tablet’s graphical user interface. Then Section 3.2.2 addresses the forms of user input used in Newton’s Tablet. This is followed by a description of Newton’s Tablet’s stroke recognition algorithms in Section 3.2.3. Section 3.2.4 then discusses the beautification process.

3.2.1 The Graphical User Interface Layout

Here a brief overview of Newton’s Tablet interface is presented. Figure 3.9 shows this interface. The overall Newton’s Tablet window is divided into three main sections.

The first section of the window runs along the top. There is an assortment here of menu buttons used throughout the program, a status bar which prompts the student with brief instructions for each stage, and a pair of status bar navigation buttons for re-examining previous messages. Table 3.1 describes each of the menu buttons.

The next section of the window is the free-body diagram canvas. This is where all of the student’s work for constructing free-body diagrams takes place. Not only does this canvas capture student stroke data, but the original system image and feedback for the student’s work are also rendered here. A small box showing the current problem’s original, annotated system image is situated in the top-left corner, and immediately below that is an additional box containing a textual description for the current problem. The student can dynamically collapse and expand both of these boxes for space efficiency. The free-body diagram canvas may also contain additional buttons along the left side for certain stages.

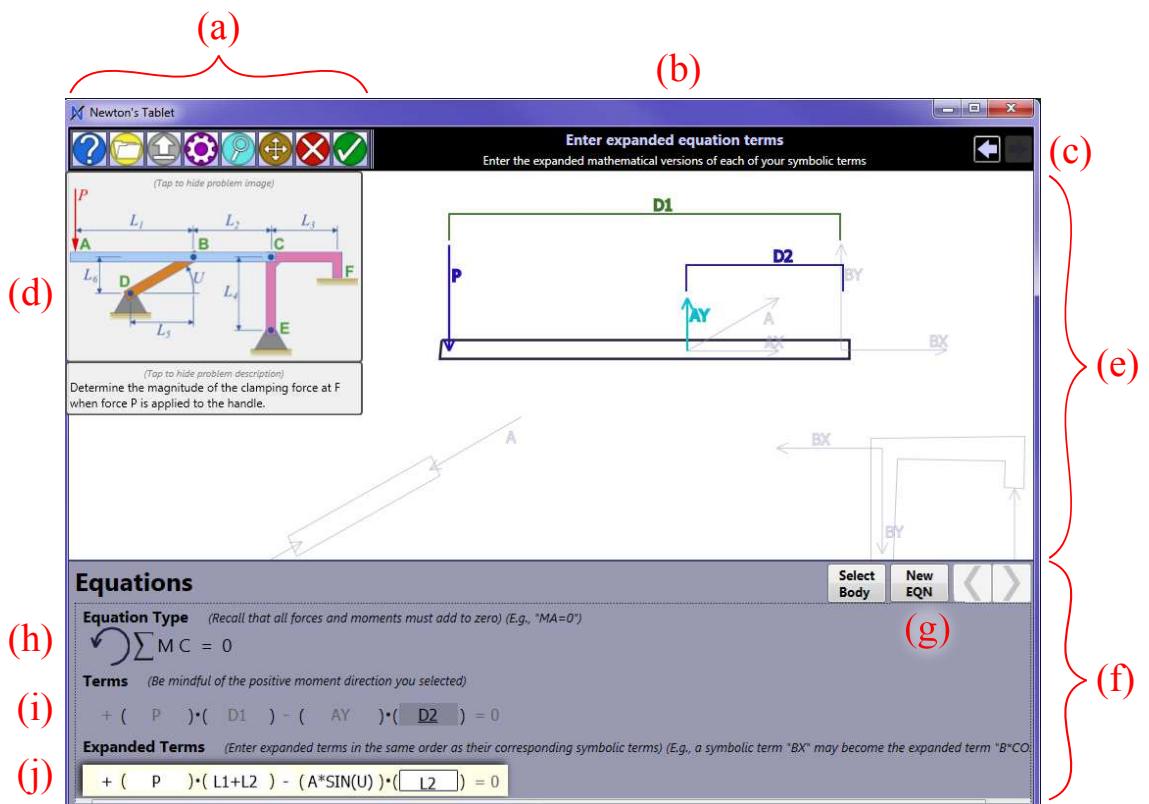


Figure 3.9: Newton's Tablet Graphical User Interface. (a) Menu buttons for use in most stages of the program. (b) Status bar where brief messages are displayed which prompt the student with directions for the current stage. (c) Buttons for scrolling through recently displayed status-bar messages. (d) Boxes containing the original problem image and description. Additional buttons may appear below these boxes depending on the stage. (e) Free-body diagram canvas for drawing free-body diagrams. (f) Equation panel used for constructing equilibrium equations. (g) Buttons used to switch between work on different equations. (h) The equation type. (i) The workspace for entering the symbolic form of an equilibrium equation. (j) The workspace for entering the expanded form of an equilibrium equation.

Menu Button	Functionality
General Help Mode Button	Toggle General Help Mode (Section A.2.3)
Load Problem Button	Select a new statics problem to work on (Section D.2)
Upload Button	Submit the student's to a file server
Settings Button	Primarily for development purposes such as stage skipping
Query Mode Button	Toggle Query Mode (Section A.2.4)
Drag Mode Button	Toggle Drag Mode (Section A.2.1)
Erase Mode Button	Toggle Erase Mode (Section A.2.2)
Check Work Button	Check the student's work for the current stage (Section A.2.6)

Table 3.1: Menu Buttons for Newton's Tablet. The menu buttons are shown in the top-left corner of the window. They are listed here in order from left to right.

The final section of the window is the equation-entry panel. This panel is only displayed after the student has completed the free-body diagram half of the problem. In the top-right corner of this panel are a pair of buttons used for switching between equations which are in-progress or have been completed for the currently selected body. These sit next to a similar button used for starting work on new equations for the current body. Also in this corner is a button used for switching to solve equations for another body. In the top-left corner of this panel is the region where the student enters the equation type and possibly the positive moment direction if the type specifies a moment equation (Sections [A.1.9](#) and [A.1.10](#)). Then, below the equation type is a region where the student enters the symbolic form of the current equation (Section [A.1.15](#)). Finally, the area where the student enters the expanded form of the current equation is at the very bottom of the equation panel (Section [A.1.16](#)).

3.2.2 Student Input

The primary means of student input in the Newton’s Tablet program occurs via hand-drawn strokes within the free-body diagram canvas. Although we designed the system to be used with a stylus on a tablet device, a mouse can also be used with a standard PC. The student uses longer strokes throughout the system for sketching solutions on the canvas and uses very short strokes—taps—for additional logical input or to query the system for feedback.

Previous research [38, 32, 40, 18] with similar tutoring systems has found that inaccuracies from the hand-writing recognition engine often present a substantial barrier to using the system effectively. Because of this, Newton’s Tablet accepts textual input through the keyboard. Obviously, hand-drawn text entry would be a more natural form of input and would likely aid in the learning process, but the development of a significantly more accurate online recognition engine was outside the scope of this current research. With Newton’s Tablet, we made every attempt to seamlessly integrate textboxes for this keyboard-based text entry in a natural fashion. There are two types of text entry in Newton’s Tablet: labels for various free-body diagram shapes and text for equation entry.

When entering textual labels for free-body diagram shapes, a textbox is automatically generated next to the shape as soon as it is drawn. This transition is relatively smooth and intuitive.

For entering text for equations, Newton’s Tablet compromises between two conflicting concerns. The first concern is to scaffold student work. By breaking an equation apart into a structured collection of textboxes for each separate term, the system can make

more-definite decisions regarding the student thought process at each part of the equation. This in turn enables the system to present more-targeted and more-helpful feedback in response to student errors.

The second concern is to present a natural user interface which mimics the use of the traditional pen-and-paper interface for statics. Unfortunately, segmenting the text-entry area into a long sequence of textboxes does not help to emulate the traditional drawing space for constructing equilibrium equations. To somewhat remedy this unnatural design, additional textboxes for a new term are only generated after text has been entered in textboxes for all other terms. Furthermore, once the student has correctly constructed an equation, these textboxes are hidden and the structure of the equation is left appearing natural and cohesive.

3.2.3 Stroke Recognition

Throughout Newton's Tablet, there are five types of gestures the student can draw on the free-body diagram canvas. These are: traces, arrows, brackets, taps, and erasures. The system needs to be able to distinctly recognize strokes representing each of these shapes.

During the Boundary-Trace Stage (Section A.1.2), in order to identify unambiguously a system boundary, the student traces this boundary directly on top of the original problem image (Figure 3.10). This forces the student to make his/her thoughts concrete as to what should be considered a relevant body, and this allows Newton's Tablet to more readily recognize which body is being represented by the student's free-body diagram. The goal, then, of the trace recognition algorithm is to determine whether the student's trace

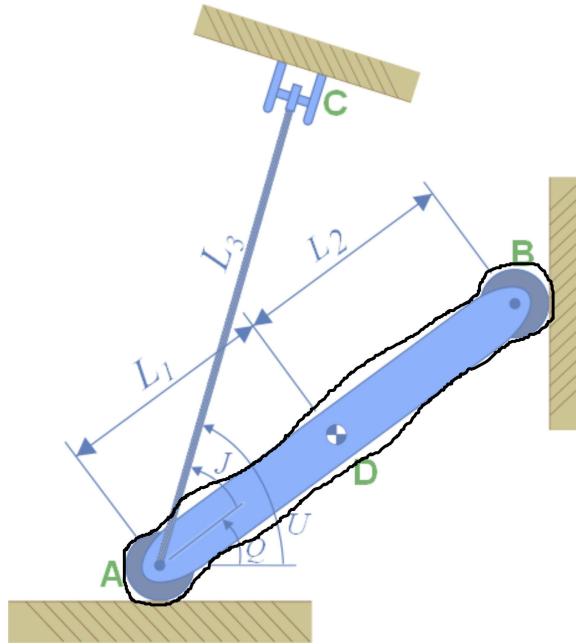


Figure 3.10: A Correct Boundary Trace. The student’s trace correctly identifies the relevant system boundary.

correctly identifies a system boundary.

Newton’s Tablet represents each distinct element of the original problem image as a polygon in the problem encoding. Also, all possible unions of these polygons are stored in the encoding. Some of these polygons represent valid bodies, which are part of some correct solution for the problem, and some of these polygons represent invalid bodies, which can be used to provide more-targeted feedback as to why a matching student trace is incorrect. Then, in order to critique a student’s trace, the recognition algorithm can test whether the trace matches any of the known polygons.

The basic idea for determining whether a trace matches a polygon is to check whether each point from the sequence of points in the trace is near some point from the

polygon, and vice versa. In other words, it is a check for the approximate coincidence between two point sets. Newton’s Tablet actually uses the Hausdorff distance metric to calculate how well a trace matches a boundary.

The Hausdorff distance between two sets of points is the greatest distance from a point in the first set to the closest point from the second set. More formally, the Hausdorff distance between finite sets A and B is a maximin function, defined as:

$$H(A, B) = \max(h(A, B), h(B, A)),$$

where

$$h(A, B) = \max_{a \in A} \left(\min_{b \in B} \|a, b\| \right),$$

and $\|a, b\|$ represents the distance—typically the Euclidean distance—between the points a and b . $h(A, B)$ represents the greatest distance that exists from a point in A to its nearest point in B . This is known as the directed Hausdorff distance from A to B . If $h(A, B) = d$, then every point in A has a corresponding point from B that is located within a distance of d . Note that typically $h(A, B) \neq h(B, A)$. The Hausdorff distance is then the maximum of these two directed distances.

The trace recognition algorithm loops over each boundary polygon, rejects any boundary whose Hausdorff distance from the trace is greater than some threshold, and returns the first boundary whose Hausdorff distance is less than the threshold.

Previous research [38] used an area-based approach to solve the problem of recognizing which body is represented by a student’s trace. This algorithm considered the intersection of the areas of the boundary and the trace polygons (Figure 3.11). A trace was

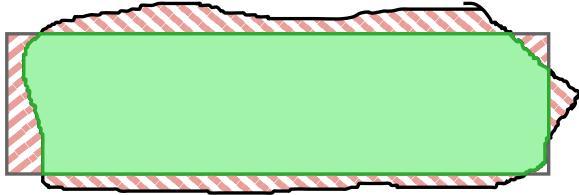


Figure 3.11: Area-Based Trace Recognition. An alternative recognition strategy could consider the area of the intersection between two polygons. Areas in green represent matching areas between the trace and the underlying body, while crosshatched areas represent non-matching areas.

then recognized as matching a boundary if 80% of the area of the boundary polygon was contained in the trace polygon. Additional checks were included to ensure that the trace polygon did not completely include an enlarged version of the boundary polygon and did not completely include a shrunken version of the boundary polygon. However, calculating the area of the intersection between two arbitrary polygons is computationally intensive. The boundary-based approach developed for this current research is much more efficient and is better suited for the Livescribe smartpen platform.

There are some important optimizations to mention in the trace recognition algorithm. The policy of accepting the first matching boundary polygon is the algorithm's first important optimization. Because fewer polygons are considered, this results in a decrease in recognition time in the average case by a little more than a factor of two. A side effect from this optimization is the possibility that there exists another boundary which is actually a closer match to the student's trace than the selected boundary. However, most problem boundaries are not similar enough for this to be a problem, and the beautification process (Section 3.2.4) will ensure that there is no confusion between what the student thinks a trace represents and what the system thinks that trace represents.

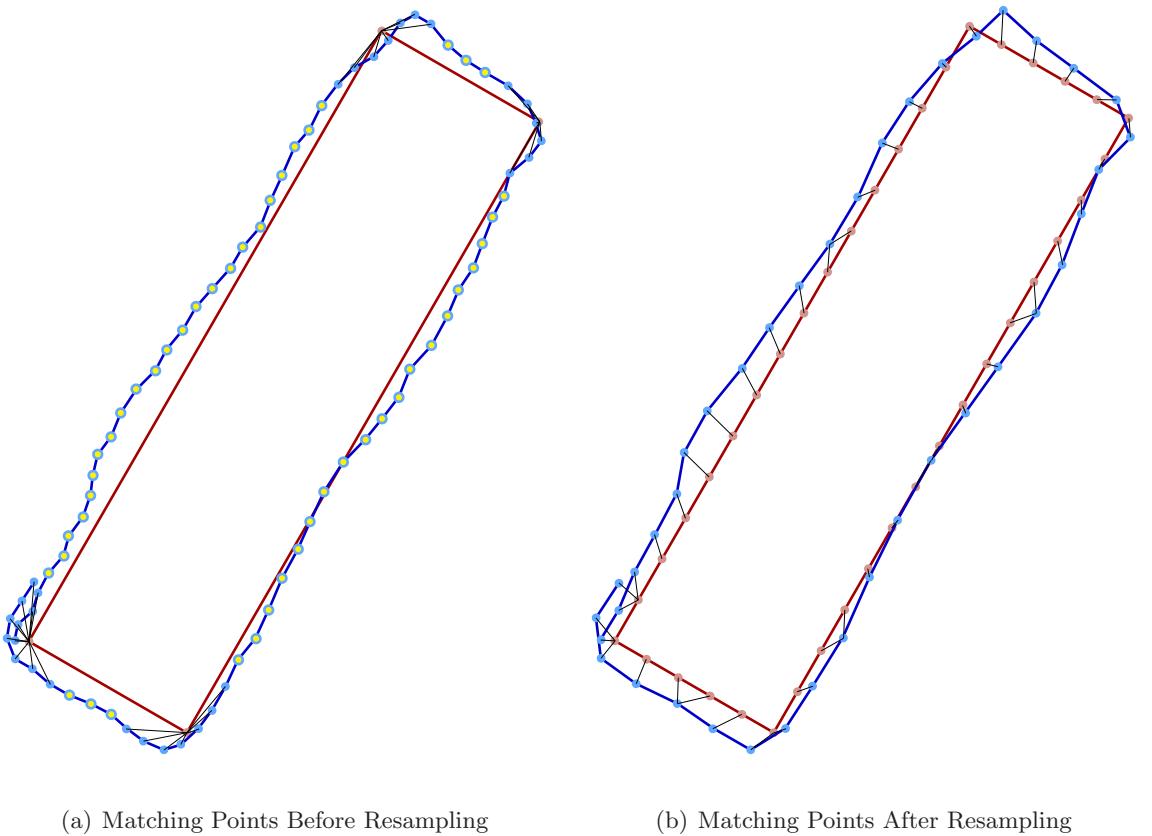


Figure 3.12: The Importance of Resampling for Matching Points. In these two examples, the blue points represent the hand-drawn stroke, and the red points represent the boundary of the underlying body. Lines are drawn from a point in one set to the closest point in the other set if the closest point is within the distance threshold. The points with a yellow center represent points that do not have any close matches from the other set. (a) Many of the points from a densely sampled trace stroke will not match any of the points from a sparsely sampled boundary polygon. (b) Resampling the trace and the boundary polygon to the same number of points—40 in this case—greatly increases the likelihood of points having matches.

The next optimization comes from resampling. Resampling is the process of either increasing or decreasing the number of points in a given sequence, such that adjacent points in the resulting sequence are roughly evenly spaced. The number of points contained in the resulting sequence is a parameter to the function. Newton’s Tablet resamples student traces—and the underlying body polygons—to 64 points. Resampling is actually critical for the success of the trace recognition algorithm, because boundary polygons often consist of a small number of points that are spread out over a distance much greater than the threshold for Hausdorff distance failure. This means that any trace point lying in the middle of a line segment between such spread-out boundary points will likely not have a match even though the trace point may indeed lie right on the edge of the boundary polygon. By resampling both the trace and the boundary polygon to include the same number of points, this scenario becomes extremely unlikely. Figure 3.12 demonstrates why resampling is important for finding matching points. Resampling is an optimization for trace recognition, because student traces typically include far more points before being resampled, meaning that exponentially fewer distance calculations between points will need to be performed while looping over the boundary polygons. The resampling algorithm used in Newton’s Tablet is taken from Wobbrock et al. [70].

Another optimization comes from using a simple heuristic when searching for a matching point. A naive approach for finding a matching nearby point would always start the search at the beginning of the entire point collection. However, when trying to find a matching point, this algorithm instead starts the search from within the vicinity of the point from the previous match. Specifically, points within three indices from the previous

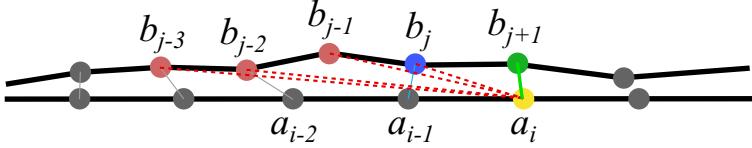


Figure 3.13: Searching Based on the Previous Match. This example shows a search for a match for the point a_i (the yellow point). The search starts three points before the match from the previous search, which in this case was b_j (the blue point). A match is eventually found at point b_{j+1} (the green point). The red points represent points that were checked and were not close enough. The dashed, red lines represent the failed distances that were checked for a_i . The other lines represent the distances to a match that was actually accepted for a point.

match are considered first. For example, when trying to match a point a_i from set A to any point from set B , the algorithm starts the search for a point from B within the vicinity of the match that was found for point a_{i-1} . For example, if b_j was the matching point found for point a_{i-1} , then the order in which the points from B will be searched to find a match for a_i is $b_{j-3}, b_{j-2}, b_{j-1}, b_j, b_{j+1}, \dots$. Figure 3.13 demonstrates the use of this heuristic. Most matching points are indeed found within the vicinity of the match for the previous point, and this strategy greatly improves recognition runtime.

The trace recognition algorithm further decreases its average-case runtime by halting the search for a matching point as soon as one is found. However, this is not fully true for the reciprocal case. If a match is not found for a point in the first set, then the algorithm does indeed abandon attempts to find a match for the remaining points in this first set. However, an attempt is then still made to determine whether all of the points in the second set have a match within this failed first set. If this is the case, then the system can provide more-specific feedback to the student. For example, if all of the trace points match to a point in some boundary, but there exist points in that boundary that do not

match to any point in the trace, then it is likely that the student's trace breaks a rigid body (Figure 3.14(a)). Reciprocally, if all of the points in some boundary match to some point in the trace, but there exist points in the trace that do not match to that boundary, then it is likely that the student's trace includes some external support material (Figure 3.14(b)). Both of these cases represent important conceptual errors, about which the student needs to be informed.

Newton's Tablet also allows the student to use multiple strokes when tracing a boundary. When multiple strokes are used, the trace-recognition algorithm simply concatenates strokes together in the order they were drawn. Assuming the student draws his/her trace strokes in order around the perimeter of the boundary, this simple concatenation does not present any problems. This policy is robust to collinear over-stroking (Figure 3.15(a)) and collinear gaps between adjacent strokes (Figure 3.15(b)). The former issue is not a problem, because point redundancy and ordering does not affect recognition accuracy as long as all trace points lie along the perimeter of the boundary. The latter issue does not present a problem, because the resampling process will fill in the gap.

The student draws brackets during the Moment-Arm-Drawing Stage (Section A.1.14). For bracket recognition, Newton's Tablet uses the Protractor gesture recognizer [42], which is a template-based recognition algorithm. This recognizer was chosen because it is robust to changes in size and orientation and is relatively easy to implement. See Garcia [24] for a more detailed description of the use of the Protractor recognizer in Newton's Tablet.

The student draws arrows during the Force-Drawing Stage and the Component-Break-Down Stage (Sections A.1.7 and A.1.12, respectively). For arrow recognition, New-

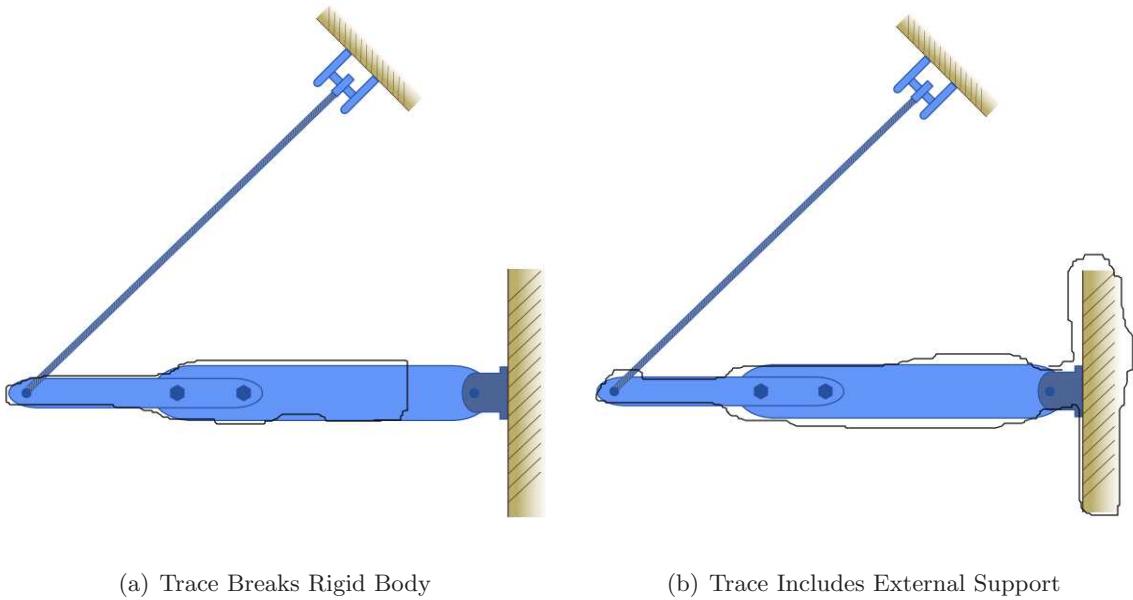


Figure 3.14: Asymmetric Trace-Boundary Matches. In (a), the trace points all match to a boundary point, but not all boundary points match to a trace point. In (b), the boundary points all match to a trace point, but not all trace points match to a boundary point.

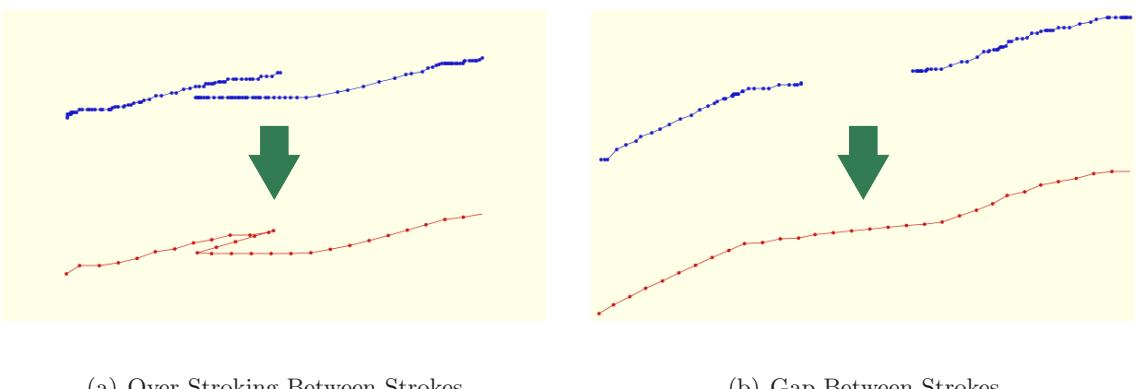


Figure 3.15: Multi-Stroke Trace Merging. In both cases, the blue dots and lines on the top represent the points from the original strokes, while the red dots and lines on the bottom represent the points in the single resulting merged and resampled stroke.

ton’s Tablet has the policy of simply recognizing any stroke that is long enough. Earlier versions of Newton’s Tablet also used Protractor for recognizing force arrows. Unfortunately, there is a lot of variance in how students draw arrows, so recognition accuracy was not as high for recognizing arrows as it is for recognizing brackets; there were too many false negatives. Furthermore, even once a stroke has been successfully recognized as being an arrow, additional processing is needed in order to determine the location of the arrow’s tip. Newton’s Tablet was using a curvature-based approach for finding an arrow’s tip, but this was also frequently inaccurate. With the newer, simplistic recognizer, the start of the stroke becomes the position of the arrow’s tail, and the direction of the arrow is then calculated via a least-squares line fit. Very few students ever draw the wrong shape when they should be drawing an arrow, so this simplistic recognition algorithm yields few false positives and is far more usable overall. See Garcia [24] for a more detailed description of arrow recognition in Newton’s Tablet.

The student draws taps during most stages. Taps are by far the simplest gesture to recognize. All strokes captured by the system are filtered according to how many points they contain. If the number of points in a stroke is less than a certain threshold—Newton’s Tablet uses a threshold of 20 points—then the stroke is treated as a tap occurring at the centroid of the stroke. Otherwise, the stroke is not treated as a tap, and may instead be subject to further recognition. After a stroke has been recognized as being a tap, Newton’s Tablet calculates whether the tap location intersects with any of the elements on the canvas. The specific point-intersection algorithm used depends upon the given shape. For determining whether a point intersects with a circle, the distance from the tap point to the circle’s

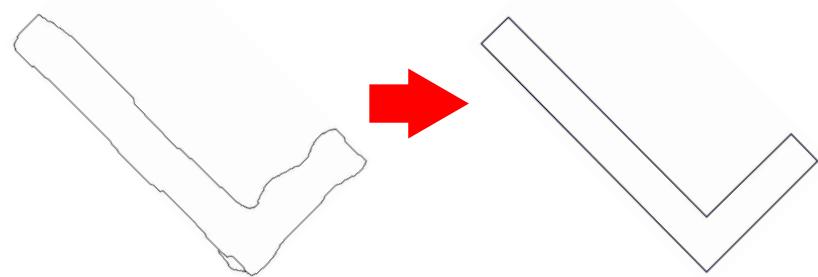
center is compared to the circle’s radius. For determining whether a point intersects with an axially aligned rectangle, the tap point is compared to the minimum and maximum horizontal and vertical dimensions of the rectangle. For the point-in-polygon calculation, Newton’s Tablet uses a standard winding number algorithm [1].

The student can also draw erasure strokes during many stages. In order to test whether an object on the free-body diagram canvas is “erased” by an erasure stroke, Newton’s Tablet calculates whether any of the segments of the stroke intersect with the object. See Section A.2.2 for a description of the erasure functionality.

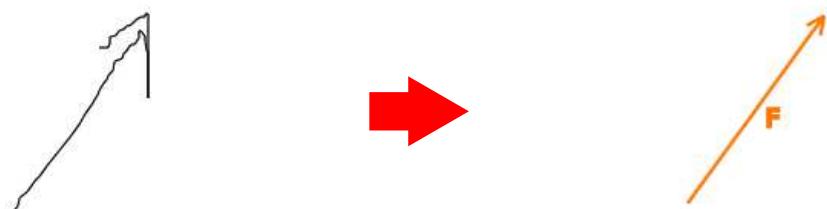
3.2.4 Beautification

Beautification is the process of replacing a hand-drawn stroke with a machine-drawn representation. Figure 3.16 shows each of the different shapes Newton’s Tablet asks the student to draw. The left column shows the student-drawn strokes and the right column shows the beautified versions.

The primary reason for beautification is to help inform the student of how the system has interpreted a gesture [38]. For example, in Figure 3.16(a) the student has drawn a somewhat sloppy trace that could be interpreted as either including or not including the support structure on the right. By replacing the student’s stroke with the polygon from the traced body, the system removes any ambiguity about what was recognized. The same principle holds for brackets and arrows. For example, the direction of a noisy arrow stroke can often be confused, but by replacing the stroke with a beautified version all ambiguity is removed.



(a) Boundary Trace



(b) Force Arrow



(c) Moment-Arm Bracket



(d) Positive-Moment-Direction Arrow

Figure 3.16: Beautification. The original hand-drawn stroke is shown on the left and the beautified version is shown on the right.

Another important justification of beautification is to maintain neatness. A hand-drawn workspace can quickly become cluttered and difficult to interpret unless the student takes great care and foresight to organize his/her strokes and to draw neatly. This is typically not the case, especially for novice students who are still learning how to construct free-body diagrams.

All beautified shapes that the student labels are given a randomly selected color in order to offer more visual distinction to different elements. The color selection process uses heuristics in order to generate random colors that have high color contrast with the canvas and with previously generated colors.

3.3 Tutorial Feedback

There are two fundamental types of feedback provided by Newton's Tablet. The first type is a response to an input event, and is presented in the form of either a temporary or a persistent message in the status bar at the top of the window. Examples of this type of message include the temporary message "Successfully recognized your boundary trace," when the student correctly traces a system boundary, and the persistent message "There are problems with your selected forces: tap on a highlight to learn more," when the student makes a mistake during the Force Identification Stage.

The second type of feedback is the result a conceptual analysis of the student's work. This occurs when the student believes he/she has completed the work for the current stage and taps the Check Work button. If there are errors, the system highlights their locations. These highlights can take the form of polygons over a body, rectangles over

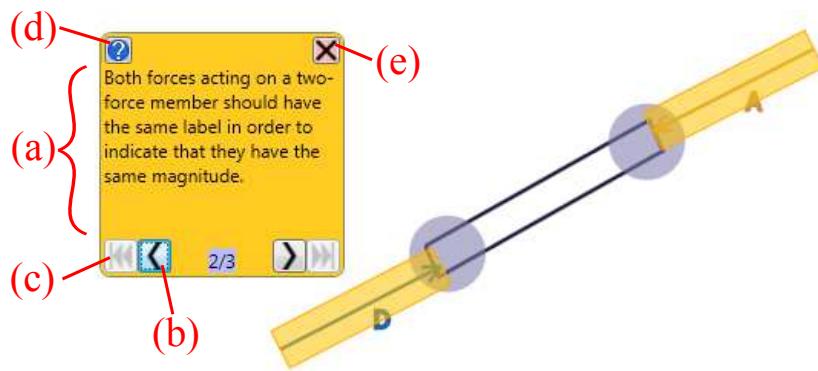


Figure 3.17: An Error Help Message. (a) The current textual message in this error's message sequence. (b) Navigational buttons for scrolling through the sequence of messages for this error. (c) Sometimes it is possible for multiple conceptual errors to be present simultaneously in the same location. When this is the case, these skip buttons can be used to transition between the errors. (d) This button opens a relevant general help page with more information on the underlying concept of the current problem error. (e) This button can be used to close the error message box.

arrows or brackets, circles over specific points on the free-body diagram, or a background-color change for a textbox. It is also possible for a single error to be based upon multiple locations. For example, when the student labels two unrelated force arrows with the same letter, the location of the error is actually at both arrows. In this case, both are highlighted.

Once Newton's Tablet displays these error highlights, the student can tap on them to query the system for explanations. The explanations are presented inside of a small box that appears next to the highlight. Figure 3.17 shows an example. The text box in the figure contains a sequence of textual messages that explain in progressively greater detail what is wrong with the student's work. The student can scroll from one message in this sequence to the next via buttons along the bottom of the box.

There is an important reason for presenting error explanations near the locations of the errors. Previous research [31, 38] suggests that presenting a direct spatial connection

such as this between an otherwise abstract textual error message and the student’s actual work leads to greater student understanding of the underlying concepts. When presented only with a text-based list of error descriptions, students often will not expend the effort to decipher the conceptual information of the text. By presenting errors according to their locations, the student is far more likely to form the cognitive associations between the underlying statics concepts and the actual work in the workspace.

Presenting feedback in the form of messages that gradually increase in specificity accomplishes two things. First, it provides the student with an opportunity to try to fix the error on his/her own before viewing detailed feedback. Second, it focuses the student’s attention on one piece of information at a time. This can make what would otherwise be an intimidating wall of text more approachable.

As an example of the sequence of messages presented for an error, here are the messages displayed when the student incorrectly selects a point inside the system boundary as a point of interaction during the Point of Interaction Location Stage (Section A.1.3): “This point of interaction is internal to your system,” “Internal system interactions are those which reveal complex internal forces,” and “These interactions should not appear on the free-body diagram.”

Other types of errors include things such as drawing force arrows in the wrong direction or using the wrong trigonometric function for an expanded equation term. See Appendix C for a complete enumeration of the conceptual errors addressed by Newton’s Tablet.

Another useful feature of the problem error box is the general help mode button in

the top-left corner of the box. If the student taps this button, then the General Help Mode window is displayed (Section [A.2.3](#)), and its content is set to whichever page is relevant to the underlying concepts of the current problem error.

Chapter 4

Newton’s Pen: The System for Livescribe Smartpens

The version of the tutorial system that we designed to run on Livescribe smartpens is called Newton’s Pen (Figure 4.1). In developing Newton’s Pen, we modified the instructional design and user interface design from Newton’s Tablet to work within the constraints of the smartpen platform. Because of the limited computational resources of the smartpen and its novel user interface capabilities, significant modifications to the system design and implementation were necessary. This chapter describes these changes.

Newton’s Pen is written in Java using the Livescribe application programming interface. Programs that run on the Livescribe smartpen platform are called penlets. Livescribe smartpens must be used in conjunction with special pages (Figure 1.2) that have been printed with an Anoto dot pattern [26]. This dot pattern enables a camera at the tip of the smartpen to determine which page the pen is on, and its location on that page. This,



Figure 4.1: The Livescribe EchoTM Smartpen. This is one of the two models of Livescribe smartpens used with Newton’s Pen; The other was the Livescribe PulseTM Smartpen. (a) OLED display. (b) Speakers. (c) A camera in the pen tip reads the dot pattern on the page. (d) Ink cartridge. (e) Micro-USB port. (f) Power button.

in turn, enables the pen to record a digital copy of all ink the student draws. Newton’s Pen interprets the student’s ink as he/she draws in order to evaluate the correctness the work. Tutorial feedback is presented via audio messages from an integrated speaker and small images and textual messages scrolled across a display on the barrel of the pen.

Because Newton’s Pen does not have a large screen on which to dynamically display information, it is more difficult for the pen to provide clear tutorial feedback. For example, a typical human tutor often indicates the location of the error and draws explanatory information. Newton’s Tablet can easily provide this functionality. However, Newton’s Pen can only present brief text or audio messages, or it can reference external material for explanations that are inherently less contextually relevant. Furthermore, Newton’s Pen cannot provide extended messages without slowly scrolling them across the small, integrated display. An ideal tutorial interface needs to be both natural and powerful enough to provide immediately understandable feedback within the context of the student’s current work. The Livescribe smartpen platform does not meet this requirement.

We designed Newton’s Pen to provide instruction for both free-body diagrams and equilibrium equations. However, because of memory limitations, we had to remove the support for equilibrium equations. With the support included, the software often encountered memory errors.

Section 4.1 describes the user interface used in Newton’s Pen. Section 4.2 follows this with an explanation the problem-solving decomposition used in Newton’s Pen and discusses its pedagogical significance. A more in-depth discussion of the inherent challenges of the Livescribe smartpen platform is then presented in Section 4.3. Section 4.4 concludes this chapter with a discussion of the tutorial feedback system in Newton’s Pen.

Appendix F describes auxiliary documentation, which was provided in tandem with the Newton’s Pen software. This documentation was intended to facilitate both using the system and understanding the concepts.

Appendix B provides more details about the logical stages and functional modes used in Newton’s Pen. This gives a particular focus on how they differ from their counterparts in Newton’s Tablet. The same problem set from Newton’s Tablet was used with Newton’s Pen and is presented in Section D.2.

4.1 The User Interface

This section begins by describing the worksheets used with Newton’s Pen (Section 4.1.1). This is followed by a description of the forms of output employed by Newton’s Pen (Section 4.1.2). For a discussion of the problems associated with not being able to remove ink, see Section 4.3.3.

4.1.1 The Worksheets

When using Newton’s Pen, the student interacts with two types of worksheets: problem-description and free-body diagram worksheets. The former (Figure 4.2) contains the problem statement and a series of “buttons” with which the student interacts with the penlet. The buttons, which are the same for every problem, are described in Table 4.1. The buttons include a keyboard for text entry.

Each free-body diagram worksheet (Figure 4.3) contains two workspaces in which the student draws his/her free-body diagrams. These workspaces contain a faded image of the system from the problem description, over which the student traces relevant system boundaries. Workspaces also contain two small textboxes in their top-right corners: one textbox is used by the student to record the name of the system represented by the free-body diagram, while the other is used to record whether or not the system is a two-force member. The student’s problem solution is drawn in one of these workspaces.

The penlet can determine which problem the student is working on as soon as the student taps on a new dot-patterned page. This is because Newton’s Pen associates each worksheet for each problem with a specific page address, which is uniquely determined by that page’s dot pattern.

Multiple free-body diagram workspaces are provided for each problem. This enables the student to construct multiple free-body diagrams for multi-body problems. It also enables the student to start over with a clean workspace if the student makes errors.

Homework 5 Problem Description Problem 2

Instructions: The elements of a rear suspension for a front wheel drive car are shown in the figure. Determine the magnitude of the force at each joint if the normal force N exerted on the tire is known.

CHECK WORK	ERASE	DONE YES	NO	BACKSPACE	UP	DOWN	RIGHT	
				Applied Force (AF)	Tension Force (TF)			
				Weight Force (WF)	Roller Joint (RJ)			
				Smooth Contact (SC)	Pivot Joint (PJ)			
				Slider Joint (SJ)	POI Done			
				Start Over		Recent Messages		
				SFX				

Figure 4.2: A Problem-Description Worksheet. (a) Assignment and problem number. (b) A large, image of the problem system. (c) A textual description of the problem. (d) General, system-use buttons. These help the student to interact with the tutoring system within each stage. The keyboard is used for text entry.

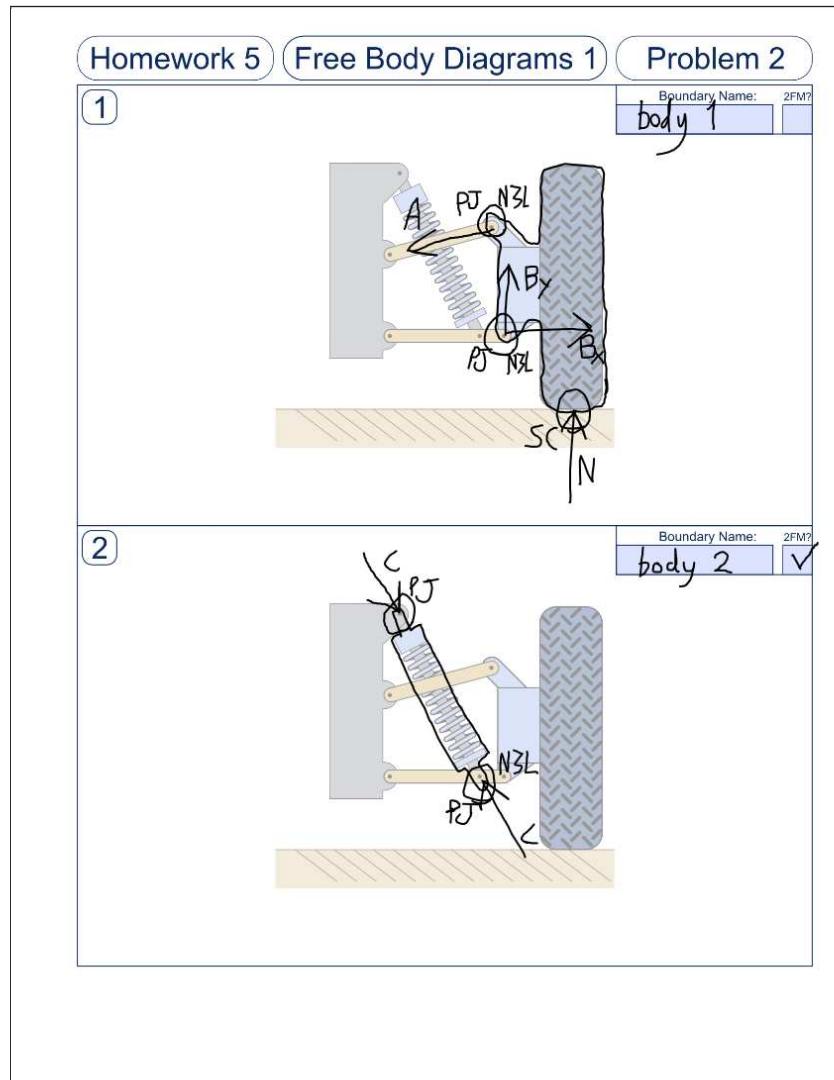


Figure 4.3: A Free-Body Diagram Worksheet. (a) Assignment and problem number. (b) A workspace for drawing a free-body diagram. (c) A faded image of the given system in this problem. (d) A textbox in which the student records the name of the system modeled in this workspace. (e) A textbox in which the student records whether the body in this workspace is a two-force member. (f) The number of this workspace.

Menu Button	Functionality
Check Work Button	Trigger evaluation of the correctness of the student's work for the current stage (Section 4.4)
Erase Button	Toggle Erase Mode (Section B.2.1)
Yes and No Buttons	Respond to system prompts
Directional Buttons	Navigate through various menus presented on the pen's display
Keyboard Buttons	Enter labels for forces (Section B.2.3)
POI Type Buttons	Classify points of interaction (Section B.1.2)
Start Over Button	Inform the system to ignore all work for the current problem
Recent Messages Button	Scroll through recently displayed messages, which may have been too brief or over-written by a later message
Sound Effects Button	Toggle sound clips for the special notification types (Figure 4.4)

Table 4.1: Newton's Pen Paper Buttons. Each problem-description worksheet contains a set of paper-based buttons.

4.1.2 Providing Output

One fundamental difference between the output of Newton's Tablet and Newton's Pen is that Newton's Pen does not have a large screen on which to render arbitrary images and visually manipulate student input. Instead, the majority of output in Newton's Pen occurs in the form of textual messages that are rendered on the smartpen's display. If these messages are longer than the length of the display, then they are slowly scrolled. Similar to Newton's Tablet, two types of textual messages are presented: temporary messages which occur after every student stroke, and persistent, stage-specific messages that are displayed when the temporary messages time out.

Newton's Pen also has a small collection of audio clips—simple sound effects less than 0.5 seconds in length—which are played via the smartpen's integrated speaker. Each of these audio clips corresponds to a specific type of notification event. Using a small set of special notification event types enables the student to instantly ascertain the result of

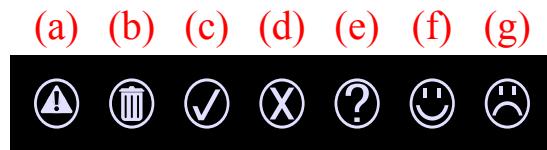


Figure 4.4: The Pen Notification Icons. (a) System Error Notification. (b) Stroke Erased Notification. (c) Correct Work Notification. (d) Incorrect Work Notification. (e) Information Notification. (f) Recognition Succeeded Notification. (g) Recognition Failed Notification.

most events without needing to read the entire textual message. These special notification events also each have a corresponding icon that is displayed on the display at the start of the event’s textual message (Figure 4.4). The seven notification event types used by Newton’s Pen are: (1) a System Error Notification, which occurs any time some operating system or Java exception occurs; (2) a Stroke Erased Notification, which occurs whenever the student successfully “erases” a stroke (Section B.2.1); (3) a Correct Work Notification, which occurs when the student successfully completes all of the work for the current stage; (4) an Incorrect Work Notification, which occurs when there are errors in the student’s work for the current stage; (5) an Information Notification, which occurs whenever the student taps on part of the free-body diagram in order to query the system for information about the objects drawn at that location; (6) a Recognition Succeeded Notification, which occurs whenever the student draws a stroke that Newton’s Pen is able to identify as valid input; (7) a Recognition Failed Notification, which occurs whenever the student draws a stroke that Newton’s Pen interprets as invalid.

One last form of output used by Newton’s Pen is an auxiliary booklet, which provides more in-depth explanations of how to use the system and descriptions of gen-

eral statics concepts. This auxiliary material (Appendix F) is occasionally referenced by Newton's Pen's textual messages.

4.2 The Problem-Solving Decomposition

The instructional design for Newton's Pen is similar to that of Newton's Tablet. This section describes how Newton's Pen guides student through the various problem-solving stages, with an emphasis on how this differs from the approach used in Newton's Tablet. (See Section 3.1 for a more in-depth discussion of the pedagogical motivation for the instructional design.)

To create a free-body diagram, the student must first define the boundary of the system to be analyzed. The student does this by using the pen to trace the boundary directly on the problem image. It is, of course, impossible for the student to drag the boundary to an empty work area. Thus, the system provides multiple workspaces, each with its own faint image of the problem (Figure 4.5). Each workspace is used to create a single free-body diagram . The problem images are faint to avoid cluttering the workspaces. If a boundary is recognized as defining a valid system, Newton's Pen prompts the student to write a name for the system in a textbox in the corner of the workspace. When Newton's Pen provides tutorial feedback, it is then able to refer to a free-body diagram by its name.

Next, the student must locate and classify all of the points of interaction on each body (Figure 4.6). Whereas Newton's Tablet requires the student to first locate all points of interaction and then classify them, Newton's Pen instead requires the student to locate, classify, and label one point of interaction before considering another. This approach enables

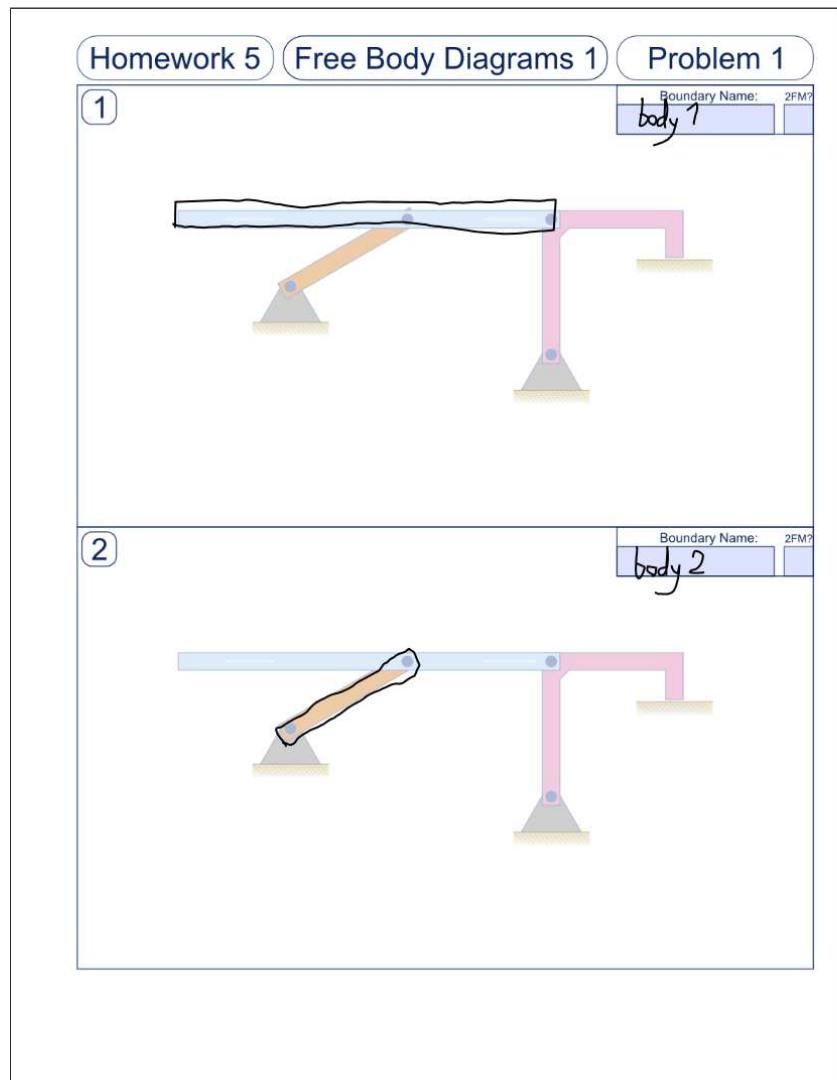


Figure 4.5: Identifying Boundaries.

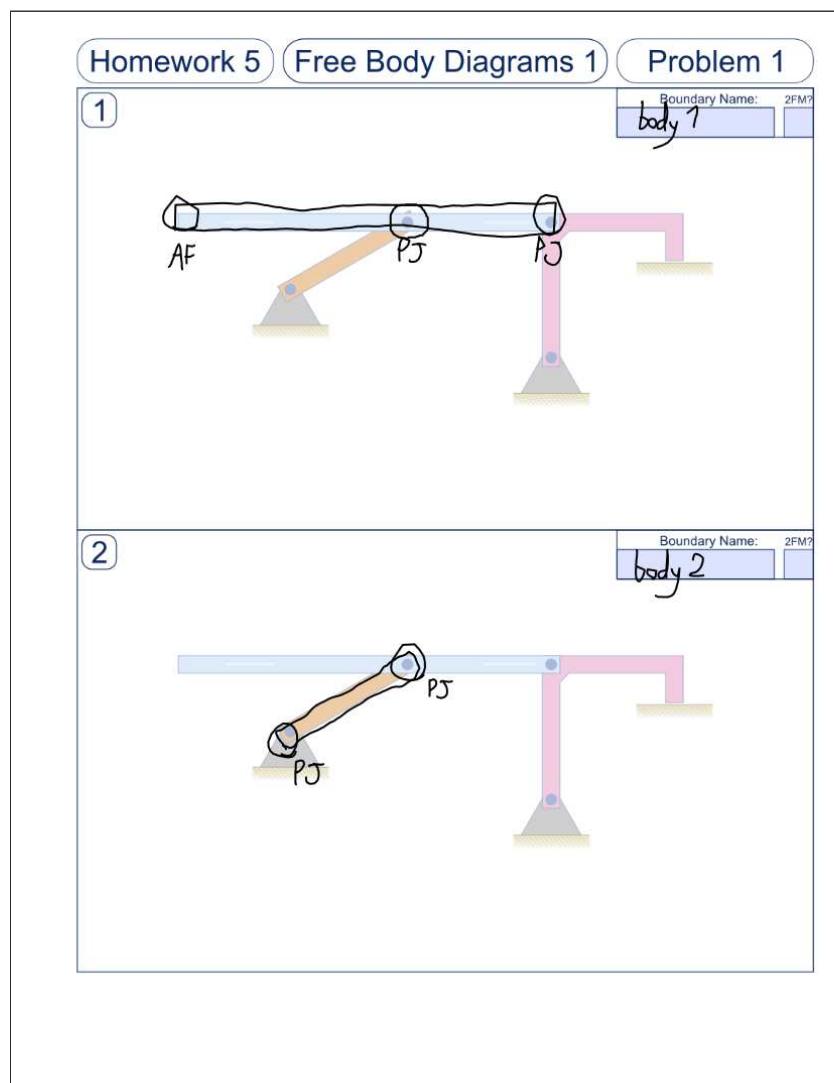


Figure 4.6: Identifying Points of Interaction.

Newton's Pen to provide feedback on errors related to a specific point of interaction without the need to specify its location.

In Newton's Pen, the student identifies the location of a point of interaction by circling it rather than by tapping on it. This helps the student to later remember where he/she located the points of interaction. The student then identifies the type of a point of interaction by tapping with the pen on the appropriate type button from the problem description page. After the student correctly circles and classifies a point of interaction, Newton's Pen requires the student to write a label near it. This label specifies the interaction type. For example, if a point of interaction is a pivot joint, the student would label it "PJ."

Next, Newton's Pen requires the student to identify all two-force members and Newton's-third-law pairs. As with Newton's Tablet, the student simply taps with the pen on a body or point of interaction to identify it as being a two-force member or part of a Newton's-third-law pair. However, after correctly identifying a two-force member or Newton's-third-law pair, Newton's Pen requires the student to label it. The student labels all points of interaction which are involved in a Newton's-third-law pair with "N3L," and labels all two-force members with a check mark in a special textbox in the corner of that body's workspace.

Finally, the student models each of the forces. To model a force, the student draws an arrow and then writes a label near it. For other labels, such as the body labels in Figure 4.6, no character recognition is needed because the system prompts the student to write specific text. Here, by contrast, the student can choose which label to assign to each force arrow. We initially used the handwriting recognizer built into the smartpen to interpret the

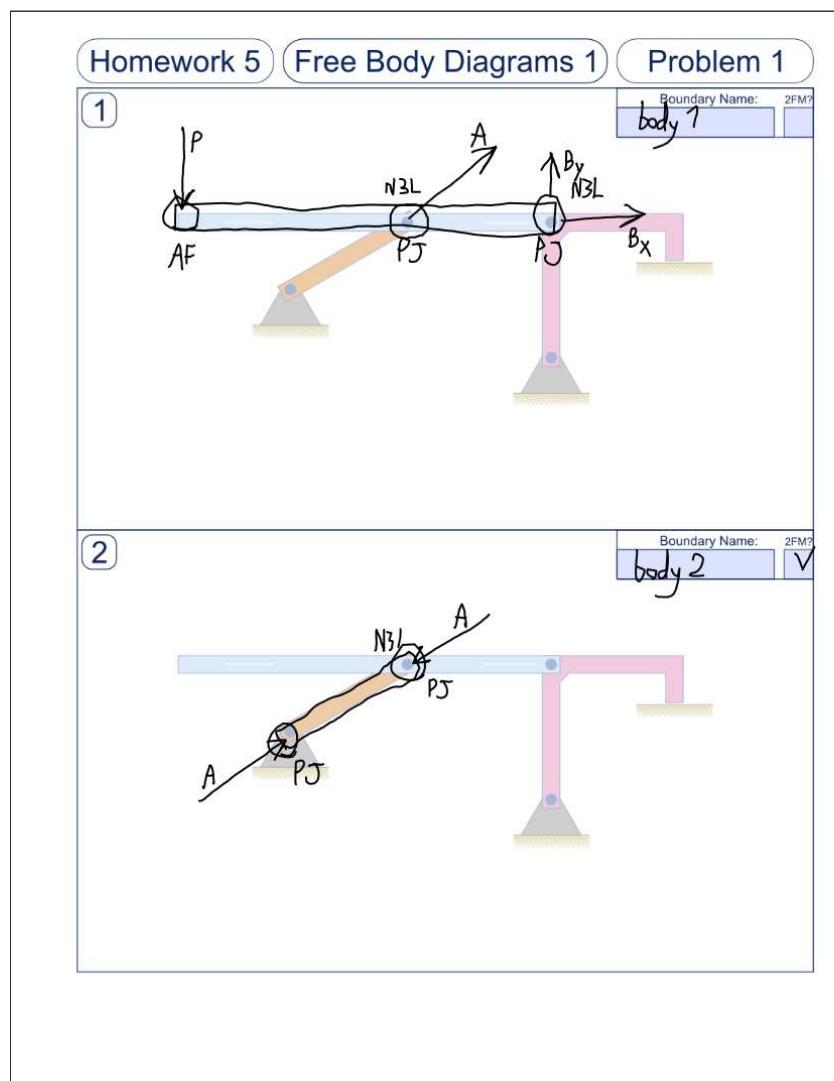


Figure 4.7: Identifying Forces.

force labels. However, this led to frequent memory errors (See Section 4.3.4). Thus, after the student writes a label, he/she uses a keyboard printed the problem description page to reenter it as text. The handwritten labels on the page are for the student's use, and the text is for the program's use.

Because it was necessary to eliminate the code which supports instruction on equilibrium equations, once the student has correctly drawn each of the force arrows, no more instruction is provided.

4.3 Challenges of Developing for the Livescribe Platform

The Livescribe smartpen environment affords the opportunity to create a tutorial system which operates with true pen-and-paper interaction. However, this platform also presents many pedagogical and technical challenges.

The most significant difference between this platform and the tablet platform is that the paper interface of the smartpen platform is visually static. A worksheet with relevant system images can be pre-printed over dot-patterned paper, but these worksheets cannot be changed at runtime, and the student's ink is indelible. Sections 4.3.3, 4.3.1, and 4.3.2 describe these issues. Section 4.3.4 then discusses many of the challenges in programming on the Livescribe smartpen platform.

4.3.1 Communicating Locations

One of the most difficult problems is informing the student of the location of an error on a free-body diagram. We considered a few solutions to this problem. For

example, Newton’s Pen could include a grid overlay on top of each workspace and refer to errors according to their nearest column and row. Alternatively, Newton’s Pen could include position labels at all possibly significant locations on the system diagram and refer to errors according to their nearest position label. Newton’s Pen could also attempt to provide feedback on the correctness of all ink immediately after it is drawn. This approach is effective because it avoids the need to specify the location of the error. However, it is not always possible to diagnose errors immediately after the ink is written. Sometimes errors cannot be detected until other parts of the problem have been solved. When possible, Newton’s Pen provides immediate feedback. When this is not possible, it uses labels to locate the errors. Some of the labels are pre-printed on the problem description, while others are written by the student during problem solving. For example, the problem description may mention something along the lines of “A tension force is applied to the bar at point A...”, in which case point A will be pre-labeled. For another example, when the student draws a force arrow and then gives it the label “F”, the system can also use this label to describe the location of a potential error.

In part to facilitate specifying the location of errors, Newton’s Pen asks the student to label all free-body diagrams, points of interaction, and forces immediately after they are drawn. Because Newton’s Pen asks the student to label each object immediately after it is drawn, there is no ambiguity about the associate between labels and objects. Nevertheless, the system does still check that the student’s label is indeed proximate enough to be associated with the given object.

4.3.2 Multi-Body Problems

The need for the student to work on multiple free-body diagrams for multi-body problems presents another problem for Newton's Pen. In ordinary paper-and-pencil solutions, the student simply draws the free-body diagrams separated from each other. However, with Newton's Tablet and Newton's Pen, it is important for the student to trace a boundary directly on top of the original problem image, so that the system can readily recognize the boundary. Newton's Tablet then allows the student to drag the boundary away to an open area of the workspace. However, Newton's Pen cannot change the ink on the page. Instead, the student is required to trace each new boundary in a separate workspace. This requires the use of additional free-body diagram worksheets for each multi-body problem.

4.3.3 Erroneous and Messy Ink

Another challenge with Newton's Pen is that the ink is indelible. Newton's Tablet can delete ink. For example, once an object has been recognized, Newton's Tablet removes the ink and replaces it with "beautified" shapes. It also removes ink when it is no longer relevant to the current problem-solving stage or when it is not recognized as valid input. Additionally, the student may erase ink. Newton's Pen, however, cannot remove or change ink.

In addition to resulting in a less cluttered, more legible free-body diagram, beautification ensures that the student instantly sees how the system has interpreted his/her work (Section 3.2.4). For example, if the student draws an arrow that is somewhat crooked or messy, then the system might believe the arrow to be drawn in one direction while the

student believes the arrow to be drawn in another. Because Newton's Pen extracts high-level features from the ink in order to represent an arrow internally, it is common for the student and the system to disagree when interpreting ink. Figure 4.8 demonstrates this phenomenon.

It is helpful for the tutorial system to be able to remove old ink that was relevant for an earlier stage. This prevents unnecessary confusion and allows the student to encounter fewer distractions while focusing only on the concepts that are addressed in the current stage. The Livescribe smartpen environment cannot provide this functionality.

Handling erroneous strokes is one of the toughest problems with Newton's Pen. The most obvious policy is for Newton's Pen to simply ignore invalid strokes and to then inform the student when a stroke is ignored. However, the student will inevitably forget exactly which strokes represent information that is still relevant. This confusion caused from not being able to remember which strokes are invalid can pose a substantial barrier to student learning.

Newton's Pen has two solutions to remedy some of the confusion caused from the

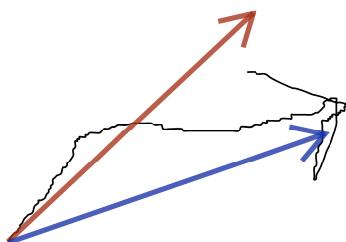


Figure 4.8: Discrepancy in Beliefs of Stroke Representations. The black stroke (middle) is an example of a hand-drawn force arrow. The red arrow (top) represents how Newton's Pen might interpret a stroke, while the blue arrow (bottom) represents what the student might have actually intended.

permanence of strokes. First, the student always has the option to abandon all work on a problem and to start over in a new workspace or on a new page. Second, Newton’s Pen provides the ability for the student to tap on any part of a free-body diagram at any time in order to query the system about what it believes is the current state represented by that part of the free-body diagram. An example of why this might be useful is if the student is unsure of whether a point-of-interaction circle he/she previously drew actually represented a valid or invalid location.

Newton’s Pen also has an Erase Mode (Section [B.2.1](#)). This allows the student to draw a stroke through any of the work for the current stage in order to inform the system to remove that work from consideration. An early version of Newton’s Pen automatically transitioned to Erase Mode whenever the student drew any strokes that were unsuccessfully recognized as valid input. The goal of this behavior was to ensure that the student had crossed out all invalid strokes in order to be less confused about a workspace that may have contained a lot of mistakes. However, this automatic transition to an unexpected mode made the system very difficult for students to use, and forcing the student to add extra ink—in the form of cross-outs—failed to make the workspace more legible afterward. In the final version of Newton’s Pen, invalid strokes are simply ignored, and the student is advised to cross them out if he/she wishes. For most stages, the system then only saves strokes that represent correct work, which means that the formal Erase Mode is not needed, because the student should not need to tell the system to remove correct work from consideration. A small exception is in the Force Drawing Stage (Section [B.1.5](#)), where Newton’s Pen does allow the student to draw arrows which may not actually be a part of the final solution.

4.3.4 Programming Limitations

The most significant challenge in developing for the Livescribe platform was the limited system memory. Because of memory constraints, Newton’s Pen does not provide instruction for constructing equilibrium equations. We actually implemented this part of the system, but it did not operate reliably within the available memory. We considered separating the system into two programs, one for free-body diagrams and one for equilibrium equations. However, it was not possible to create the second system, because it would still need much of the data from the free-body diagram system.

Newton’s Pen cannot load problem encodings from the same XML files as Newton’s Tablet. This would require the system to hold both the XML document object model and the corresponding class objects in memory simultaneously. So we devised a new problem encoding format which directly hard-coded a problem’s data into a Java class file. This significantly reduced the load on system memory. However, these class files are always present in memory, whereas the XML files were only present in memory while initially loading the problem. In the final version of Newton’s Pen, we provided a separate version of the program for each homework assignment, containing only the problems for that assignment.

Newton’s Pen is unable to hold raw stroke data in system memory—raw stroke data consist of many timestamped x/y-coordinates. Instead, the system extracts key features from each stroke to save in memory, and then writes out the raw stroke data to a file.

Another consequence of the smartpen’s limited memory is the inability to utilize the built-in handwriting-recognition engine. This functionality, provided by the operating system, must be disabled in Newton’s Pen, because it consumes too much system memory.

This fundamentally changes how Newton's Pen has the student enter labels for forces. The student still writes the labels so that he/she can remember them later. However, Newton's Pen cannot understand these labels. Thus the student must re-enter them using a keyboard printed on the problem description worksheet.

4.4 Tutorial Feedback

Newton's Tablet presents conceptual errors as highlights directly over the location of the error, but this is not possible in Newton's Pen. Instead, Newton's Pen presents conceptual errors as a textual list on the pen's display. The student can use the directional buttons on the Problem-Description worksheet to navigate vertically through the different errors, and can navigate horizontally to view more-detailed information for a given error. Each error consists of at least three separate messages: the first describes the location of the error in the free-body diagram in reference to body names, point-of-interaction names, and force labels (Section 4.3.1); the second describes the type of the error and references the auxiliary booklet for an explanation of this type of error; and any following messages offer progressively more detailed explanations of the conceptual error.

Here is an example of the message sequence for a Missing Applied Force error in Newton's Pen:

1. Error @ Point A
2. [Section II.C] Missing applied force
3. There is a force missing

4. The missing force was given in the problem statement
5. All external forces applied to the system must be drawn on the free-body diagram
6. See Ref. Section II.C

Appendix C presents a discussion of each of the conceptual errors evaluated in each stage of Newton's Tablet. The types of conceptual errors evaluated in Newton's Pen are very similar.

Chapter 5

User Studies

We conducted two separate user studies using the Newton’s Tablet and Newton’s Pen systems in order to assess their educational value. This chapter discusses the results of these studies.

Section 5.1 describes the use of these systems as required coursework in an undergraduate course at the University of California, Riverside. In this first study, Newton’s Tablet showed more-promising results than Newton’s Pen, so we conducted a follow-up study focused exclusively on Newton’s Tablet. This second user study is described in Section 5.2.

While Newton’s Tablet and Newton’s Pen are used, they generate detailed log files. An entry is created in a log file for every user event and for other pieces of significant system information. These log files contain invaluable data for assessing the educational value of these systems. These data can also offer insights to other aspects of our systems, such as which user-interface features were more effective than others and which statics concepts are

the most difficult for students to tackle. A comprehensive analysis of the log files generated in our two user studies is forthcoming in Garcia [24].

5.1 Use in an Introduction to Statics Course

5.1.1 Experimental Setup

In our first user study, we used both the Newton's Tablet and the Newton's Pen systems as a required component of two separate homework assignments in a sophomore-level introductory statics course (ME10) at the University of California, Riverside in the Winter of 2013. In this course, we only deployed the free-body diagram portions of the systems. We did this because the pen version was unable to support the equation portion, and deploying the equation portion on only the tablet version would lessen our ability to make an unbiased comparison of the two systems. Problems one through six were used in the first assignment, while problems seven through nine were used in the second assignment. The teaching assistants used problem zero for demonstrating how to use the systems. Section D.2 provides a description of these problems.

5.1.2 Student Performance

We gave the Newton's Tablet system to a group of 26 students and the Newton's Pen system to a different group of 28 students. Of these students, 20 from the Newton's Tablet group and 23 from the Newton's Pen group reported using the system at least once during one of the two assignments.

The data sets collected from the ME10 user study consist of: extensive log files from each student using either system, the actual grades assigned to each student throughout the course, and the answers students gave in an attitudinal survey at the end of the course. A comprehensive analysis of each of these data sets is forthcoming in Garcia [24]. This later analysis will evaluate the performance of the students and the educational value of Newton's Tablet and Newton's Pen.

5.1.3 Attitudinal Survey

At the end of the ME10 course, the students completed surveys that asked them to describe various aspects of the systems on a scale from one to five. The questions given to the students in the Newton's Tablet group were slightly different from those given to the Newton's Pen group. Tables 5.1, 5.2, and 5.3 describe the attitudinal survey used in the statics course for students who used Newton's Pen. Tables 5.4, 5.5, and 5.6 describe the attitudinal survey used in the statics course for students who used Newton's Tablet. Each of these tables lists the questions, their average responses, and their standard deviations. As a pre-processing step, we first removed the data for all students who reported not using the tutorial software.

Students responded positively to almost all aspects of both tutorial systems. In general, they found both systems to be helpful tools for learning. The only negative reactions indicate that the Newton's Pen system is somewhat frustrating and difficult for students to use. One particularly negative exception is the student response to the question "How similar is this system to working with paper and pencil?" from the Newton's Pen survey

Survey Question	Scale (1 — 5)	Mean	Std Dev
How often did you use the supplemental reference materials (user guides, quick reference sheet, videos, etc.) posted on iLearn?	Never — Very often	2.00	1.10
How useful were the supplemental reference materials for learning to use the system?	Very useless — Very useful	3.26	1.07
How easy was it to learn to use the program?	Very difficult — Very easy	3.83	1.05
Did the system provide clear guidance about what you needed do during each stage of problem solving?	Very unclear — Very clear	3.04	1.12
Were the problem-solving stages ordered in an organized, logical manner?	Very unorganized & illogical — Very organized & logical	4.00	0.51
How helpful was the progression of the problem-solving stages in guiding you through the process of creating a free-body diagram?	Very unhelpful — Very helpful	3.74	1.07
I felt that the step-by-step process that the tutorial system walked me through helped me understand fundamental statics concepts.	Strongly disagree — Strongly agree	3.48	1.02
How easy was it to trace system boundaries?	Very difficult — Very easy	3.74	1.03
How easy was it to draw force arrows?	Very difficult — Very easy	3.78	0.59
How easy was it to label force arrows?	Very difficult — Very easy	3.70	1.16
How easy was it to select and label the interaction types for each POI?	Very difficult — Very easy	3.83	1.01
How easy was it to use the paper keyboard and other buttons on the problem description page?	Very difficult — Very easy	3.52	1.06
How helpful were the pen sound effects as a form of feedback?	Very difficult — Very easy	3.30	1.00
How easy was it to read and understand the messages displayed on the pen LCD display?	Very difficult — Very easy	3.65	0.96
Is the program's tutorial feedback (action prompts & error messages) presented in a useable form?	Very unusable — Very useable	3.22	1.14

Table 5.1: Newton's Pen Attitudinal Survey Results from ME 10 (Part 1). All questions were answered on a scale from one to five, with one being the most negative response and five being the most positive.

Survey Question	Scale (1 — 5)	Mean	Std Dev
How often did you read through all of the relevant messages for a given error?	Never — Always	3.91	1.06
How helpful were the error messages in guiding you to obtain the correct free-body diagram?	Not at all helpful — Very helpful	3.26	1.07
How easy was it to navigate through the error messages?	Very difficult — Very easy	2.87	1.23
How easy was it to locate the errors described in the error messages?	Very difficult — Very easy	2.65	1.09
How often did you find it more convenient to identify the errors yourself instead of reading the error messages?	Never — Always	3.04	1.04
After completing all of the problem-solving stages, what was the final appearance of the worksheet?	Very cluttered — Very organized	2.26	0.94
What is your overall opinion of the usability of this program?	Very unusable — Very useable	3.57	0.92
How useful is the program for learning to select suitable system boundaries?	Very useless — Very useful	3.70	0.75
How useful is the program for learning to identify the forces on free-body diagrams?	Very useless — Very useful	3.74	0.61
Was the process of explicitly identifying Points of Interaction (POIs) helpful to your learning?	Very unhelpful — Very helpful	3.87	0.74
Was the process of explicitly identifying the types of interactions at each POI helpful to your learning?	Very unhelpful — Very helpful	3.83	0.82
Was the process of explicitly identifying Two-Force Members (TFMs) helpful to your learning?	Very unhelpful — Very helpful	4.04	0.70
Was the process of explicitly identifying Newton's Third-Law Pairs (N3L) helpful to your learning?	Very unhelpful — Very helpful	3.83	0.76
Using the tutorial system increased my confidence in my abilities to correctly construct free-body diagrams.	Strongly disagree — Strongly agree	3.57	1.01
Using the tutorial system increased my motivation to complete the homework assignment.	Strongly disagree — Strongly agree	2.91	0.93
Using the tutorial system helped me better understand subsequent homework problems.	Strongly disagree — Strongly agree	3.13	0.90

Table 5.2: Newton's Pen Attitudinal Survey Results from ME 10 (Part 2). All questions were answered on a scale from one to five, with one being the most negative response and five being the most positive.

Survey Question	Scale (1 — 5)	Mean	Std Dev
How similar is this system to working with paper and pencil?	Very dissimilar — Very similar	2.70	1.30
Do you find this kind of pen-based user interface preferable to a traditional computer interface based on a keyboard and mouse?	Strongly prefer traditional interface — Strongly prefer pen-based interface	3.17	1.13
What is your overall opinion of the usefulness of this program for learning to solve equilibrium problems?	Very useless — Very useful	3.78	0.78
If this system had been available for other assignments in the class, how likely would you have been to use it?	Very unlikely — Very likely	3.40	1.34
How likely would you be to use this kind of instructional system for other subjects if such systems were available to you?	Very unlikely — Very likely	3.35	1.31
Using the tutorial system was...	Very terrible — Very wonderful	3.05	0.72
Using the tutorial system was...	Very difficult — Very easy	3.05	1.05
Using the tutorial system was...	Very frustrating — Very satisfying	2.24	0.81
Using the tutorial system was...	Very dull — Very stimulating	3.09	0.80

Table 5.3: Newton's Pen Attitudinal Survey Results from ME 10 (Part 3). All questions were answered on a scale from one to five, with one being the most negative response and five being the most positive.

Survey Question	Scale (1 — 5)	Mean	Std Dev
How often did you use the supplemental reference materials (user guides, quick reference sheet, videos, etc.) posted on iLearn?	Never — Very often	1.68	0.98
How useful were the supplemental reference materials for learning to use the system?	Not at all useful — Very useful	2.68	0.98
How easy was it to learn to use the program?	Very difficult — Very easy	4.11	0.79
Did the system provide clear guidance about what you needed do during each stage of problem solving?	Very unclear — Very clear	3.60	0.73
Were the problem-solving stages ordered in an organized, logical manner?	Very unorganized & illogical — Very organized & logical	3.75	0.94
How helpful was the progression of the problem-solving stages in guiding you through the process of creating a free-body diagram?	Very unhelpful — Very helpful	3.75	0.83
I felt that the step-by-step process that the tutorial system walked me through helped me understand fundamental statics concepts.	Strongly disagree — Strongly agree	3.65	0.96
How easy was it to trace and drag system boundaries?	Very difficult — Very easy	3.70	0.84
How easy was it to draw force arrows?	Very difficult — Very easy	3.35	0.79
How easy was it to label force arrows?	Very difficult — Very easy	3.95	0.92
How easy was it to select the interaction types for each POI?	Very difficult — Very easy	3.55	1.12
How easy was it to understand and use the menu buttons?	Very difficult — Very easy	3.7	0.95
How easy was it to read and understand the messages displayed in the top status bar?	Very difficult — Very easy	3.8	0.81
Is the program's tutorial feedback (action prompts, error messages, and general help) presented in a useable form?	Very unusable — Very useable	3.6	0.73
How often did you read through all of the relevant messages for a given error?	Never — Always	3.55	1.12

Table 5.4: Newton's Tablet Attitudinal Survey Results from ME 10 (Part 1). All questions were answered on a scale from one to five, with one being the most negative response and five being the most positive.

Survey Question	Scale (1 — 5)	Mean	Std Dev
How helpful were the error messages in guiding you to obtain the correct free-body diagram?	Not at all helpful — Very helpful	3.5	0.97
How easy was it to navigate through the error messages?	Very difficult — Very easy	2.8	1.03
How easy was it to locate the errors described in the error messages?	Very difficult — Very easy	3.25	1.22
How often did you find it more convenient to look only for the error location highlights and identify the errors yourself instead of reading the error messages?	Never — Always	3.26	1.02
How easy was it to use the traditional computer interface (mouse & keyboard)?	Very difficult — Very easy	3.5	1.24
What is your overall opinion of the usability of this program?	Very unusable — Very useable	3.6	0.8
How useful is the program for learning to select suitable system boundaries?	Very useless — Very useful	4.1	0.77
How useful is the program for learning to identify the forces on free-body diagrams?	Very useless — Very useful	4.1	0.89
Was the process of explicitly identifying Points of Interaction (POIs) helpful to your learning?	Very unhelpful — Very helpful	4.00	0.89
Was the process of explicitly identifying the types of interactions at each POI helpful to your learning?	Very unhelpful — Very helpful	3.85	1.06
Was the process of explicitly identifying Two-Force Members (TFMs) helpful to your learning?	Very unhelpful — Very helpful	3.7	1.00
Was the process of explicitly identifying Newton's Third-Law Pairs (N3L) helpful to your learning?	Very unhelpful — Very helpful	3.7	0.84
Using the tutorial system increased my confidence in my abilities to correctly construct free-body diagrams.	Strongly disagree — Strongly agree	3.7	0.84
Using the tutorial system increased my motivation to complete the homework assignment.	Strongly disagree — Strongly agree	3.37	0.98
Using the tutorial system helped me better understand subsequent homework problems.	Strongly disagree — Strongly agree	3.70	1.00
What is your overall opinion of the usefulness of this program for learning to solve equilibrium problems?	Very useless — Very useful	3.85	1.01

Table 5.5: Newton's Tablet Attitudinal Survey Results from ME 10 (Part 2). All questions were answered on a scale from one to five, with one being the most negative response and five being the most positive.

Survey Question	Scale (1 — 5)	Mean	Std Dev
If this system had been available for other assignments in the class, how likely would you have been to use it?	Very unlikely — Very likely	3.35	1.11
How likely would you be to use this kind of instructional system for other subjects if such systems were available to you?	Very unlikely — Very likely	3.75	0.70
Using the tutorial system was...	Very terrible — Very wonderful	3.50	0.67
Using the tutorial system was...	Very difficult — Very easy	3.30	1.00
Using the tutorial system was...	Very frustrating — Very satisfying	2.95	0.74
Using the tutorial system was...	Very dull — Very stimulating	3.33	0.75

Table 5.6: Newton’s Tablet Attitudinal Survey Results from ME 10 (Part 3). All questions were answered on a scale from one to five, with one being the most negative response and five being the most positive.

(Table 5.3). Students felt that the Newton’s Pen system was not a natural environment for statics and was very different from their typical experiences with pen and paper. In contrast, students felt that the Newton’s Tablet system was natural and easy to use. This result contradicts the original hypothesis that the Newton’s Pen system would relate more directly to the traditional pen-and-paper interface and would therefore yield greater learning gains. However, the rest of the results are encouraging in that they indicate that these systems are effective tools for learning.

5.2 Supplementary User Study

5.2.1 Experimental Setup

In our second user study, we had students use Newton’s Tablet in an additional 90-minute user study conducted at the University of California, Riverside in the spring of 2013. An attempt was made to target the study toward students who had already taken the University of California, Riverside course Introduction to Mechanical Engineering (ME2) but who had not yet taken ME10. Ideally, these students were primed with enough knowledge to understand the tutorial system, but did not already have a complete knowledge of statics. That is, there was potential for these students to learn from Newton’s Tablet. We compensated the students in this user study with a \$15 gift certificate for on-campus dining.

The user study consisted of five parts: first, the student completed a single pretest problem; the student was then guided by a proctor through the process of using Newton’s Tablet on problem zero; next, the student independently completed problems one and three with Newton’s Tablet; lastly, the student completed a single posttest problem. We used both the free-body diagram portion and the equilibrium-equation portion of Newton’s Tablet in this user study. We included the pre- and posttests for the purpose of measuring the participants’ learning gains. Figure 5.1 shows the problem description given for the pre- and posttest. The given problem description was identical for both tests. Using the same problem for both the pre- and posttest allowed us to quantitatively measure the learning gains developed while using the Newton’s Tablet system. We evaluated student

performance on the pre- and posttest using scales from zero to five, and we scored the free-body-diagram and equilibrium-equation portions separately. Table 5.7 shows the rubric we used to score the free-body-diagram portions, and Table 5.8 shows the rubric we used to score the equilibrium-equation portions.

We chose not to use Newton's Pen as part of this second study. This was largely because of the study's small size, but also because results from the earlier ME10 study had hinted at the tablet system being a more effective learning tool.

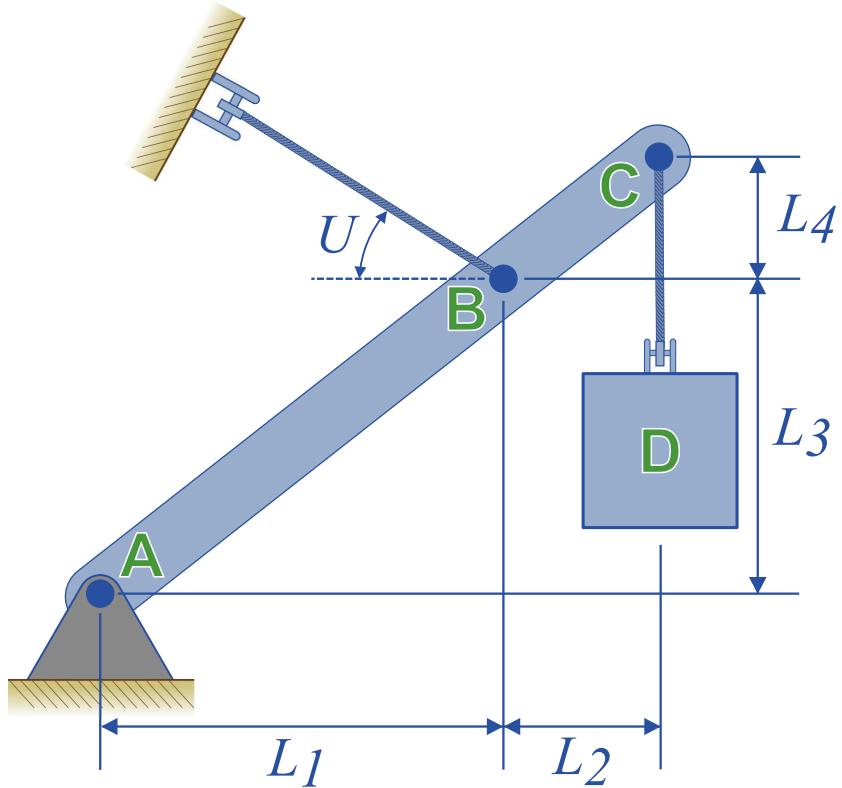


Figure 5.1: The Pre- and Posttest Problem Description. If the mass of crate D is 50 kg, what is the magnitude of the tension in the rope which connects to the bar at point B? Assume that the bar itself is massless and that the system shown is in static equilibrium. $g = 9.81 \text{ m/s}^2$; $U = 30^\circ$; $L_1 = 3 \text{ m}$; $L_2 = 1 \text{ m}$; $L_3 = 2 \text{ m}$; $L_4 = 0.75 \text{ m}$.

Score	Criteria
0	Nothing drawn
1	No understanding, but something drawn
2	Minimal understanding; e.g., the diagram was missing many forces or the diagram did not isolate the relevant system boundary
3	Some understanding; e.g., the body and some forces were drawn, but there were significant errors or omissions
4	Good understanding; i.e., the diagram was mostly correct, but there were some minor errors
5	Correct solution; the diagram was complete and correct

Table 5.7: User Study Pre- and Posttest Free-Body Diagram Scoring Rubric.

Score	Criteria
0	Nothing written
1	No understanding, but something written
2	Minimal understanding; i.e., the equations demonstrated some notion of equilibrium but were largely incorrect
3	Some understanding; i.e., the equations demonstrated a clear notion of equilibrium and were partially correct
4	Good understanding; i.e., the equations were mostly correct, but there were some minor errors
5	Correct solution; the equations were complete and correct

Table 5.8: User Study Pre- and Posttest Equation Scoring Rubric.

5.2.2 Student Performance

Ten students participated in the supplementary user study. The free-body diagrams and equations from the pre- and posttests from these students were evaluated using the scales shown in Tables 5.7 and 5.8. Table 5.9 and Figure 5.2 show the average pre- and posttest scores from all participants using these grading rubrics. However, three of the user-study participants attained perfect scores on the pretest, and therefore Newton's Tablet could not offer any learning gains for these students according to this rubric. For this reason, Table 5.10 and Figure 5.3 show the average pre- and posttest scores after removing the students who attained perfect scores on the pretest. All students who attained perfect scores also attained perfect scores on the posttest. Table 5.11 shows the average time spent on the pre- and posttests.

A comprehensive analysis of the extensive log files collected during this user study is forthcoming in Garcia [24].

These results from the pre- and posttests show that Newton's Tablet imparts

		Mean	Std Dev
Free-Body Diagram	Pretest	2.4	1.80
	Posttest	4.7	0.64
Equations	Pretest	3.0	1.41
	Posttest	4.7	0.64

Table 5.9: User Study Pre- and Posttest Results (All Students). This table shows the pre- and posttest results from all students in the user study. Note: pre-processing has not been performed on these data to remove the three students who attained perfect scores on the pretest. Figure 5.2 displays these data in graphical form.

		Mean	Std Dev
Free-Body Diagram	Pretest	1.29	0.70
	Posttest	4.57	0.73
Equations	Pretest	2.14	0.64
	Posttest	4.57	0.73

Table 5.10: User Study Pre- and Posttest Results (Pre-processed). This table shows the pre- and posttest results after pre-processing the data by removing the three students who attained perfect scores on the pretest. Figure 5.3 displays these data in graphical form.

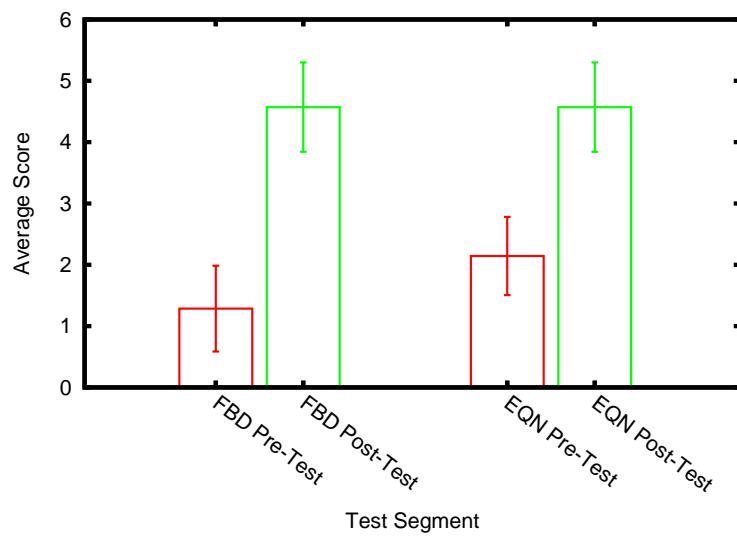


Figure 5.2: User Study Pre- and Posttest Results (All Students). This graph shows the pre- and posttest results from all students in the user study. Note: pre-processing has not been performed on these data to remove the three students who attained perfect scores on the pretest. Table 5.9 displays these data in graphical form.

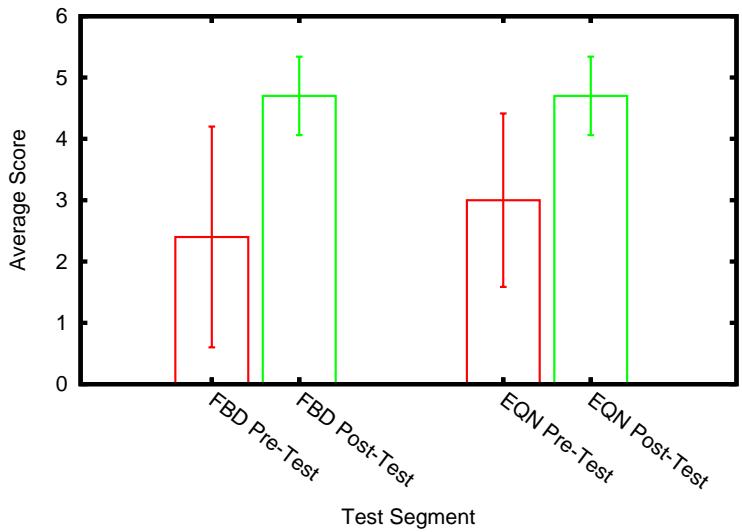


Figure 5.3: User Study Pre- and Posttest Results (Pre-Processed). This graph shows the pre- and posttest results after pre-processing the data by removing the three students who attained perfect scores on the pretest. Table 5.10 displays these data in numerical form.

Pretest Time (minutes)	Posttest Time (minutes)	Reduction (minutes)	Reduction (%)
9.32	8.27	1.05	11.3

Table 5.11: User Study Pre- and Posttest Times. The average solution time pre- and posttest problems, along with the reduction of the average time from pre- to posttest.

significant learning gains to students using the system. Students score higher on the same material after using the tutorial system (Figure 5.2). The improvements in performance are even more dramatic when we remove from the analysis the students who attained perfect scores on the pretest (Figure 5.3). Furthermore, students take slightly less time to complete the same material after using the tutorial system (Table 5.11).

5.2.3 Attitudinal Survey

At the end of the supplementary user study, the students completed a survey which asked them to describe various aspects of the system on a scale from one to five. The questions in this survey are slightly different from those given in the statics-course user studies (Section 5.1.3). Tables 5.12, 5.13, and 5.14 describe the attitudinal survey used in the supplementary user study. Each of these tables list the questions, their average responses, and their standard deviations.

Students responded positively to almost all aspects of Newton’s Tablet. None of the survey questions produced negative reactions. The lowest average score for a question was 3.4, and this question—“How often did you find it more convenient to look only for the error location highlights and identify the errors yourself instead of reading the error messages?”—related more to how the students used the system and less to how well the system performed. This means that the students found the Newton’s Tablet system to be natural, easy to use, and helpful for learning in all aspects we tested.

Survey Question	Scale (1 — 5)	Mean	Std Dev
How easy was it to learn to use the program?	Very difficult — Very easy	4.10	0.94
Did the system provide clear guidance about what you needed do during each stage of problem solving?	Very unclear — Very clear	4.40	0.49
Were the problem-solving stages ordered in an organized, logical manner?	Very unorganized & illogical — Very organized & logical	4.70	0.46
How helpful was the progression of the problem-solving stages in guiding you through the process of creating a free-body diagram?	Very unhelpful — Very helpful	4.00	0.89
I felt that the step-by-step process that the tutorial system walked me through helped me understand fundamental statics concepts.	Strongly disagree — Strongly agree	4.00	0.77
How easy was it to trace and drag system boundaries?	Very difficult — Very easy	4.00	1.10
How easy was it to draw force arrows?	Very difficult — Very easy	4.60	0.49
How easy was it to label force arrows?	Very difficult — Very easy	4.60	0.66
How easy was it to draw moment arm brackets?	Very difficult — Very easy	4.40	0.49
How easy was it to select the interaction types for each POI?	Very difficult — Very easy	4.60	0.66
How easy was it to understand and use the menu buttons?	Very difficult — Very easy	4.20	0.98
How easy was it to enter symbolic equations?	Very difficult — Very easy	4.20	0.60
How easy was it to enter expanded equations?	Very difficult — Very easy	4.00	0.77
How easy was it to read and understand the messages displayed in the top status bar?	Very difficult — Very easy	4.20	1.08
Is the program's tutorial feedback (action prompts, error messages, and general help) presented in a useable form?	Very unusable — Very useable	4.20	0.87

Table 5.12: Newton's Tablet Attitudinal Survey Results from User Study (Part 1). All questions were answered on a scale from one to five, with one being the most negative response and five being the most positive.

Survey Question	Scale (1 — 5)	Mean	Std Dev
How often did you read through all of the relevant messages for a given error?	Never — Always	3.80	0.87
How helpful were the error messages in guiding you to obtain the correct free-body diagram?	Not at all helpful — Very helpful	3.70	1.19
How easy was it to navigate through the error messages?	Very difficult — Very easy	4.20	0.60
How easy was it to locate the errors described in the error messages?	Very difficult — Very easy	4.00	0.77
How often did you find it more convenient to look only for the error location highlights and identify the errors yourself instead of reading the error messages?	Never — Always	3.40	0.66
How easy was it to use the tablet and keyboard interface?	Very difficult — Very easy	4.00	1.10
What is your overall opinion of the usability of this program?	Very unusable — Very useable	4.40	0.66
How useful is the program for learning to select suitable system boundaries?	Very useless — Very useful	4.50	0.50
How useful is the program for learning to identify the forces on free-body diagrams?	Very useless — Very useful	4.80	0.40
Was the process of explicitly identifying Points of Interaction (POIs) helpful to your learning?	Very unhelpful — Very helpful	4.20	0.87
Was the process of explicitly identifying the types of interactions at each POI helpful to your learning?	Very unhelpful — Very helpful	4.30	1.00
Using the tutorial system increased my confidence in my abilities to correctly construct free-body diagrams.	Strongly disagree — Strongly agree	4.30	0.64
What is your overall opinion of the usefulness of this program for learning to solve equilibrium problems?	Very useless — Very useful	4.30	0.90
How likely would you be to use this kind of instructional system for other subjects if such systems were available to you?	Very unlikely — Very likely	3.90	1.22

Table 5.13: Newton’s Tablet Attitudinal Survey Results from User Study (Part 2). All questions were answered on a scale from one to five, with one being the most negative response and five being the most positive.

Survey Question	Scale (1 — 5)	Mean	Std Dev
Using the tutorial system was...	Very terrible — Very wonderful	4.10	0.70
Using the tutorial system was...	Very difficult — Very easy	4.00	0.77
Using the tutorial system was...	Very frustrating — Very satisfying	3.70	0.78
Using the tutorial system was...	Very dull — Very stimulating	3.70	1.00

Table 5.14: Newton’s Tablet Attitudinal Survey Results from User Study (Part 3). All questions were answered on a scale from one to five, with one being the most negative response and five being the most positive.

Chapter 6

Conclusion

This dissertation has presented two versions of an intelligent pen-based tutoring system for the construction of static free-body diagrams and the derivation of their corresponding equilibrium equations. One version of this system runs on Livescribe smartpens, while the other runs on Windows tablet PCs. These systems are designed to decompose the traditional problem-solving process into fine-grained sub-steps that require the student to explicitly identify the fundamental concepts involved in the problem. This decomposition allows these systems to better model student cognition and, therefore, also provide more-targeted didactic feedback. We also designed these systems around the use of natural, pen-based interfaces, which emulate the traditional pen-and-paper interface used for solving statics problems.

The novel contributions of this research consist of: (1) a robust exploration of the efficacy of the Livescribe smartpen platform as an ITS for the discipline of statics; (2) a pedagogical technique which requires the student to explicitly identify fundamental concepts in

isolation for the purpose of better scaffolding student learning; (3) a location-based, sequential error-feedback system which facilitates the creation of cognitive connections between the student's actual work and the underlying statics concepts.

We used these two ITSs in two separate user studies. The first user study took place as a mandatory component in an introductory statics course at the University of California, Riverside. The second user study was distinct from the first and took place the following quarter. The results from these experiments indicate that both systems are valuable tools for student learning. Pre- and posttest results from the second user study show significant learning gains for students who used Newton's Tablet to work through a few problems. Attitudinal surveys from both studies suggest that the instructional model is effective in both Newton's Tablet and in Newton's Pen. These surveys also indicate that the user-interface design in Newton's Tablet is effective. Unfortunately, the survey results show that students did not find Newton's Pen to be as intuitive and easy to use as Newton's Tablet.

Many promising avenues exist for future work with this pen-based statics ITS research. The current systems do not support problems that depend on flexible members. Accounting for such members would greatly increase the logical complexity of checking student work, but a future system might tackle this difficulty. A future system could also allow for additional practice on the parts of the problem-solving process that prove especially difficult for the student. In particular, extra practice identifying moment arms and deriving moment-arm equilibrium equations would be helpful for most students. The current systems do not know when the student has derived a sufficient number of equilibrium equations to

solve for the unknowns given in the problem description. It could be useful for a future system to track the student’s progress toward solving for these unknowns. In addition, further user studies could be performed in order to develop a better understanding of which aspects of the system contribute the most to student learning and how this system compares to other tutorial techniques.

The pen-and-paper interface of Newton’s Pen is even more natural to the domain of statics than the stylus-and-tablet interface employed by Newton’s Tablet. Accordingly, the initial hypothesis of this research was that—according to cognitive load theory[52]—the more-natural Livescribe smartpen would allow for greater student learning gains from this tutorial system. However, results from using these systems suggest that the Newton’s Pen interface may be too limited to function as effectively as a tutoring device as Newton’s Tablet. This is primarily due to the indelible nature of ink and the lack of a large dynamic display with the Newton’s Pen interface. Because Newton’s Pen does not utilize a dynamic display for rendering ink, student mistakes cannot be removed and tutoring feedback cannot be shown spatially—it can only be explained via text. An ideal tutorial interface needs to be both natural and powerful enough to provide immediately understandable feedback within the context of the student’s current work. This research has found that the Livescribe smartpen platform does not meet the latter goal, and this significantly impairs its effectiveness as an ITS. For this reason, it seems unlikely that further work with ITSs for statics on the current Livescribe smartpen platform will yield fruitful pedagogical results. Perhaps more success could be had with the Livescribe smartpen platform within a domain such as algebra, which presents fewer of the spatial challenges inherent with free-body dia-

grams in statics. Nevertheless, results from Newton's Tablet have proven quite promising, and we intend to perform further research using this educational platform.

Bibliography

- [1] DG Alciatore and Rick Miranda. A winding number and point-in-polygon algorithm. Technical report, Technical report, Colorado State University, 1995.
- [2] Vincent Aleven. Rule-based cognitive modeling for intelligent tutoring systems. In Roger Nkambou, Jacqueline Bourdeau, and Riichiro Mizoguchi, editors, *Advances in Intelligent Tutoring Systems*, volume 308 of *Studies in Computational Intelligence*, pages 33–62. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-14362-5.
- [3] Vincent Aleven, Bruce M. McLaren, Jonathan Sewall, and Kenneth R. Koedinger. A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education*, 19(2):105–154, 2009.
- [4] Christine Alvarado and Randall Davis. Resolving ambiguities to create a natural computer-based sketching environment. In *ACM SIGGRAPH 2006 Courses*, New York, NY, USA, 2006. ACM. ISBN 1-59593-364-6.
- [5] John R. Anderson. *The Architecture of Cognition*. Harvard University Press, 1983. ISBN 0674044258.

- [6] R. Anderson, L. McDowell, and B. Simon. Use of classroom presenter in engineering courses. In *Frontiers in Education, 2005. FIE '05. Proceedings 35th Annual Conference*, 2005.
- [7] Lisa Anthony, Jie Yang, and Kenneth R. Koedinger. Evaluation of multimodal input for entering mathematical equations on the computer. In *CHI '05 Extended Abstracts on Human Factors in Computing Systems*, pages 1184–1187, New York, NY, USA, 2005. ACM. ISBN 1-59593-002-7.
- [8] Lisa Anthony, Jie Yang, and Kenneth R. Koedinger. A paradigm for handwriting-based intelligent tutors. *International Journal of Human-Computer Studies*, 70(11):866–887, 2012. ISSN 1071-5819.
- [9] D. Arnow and O. Barshay. Webtoteach: an interactive focused programming exercise system. In *Frontiers in Education Conference, 1999. FIE '99. 29th Annual*, volume 1, 1999.
- [10] Olufunmilola Atilola, Martin Field, Erin McTigue, Tracy Hammond, and Julie Linsey. Mechanix: A sketch recognition truss tutoring system. In *ASME 2011 International Design Engineering Technical Conferences & Computers and Information Engineering Conference*, 2011.
- [11] Robert K. Atkinson, Sharon J. Derry, Alexander Renkl, and Donald Wortham. Learning from examples: Instructional principles from the worked examples research. *Review of Educational Research*, 70:181–214, 2000.

- [12] Ferdinand Pierre Beer, Elwood Russell Johnston, David F. Mazurek, and Elliot Eisenberg. *Vector Mechanics for Engineers: Statics*. McGraw-Hill Science/Engineering/Math, 9th edition, 2009.
- [13] William Billingsley, Peter Robinson, Mark Ashdown, and Chris Hanson. Intelligent tutoring and supervised problem solving in the browser. In *Proceedings of ICWI*, pages 806–810, 2004.
- [14] Brian P. Butz, M. Duarte, and S. M. Miller. An intelligent tutoring system for circuit analysis. *Education, IEEE Transactions on*, 49(2):216–223, 2006. ISSN 0018-9359.
- [15] Salman Cheema and Joseph LaViola. Physicsbook: a sketch-based interface for animating physics diagrams. In *Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces*, pages 51–60, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1048-2.
- [16] Salman Cheema and Jr. LaViola, Joseph J. Applying mathematical sketching to sketch-based physics tutoring software. In Robyn Taylor, Pierre Boulanger, Antonio Krger, and Patrick Olivier, editors, *Smart Graphics*, volume 6133 of *Lecture Notes in Computer Science*, pages 13–24. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13543-9.
- [17] Yu-Wei Cheng. Design of a pen-based system to support peer assessment of mathematics problem solving. Master’s thesis, National Central University, 2009.
- [18] Ruwanee De Silva. Kirchhoff’s pen - a pen based circuit analysis tutor. Master’s thesis, University of California, Riverside, 2008.

- [19] Ruwanee De Silva, David Tyler Bischel, WeeSan Lee, Eric J. Peterson, Robert C. Calfee, and Thomas F. Stahovich. Kirchhoff's pen: a pen-based circuit analysis tutor. In *Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling*, pages 75–82, New York, NY, USA, 2007. ACM.
- [20] John Dunlosky, Katherine A. Rawson, Elizabeth J. Marsh, Mitchell J. Nathan, and Daniel T. Willingham. Improving students learning with effective learning techniques promising directions from cognitive and educational psychology. *Psychological Science in the Public Interest*, 14(1):4–58, 2013.
- [21] Robert G. Farrell, John R. Anderson, and Brian J. Reiser. An interactive computer-based tutor for lisp. In *Proceedings of the Fourth Annual Conference on Artificial Intelligence*, pages 106–109, 1984.
- [22] Kenneth Forbus, Jeffrey Usher, Andrew Lovett, Kate Lockwood, and Jon Wetzel. Cogsketch: Open-domain sketch understanding for cognitive science research and for education. In *Proceedings of the fifth eurographics workshop on sketch-based interfaces and modeling*, 2008.
- [23] Kenneth D. Forbus, Ronald W. Ferguson, and Jeffery M. Usher. Towards a computational model of sketching. In *Proceedings of the 6th international conference on Intelligent user interfaces*, pages 77–83, New York, NY, USA, 2001. ACM. ISBN 1-58113-325-1.
- [24] Reyna Garcia. (forthcoming). Master’s thesis, University of California, Riverside, December 2013.

- [25] Leslie Gennari, Levent Burak Kara, Thomas F. Stahovich, and Kenji Shimada. Combining geometry and domain knowledge to interpret hand-drawn diagrams. *Computers & Graphics*, 29(4):547–562, 2005. ISSN 0097-8493.
- [26] Anoto Group. Anoto annual report 2011, 2011.
- [27] Yonca Haciahmetoglu and Francis Quek. Real-time control of a remote virtual tutor using minimal pen-gestures. In Vincent Aleven, Judy Kay, and Jack Mostow, editors, *Intelligent Tutoring Systems*, volume 6095 of *Lecture Notes in Computer Science*, pages 334–336. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-13436-4.
- [28] Tracy Hammond and Randall Davis. Tahuti: a geometrical sketch recognition system for uml class diagrams. In *ACM SIGGRAPH 2006 Courses*, New York, NY, USA, 2006. ACM. ISBN 1-59593-364-6.
- [29] Ji Hu, Michael Schmitt, Christian Willems, and Christoph Meinel. A tutoring system for it-security. In *Proceedings of the 3rd World Conference in Information Security Education*, pages 51–60, 2003.
- [30] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *ACM SIGGRAPH 2007 courses*, New York, NY, USA, 2007. ACM. ISBN 978-1-4503-1823-5.
- [31] Jooyoung Jang, Christian D. Schunn, and Timothy J. Nokes. Spatially distributed instructions improve learning outcomes and efficiency. *Journal of educational psychology*, 103(1):60, 2011.

- [32] Josiah Jordan. An intelligent tutoring system for static equilibrium equations. Master's thesis, University of California, Riverside, 2010.
- [33] Levent Burak Kara and Thomas F. Stahovich. Hierarchical parsing and recognition of hand-sketched diagrams. In *ACM SIGGRAPH 2007 courses*, New York, NY, USA, 2007. ACM. ISBN 978-1-4503-1823-5.
- [34] Levent Burak Kara, Leslie Gennari, and Thomas F. Stahovich. A sketch-based tool for analyzing vibratory mechanical systems. *Journal of Mechanical Design*, 130(10), September 2008.
- [35] J. A. Landay and B. A. Myers. Sketching interfaces: toward more human interface design. *Computer*, 34(3):56–64, 2001. ISSN 0018-9162.
- [36] James A. Landay. Silk: sketching interfaces like krazy. In *Conference Companion on Human Factors in Computing Systems*, pages 398–399, New York, NY, USA, 1996. ACM. ISBN 0-89791-832-0.
- [37] Joseph J. LaViola and Jr. Mathematical sketching: A new approach to creating and exploring dynamic illustrations, 2005.
- [38] Chia-Keng Lee. Statics intelligent tutoring system. Master's thesis, University of California, Riverside, 2011.
- [39] Jae Seung Lee and Jin Hyung Kim. Pen-based intelligent tutor for simultaneous equations. In *Chinese Conference on Pattern Recognition*, pages 1–5, 2009.

- [40] WeeSan Lee. *Enabling Instructional Applications on Pentop Computers*. PhD thesis, University of California, Riverside, 2009.
- [41] WeeSan Lee, Ruwanee De Silva, Eric J. Peterson, Robert C. Calfee, and Thomas F. Stahovich. Newton's pen: a pen-based tutoring system for statics. In *Proceedings of the 4th Eurographics workshop on Sketch-based interfaces and modeling*, pages 59–66, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-915-9.
- [42] Yang Li. Protractor: a fast and accurate gesture recognizer. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2169–2172, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-929-9.
- [43] M. Masry, D. J. Kang, I. Susilo, and H. Lipson. A freehand sketching interface for progressive construction and analysis of 3d objects. In *Proceedings of AAAI Fall Symposium - Making Pen-Based Interaction Intelligent and Natural*, pages 98–105, 2004.
- [44] Nicholas Matsakis. Recognition of handwritten mathematical expressions. Master's thesis, Massachusetts Institute of Technology, 1999.
- [45] Richard E. Mayer, Kathryn Steinhoff, Gregory Bower, and Rebecca Mars. A generative theory of textbook design: Using annotated illustrations to foster meaningful learning of science text. *Educational Technology Research and Development*, 43(1):31–41, 1995.
- [46] Barbara Means, Yukie Toyama, Robert Murphy, Marianne Bakia, and Karla Jones. Evaluation of evidence-based practices in online learning: A meta-analysis and review of online learning studies. Technical report, US Department of Education, 2010.

- [47] Debra K. Meier, Karl J. Reinhard, David O. Carter, and David W. Brooks. Simulations with elaborated worked example modeling: Beneficial effects on schema acquisition. *Journal of Science Education and Technology*, 17(3):262–273, 2008. ISSN 1059-0145.
- [48] J. L. Meriam and L. G. Kraige. *Engineering Mechanics - Statics*. John Wiley & Sons, Inc., 6th edition, 2007.
- [49] Jeroen J. G. Merrinboer and John Sweller. Cognitive load theory and complex learning: Recent developments and future directions. *Educational Psychology Review*, 17(2):147–177, 2005. ISSN 1040-726X.
- [50] Antonija Mitrovic. Learning sql with a computerized tutor. In *Proceedings of the twenty-ninth SIGCSE technical symposium on Computer science education*, pages 307–311, New York, NY, USA, 1998. ACM. ISBN 0-89791-994-7.
- [51] Shankar Narayanaswamy. *Pen and Speech Recognition in the User Interface for Mobile Multimedia Terminals*. PhD thesis, University of California, Berkeley, 1996.
- [52] Sharon Oviatt, Alex Arthur, and Julia Cohen. Quiet interfaces that help students think. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 191–200, New York, NY, USA, 2006. ACM. ISBN 1-59593-313-1.
- [53] Jan L. Plass, Roxana Moreno, and Roland Brnken. *Cognitive Load Theory*. Cambridge University Press, 32 Avenue of the Americas, New York, NY 10013-2473, USA, 2010.
- [54] Robert J. Roselli, Larry Howard, Bryan Cinnamon, Sean Brophy, Patrick Norris, Megan Rothney, and Derek Eggers. Integration of an interactive free body diagram

assistant with a courseware authoring package and an experimental learning management system. In *Proceedings of American Society for Engineering Education Annual Conference & Exposition*, 2003.

- [55] Sue Vi Rosser and Christine Valle. Intel: Interactive toolkit for engineering learning, 2007. URL <http://intel.gatech.edu/>.
- [56] Steve Smithies, Kevin Novins, and James Arvo. A handwriting-based equation editor. In *Proceedings of Graphics Interface*, pages 84–91. Morgan Kaufmann Publishers Inc, 1999.
- [57] Georges Span. Lites, an intelligent tutoring system for legal problem solving in the domain of dutch civil law. In *Proceedings of the 4th International Conference on AI and Law*, pages 76–81, 1993.
- [58] S. Suebnukarn and P. Haddawy. Comet: A collaborative tutoring system for medical problem-based learning. *Intelligent Systems, IEEE*, 22(4):70–77, 2007. ISSN 1541-1672.
- [59] Siriwan Suebnukarn and Peter Haddawy. A collaborative intelligent tutoring system for medical problem-based learning. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, pages 14–21, 2004.
- [60] Patrick Suppes, Mona Morningstar, et al. Computer-assisted instruction. *Science*, 166(903):343–350, 1969.
- [61] John Sweller and Graham A. Cooper. The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction*, 2(1):59–89, 1985.

- [62] John Sweller, Paul Ayres, and Slava Kalyuga. *Cognitive Load Theory*. Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA, 2011.
- [63] Victoria Tsiriga and Maria Virvou. Transferring a standalone intelligent algebra tutor over the www. In *Proceedings of the 4th International Workshop on Computer Science and Information Technologies*, 2002.
- [64] Leonard Uhr. Teaching machine programs that generate problems as a function of interaction with students. In *Proceedings of the 1969 24th national conference*, pages 125–134, New York, NY, USA, 1969. ACM.
- [65] Stephanie Valentine, Francisco Vides, George Lucchese, David Turner, Hong-hoe Kim, Wenzhe Li, Julie Linsey, and Tracy Hammond. Mechanix: A sketch-based tutoring system for statics courses. In *The Twenty-Fourth Conference on Innovative Applications of Artificial Intelligence, Toronto, Ontario, Canada*, 2012.
- [66] Kurt VanLehn. The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education*, 16(3):227–265, 2006.
- [67] Kurt VanLehn. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist*, 46(4):197–221, 2011.
- [68] Kurt VanLehn, Collin Lynch, Kay Schulze, Joel A. Shapiro, Robert Shelby, Linwood Taylor, Don Treacy, Anders Weinstein, and Mary Wintersgill. The andes physics tutoring system: Lessons learned. *International Journal of Artificial Intelligence in Education*, 15(3), 2005.

- [69] Etienne Wenger. *Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987. ISBN 0-934613-26-5.
- [70] Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, pages 159–168, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-679-0.
- [71] Panrong Yin, Kenneth D. Forbus, Jeffrey Usher, Brad Sageman, and B. Jee. Sketch worksheets: A sketch-based educational software system. In *Proceedings of the 22nd Innovative Applications of Artificial Intelligence Conference, Atlanta*, 2010.
- [72] Robert Zeleznik, Andrew Bragdon, Ferdi Adeputra, and Hsu-Sheng Ko. Hands-on math: a page-based multi-touch and pen desktop for technical work and problem solving. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, pages 17–26, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0271-5.
- [73] Robert C. Zeleznik, Kenneth P. Herndon, and John F. Hughes. Sketch: an interface for sketching 3d scenes. In *ACM SIGGRAPH 2007 courses*, New York, NY, USA, 2007. ACM. ISBN 978-1-4503-1823-5.

Appendix A

Newton's Tablet Stages and Modes

This appendix describes in detail each of the logical stages and functional modes used in Newton's Tablet. Section [A.1](#) describes the system stages. The progression of these stages is what guides the student through the novel problem-solving decomposition of this research. Section [A.2](#) describes the system modes. These modes represent common functional states in which the system can be during nearly any given stage.

A.1 Newton's Tablet Stages

Newton's Tablet decomposes the process of constructing statics free-body diagrams and equilibrium equations into atomic sub-steps. This section enumerates and discusses the logical system stages that represent these problem-solving sub-steps. Figure [A.1](#) shows a state diagram for the free-body diagram stages, and Figure [A.8](#) shows a state diagram for the equilibrium-equation stages. The section for each stage begins by stating the prompt displayed in the status bar at the start that stage, and a screenshot is shown for each stage,

which displays the relevant work for that stage using the Toggle Clamp problem from the problem set as an example. See Appendix D.2 for a description of each of the problems created for use with Newton’s Tablet.

During almost every stage, the student first enters some work—either on the canvas or in textboxes—and then the student taps the Check Work button to prompt the system for conceptual feedback on his/her work for the current stage. See Section 3.3 for a description of the conceptual feedback system, and see Appendix C for a complete enumeration of the types of conceptual errors evaluated during each stage. A few stages, however, have no right or wrong answer, and automatically transition once the student enters any work. For example, in the Positive Moment Direction Drawing Stage, the system automatically transitions once the student indicates either direction.

Some of the equation stages are only relevant for moment equations and are skipped if the current equation is a force equation. Similarly, some of the free-body diagram stages are only relevant for multi-body problems and will be skipped if the current problem is a single-body problem. This does not necessarily present the pedagogical problem of giving away too much information to the student, because we designed the Newton’s Tablet problem set so that students would work on the single-body problems before having seen in lecture how to model a multi-body problem.

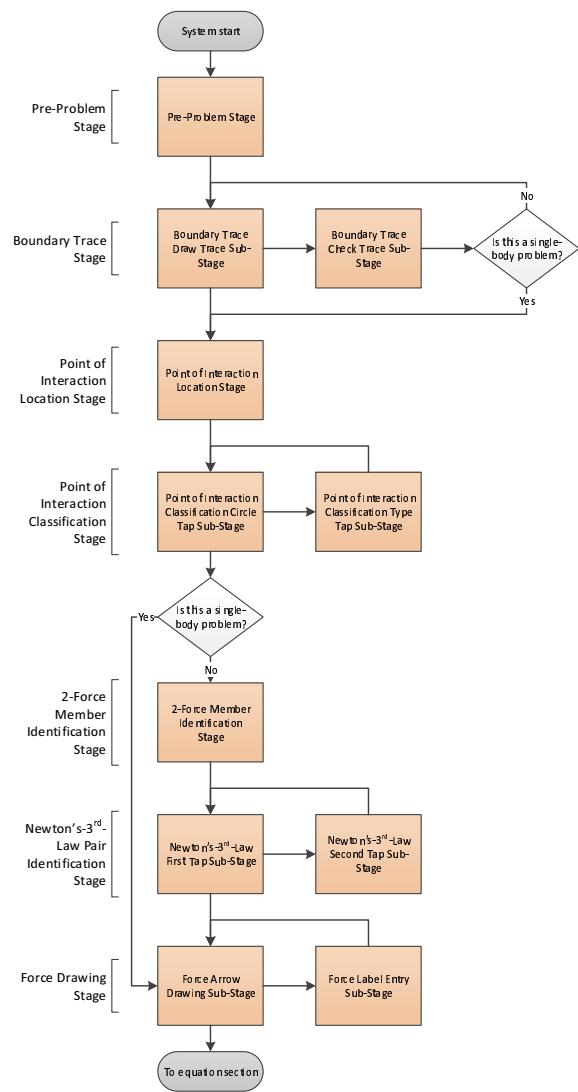


Figure A.1: State Diagram of the Free-Body Diagram Stages in Newton's Tablet.

A.1.1 Pre-Problem Stage

Stage Instructions: “Use the open button on the top left to select a new problem”

The first thing the student sees upon starting the program is a blank canvas and the menu bar at the top of the window. At this point, the student selects a problem via the Load Problem button. Once the student selects a problem, the problem-description and annotated-problem-image boxes appear, and a larger, un-annotated version of the problem image is rendered on the canvas.

In addition, Newton’s Tablet automatically starts in General Help mode (Section A.2.3). This serves as a helpful tutorial for how to use the mechanics of the system in case it is the student’s first time using Newton’s Tablet.

A.1.2 Boundary Trace Stage

Stage Instructions 1: “Trace the boundary of the intended system”

Stage Instructions 2: “When you are done tracing the boundary, tap the Check Body Trace button”

The first logical step in constructing a free-body diagram is to identify the boundaries of the bodies that are being modeled. In the Boundary Trace Stage, the student does this by tracing around the appropriate bodies. Because the system allows the student to use multiple strokes when tracing boundaries, there needs to be some action that prompts the system to attempt recognition once all of the appropriate strokes for the current trace have been drawn. The student triggers this recognition by tapping on the Check Body Trace

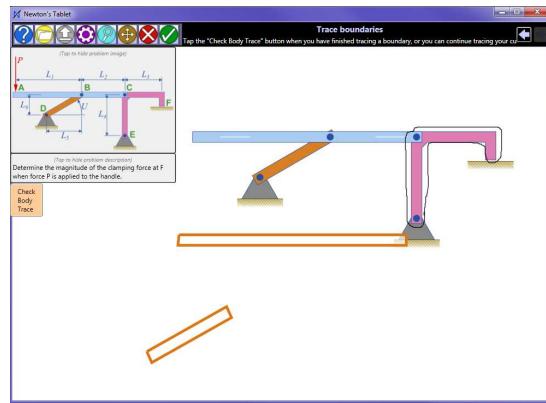


Figure A.2: Boundary Trace Stage Screenshot.

button. This button appears at the start of this stage and is removed at the end. Thus, this stage embodies two logical sub-stages: the Boundary Trace Draw Trace Sub-Stage and the Boundary Trace Check Trace Sub-Stage.

If the current problem is a multi-body problem, then upon a successful boundary trace recognition, the program allows the student to drag the boundary away from the underlying image (Section A.2.1). This prevents the student from leaving all of the boundaries stacked over top one another, which in turn makes for a more organized workspace. Spacing apart the boundaries mimics the standard practice when drawing free-body diagrams with pen on paper. After re-positioning the boundary, the student is free to then trace another boundary.

If the current problem is a single-body problem, then the student does not need the option of tracing additional boundaries. So with single-body problems, the system automatically transitions to the next stage as soon as the student successfully traces a valid boundary.

See Section 3.2.3 for a detailed description of this system’s trace recognition algorithm.

A.1.3 Point of Interaction Location Stage

Stage Instructions: “Tap on the point(s) where external forces act on your system”

After identifying the system boundaries, the next logical step in constructing a free-body diagram is to identify the locations of all of the external forces acting on the system boundaries. These are called points of interaction. To do this, the student simply taps on all of the locations on each body where there is a point of interaction.

Once the student taps on a point of interaction, a dot appears at that point on the boundary. These dots persist through the remainder of the free-body diagram stages.

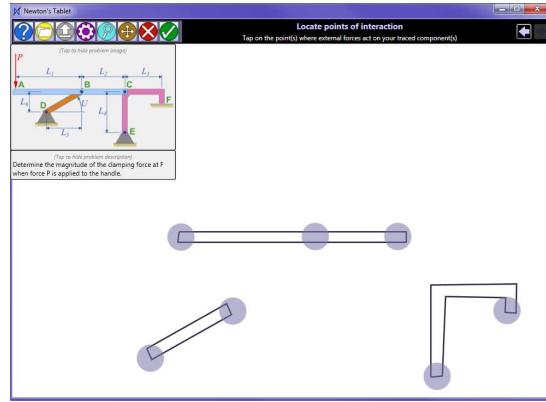


Figure A.3: Point of Interaction Location Stage Screenshot.

A.1.4 Point of Interaction Classification Stage

Stage Instructions 1: “Tap on an interaction point to add a label”

Stage Instructions 2: “Select the interaction type from the menu”

After identifying where each of the points of interaction in the system are located, the next logical step is to determine the type of each of these points of interaction. There are many potential types of interactions which could be modeled in a free-body diagram; some of the most common types of interactions include: applied forces, weight forces, smooth contacts, slider joints, tension forces, roller joints, pivot joints, and friction forces. None of the problems in the Newton’s Tablet problem set considers friction, so it is not presented as an option. All of the other point-of-interaction types just mentioned are available for selection during this stage. These interaction types are dynamically displayed as buttons along the left side of the canvas during this stage.

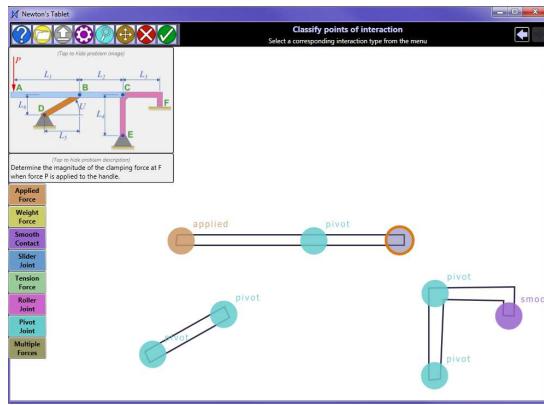


Figure A.4: Point of Interaction Classification Stage Screenshot.

In order to classify a point of interaction, the student first taps on the point of interaction being classified. Then the student taps on whichever button represents the type of this point of interaction. If there are actually multiple forces of different interaction types acting on the boundary at a point of interaction, then the student first taps on the Multiple Forces button, and then on each of the appropriate type buttons. Thus, two logical sub-stages comprise this stage: the Point of Interaction Classification Circle Tap Sub-Stage and the Point of Interaction Classification Type Tap Sub-Stage.

The interaction type buttons are color-coded, and after the student classifies a point of interaction, the dot for that point of interaction becomes the corresponding color and is labeled with the name of the type. This makes it easier for the student to readily identify which type(s) he/she has previously classified a point of interaction with. These colors and labels are removed after this stage.

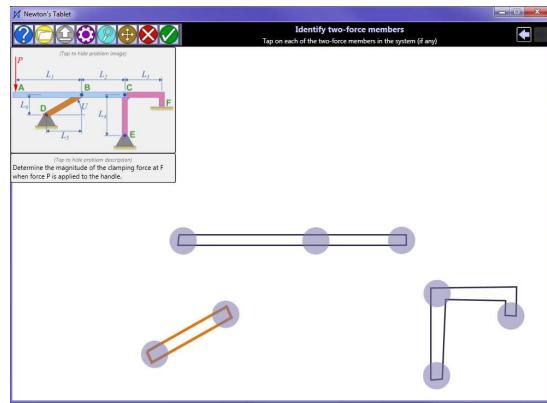


Figure A.5: Two-Force Member Identification Stage Screenshot.

A.1.5 Two-Force Member Identification Stage

Stage Instructions: “Tap on each of the two-force members in the system (if any)”

If the problem under consideration is a multi-body problem, then an important logical step is to identify any two-force members that may be a part of system. The forces used to model two-force members—and also those for Newton’s-third-law pairs—always act with equal magnitude and in opposite directions. Identifying these concepts allows the resulting free-body diagram to be greatly simplified. To identify two-force members, the student simply taps on a body to identify it as being a two-force member. A second tap will then un-identify it. When the student identifies a two-force member, the system gives it a bold, orange border. This helps the student to readily remember which two-force members he/she has identified. Because two-force members will not exist in single-body problems, Newton’s Tablet skips this stage unless the current problem is a multi-body problem.

A.1.6 Newton's-Third-Law Pair Identification Stage

Stage Instructions 1: “Tap on an interaction point for a Newton’s-third-law pair (if any)”

Stage Instructions 2: “Tap on the other interaction point for this Newton’s-third-law pair”

Another important logical step when considering a multi-body problem is to identify which of the points of interaction form Newton’s-third-law pairs. To do this, the student first taps on the first point of interaction in a pair, and then taps on the second point of interaction in the pair. Thus, this stage consists of two logical sub-stages: the Newton’s-Third-Law First Tap Sub-Stage and the Newton’s-Third-Law Second Tap Sub-Stage. This stage is also irrelevant and is skipped for single-body problems

Once the student has identified two points of interaction as forming a Newton’s-third-law pair, they are both given a matching random color. This helps the student remember which Newton’s-third-law pairs he/she has already identified. These dynamic colors are removed after this stage.

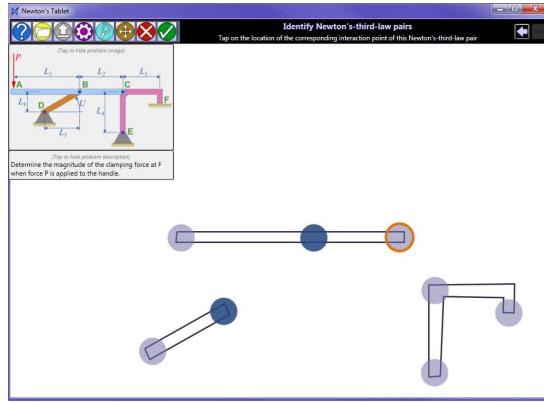


Figure A.6: Newton’s-Third-Law Pair Identification Stage Screenshot.

A.1.7 Force Drawing Stage

Stage Instructions 1: “Draw (single-stroke) arrows for the external forces acting on your system”

Stage Instructions 2: “Enter a label for the force. Press Enter when finished”

Finally, in this stage, the student has performed all preliminary free-body diagram analysis of the system, and the student now identifies the actual forces acting on each of the system boundaries. To do this, the student draws an arrow to represent the location and direction of a force. After drawing a force arrow, the student enters a label for the force. Label entry occurs via the keyboard and in a textbox that dynamically appears next to the force arrow. Thus, this stage contains two logical sub-stages: the Force Arrow Drawing Sub-Stage and the Force Label Entry Sub-Stage.

The arrow recognizer used in Newton’s Tablet is an implementation of the Protractor recognizer [42], and requires the student to draw arrows as a single stroke. See

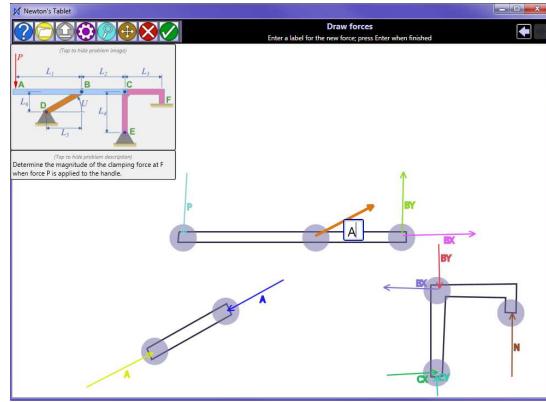


Figure A.7: Force Drawing Stage Screenshot.

Garcia [24] for a detailed description of this arrow recognition algorithm. The conceptual checks of the student’s work for this stage consider the locations of the force arrows, the directions of the force arrows, and the labels of the force arrows. To determine the location where a force arrow interacts with a body, both the head and tail positions of the arrow must be considered. The system uses whichever end of the arrow is closer to a point of interaction as the conceptual location of the force.

Once this stage completes, the student’s force arrows are given uniform lengths and snapped to the actual force directions. In addition, two force arrows are given the same color if they both belong to the same Newton’s-third-law pair or act on the same two-force member. These force arrows persist on the canvas for the remainder of the problem.

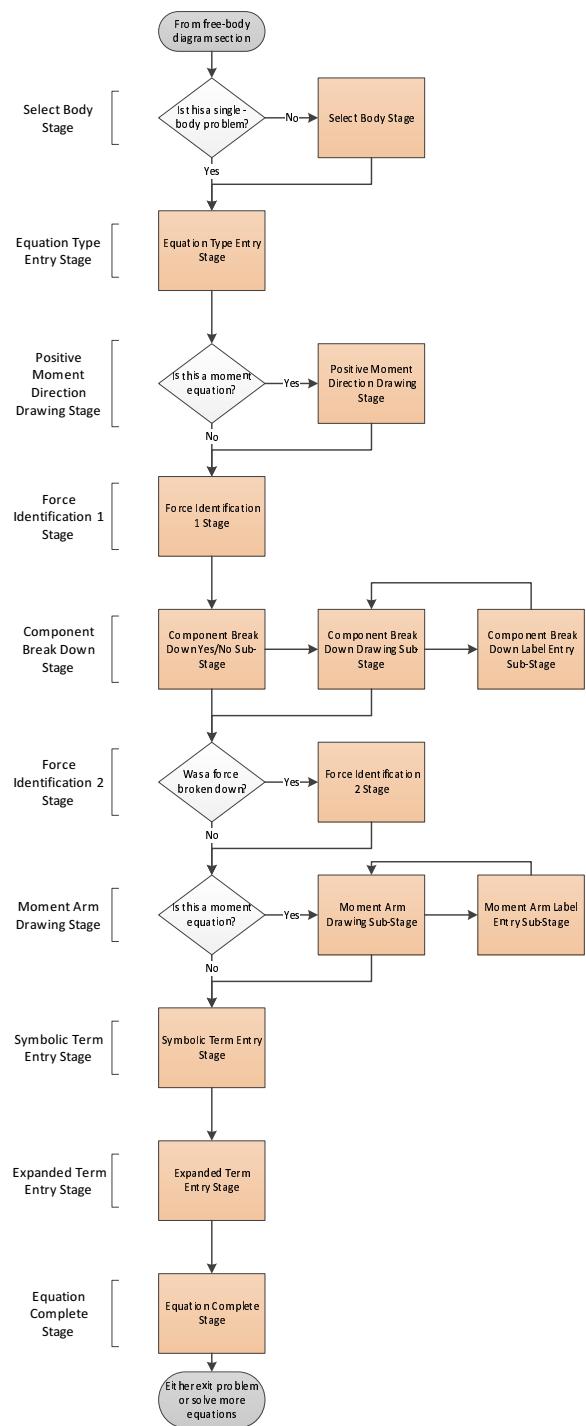


Figure A.8: State Diagram of the Equilibrium-Equation Stages in Newton's Tablet.

A.1.8 Body Selection Stage

Stage Instructions: “Select the body corresponding to the equation you would like to derive”

The Body Selection Stage marks the end of the free-body diagram portion of the system, and marks the beginning of the equilibrium-equation portion of the system. Newton’s Tablet now displays the equation panel along the bottom of the window. At this point in the problem-solving process, the student must choose which body to work on, and which equilibrium equation to solve for that body.

To select a body to work on, the student simply taps on the body. The selected body and its forces remain bold and colored, but the other bodies and forces in the system become thin and greyed-out. This helps the student readily remember which body he/she is working on. After the student has selected a body, the system automatically transitions to the next stage.

At any given time during the equation portion of the system, if the student decides to instead work on an equation for a different body, he/she can tap on the Select Body button, which then transitions the system back to this stage. Because a single-body problem only has one boundary to choose from, this stage—and the Select Body button—is only available for multi-body problems.

A.1.9 Equation Type Entry Stage

Stage Instructions: “Enter the type for the equation for this body”

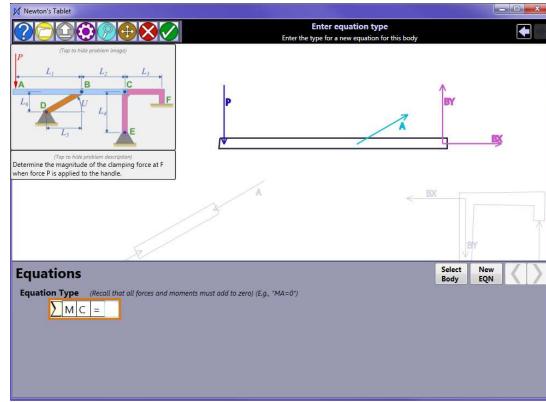


Figure A.9: Equation Type Entry Stage Screenshot.

After having chosen a body to work on, the student then decides which equilibrium equation to solve for that body. The student uniquely describes the identity of the equation being considered by writing a special equation-type meta-equation. Because the Newton's Tablet system only considers static problems that are under equilibrium from external forces, this equation-type meta-equation always states that the sum of the forces or moments under consideration is equal to zero. Moments are only considered around a single, given point at a time, and forces are only considered in a single, given, axially aligned direction at a time. For Newton's Tablet, the positive directions of the coordinate axes are given in the problem description, and the student is not allowed the freedom to change these. This greatly reduces the complexity of evaluating the student's equation work.

The student enters this equation-type meta-equation in textboxes that dynamically appear at the top-left of the equation panel. These textboxes persist—but in a more natural, cohesive, and un-editable form—after this stage. Newton's Tablet gives a summation sign to the left of the textboxes for two reasons: it helps to clarify what the student needs to

enter in the textboxes, and it eliminates the need to enter the sigma character using the keyboard.

A.1.10 Positive Moment Direction Drawing Stage

Stage Instructions: “Draw the positive moment direction”

Once the specific equilibrium equation under consideration has been determined, if a moment equation was selected, then the next logical step is to determine whether moments will be considered positive in the clockwise or counter-clockwise direction for the current equilibrium equation. To enable this, Newton’s Tablet dynamically displays a small ink canvas to the left of the equation type. The student then draws a curved arrow in this canvas. The system calculates and saves whether the overall direction of this stroke was clockwise or counter-clockwise. Newton’s Tablet allows the student to retrospectively change this positive moment direction by re-drawing it until the end of the symbolic equation entry stage. If the student selected a force equation in the Equation Type Entry Stage, then this stage is irrelevant and is skipped. After the student draws the direction stroke, the system automatically transitions to the next stage.

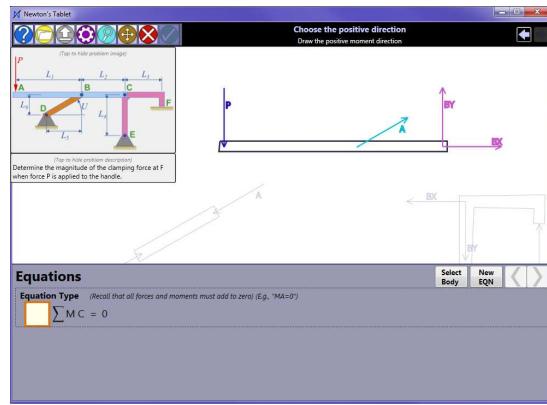


Figure A.10: Positive Moment Direction Drawing Stage Screenshot.

A.1.11 Force Identification 1 Stage

Stage Instructions: “Tap on all forces relevant to this equation”

Before attempting to write the actual equilibrium equation, an important conceptual step is to identify which of the forces acting on the body are relevant for the current equation. To do this, the student simply taps each of the relevant force arrows. When the student has selected a force arrow, the system gives it a bold, orange line. This helps the student to readily remember which forces he/she has selected. The student can tap again on a selected arrow to un-select it.

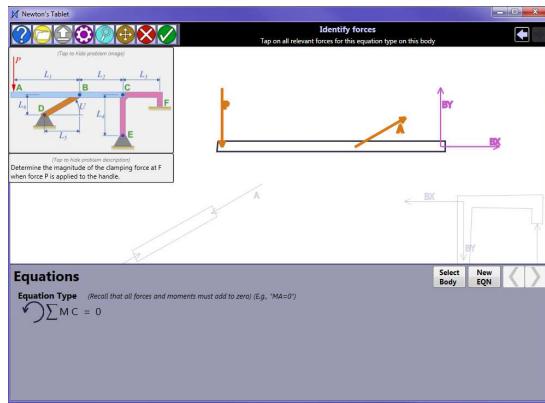


Figure A.11: Force Identification 1 Stage Screenshot.

A.1.12 Component Break Down Stage

Stage Instructions 1: “Draw force component arrows as needed”

Stage Instructions 2: “Enter a label for the force component. Press Enter when finished”

After identifying which forces are involved in the current equilibrium equation, the next logical step is to determine whether any of these forces need to be decomposed into axially aligned components. To do this, first Newton’s Tablet asks the student—via a dialogue box—whether any of the forces for the current equation need to be broken down. Then, if any of the forces do indeed need to be broken down, the student draws and labels the appropriate force arrows for the decomposed force components. This is accomplished in the same manner as in the Force Drawing stage (Section A.1.7). Thus, this stage encompasses three logical sub-stages: the Component Break Down Yes/No Sub-Stage, the Component Break Down Drawing Sub-Stage, and the Component Break Down Label Entry Sub-Stage.

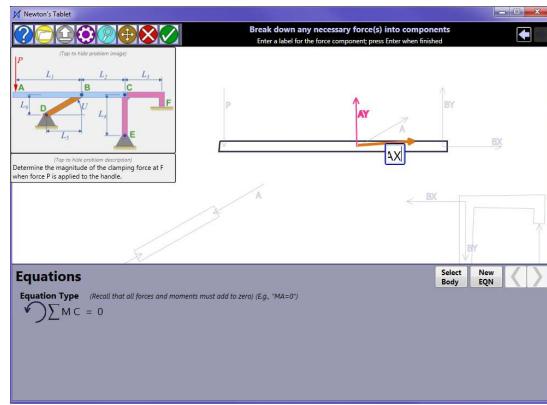


Figure A.12: Component Break Down Stage Screenshot.

A.1.13 Force Identification 2 Stage

Stage Instructions: “Select the newly-drawn components that are relevant to this equation”

If a relevant force for the current equation was indeed broken down, then it is important to identify which of its two broken-down components is relevant for this equation. To do this, the student simply repeats the same process as in Force Identification 1 Stage (Section A.1.11). However, for all of the forces which were previously identified, Newton’s Tablet starts this stage with them still selected—unless they were just broken down in the Component Break Down Stage.

If no forces were decomposed in the Component Break Down Stage, then this stage is unnecessary and is skipped.

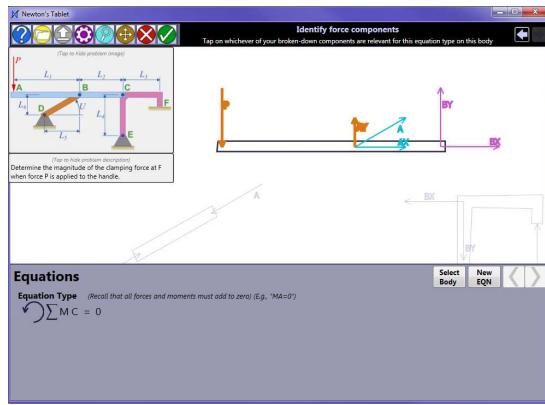


Figure A.13: Force Identification 2 Stage Screenshot.

A.1.14 Moment Arm Drawing Stage

Stage Instructions 1: “Draw square brackets to represent each of the relevant moment arms. You may draw references lines if needed”

Stage Instructions 2: “Enter a label for the new moment arm. Press Enter when finished”

At this point in the problem-solving process, if the equation under consideration is a moment equation, then before the actual equilibrium equation can be written out, the locations and directions of the relevant moment arms still need to be determined. To do this, the student draws a bracket to represent a moment arm. The student needs to align this bracket in the correct direction, and needs to start it near the force to which this moment belongs and end it near the point of interaction about which this equation is based. After drawing a moment arm bracket, the student then labels the moment arm. The reason for labeling the moment arm here is so that the student can reference the conceptual arm itself

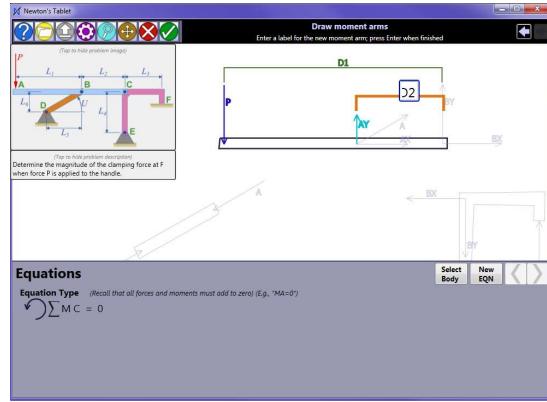


Figure A.14: Moment Arm Drawing Stage Screenshot.

in the Symbolic Term Entry Stage—as opposed to the representation of this moment arm using dimensions from the problem description, which will be used in the Expanded Term Entry Stage. Thus, two logical sub-stages comprise this stage: the Moment Arm Drawing Sub-Stage and the Moment Arm Label Entry Sub-Stage. If the current equilibrium equation is a force equation, then this stage is unnecessary and is skipped.

Newton’s Tablet uses the same recognizer for the moment-arm-bracket gestures as was used for the arrow gestures. See Garcia [24] for a detailed description of this bracket recognition algorithm.

A.1.15 Symbolic Term Entry Stage

Stage Instructions: “Enter the symbolic equation terms for each of the forces you just identified”

At this point in the problem-solving process, the student has completed all of the preliminary analysis for actually writing the equilibrium equation. However, Newton’s

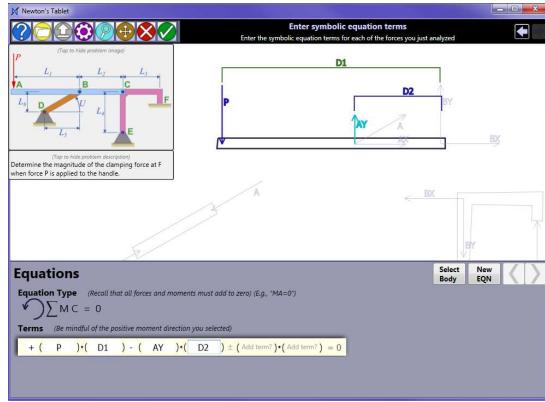


Figure A.15: Symbolic Term Entry Stage Screenshot.

Tablet makes an important pedagogical distinction here between the concepts of the symbolic form of an equation and the expanded form of an equation. The student is required to enter the symbolic form of an equation before entering the expanded form.

Throughout all of the previous stages, the student has been constructing a free-body diagram that includes labeled sketches of all forces and moment arms relevant for the current equilibrium equation. The symbolic form of this equation references only the force and moment arm labels drawn by the student in the free-body diagram.

The expanded form of an equilibrium equation represents a translated version of the symbolic equation, in which all of the terms use dimensions, forces, and angles from the original problem description rather than their symbolic equivalents from the free-body diagram. More specifically, moment arm lengths and directions will be represented according to problem-description dimensions and possibly by trigonometric functions applied to problem-description angles, and angular forces will be decomposed using trigonometric functions into their axially aligned components.

For an example of the difference between the symbolic and expanded forms of an equation, consider Figure 3.7.

For entering text for equations, Newton's Tablet compromises between two conflicting concerns. The first concern is to scaffold student work. By breaking an equation apart into a structured collection of textboxes for each separate term, the system can make more-definite decisions regarding the student thought process at each part of the equation. This in turn enables the system to present more-targeted and more-helpful feedback in response to student errors.

The second concern is to present a natural user interface which mimics the use of the traditional pen-and-paper interface for statics. Unfortunately, segmenting the text-entry area into a long sequence of textboxes does not help to emulate the traditional drawing space for constructing equilibrium equations. To somewhat remedy this unnatural design, additional textboxes for a new symbolic term are only generated after text has been entered in textboxes for all other symbolic terms. Furthermore, once the student has finished constructing the symbolic equation, Newton's Tablet hides these textboxes, and the structure of the symbolic equation is left appearing natural and cohesive.

The format of the set of textboxes generated for a given term depends on whether the given equation is a force equation or a moment equation. If it is a force equation, then a single term consists of only two textboxes: the first textbox represents the sign of the term and accepts only a single '-' or '+' character; and the second textbox represents the magnitude and direction of the force represented by this term. If the given equation is a moment equation, then a single term consists of three textboxes. The first textbox, once

again, represents the sign of the term. The second textbox now represents either the same type of force information as the second force term textbox or the direction and length of the moment arm belonging to this force. The third textbox represents whichever of the two options was not used in the second textbox.

A.1.16 Expanded Term Entry Stage

Stage Instructions: “Enter the expanded mathematical versions of each of the symbolic terms”

Once the student has identified the symbolic terms of the equilibrium equation, he/she is ready to write out the final expanded form of the equation. To do this, the student uses the keyboard to enter the expanded terms in textboxes that have appeared below the symbolic terms. Newton’s Tablet requires the student to enter the expanded form of a term in the textbox below the textbox for the corresponding symbolic term. This allows for more-targeted evaluation and feedback of the student’s work. In order to help guide the student to enter expanded terms in the same order as their symbolic counterparts, when one of the expanded-term textboxes is given focus, the corresponding symbolic-term textbox is emphasized with an underline and a darker background. Similar to the symbolic equation, after the student has finished correctly entering the expanded equation, the segmented textboxes are hidden, and the expanded equation is left appearing natural and cohesive.

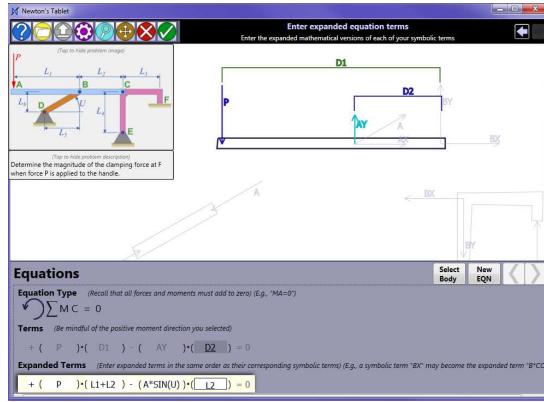


Figure A.16: Expanded Term Entry Stage Screenshot.

A.1.17 Equation Complete Stage

Stage Instructions: “Congratulations! You have completed this equation!”

After completing all of the work for an equilibrium equation, the student has the option of solving another equation. He/she can accomplish this via either the Select Body button or the New EQN button at the top right of the equation panel. The former allows the student to select a new body for which to solve equilibrium equations, while the latter allows the student to solve another equilibrium equation for the currently selected body. The student can actually use these buttons at any stage in the equation-solving process, and if the student leaves an equation in a state of partial completion, he or she can later return to finish it via the left and right arrow buttons at the top right of the equation panel. If the student wishes to return to a previous equation for a body other than the one currently selected, he or she would first need to tap the Select Body button. This transitions the program to the Body Selection Stage, in which the student can select another body.

A.2 Newton's Tablet Modes

In addition to the sequence of logical stages discussed in Section A.1, there is a finite set of functional modes in which the system can be at nearly any given time. Unlike the disjoint stages, the system can be in more than one of these modes simultaneously, and the student can toggle these modes on and off within nearly any stage. For example, when error highlights are being displayed after the student has tapped on the Check Work button, the system is actually in Error Mode, and the student can then leave Error Mode by tapping again on the Check Work button. Each of these modes can be toggled on and off by tapping on its corresponding button in the menu bar at the top of the window. This section now describes each of these modes.

A.2.1 Drag Mode

In Drag Mode, the student can drag a system boundary in order to re-position it within the canvas. This is especially helpful with multi-body problems in order to keep the free-body diagram organized. Drag Mode is toggled on and off via the Drag Mode Button at the top-left of the window.

A.2.2 Erase Mode

In Erase Mode, the student can erase any of his/her work that was entered within the current stage. Erasing is accomplished by drawing a stroke that intersects with whatever shape is to be erased. It is important to note that Newton's Tablet never allows the student to erase work from previous stages, because such work is guaranteed to be correct. The

student can also erase work without entering Erase Mode simply by using the right button of the stylus or mouse to draw the stroke rather than the left. Erase Mode is toggled on and off via the Erase Mode Button at the top-left of the window.

A.2.3 General Help Mode

In General Help Mode, a large box is displayed which provides a detailed description of the concepts involved in the current stage and also a general explanation of what the system expects the student to do to enter the work for the current stage. This box contains images, text, and hyperlinks to other general-help conceptual pages. In order to see more clearly more-detailed general-help images, the student can tap on these images to expand them along with the overall general-help box. Newton's Tablet contains a large collection of general-help pages containing textual descriptions and descriptive images for many different important statics concepts.

The General Help box is automatically displayed when the program starts, and its contents is updated with the transition of each logical stage in order to remain relevant for the current task. Because the General Help box automatically updates with a description for the current stage, it can be useful as a tutorial for students who are using Newton's Tablet for the first time.

General-help content also ties into specific problem error messages. If General Help Mode is on when a problem error box is displayed, then the general-help content automatically updates to whichever page is relevant to that problem error. Otherwise, if General Help Mode is not already open when the student opens a new problem error box,

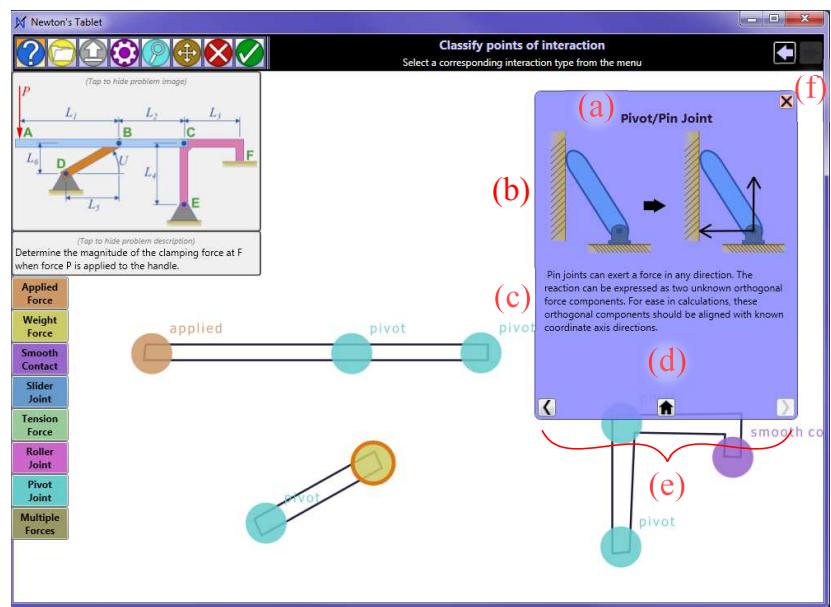


Figure A.17: A General Help Mode Screenshot. (a) The title of the current general-help page. (b) An optional image for this page. (c) The detailed textual description for this page. (d) This is where hyperlinks to other general-help pages would appear. (e) These navigational buttons allow the student to progress to adjacent, related general-help pages or to return to the home page. (f) This button allows the student to exit General Help Mode.

then the student can tap on the General Help Mode button inside of the problem error box to have the appropriate page displayed. General Help Mode can also be toggled on and off via the General Help Mode Button at the top-left of the window.

A.2.4 Query Mode

The purpose of this mode is to allow the student to query information that has previously been analyzed for a given part of the free-body diagram. To query information about an object in the free-body diagram, the students simply hovers the mouse cursor over that object, and then a tool-tip box appears near the object. The reason this querying may be needed is because it is frequently the case that work from a given stage is removed from canvas after the completion of that stage, and thus the student may forget what was previously analyzed. For example, after the Newton's-Third-Law Pair Identification Stage, the points of interaction that were in Newton's-third-law pairs are returned to their original colors. By later querying these points of interaction, the student can see that they were identified as being in a Newton's-third-law pair. Query Mode is toggled on and off via the

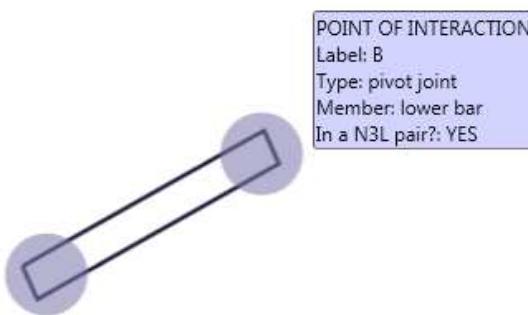


Figure A.18: A Query Box.

Query Mode Button at the top-left of the window. See Figure A.18 for an example of what is shown in a query box.

A.2.5 Scratch Work Mode

In Scratch Work Mode, the student can draw whatever strokes he/she desires, and Newton's Tablet will leave them completely un-analyzed. The normal system behavior outside of Scratch Work Mode is for all strokes to be analyzed and then either replaced with a beautified representation of the stroke (Section 3.2.4) or discarded if they are do not represent anything meaningful for the current stage. These strokes will remain on the canvas, and this allows the student to visually work out any thoughts within the system. Scratch Work Mode is toggled on and off within the Settings Menu.

A.2.6 Error Mode

In Error Mode, Newton's Tablet highlights the locations of all errors in the student's work for the current stage. The student can then tap on one of these error highlights to receive didactic feedback regarding the specific conceptual error. Error Mode can be toggled on and off either via the Check Work Button at the top-left of the window or by tapping outside of one of the error highlights. For a more detailed description of conceptual error feedback, see Section 3.3.

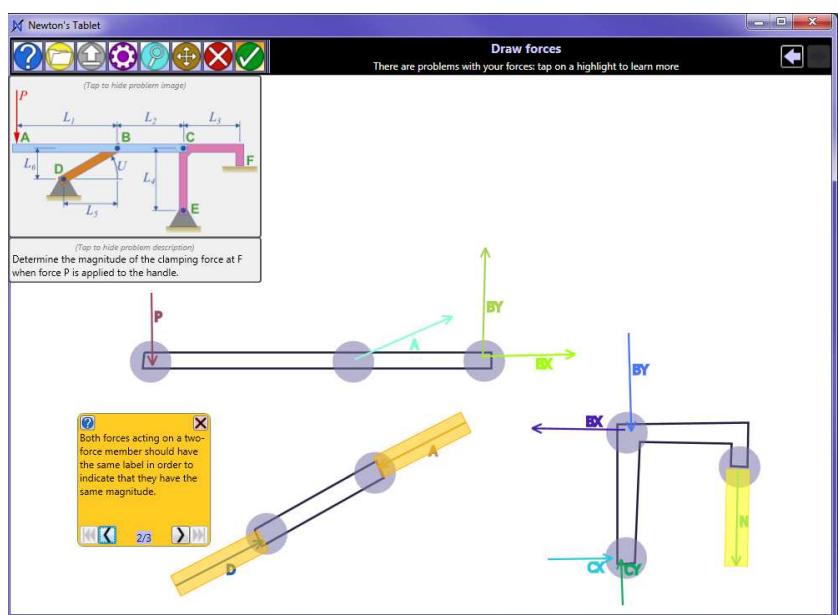


Figure A.19: An Error Mode Screenshot.

Appendix B

Newton's Pen Stages and Modes

This appendix describes in detail each of the logical stages and functional modes used in Newton's Pen. Section B.1 describes the system stages. The progression of these stages is what guides the student through the novel problem-solving decomposition of this research. Section B.2 describes the system modes. These modes represent common functional states in which the system can be during nearly any given stage.

B.1 Newton's Pen Stages

The problem-solving stage progression of Newton's Pen closely follows that of Newton's Tablet. The most notable exception is that the Newton's Pen stages prompt the student to label each free-body diagram element immediately after drawing it. This allows the system to later reference particular elements by name. There are two fundamentally different types of labels in Newton's Pen. The first type involves labels which the system chooses and prompts the student to draw. The system does not at all attempt to recognize

the text of these labels. The second type involves labels that the student chooses and the system does need to evaluate the text of these labels. In order to obtain a textual value for this type of label, the system prompts the student to use the keyboard buttons to enter the value of the label immediately after drawing the ink for the label. For both types of labels, in order to signal the completion of the strokes for a label, the student can either perform a double-tap anywhere on the page or the student can wait for a small period. Section 4.3.1 provides a more detailed explanation of how Newton's Pen references a specific free-body diagram element.

Another modification in Newton's Pen is that messages presented to the student need to be more concise in order to not take too much time to scroll across the small display. To facilitate this goal, Newton's Pen gives acronyms to various common free-body diagram concepts: points of interaction are called POIs, two-force members are called 2FMs, and Newton's-third-law pairs are called N3L pairs.

Figures B.1 and B.2 show flowcharts of the two sub-stages in the Boundary Identification Stage, which serve as good examples for how most of the Newton's Pen stages operate.

Figure B.3 shows a state diagram of the logical stages in Newton's Pen.

The following descriptions of the Newton's Pen stages do not focus on describing the rationale or many of the details of the stage breakdown, but instead focus on how these stages differ from the corresponding stages in Newton's Tablet.

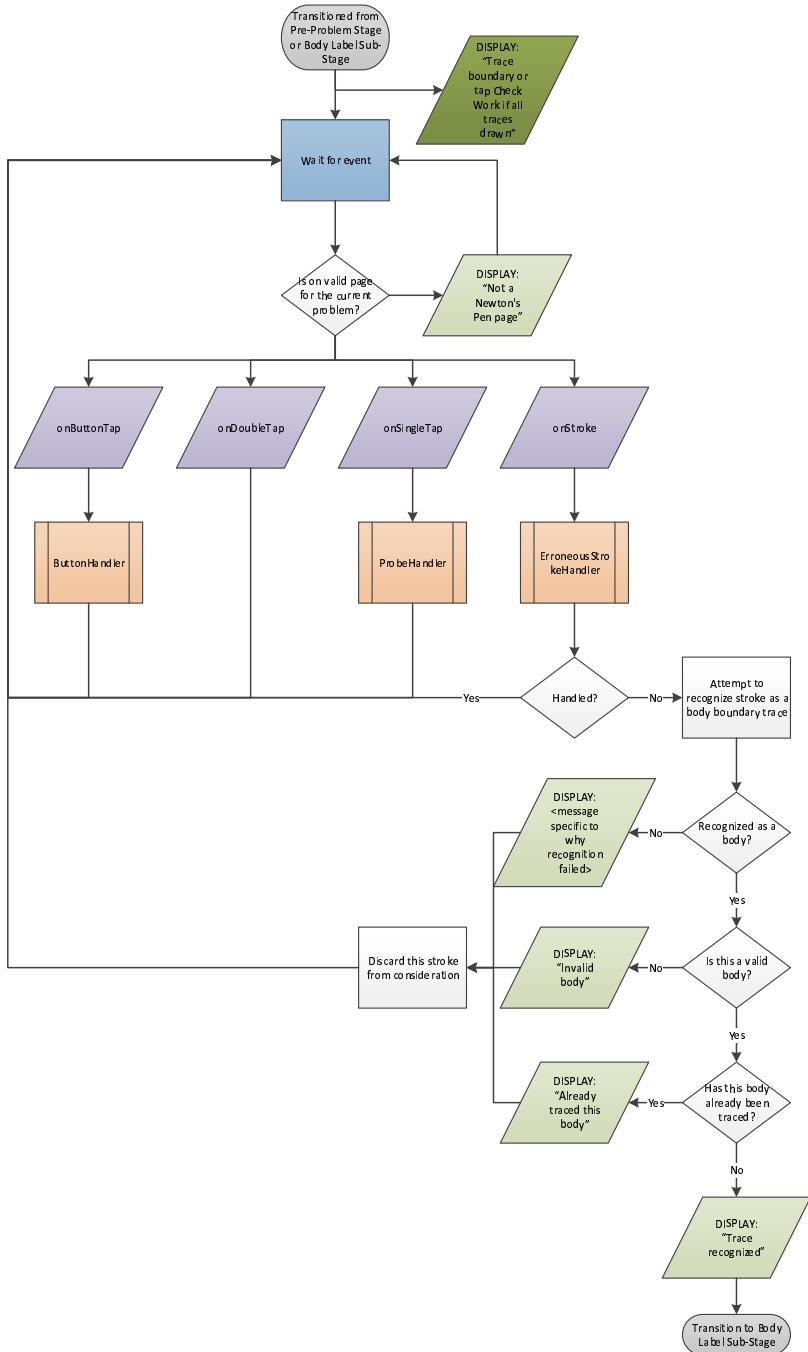


Figure B.1: Flowchart of the Boundary Trace Sub-Stage Logic.

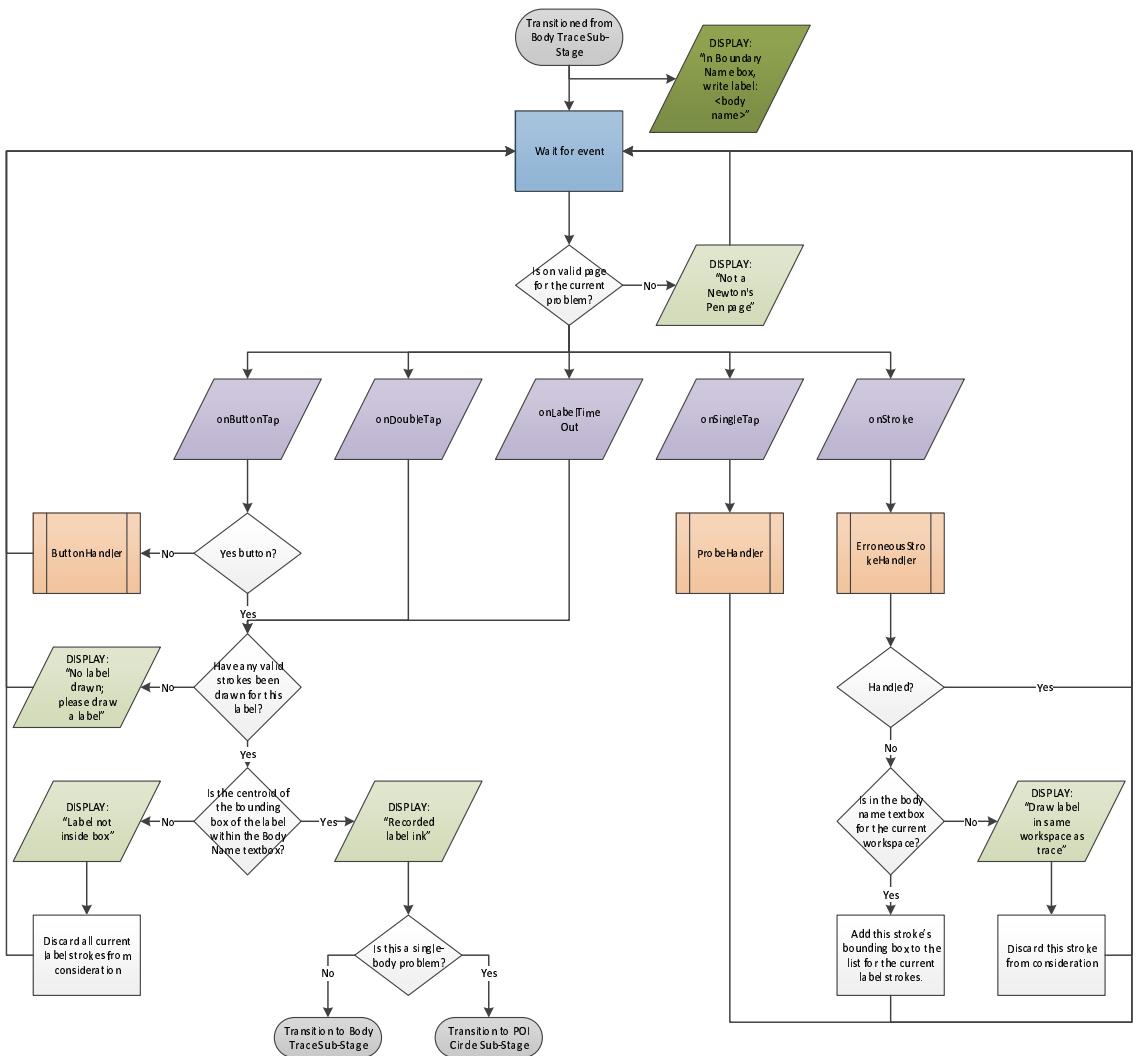


Figure B.2: Flowchart of the Boundary Label Sub-Stage Logic.

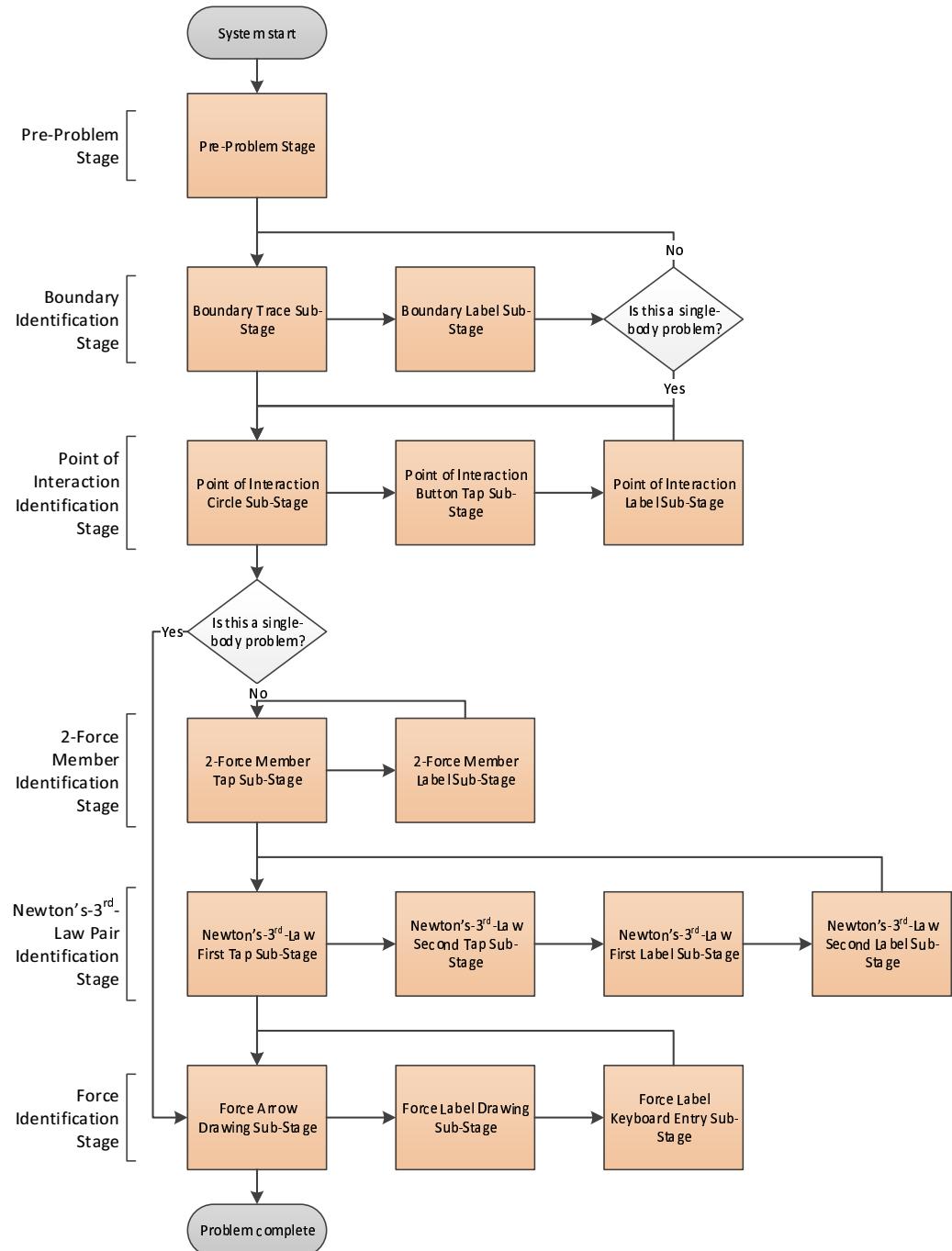


Figure B.3: State Diagram of the Free-Body Diagram Stages in Newton's Pen.

B.1.1 Boundary Identification Stage

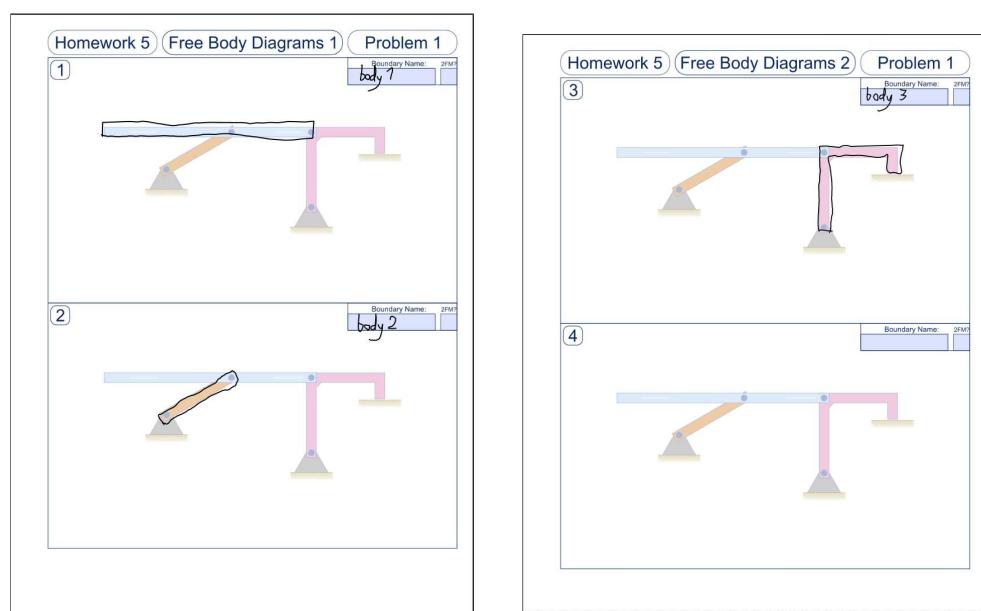
Stage Instructions 1: “Trace boundary or tap Check Work if all traces drawn”

Stage Instructions 2: “In Boundary Name box, write label: <*body name*>”

The Boundary Identification Stage in Newton’s Pen differs in three main ways from the same stage in Newton’s Tablet. Whereas Newton’s Tablet allows the student to use multiple strokes to trace a system boundary, Newton’s Pen only allows single-stroke traces. This eliminates a sub-stage and creates slightly less confusion for the student. Then, after tracing a successful system boundary, the student is prompted with a specific boundary name to write in the Boundary Name Textbox for this boundary. The system does not recognize the label strokes. If the current problem is a multi-body problem, then Newton’s Pen prompts the student to trace a new boundary in a separate free-body diagram workspace.

This stage embodies two logical sub-stages: the Boundary Trace Sub-Stage and the Boundary Label Sub-Stage.

See Section 3.2.3 for a detailed description of the trace recognition algorithm.



(a) Free-Body Diagram Worksheet 1

(b) Free-Body Diagram Worksheet 2

Figure B.4: Boundary Identification Stage Example Work.

B.1.2 Point of Interaction Identification Stage

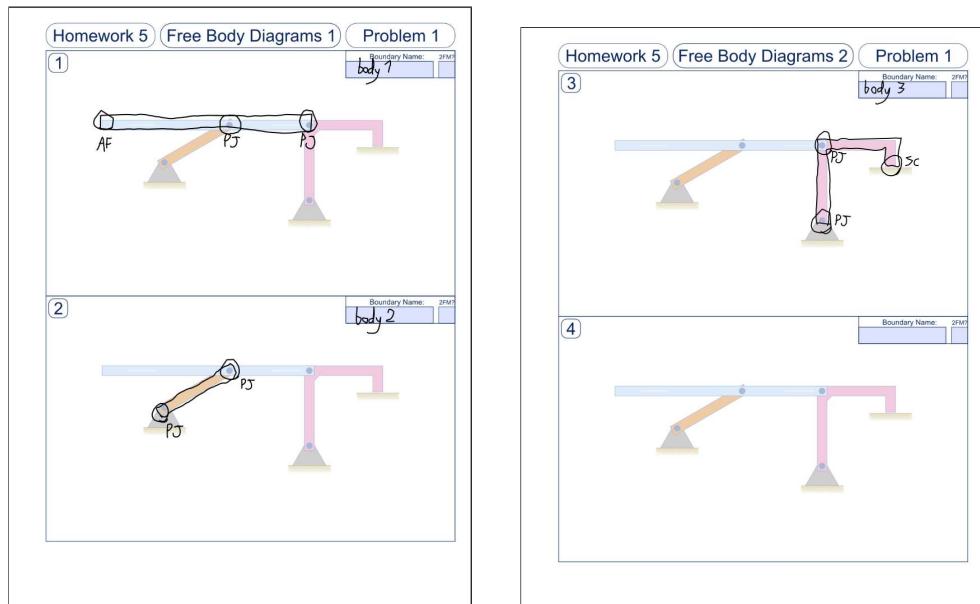
Stage Instructions 1: “Circle POI or tap Check Work if all POIs drawn”

Stage Instructions 2: “Select POI type or tap POI Done if all types selected”

Stage Instructions 3: “Near this POI, write label: <*poi name*>”

The Point of Interaction Identification Stage in Newton’s Pen differs in three main ways from the same stage in Newton’s Tablet. The most significant modification is that Newton’s Pen merges the point-of-interaction location and classification stages into a single stage. Newton’s Pen combines the stages so that the student can label the points of interaction according to their types. This follows the consistent paradigm used throughout Newton’s Pen of drawing a shape and then labeling it, which allows the pen to almost always reference a free-body diagram element by its name. Newton’s Pen also prompts the student to circle the locations of the points of interactions—rather than tapping on them as in Newton’s Tablet. This helps the student to later remember where the points of interaction are located. Lastly, Newton’s Pen prompts the student to label the point-of-interaction circles according to their types. For example, the student is prompted to label the point of interaction of a roller joint as “RJ.”

This stage embodies three logical sub-stages: the Point of Interaction Circle Sub-Stage, the Point of Interaction Button Tap Sub-Stage, and the Point of Interaction Label Sub-Stage.



(a) Free-Body Diagram Worksheet 1

(b) Free-Body Diagram Worksheet 2

Figure B.5: Point of Interaction Identification Stage Example Work.

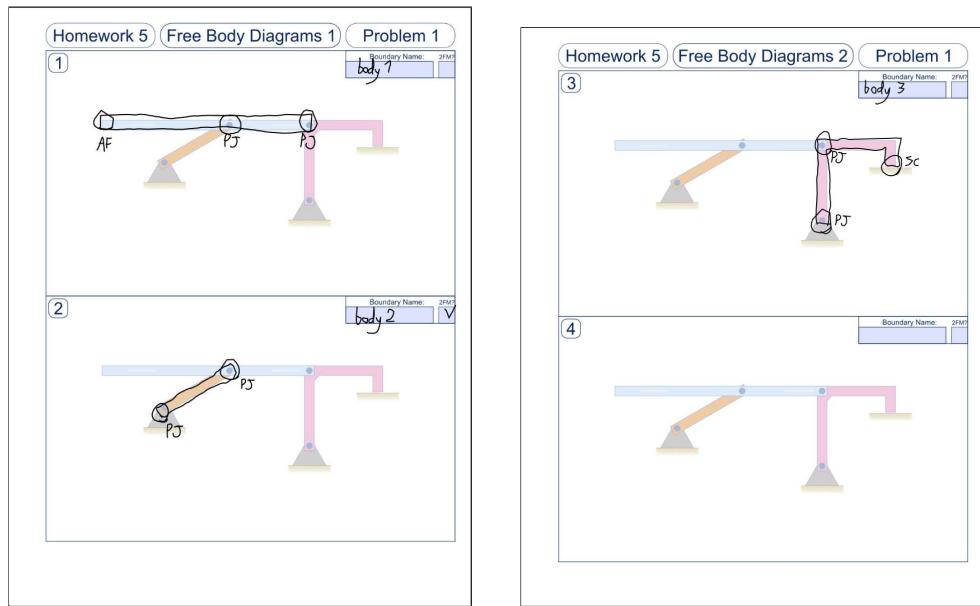
B.1.3 Two-Force Member Identification Stage

Stage Instructions 1: “Tap on a 2FM (if any). Tap Check Work when all 2FMs are identified”

Stage Instructions 2: “Draw a check mark in the 2FM box”

The Two-Force Member Identification Stage in Newton’s Pen differs from the same stage in Newton’s Tablet only in that Newton’s Pen prompts the student to draw a checkmark in the “2FM?” textbox after successfully tapping on the workspace of a two-force member.

This stage embodies two logical sub-stages: the Two-Force Member Tap Sub-Stage and the Two-Force Member Label Sub-Stage.



(a) Free-Body Diagram Worksheet 1

(b) Free-Body Diagram Worksheet 2

Figure B.6: Two-Force Member Identification Stage Example Work.

B.1.4 Newton's-Third-Law Pair Identification Stage

Stage Instructions 1: “Select (tap) 1st POI in a Newton’s 3rd Law pair (if any). Tap Check Work when all pairs have been identified”

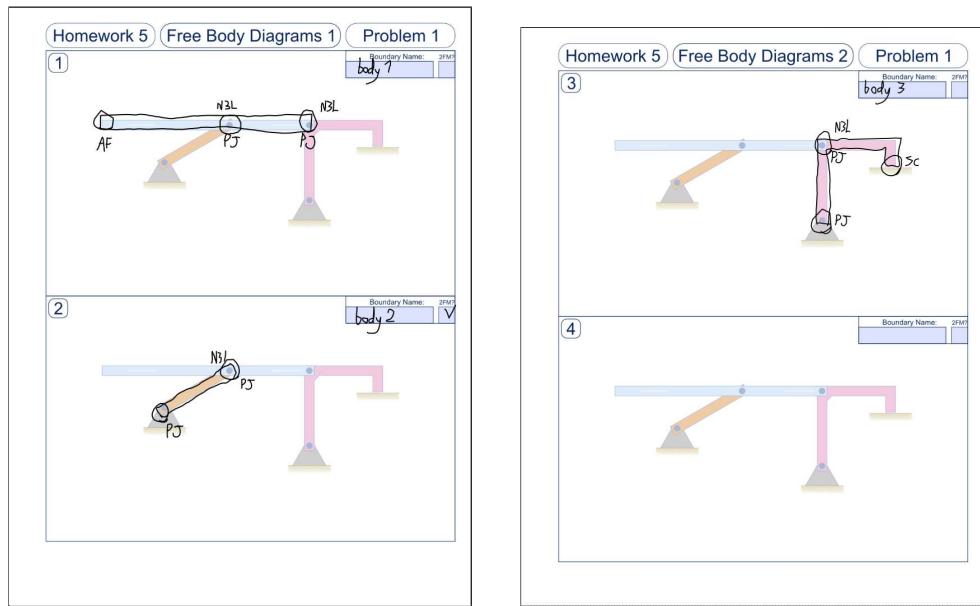
Stage Instructions 2: “Select (tap) 2nd POI in this Newton’s 3rd Law pair”

Stage Instructions 3: “Label 1st POI in this Newton’s 3rd Law pair: N3L”

Stage Instructions 4: “Label 2nd POI in this Newton’s 3rd Law pair: N3L”

The Newton’s-Third-Law Pair Identification Stage in Newton’s Pen differs from the same stage in Newton’s Tablet only in that Newton’s Pen prompts the student to label each point of interaction in a pair with “N3L” after successfully identifying the two points of the pair.

This stage embodies four logical sub-stages: the Newton’s-Third-Law First/Second Tap Sub-Stages and the Newton’s-Third-Law First/Second Label Sub-Stages.



(a) Free-Body Diagram Worksheet 1

(b) Free-Body Diagram Worksheet 2

Figure B.7: Newton's-Third-Law Pair Identification Stage Example Work.

B.1.5 Force Identification Stage

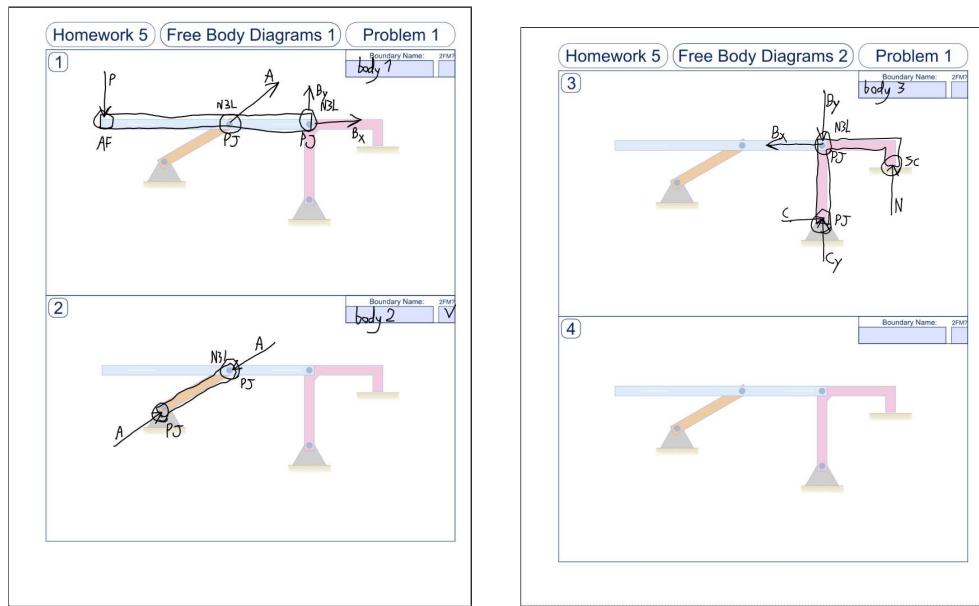
Stage Instructions 1: “With a single stroke, draw force arrow. Tap Check Work when all forces drawn”

Stage Instructions 2: “Near arrow, label this force”

Stage Instructions 3: “Now use the keyboard to enter your text”

The Force Identification Stage in Newton’s Pen differs from the same stage in Newton’s Tablet only in that Newton’s Pen prompts the student to also use the keyboard buttons to enter the textual value of a force’s label immediately after drawing the label. In order to have the student enter the textual value of a force label, Newton’s Pen essentially enters Text Edit Mode (Section [B.2.3](#)).

This stage embodies three logical sub-stages: the Force Arrow Drawing Sub-Stage, the Force Label Drawing Sub-Stage, and the Force Label Keyboard Entry Sub-Stage.



(a) Free-Body Diagram Worksheet 1

(b) Free-Body Diagram Worksheet 2

Figure B.8: Force Identification Stage Example Work.

B.1.6 Equation Stages

We designed additional Newton's Pen stages for guiding the student through the process of solving equilibrium equations after the free-body diagram stages have been completed. However, Livescribe smartpens do not have enough system memory to support the implementation of these stages. Therefore, we left these stages un-implemented, and they will not be described here.

B.2 Newton's Pen Modes

The functional modes of Newton's Pen are similar to those of Newton's Tablet. However, they are slightly more constrained in that the system can only be in a single mode at a given time. That is, the Newton's Pen modes are disjoint. This section now describes each of these modes.

B.2.1 Erase Mode

Erase Mode in Newton's Pen is identical to that of Newton's Tablet (Section A.2.2); only work drawn for the current stage can be erased, and erasing occurs by drawing a stroke which intersects with whatever shape is to be erased. Newton's Pen only allows the student to enter Erase Mode during the Force Identification Stage (Section B.1.5). This is because all other stages obey the principle that the student is only allowed to proceed after drawing correct work for the given sub-stage, and therefore, there never will be any incorrect work to erase. Erase Mode is toggled on and off via the Erase button.

B.2.2 Recent Messages Mode

Recent Messages Mode enables the student to view recent messages, which he/she may have missed. In this mode, Newton’s Pen presents a list of recent messages in reverse chronological order, and this list can be navigated via the directional buttons. The student enters this mode via the Recent Messages button.

B.2.3 Text Edit Mode

Text Edit Mode provides the student with the opportunity to edit the textual value of a label saved by the system. This does not actually prompt the student to change the ink representation of the label, but the student can do so without prompting. In order to enter Text Edit Mode, the student can double-tap on the ink of a previously entered label. In Text Edit Mode, the old textual value of a label is shown with an arrow pointing to the in-progress new textual value—for example, “FX-->F_.”

There are two fundamentally different types of labels in the Newton’s Pen. The first type of label is pre-determined by the system, and the student is prompted with what specifically to draw as the label. This type of label is used for free-body diagram system boundaries, points-of-interaction circles, and Newton’s-third-law pairs. The system does not actually attempt to recognize the ink drawn by the student for these labels, and Text Edit Mode does not apply for this class of labels.

The second type of label is chosen by the student, and the system does need to record the actual textual value of these labels. These are the labels whose textual values the student can later change via Text Edit Mode.

B.2.4 Error Mode

The purpose of Error Mode in Newton’s Pen is also to inform the student of conceptual errors in his/her work for the current stage, but it operates a little differently than Error Mode in Newton’s Tablet (Section A.2.6). Newton’s Tablet uses a spatially oriented approach that shows errors directly within the context of the student’s work. This approach is not feasible with Newton’s Pen, and so it employs a purely textual approach. Section 4.4 describes the conceptual feedback of Newton’s Pen in more detail.

Appendix C

The Conceptual Errors

This chapter enumerates and describes each of the conceptual errors that are evaluated within Newton’s Tablet; Newton’s Pen evaluates very similar concepts. Both systems contain a sequence of progressively more-specific text messages associated with each of these errors. See Sections 3.3 and 4.4 for a detailed discussion of the tutorial feedback systems used in Newton’s Tablet and Newton’s Pen, respectively.

C.1 Boundary Trace Stage Errors

Invalid Body Combination

Newton’s Tablet presents this error when the student traces an invalid combination of bodies during a multi-body problem.

Trace Broken Member

Newton’s Tablet presents this error when the student’s trace cuts through a rigid

member. A trace is determined to break a member if all of its points match to some body, but not all of that body's points match to the trace. See Section 3.14(a) for more detail.

Invalid Single Body Specific Error

Newton's Tablet presents this error when the student's trace matches an incorrect body that has been encoded with a specific error message. This enables the system to present more-specific feedback when the student identifies invalid bodies which are predicted to be particularly troublesome.

Trace Too Short

Newton's Tablet presents this error when the student's trace is not long enough to possibly be a valid trace.

Trace Too Far

Newton's Tablet presents this error when the student's trace does not form even a partial match with any system components.

Trace External Supports

Newton's Tablet presents this error when the student's trace includes external supporting structures. A trace is determined to include external supports if all of the points of some body match to this trace, but not all points of this trace match to that body. See Section 3.14(b) for more detail.

Trace Open Loop

Newton's Tablet presents this error when the student's trace strokes do not form a closed loop.

Trace No Strokes

Newton's Tablet presents this error when the student taps on the Check Work button before drawing any trace strokes.

C.2 Point of Interaction Location Stage Errors

Missing Point of Interaction

Newton's Tablet presents this error when the student leaves out one or more points of interaction.

Invalid Internal Point of Interaction

Newton's Tablet presents this error when the student identifies an invalid internal location that was specifically included in the problem encoding. This enables the system to present more-specific feedback when the student identifies internal locations which are predicted to be particularly troublesome.

Invalid External Point of Interaction

Newton's Tablet presents this error when the student identifies an invalid external location that was specifically included in the problem encoding. This enables the system to present more-specific feedback when the student identifies external locations which are predicted to be particularly troublesome.

Extra Point of Interaction

Newton's Tablet presents this error when the student identifies a location that does not correspond to any actual point of interaction for the system.

C.3 Point of Interaction Classification Stage Errors

Misclassified Point of Interaction

Newton's Tablet presents this error when the student classifies a point of interaction with the wrong type.

Unclassified Point of Interaction

Newton's Tablet presents this error when the student forgets to classify a point of interaction.

Roller Should Be Smooth Contact

Newton's Tablet presents this error when the student incorrectly classifies a point of interaction as a roller joint when the point of interaction is actually the location of a smooth contact.

Smooth Contact Should Be Roller

Newton's Tablet presents this error when the student incorrectly classifies a point of interaction as the location of a smooth contact when the point of interaction is actually a roller joint.

C.4 Two-Force Member Identification Stage Errors

Missing Two-Force Member

Newton's Tablet presents this error when the student leaves out a two-force member.

Extra Two-Force Member

Newton's Tablet presents this error when the student identifies a body that is not a two-force member.

C.5 Newton's-Third-Law Pair Identification Stage Errors

Missing Newton's-Third-Law Pair

Newton's Tablet presents this error when the student leaves out a Newton's-third-law pair.

Extra Newton's-Third-Law Pair

Newton's Tablet presents this error when the student identifies a pair of points of interaction that do not form a Newton's-third-law pair.

C.6 Force Drawing Stage Errors

Missing Force

Newton's Tablet presents this error when the student leaves out a force arrow.

Unmatched Arrow

Newton's Tablet presents this error when the student draws a force arrow too far from any point of interaction.

Wrong Direction

Newton's Tablet presents this error when the student draws a force arrow in the wrong direction.

Wrong Applied-Force Label

Newton's Tablet presents this error when the student labels an applied force—which is given a specific label in the problem description—with the wrong label.

More Than Two Forces

Newton's Tablet presents this error when the student draws more than two arrows over a single point of interaction. A single force never needs to be modeled with more than two arrows. Two arrows are needed in the case of a force that acts in an unknown direction—for example, a pivot force.

Two Arrows Where One Should Be

Newton's Tablet presents this error when the student draws two arrows over a point of interaction that is supposed to be modeled with one force arrow in a known direction.

One Arrow Where Two Should Be

Newton's Tablet presents this error when the student draws only one arrow over a point of interaction that needs to be modeled with two arrows. Forces that act in an unknown direction need to be modeled by two components.

Component Arrows Have Different Labels

Newton's Tablet presents this error when the student correctly draws two force component arrows for a force which acts in an unknown direction, but neglects to give both of these arrows the same base label—for example, “ F_x ” and “ F_y ” would be valid, but “ F_x ” and “ G_y ” would not be valid.

No X-Component Arrow

Newton's Tablet presents this error when the student correctly draws two force component arrows for a force that acts in an unknown direction but does not label either as being the x-component.

X Component Not Horizontal

Newton's Tablet presents this error when the student correctly draws two force component arrows for a force that acts in an unknown direction, and correctly labels one of them as being the x-component, but does not draw this x-component arrow to be aligned with the horizontal axis.

No Y-Component Arrow

Newton's Tablet presents this error when the student correctly draws two force component arrows for a force that acts in an unknown direction but does not label either as being the y-component.

Y Component Not Vertical

Newton's Tablet presents this error when the student correctly draws two force component arrows for a force that acts in an unknown direction, and correctly labels one

of them as being the y-component, but does not draw this y-component arrow to be aligned with the vertical axis.

Two-Force-Member Partner Has Different Label

Newton's Tablet presents this error when the student gives different labels to two arrows which both belong to the same two-force member. These arrows must have equal magnitude, and this is represented by giving them both the same label.

Newton's-Third-Law Partner Has Different Label

Newton's Tablet presents this error when the student gives different labels to two arrows which both belong to the same Newton's-third-law pair. These arrows must have equal magnitude, and this is represented by giving them both the same label.

Newton's-Third-Law Partner Not Opposite

Newton's Tablet presents this error when the student draws two force arrows that belong to the same Newton's-third-law pair in the same direction.

Duplicate Label

Newton's Tablet presents this error when the student gives the same label to un-related force arrows.

Invalid Three-Letter Label

Newton's Tablet presents this error when the student gives a three letter label to a force arrow and the last letter is neither 'x' or 'y'. Labeling a force with more than one or two letters is technically correct, but this would complicate the logic of the system, and students rarely try to give a long label to a force.

C.7 Body Selection Stage Errors

No conceptual evaluation occurs during the Body Selection Stage.

C.8 Equation Type Entry Stage Errors

Repeated Equation Type

Newton's Tablet presents this error when the student attempts to enter the type of an equation which he/she has already began or finished. Instead of re-solving the equation, the student should navigate to his/her previous entry using the equation panel directional buttons.

Invalid Axial Direction

Newton's Tablet presents this error when the student enters some character other than 'x' or 'y' for a force equation. The student should only need to consider force equations aligned with the x-axis or the y-axis.

Invalid Moment Point

Newton's Tablet presents this error when the student enters a character for a moment equation that is not a valid point-of-interaction label.

Invalid Equation Type

Newton's Tablet presents this error when the student enters a character other than 'F' or 'M' as the first letter for the equation type. These two letters represent force equations and moment equations, respectively, and there are no other types of equilibrium equations.

Missing Equals Sign

Newton's Tablet presents this error when the student leaves out the equals sign in the equation type.

Missing Zero

Newton's Tablet presents this error when the student leaves out the '0' as the last letter in the equation type.

C.9 Moment Direction Drawing Stage Errors

No conceptual evaluation occurs during the Body Selection Stage.

C.10 Force Identification 1 Stage Errors

Missing Relevant Force

Newton's Tablet presents this error when the student does not identify any force component at a point of interaction that includes a necessary force for the current equilibrium equation.

Extra Relevant Force

Newton's Tablet presents this error when the student identifies an extra force that is not relevant for the current equilibrium equation.

Too Many Force Components

Newton's Tablet presents this error when the student identifies more force components than are needed for the current equilibrium equation at a given point of interaction.

Wrong Force Component

Newton's Tablet presents this error when the student identifies the wrong force component for the current equilibrium equation at a given point of interaction.

Missing Relevant Force Component

Newton's Tablet presents this error when the student does identify one necessary force component at a given point of interaction, but leaves out another necessary force component at this point of interaction. This is common for moment equations.

C.11 Component Break Down Stage Errors

Force Already Decomposed

Newton's Tablet presents this error when the student draws additional force-component arrows for a force that has already been broken down for a previous equation.

Too Many Matched Arrows

Newton's Tablet presents this error when the student draws more than two force-component arrows for a force that needs to be broken down or any force-component arrows for a force which does not need to be broken down.

Only One Matched Arrow

Newton's Tablet presents this error when the student only draws one force-component arrow for a force that needs to be broken down.

Arrow Matched to Unknown Direction Force

Newton's Tablet presents this error when the student draws a force-component arrow

for a force that acts in an unknown direction. The student has already modeled such a force with two force-component arrows.

Arrow Matched to axially aligned Force

Newton's Tablet presents this error when the student draws a force-component arrow for a force that is axially aligned. Such a force only acts in the direction of a single axis, and, therefore, does not need to be broken down.

Not Axially Aligned

Newton's Tablet presents this error when the student draws a force-component arrow that is not horizontal or vertical.

Opposite Direction

Newton's Tablet presents this error when the student draws a force-component arrow in the opposite direction than it should be drawn.

Two Matched Horizontal Arrows

Newton's Tablet presents this error when the student draws more than one horizontal force-component arrow at a single point of interaction.

Two Matched Vertical Arrows

Newton's Tablet presents this error when the student draws more than one vertical force-component arrow at a single point of interaction.

Invalid Label Base

Newton's Tablet presents this error when the student labels a force-component arrow

with a base letter that does not match the label of the parent force, which is being broken down.

Invalid Label Subscript

Newton's Tablet presents this error when the student labels a force-component arrow with a subscript letter other than 'x' or 'y'.

Horizontal Component Not X

Newton's Tablet presents this error when the student labels a horizontal force-component arrow as 'y'.

Vertical Component Not Y

Newton's Tablet presents this error when the student labels a vertical force-component arrow as 'x'.

Must Break Down Force

Newton's Tablet presents this error when the student leaves out the force-component arrows for a force that must be broken down.

C.12 Force Identification 2 Stage Errors

The potential conceptual errors for the Force Identification 2 Stage are the same as those for the Force Identification 1 Stage.

C.13 Moment Arm Drawing Stage Errors

Not Perpendicular

Newton's Tablet presents this error when the student draws a moment-arm-distance bracket that is not perpendicular to the force to which it belongs.

Incorrect Length

Newton's Tablet presents this error when the student draws a moment-arm-distance bracket that is too long or too short.

Cannot Identify Moment Force

Newton's Tablet presents this error when the student draws a moment-arm-distance bracket that does not relate at either end to any relevant force.

Incorrect Relationship to Moment Point

Newton's Tablet presents this error when the student draws a moment-arm-distance bracket, which does indeed relate to a relevant force at one end, but does not relate to the point around which moments are being considered for the current equilibrium equation.

Missing Moment Arm

Newton's Tablet presents this error when the student leaves out a necessary moment arm.

Extra Moment Arm Bracket

Newton's Tablet presents this error when the student draws a moment-arm-distance bracket that is not matched to any relevant force or point of interaction.

Inappropriate Bracket Label

Newton's Tablet presents this error when the student gives an invalid label to a moment arm. A moment arm label may be invalid if it is the same as the label for a force or the label for a dimension or angle given in the problem description.

Same Moment Arm Label

Newton's Tablet presents this error when the student gives the same label to multiple moment arms.

C.14 Symbolic Term Entry Stage Errors

Sign Missing

Newton's Tablet presents this error when the student leaves a sign textbox empty.

Incorrect Sign

Newton's Tablet presents this error when the student enters the wrong sign for the given term.

Incorrect Character in Sign Box

Newton's Tablet presents this error when the student enters a character other than '+' or '-' in a sign textbox.

Incorrect Coefficient

Newton's Tablet presents this error when the student enters the wrong coefficient in a term textbox. This error rarely occurs, because equilibrium equations typically do not involve coefficients.

Missing Force

Newton's Tablet presents this error when the student leaves out a required force label from a term textbox.

Missing Trig

Newton's Tablet presents this error when the student does not enter any trigonometric function within a term textbox that needs one.

Unnecessary Trig

Newton's Tablet presents this error when the student includes an extra trigonometric function within a term textbox.

Incorrect Trig

Newton's Tablet presents this error when the student enters the wrong trigonometric function within a term textbox.

Incorrect Angle

Newton's Tablet presents this error when the student enters a wrong angle as the argument for a trigonometric function.

Function Missing Angle

Newton's Tablet presents this error when the student leaves out the angle argument from a trigonometric function.

Missing Moment Arm Label

Newton's Tablet presents this error when the student does not enter any moment-arm label within a term textbox that needs one.

Unnecessary Moment Arm

Newton's Tablet presents this error when the student includes an extra moment-arm label within a term textbox.

Incorrect Moment Arm Coefficient

Newton's Tablet presents this error when the student enters the wrong moment-arm label within a term textbox.

Missing Moment Arm

Newton's Tablet presents this error when the student does not enter any moment-arm label within a term textbox that needs one.

Blank Term

Newton's Tablet presents this error when the student leaves a term textbox blank but does enter a sign in the corresponding sign textbox.

Too Many Terms

Newton's Tablet presents this error when the student enters text in textboxes for more terms than the current equation includes.

Too Few Terms

Newton's Tablet presents this error when the student does not enter text in textboxes from enough terms for the current equation.

Blank Force Term

Newton's Tablet presents this error when the student leaves one of the term textboxes blank for a moment equation but does enter both a sign in the corresponding sign textbox and a moment-arm component in the other corresponding term textbox.

Blank Moment Term

Newton's Tablet presents this error when the student leaves a one of the term textboxes blank for a moment equation but does enter both a sign in the corresponding sign textbox and a force component in the other corresponding term textbox.

Expanded Form in Symbolic

Newton's Tablet presents this error when the student enters expanded term content within a symbolic term textbox.

Duplicated Term

Newton's Tablet presents this error when the student enters the same text within more than one term textbox.

Invalid Character in Equation

Newton's Tablet presents this error when the student enters an invalid character within a term textbox.

Valid Force for Another Equation

Newton's Tablet presents this error when the student enters a force label from the free-body diagram in a term textbox, but the specific force is not involved in the current equilibrium equation.

Non-broken-Down Force

Newton's Tablet presents this error when the student enters the label of an angular force, when the term expects the label of one of the broken-down components of that angular force.

Term Switched

Newton's Tablet presents this error when the student pairs a force component that is involved in the current moment equation with a moment-arm component which is also involved with the current moment equation, but these components need to be paired with some other moment arm and force. For example, Newton's Tablet presents this error if the student enters “ $-M_1 \cdot F_2 - M_2 \cdot F_1$ ” when the correct moment equation is “ $-M_1 \cdot F_1 - M_2 \cdot F_2$.”

Wrong Broken Down Force Used

Newton's Tablet presents this error when the student enters the vertical component of a force when the system expects the horizontal component of that force or vice versa.

Force Not in Equation Direction

Newton's Tablet presents this error when the student enters a force that acts in the y direction in the equilibrium equation for forces in the x direction or vice versa.

Sign in Component Box

Newton's Tablet presents this error when the student has included only a sign character within a term textbox.

C.15 Expanded Term Entry Stage Errors

The potential conceptual errors for the Expanded Term Entry Stage are the same as those for the Symbolic Term Entry Stage.

Appendix D

The Problems

This appendix begins with a discussion of the problem encodings used with the Newton’s Tablet and Newton’s Pen programs (Section [D.1](#)). This is followed by a description of each of the actual problem set used with these programs (Section [D.2](#)).

D.1 The Problem Encodings

Manually encoding a problem for Newton’s Tablet or Newton’s Pen is a time-consuming and error-prone task, and so an additional program was developed in C# for the sole purpose of assisting in the process of creating and editing the problem-encoding files for these systems. Figure [D.1](#) shows a screenshot of this program.

One notable aspect of the problem encodings is that a polygon for each possible combination of adjoining bodies in the system is encoded as an invalid boundary. The trace recognition algorithm considers these invalid boundaries as well as the valid ones in order

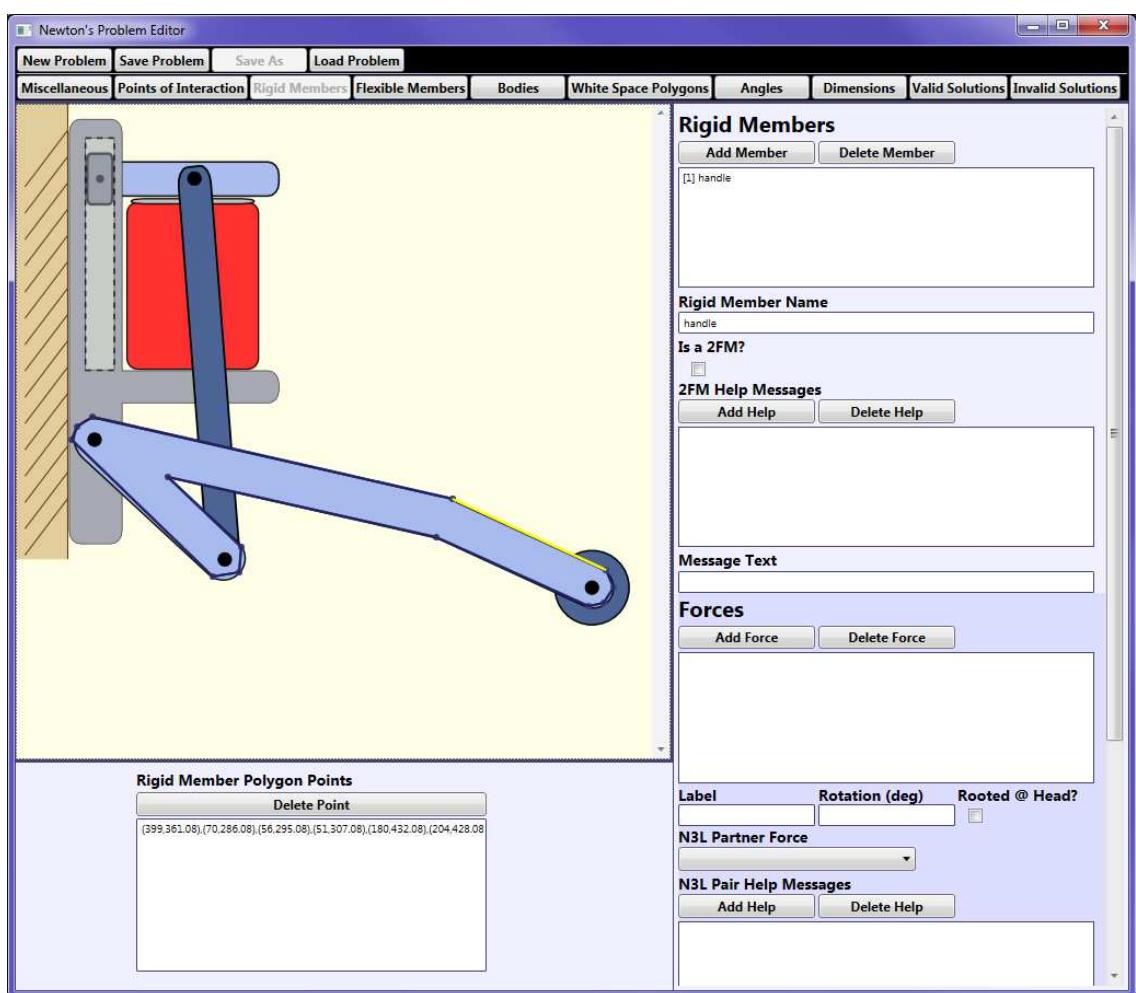


Figure D.1: The Problem Editor.

to provide more targeted error feedback. We developed a polygon-union algorithm in order to automatically generate all of these possible combinations of adjoining bodies.

The polygon union algorithm starts by checking whether any of the segments in a first polygon intersect with any of the segments in a second polygon. When an intersection is found, a new point is created at this point of intersection. By definition, this new point lies on the edge of both original polygons. For both polygons, the region on one side of this intersection is inside the resulting merged polygon, while the region on the other side of this intersection remains along the edge of the resulting merged polygon. The regions from each which lie inside the resulting merged polygon are discarded, while the regions from each which lie along the edge of the resulting merged polygon are spliced together to form the union of the two polygons.

D.1.1 The Newton’s Tablet Problem Encodings

Each problem in the problem set for Newton’s Tablet has its own XML file that contains all of the relevant data for the problem. Two image files accompany each XML file: one image contains an annotated version of the original problem system and Newton’s Tablet displays it in the problem-description box; the other image contains a version of the problem system without annotations and Newton’s Tablet renders it on the canvas during the Boundary Trace Stage (Section A.1.2).

D.1.2 The Newton's Pen Problem Encodings

The same problem-editing program that was developed to help encode problems for Newton's Tablet also creates a version of the encodings which is used in Newton's Pen. However, Newton's Pen cannot use the same XML encoding structure as Newton's Tablet. Instead, Newton's Pen uses a version of the encoding with all of the relevant data hard-coded directly into a separate Java class file for each problem. See Section 4.3.4 for a discussion of the problems presented by using XML files in Newton's Pen. Newton's Pen also requires more effort in order to create the actual free-body diagram and problem-description worksheets for each problem.

D.2 The Problem Set

We used Newton's Tablet and Newton's Pen for two homework assignments. Problems zero through six are single-body problems, and were used in the first homework set. Problems seven through nine are multi-body problems, and were used in the second homework set.

The reason the first problem is numbered as zero, is because it was used as an example problem. An instructor guided each student using either system through the process of using the system on problem zero.

All of the images are original, but we based the underlying structure of many of these problems upon problems from external sources.

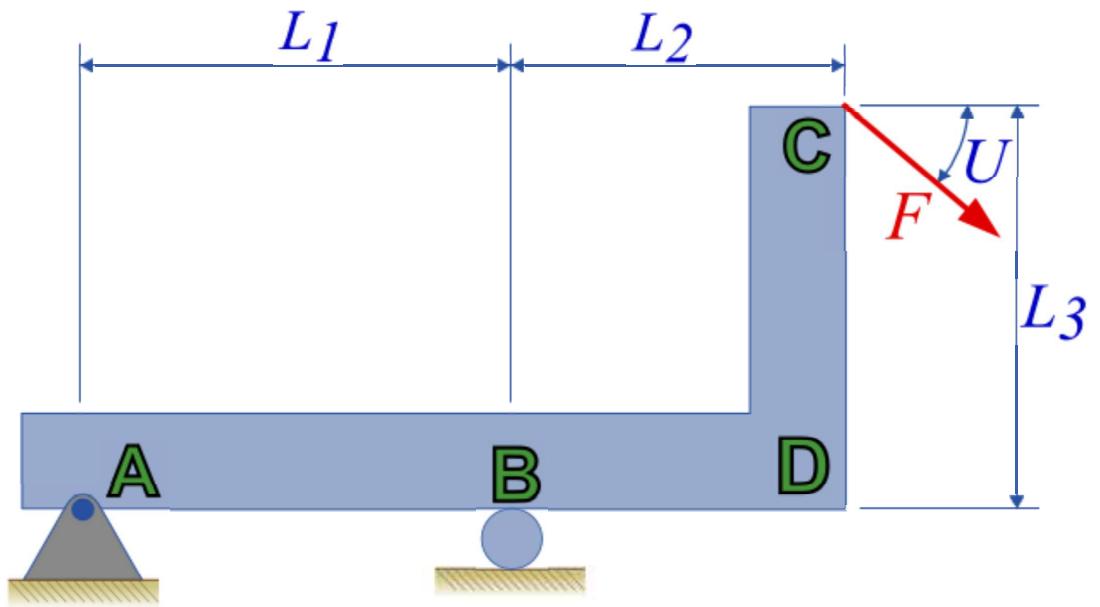


Figure D.2: Problem 0: The L-Beam.

D.2.1 Problem 0: The L-Beam

Problem Description: Assuming the magnitude of force P is known, find the magnitude of the forces at the supports.

This problem is original to this research.

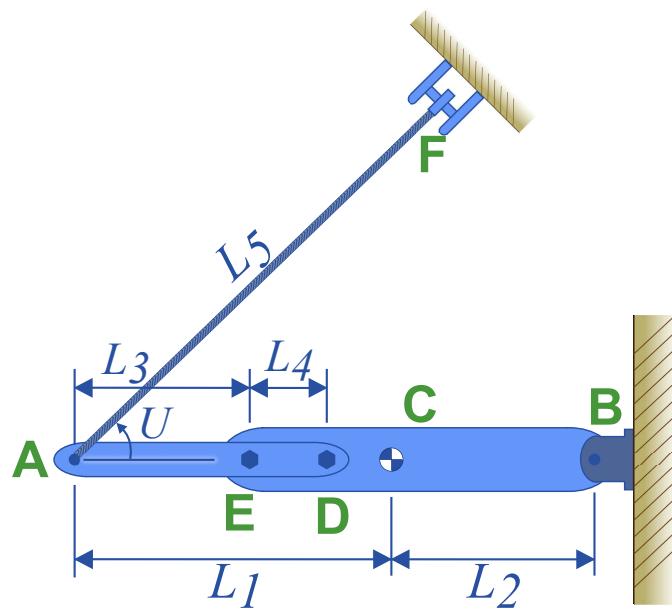


Figure D.3: Problem 1: The Bolted Bar.

D.2.2 Problem 1: The Bolted Bar

Problem Description: Find the magnitude of the tension on the supporting cable and the magnitude of the force on pin B.

This problem is original to this research.

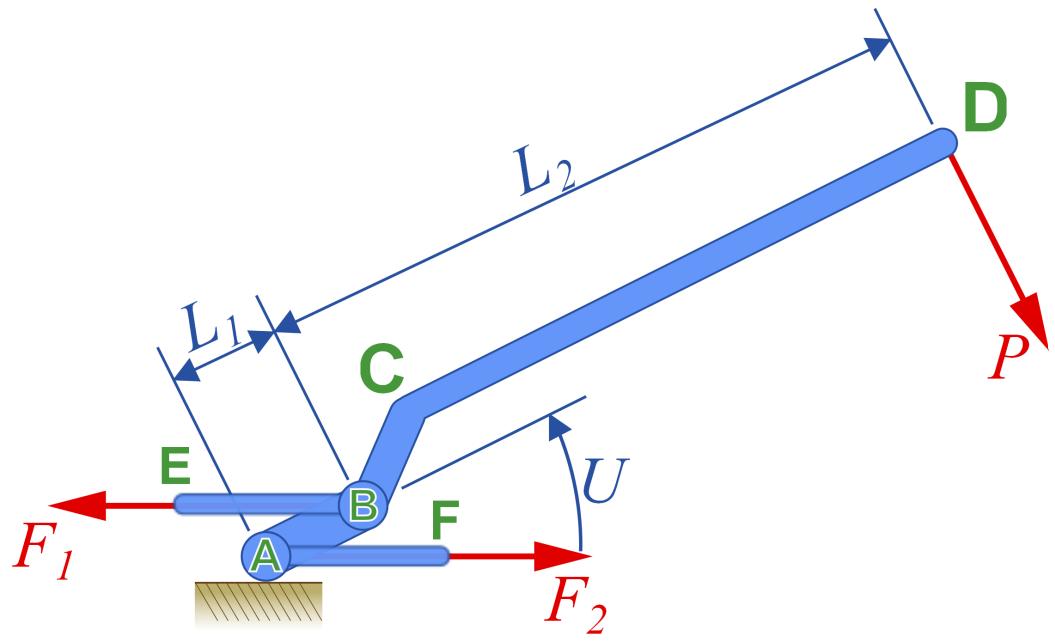


Figure D.4: Problem 2: The Load Binder.

D.2.3 Problem 2: The Load Binder

Problem Description: If tension F_1 and angle U are known, determine the force P and tension F_2 needed for this system to be in static equilibrium. Assume the surface under A is perfectly smooth.

This problem is taken from Meriam and Kraige [48].

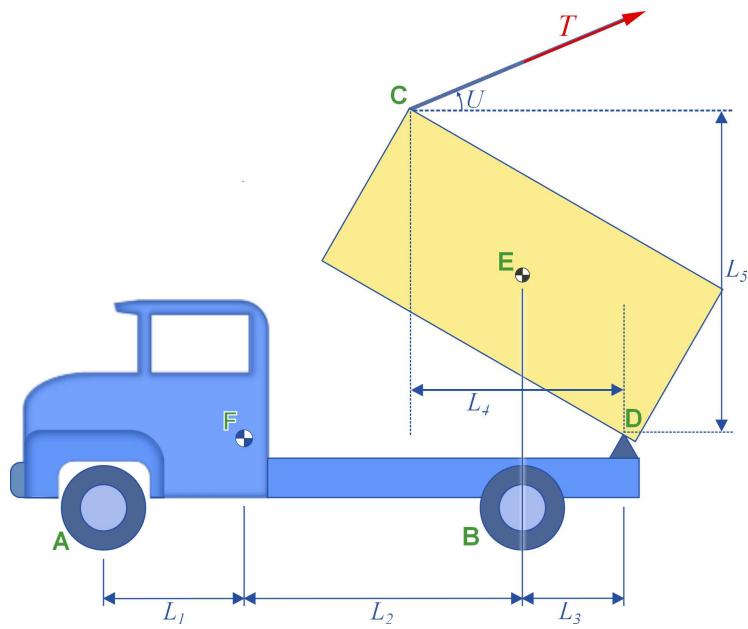


Figure D.5: Problem 3: The Truck.

D.2.4 Problem 3: The Truck

Problem Description: This truck bed has weight W_1 , located at center of mass E. The rest of the truck has weight W_2 with its center of mass located at F. Determine the magnitude of the tension force T and the force at D.

This problem is original to this research.

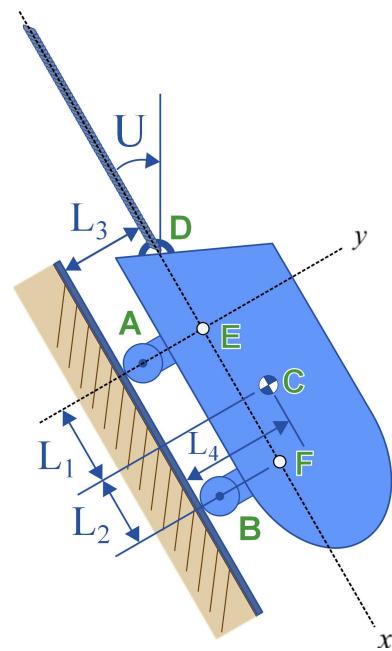


Figure D.6: Problem 4: The Cart.

D.2.5 Problem 4: The Cart

Problem Description: A loading car is at rest on a track forming an angle of U with the vertical. Car has weight W . Determine tension T and the reaction at each wheel.

This problem is taken from Beer et al. [12].

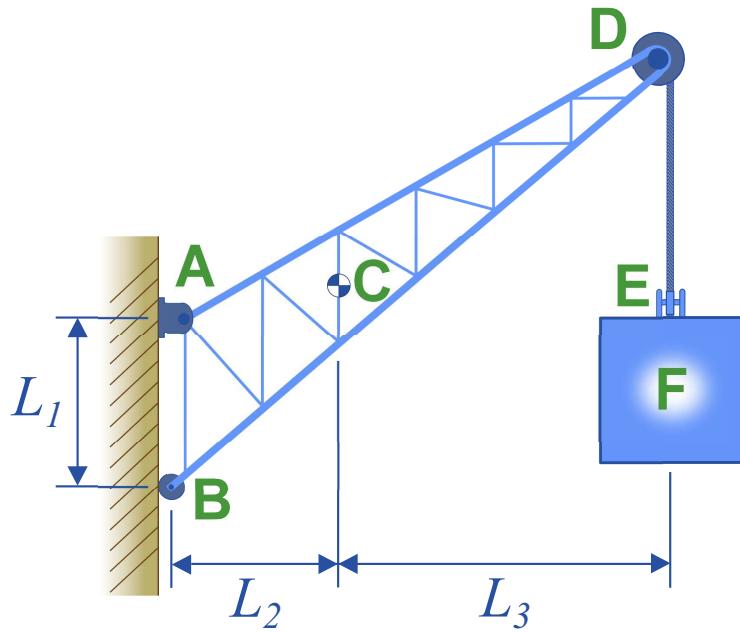


Figure D.7: Problem 5: The Crane.

D.2.6 Problem 5: The Crane

Problem Description: A fixed crane has mass of M_1 and is used to lift a crate of mass M_2 . Determine reactions at A and B.

This problem is taken from Beer et al. [12].

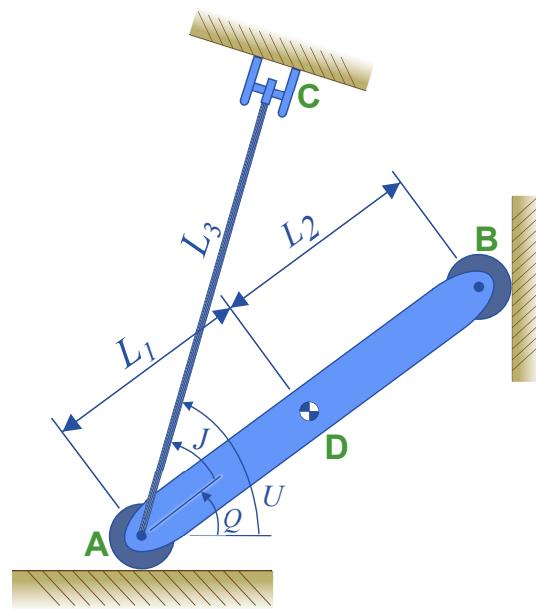


Figure D.8: Problem 6: The Un-Bolted Bar.

D.2.7 Problem 6: The Un-Bolted Bar

Problem Description: This bar with a roller at A and B is simply resting against the vertical surface at B. Find the mass of this bar. Tension(T) in the cable is known. Note: $U - Q = J$.

This problem is original to this research.

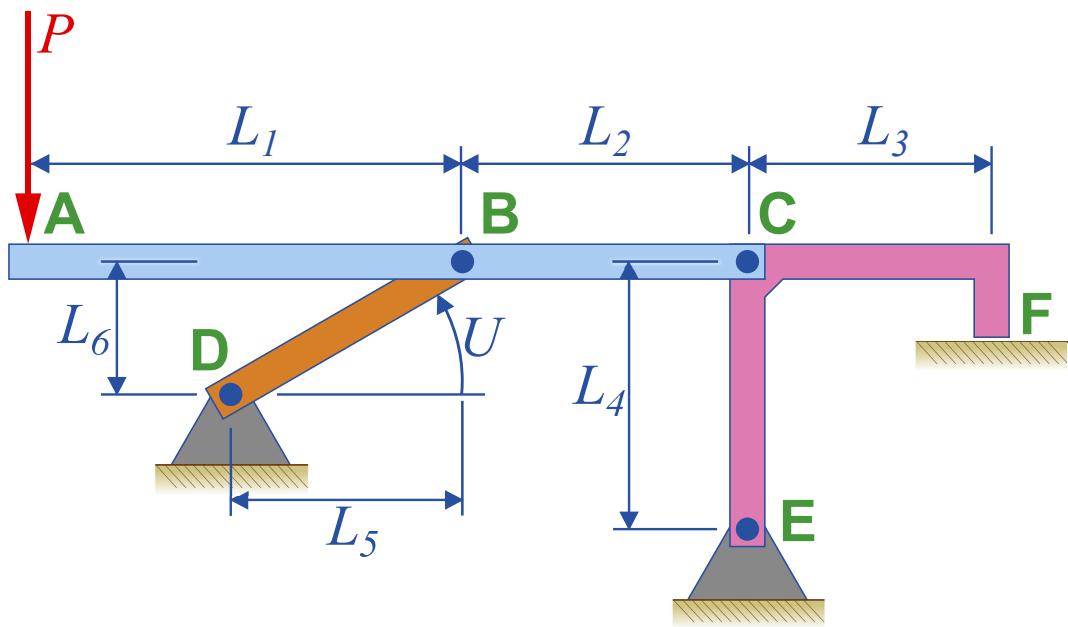


Figure D.9: Problem 7: The Toggle Clamp.

D.2.8 Problem 7: The Toggle Clamp

Problem Description: Determine the magnitude of the clamping force at F when force P is applied to the handle.

This problem is original to this research.

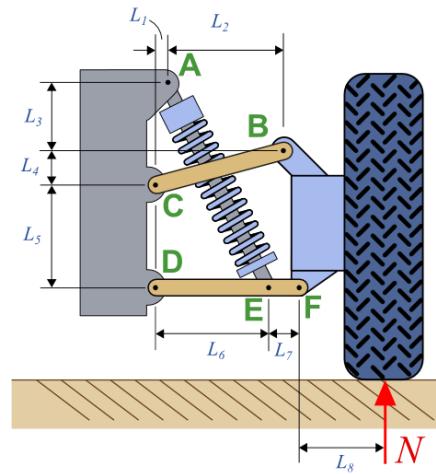


Figure D.10: Problem 8: The Wheel.

D.2.9 Problem 8: The Wheel

Problem Description: The elements of a rear suspension for a front wheel drive car are shown in the figure. Determine the magnitude of the force at each joint if the normal force N exerted on the tire is known.

This problem is taken from Meriam and Kraige [48].

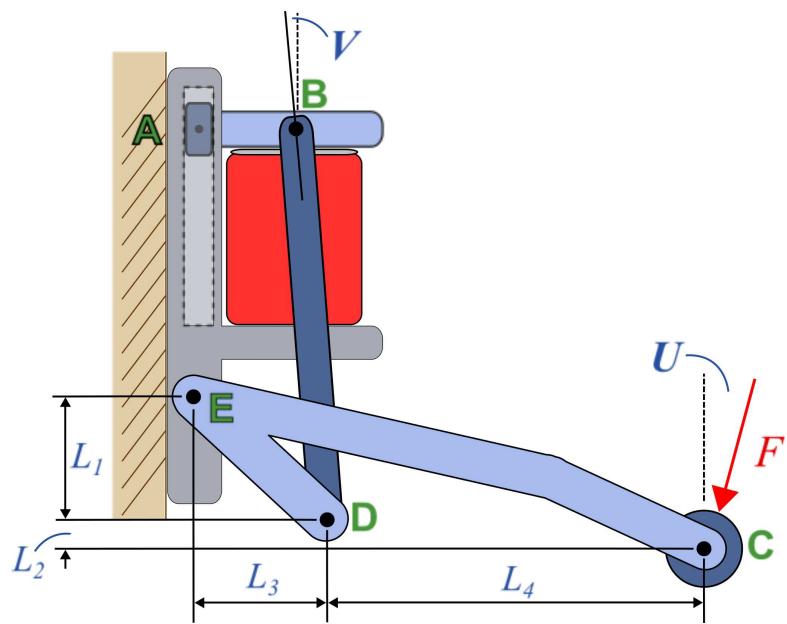


Figure D.11: Problem 9: The Can Crusher.

D.2.10 Problem 9: The Can Crusher

Problem Description: Determine the compression force exerted on the can for an applied force F when the can crusher is in the position shown. Pin B is on the vertical centerline of the can. The small square projection A of the moving jaw moves in a recessed slot in the frame and can be modeled as a slider joint.

This problem is taken from Meriam and Kraige [48].

Appendix E

Demonstrational Videos

Demonstrational videos, which walk through the processes of using the tutorial systems, can be found online at www.smarttools.engr.ucr.edu. There is both a video for Newton's Pen and a video for Newton's Tablet.

Appendix F

The Newton's Pen Documentation

This appendix presents the documentation that we provided to students to be used in tandem with the Newton's Pen software.

F.1 User Manual

This document provides in-depth information about Newton's Pen. This material both describes how to use the system and provides additional explanation for some of the more important statics concepts.

I. Using Newton's Pen

Step 1: Starting Newton's Pen

- a. Navigating to **Newton's Pen**: Tap the center of the Nav Plus on the bottom left-hand corner of the worksheet.

tap center

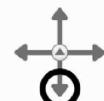


- b. **Main Menu** will be displayed on the Livescribe Smartpen.



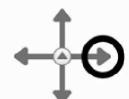
- c. Tap down once on the directional arrow to navigate to **Applications**.

tap down



- d. Tap on the right arrow to navigate the applications and tap right again to run **Newton's Pen**.

tap right



- e. Select your **Problem Description** and corresponding FBD worksheets.

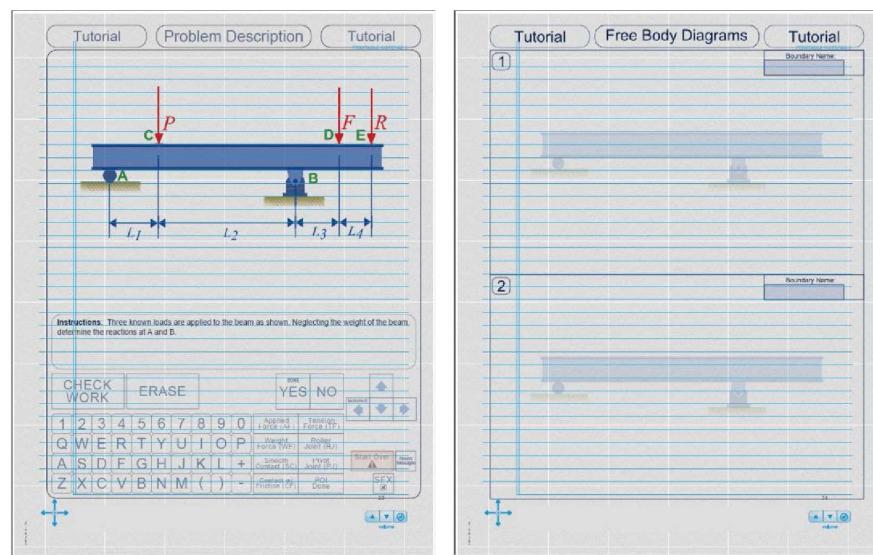
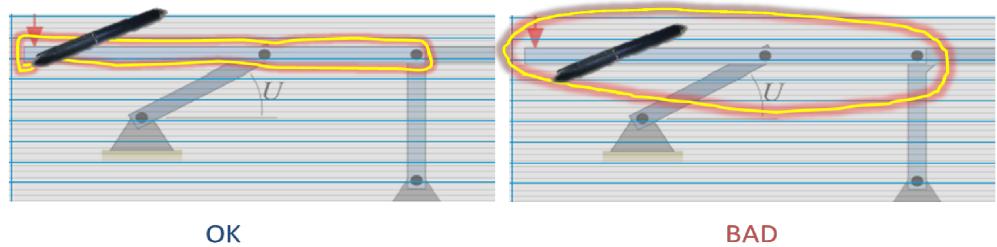


Figure F.1: Newton's Pen User Manual: Page 1

Step 2: Drawing Free Body Diagrams

- a. Using a single pen stroke, trace around the system of interest. Stay near the edges of the system. The trace does not have to be perfect. Trace neatly but **don't go too slow**. Trace recognition will happen once the pen is lifted. To read tips on how to draw FBDs, reference **Common Errors and Help**.



- b. If the system you have traced is valid, advance to **Step 3**. Otherwise, continue below to **Step 2.b.i**.
 - i. Attempt to trace around your system again. Take your time to draw neatly. Reference **Common Errors and Help** if problems persist.
 - ii. If you feel that the workspace is completely cluttered with unrecognized traces, tap the **START OVER** button on the **Problem Description Worksheet**. (Please note that tapping **START OVER** will restart the program, **ALL** work will be lost, and you must begin work in a different workspace. You will be prompted to confirm. Tap **START OVER** again if you want to start over, otherwise tap **NO** or tap in the original workspace.)

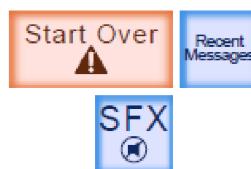
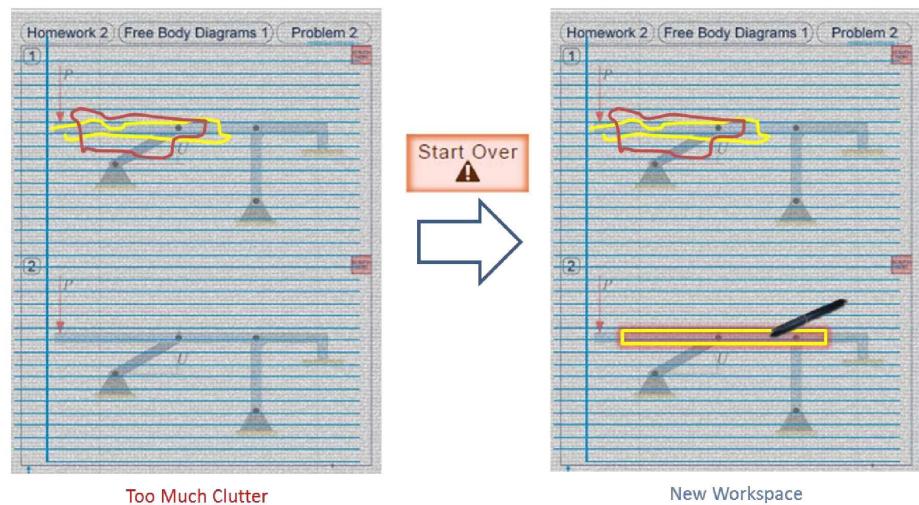


Figure F.2: Newton's Pen User Manual: Page 2



Step 3: Label Traced Boundary

- When prompted by the pen, enter the given ‘Boundary Name’ in the appropriate box.

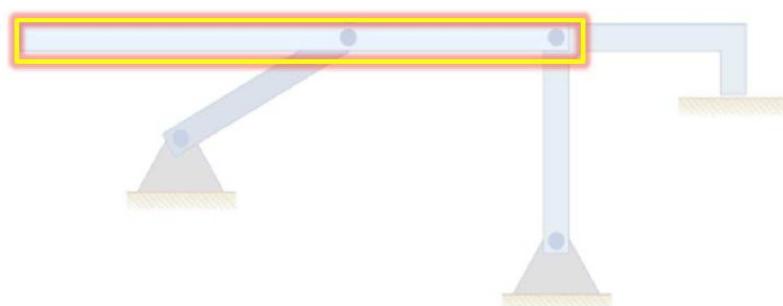
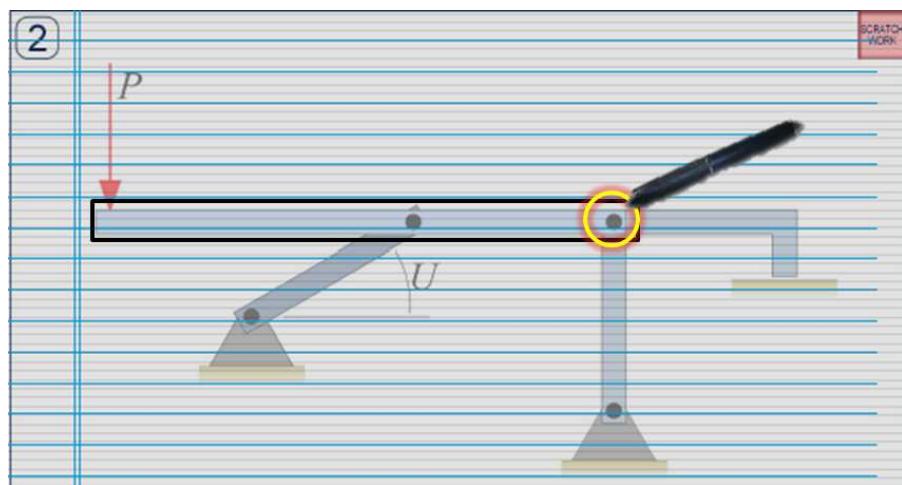


Figure F.3: Newton's Pen User Manual: Page 3

Step 4: Circle Point of Interaction

- Circle a Point of Interaction (POI) relevant to the system you have traced.



Step 5: Select and Label Interaction Types

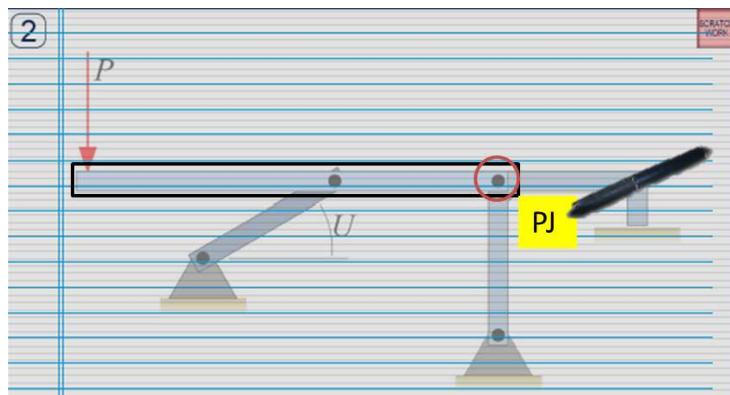
- Using the Interaction Type buttons (see below) on the Problem Description Worksheet, select ALL relevant Interaction Types for the current POI. There may be more than one interaction at a given point. For example, there may be an applied force at the same location as the weight force.

Applied Force (AF)	Tension Force (TF)
Weight Force (WF)	Roller Joint (RJ)
Smooth Contact (SC)	Pivot Joint (P.J.)
Contact w/ Friction (CF)	POI Done

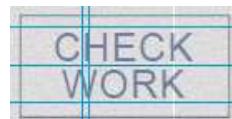
- Once you have selected all the interaction types that apply to the point of interaction you are currently working on, tap the **POI Done** button.

Figure F.4: Newton's Pen User Manual: Page 4

- c. Once the correct type(s) is/are chosen, you will be prompted to write the interaction type label next to the point of interaction.



- d. If there are other points of interaction for your traced system, identify them by repeating **Steps 4 and 5**. If you have identified all of the points of interaction, tap the **Check Work** button located on the **Problem Description Worksheet**.



- e. The pen will notify you of any errors you may have made.
 - i. If an error occurred, not all of the points of interaction on the body were circled. Refer to **Common Errors and Help** for information about navigating through the error messages. (*In general, you may use the arrows on the problem description page to navigate through error/help messages. Scrolling up and down will select different errors. Scrolling to the right will give you more information about the current error.*)
 - ii. Tap on the current workspace to exit the error messages. Return to **Steps 4 and 5** to identify the remaining point(s) of interaction.

Figure F.5: Newton's Pen User Manual: Page 5

Step 6: Draw Force Arrow

In this step you will represent each force as a labeled arrow. Each labeled arrow must be completed before beginning the next. Use the interaction types you assigned to each point of interaction to decide which forces are needed on your free body diagram.

- a. Draw an arrow representing a force on your free body diagram. **Draw the arrow with a single-pen stroke. The arrow must be drawn from tail to head.** The pen will report if the arrow was recognized. If it was not recognized, draw it again. Refer to **Common Errors and Help** for additional assistance.



- b. Once the arrow is recognized, the pen will prompt you to label it. Write the label next to the arrow. Then, when prompted by the pen, use the keyboard on the problem description page to key in the label you have just written. Tap the Yes (done) button when you have completed keying in the label. **When selecting a force label, you must follow the following rules:**
 - (i) Labels should be only **two characters long**, e.g., F2, BX, and TS.
 - (ii) Except for the names of applied forces, you **cannot reuse labels** that exist in the problem description. For example, in the figure below, you may use "P" to label the applied force, but you cannot use "U" as the label for a force. Similarly, if a point in the figure is labeled as "B", you cannot use "B" as a label. However, you can use "BX" and "BY" as labels.
 - (iii) The **two forces for a pivot**, must have the same first character; the x-component must end in "X", and the y-component must end in "Y"; e.g., "BX" and "BY".

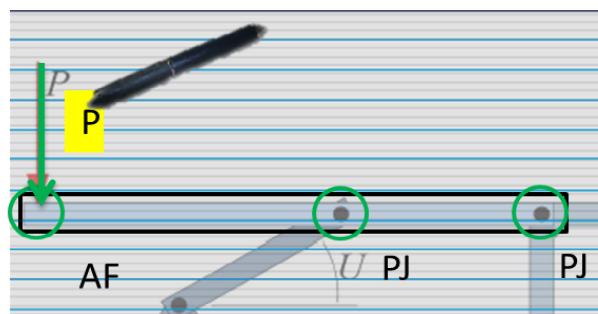
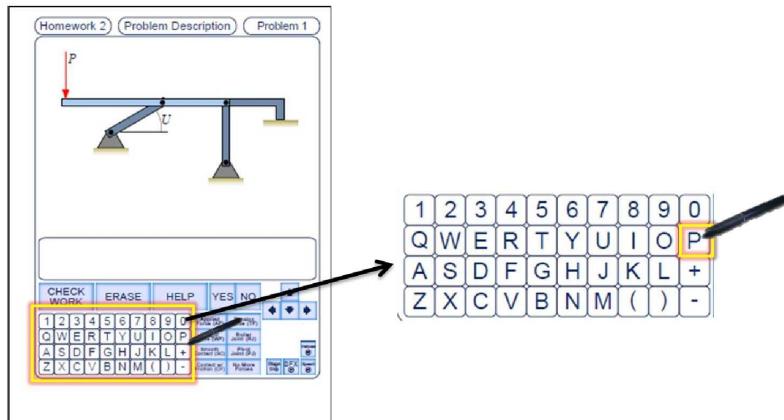


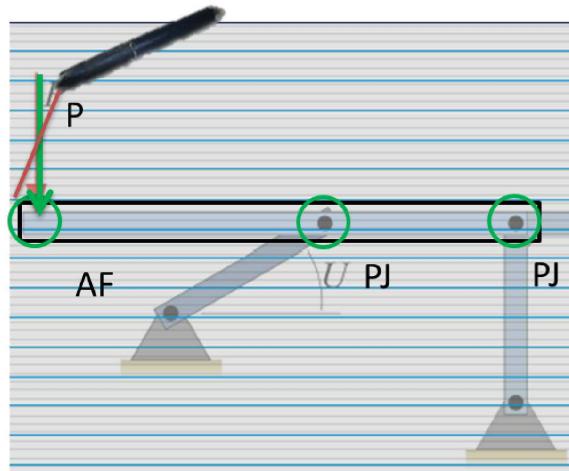
Figure F.6: Newton's Pen User Manual: Page 6



- c. If you want to erase any arrows that you have previously drawn and labeled, set the pen into **ERASE MODE** by tapping the **ERASE** button on the **Problem Description Worksheet**.



Once in **ERASE MODE**, cross out any arrows with a single stroke. Crossing out an arrow also removes the label. **Tap on the ERASE MODE BUTTON to exit ERASE MODE.**



- d. Repeat **Step 6** until all arrows have been drawn. If you believe you have drawn all the necessary arrows, tap on the **Check Work** button.



Figure F.7: Newton's Pen User Manual: Page 7

- e. If you have made any errors, a list of error messages will be displayed on the pen. Refer to **Common Errors and Help** for information about navigating through the error messages.
- (In general, you may use the arrows on the problem description page to navigate through error/help messages. Scrolling up and down will select different errors. Scrolling to the right will give you more information about the current error.)*

Otherwise, the pen will let you know that you have completed the free body diagram!

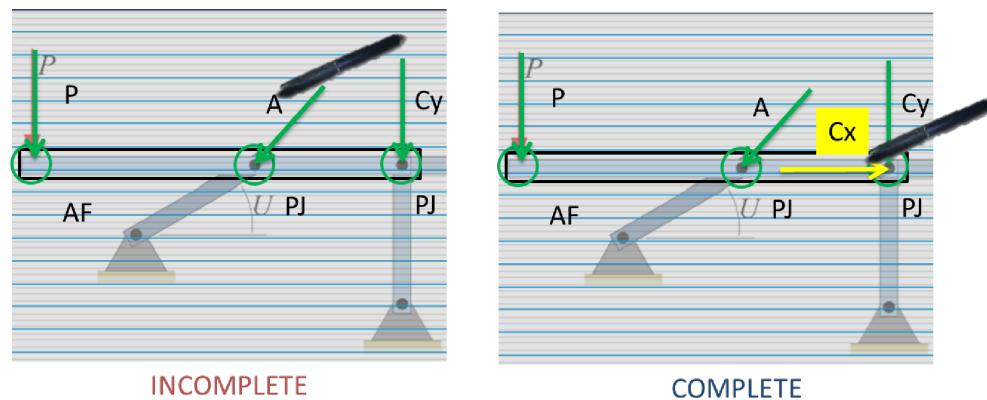


Figure F.8: Newton's Pen User Manual: Page 8

II. Concepts Covered in Newton's Pen

Helpful Tips!

A. How to Draw Free-body Diagrams

Step 1: Decide which system to isolate. Your system should involve at least one of the desired unknown quantities.

Step 2: Isolate your system by tracing its boundary. The boundary should separate the system from all other bodies, including those that support it.

Step 3: Identify points at which external objects or fields (e.g., gravity) interact with the system. Identify the types of interactions that occur at each such point.

Step 4: Represent the forces at the interaction points with force arrows.

B. Force

A force is the action of one body on another. In the example below, an operator applies a force to the system at point A.

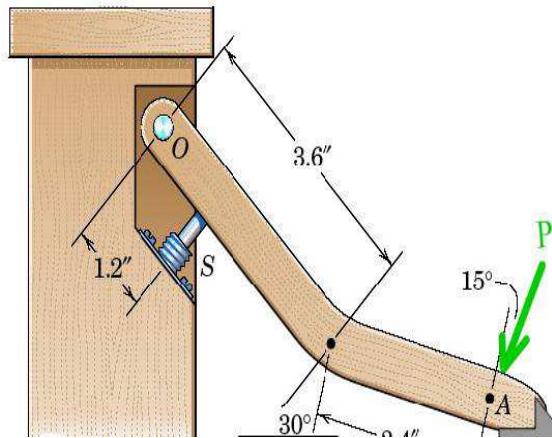
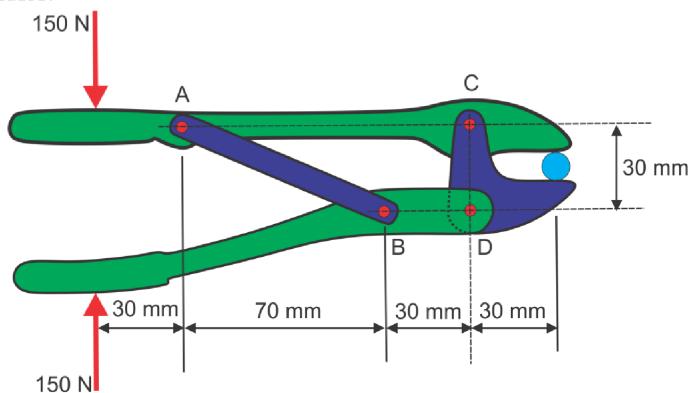


Figure F.9: Newton's Pen User Manual: Page 9

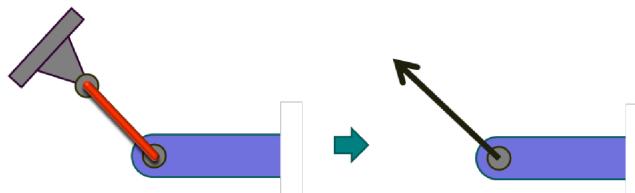
C. Applied Force

Applied forces are external forces that are imposed on the system by an external agent, such as a human, a motor, or a hydraulic cylinder. In this example, the user of the bolt cutters applies forces to the handles.



D. Flexible Elements

If your system boundary “cuts through” a flexible element, this will reveal a force. The force is aligned with the direction of the flexible element. For cables and ropes, the force is always a tension force. For springs, the force can be either tension or compression.



E. Gravity

The gravitational force on an object can be represented as a single weight force applied to the center of mass of the object. The force always points in the direction of gravity. The magnitude of the weight force is the mass of the object times the acceleration of gravity ($W = mg$).

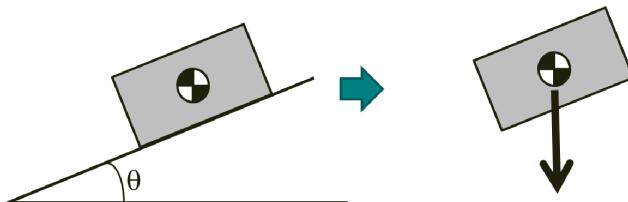
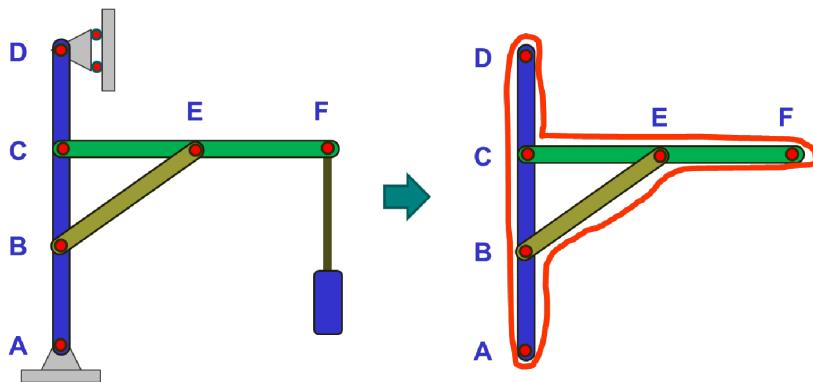


Figure F.10: Newton's Pen User Manual: Page 10

F. Internal/External Forces

Internal forces are forces that are contained within the system boundary. These forces should not appear on the free body diagram. For the red boundary shown at the right, forces at points A, D, and F are external forces and should appear on the free body diagram. Forces at points B, C, and E are internal forces and should not appear on the free body diagram.



The correct free body diagram for this example is:

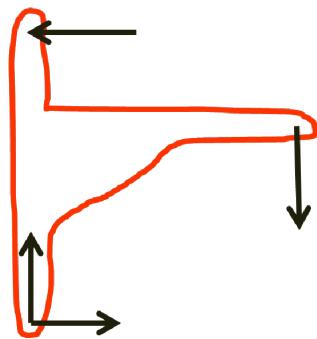
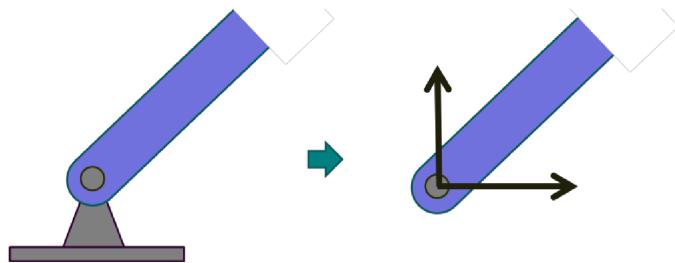


Figure F.11: Newton's Pen User Manual: Page 11

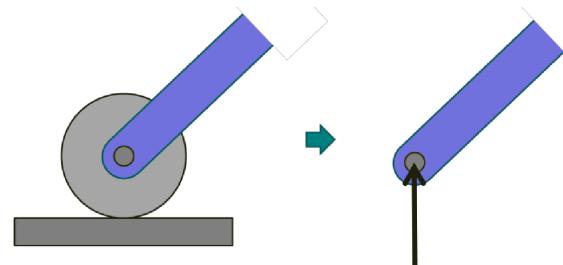
G. Pivots or Pin Joints

Pin joints can exert a force in any direction. The reaction can be expressed as two unknown orthogonal force components.



H. Roller Joint and Smooth Contact

A roller moves freely along the surface it rolls on. The reaction force that the surface applies to the object is normal (perpendicular) to the surface.



Remember, the force is always normal to the surface.

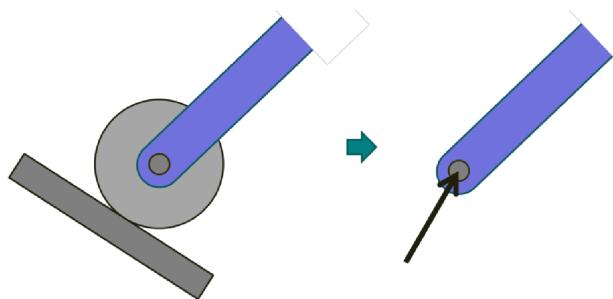


Figure F.12: Newton's Pen User Manual: Page 12

If a body is supported by a smooth surface, the surface provides a normal force just like a roller joint does.

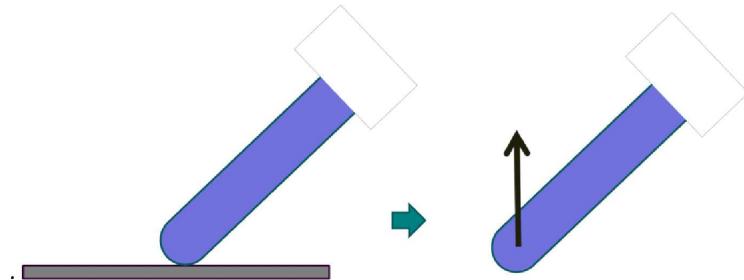
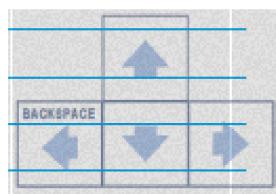


Figure F.13: Newton's Pen User Manual: Page 13

III. Common Errors and Help

A. Navigating through Error messages

1. Use the **arrow buttons on the Problem Description worksheet** to navigate through error messages.



- a. Tap down to scroll through each message.
- b. Tap right to get more detailed information about the current error

B. Problems with Tracing Free-Body Diagrams

1. You have traced an invalid collection of bodies.

- a. The FBD trace drawn below contains external elements which are not part of a valid FBD.

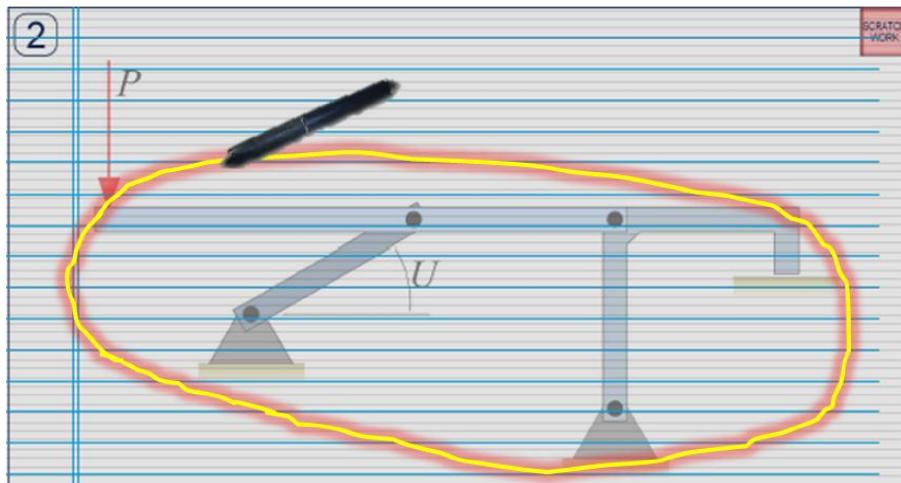
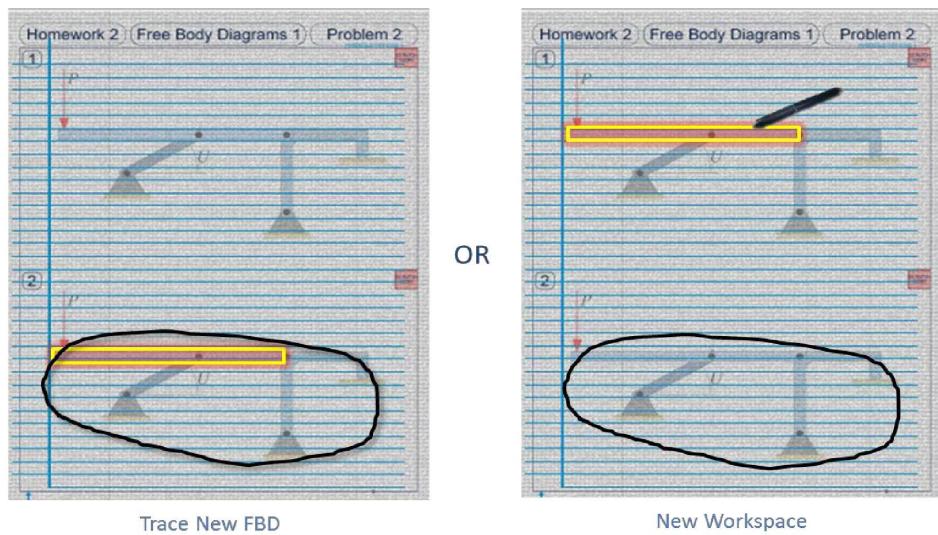


Figure F.14: Newton's Pen User Manual: Page 14

To remedy this problem, either trace a valid FBD on this workspace or trace a FBD on a new workspace.



2. You have broken a member.

a. Breaking members reveals complex internal forces.

To fix this, trace around a complete rigid body.

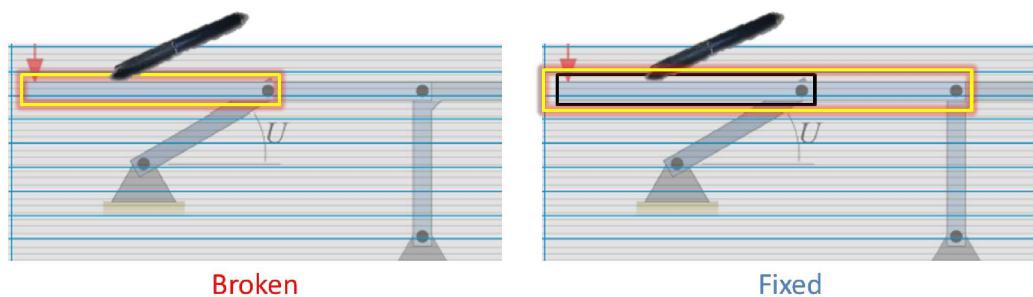


Figure F.15: Newton's Pen User Manual: Page 15

C. Problems with Drawing Arrows

1. If your arrow is not recognized, review the following suggestions:

- The arrow may be too short or too far from the circled POI.

To fix this, redraw your arrow, making sure that it is of an appropriate length and that either the head or the tail of the arrow is inside of the POI circle you have drawn.

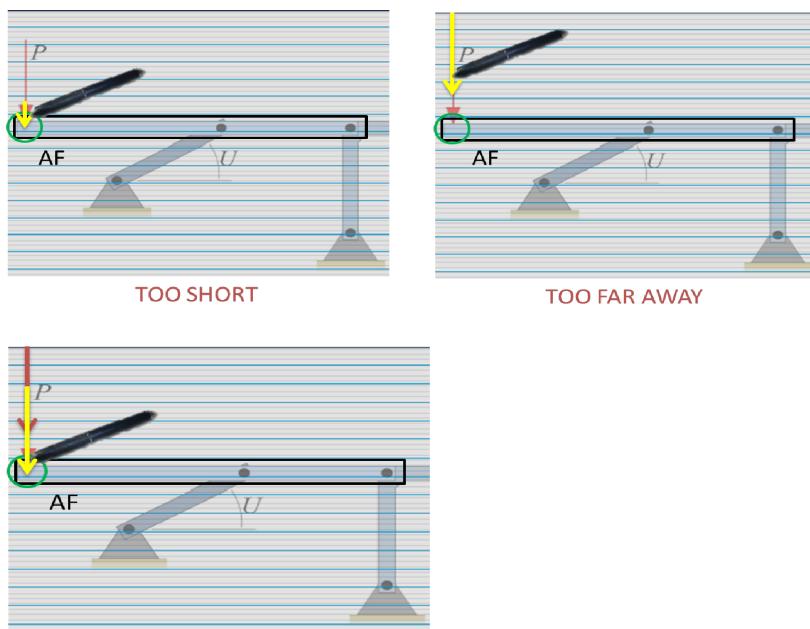


Figure F.16: Newton's Pen User Manual: Page 16

2. If you received any error messages from **Step 6**, review the following suggestions:

- The arrow you have drawn is in an incorrect direction.

To fix this, tap **ERASE MODE** and cross out the arrow with a single stroke. Then, draw the appropriate one.

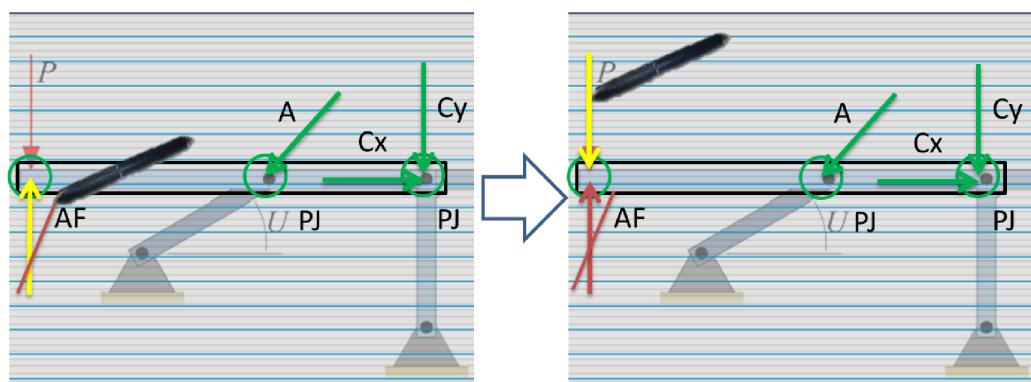
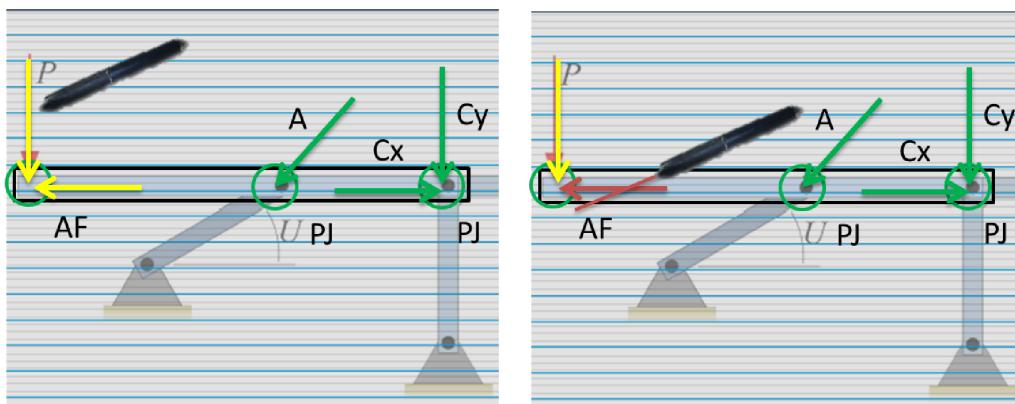


Figure F.17: Newton's Pen User Manual: Page 17

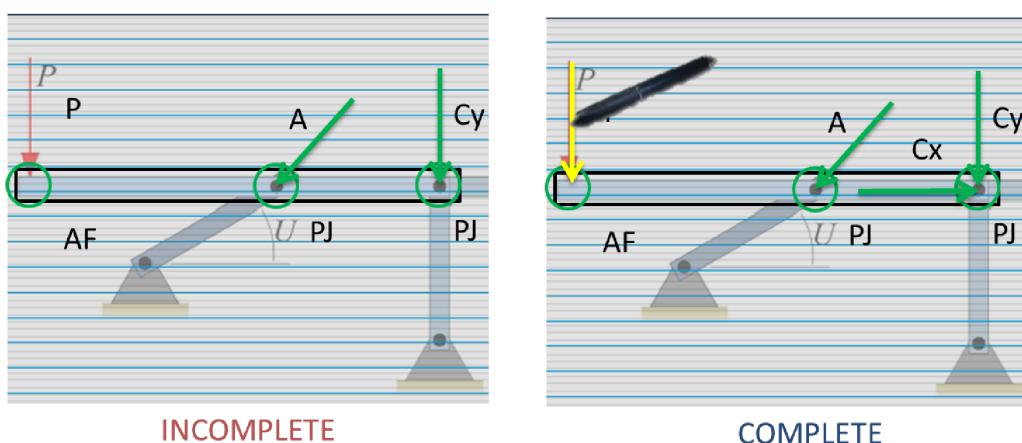
b. There are too many arrows present at a single location.

Tap **ERASE MODE** and cross out the unnecessary arrows with a single stroke.



c. An arrow at a point of interaction is missing.

To remedy this, draw and label the missing arrow(s).



3. Tap the **Check Work** button on the **Problem Description** worksheet once all of your errors have been fixed.



Figure F.18: Newton's Pen User Manual: Page 18

D. Other Buttons on Problem Description Worksheet

- a. The following are the general buttons located on the **Problem Description Worksheet**.



1. Recent Messages:

- a. Tapping **Recent Messages** allows you to navigate between messages that have been previously displayed. Tap on **Recent Messages** again to exit.



2. Sound Effects:

- a. Tapping **SFX** removes the sound effects the pen makes as work is completed throughout the Newton's Pen Tutorial system.

Figure F.19: Newton's Pen User Manual: Page 19

F.2 Quick Reference Guide

This one-page document offers a brief “cheat sheet” with some tips for using the Newton’s Pen system.

Newton's Pen

Quick Reference Guide

1. Read the pen display

- Read the pen display; it will tell you what to do next.
- After writing, always look at the display for the pen's response before continuing.

2. Point of Interaction (POI) and Interaction Types

- There may be multiple interactions to take into consideration at any particular POI.
- Using the buttons on the problem description page, select each of the relevant interaction types. (Make sure the pen responds to each button tap.)
- Tap the "POI Done" button when you have identified all relevant interaction types for the POI you are currently working on.

Applied Force (AF)	Tension Force (TF)
Weight Force (WF)	Roller Joint (RJ)
Smooth Contact (SC)	Pivot Joint (PJ)
Contact w/ Friction (CF)	POI Done

3. Drawing Arrows

- Arrows must be drawn using a single stroke.
- Draw arrows from tail to head.
- Either the tip or the tail of each arrow must be inside a POI circle.
- Make sure arrows are not too short (an inch long is good).
- If your arrow is not recognized, the pen will inform you of this. Simply draw it again.

4. Entering Arrow Labels

- After you write the label on the page, use the keyboard key in the label when prompted to do so by the pen. Tap the "YES" button when you have finished keying in the label.
- Except for the names of applied forces, you cannot reuse labels that exist in the problem description.

5. Pivot Joints (and their labels)

- Pivot Joints are modeled by two forces which are aligned with the x and y axes.
- In accordance to Newton's Pen convention, these arrows must have labels that share a base character and end in X and Y respectively. (e.g., BX and BY)

6. Erasing

- To erase an arrow (which has previously been successfully recognized and labeled), tap the "ERASE" button to enter erase mode.
- Cross out arrows using single strokes.
- Tap the "ERASE" button one more time to exit erase mode.

7. Exploring error messages

- Use the arrows on the problem description page to navigate through error/help messages.
- Scrolling up and down will select different errors.
- Scrolling to the right will give you more information about the current error.



Figure F.20: Newton's Pen Quick Reference Guide