122nd ASEE Annual
Conference & Exposition
June 14 - 17, 2015
Seattle, WA
Making Value for Society
Paper ID #14133

Work in Progress: Flatlab - An interactive learning environment for experiential learning, problem-based assessment, and dynamic instruction in engineering

Prof. Peter Goldsmith P.Eng., University of Calgary

Peter Goldsmith is an Associate Professor in Mechanical Engineering at the University of Calgary. He holds a PhD in Mechanical Engineering from the University of Toronto. His research interests are in human-computer interfaces, control theory, robotics, mechanism analysis and design, applied and pure mathematics, and engineering education.

Work in Progress: Flatlab–An interactive learning environment for experiential learning, problem-based assessment, and dynamic instruction in engineering

Peter Goldsmith
peter.goldsmith@ucalgary.ca
Dept. Mechanical Engineering
University of Calgary

Abstract

The goal of this work in progress is to design a virtual environment that integrates experiential learning with assessment and teaching. The proposed FLATLAB is a Focused Learning, Assessment, and Teaching Laboratory with a visuo-haptic interface. Its design embodies ASK (Assessment, Student, and Knowledge) centered learning. This paper focuses on a FLATLAB module for kinematics and dynamics of mechanisms, which lets students learn experientially by building and manipulating virtual planar mechanisms. The platform is initially being developed on a touch screen interface. The realism will be further enhanced by adding a haptic cursor that provides force feedback to the user's fingertips. Experiential learning in this virtual environment will be integrated with theoretical concepts so that learners will analyze mechanisms to solve for variables of interest, such as displacements, velocities, accelerations, and forces, and use FLATLAB to verify their answers. The FLATLAB platform may be used to create dynamic exams and textbooks in which students can physically interact with exam questions and instructional examples. This work in progress describes the educational requirements of the system and the technical challenges involved in the initial development of the first FLATLAB module.

1 Objectives

Effective instruction can be Assessment-centered (A), Student-centered (S), or Knowledge-centered (K)[1], but these models need not be mutually exclusive. In this work in progress, we propose a visuo-haptic learning platform that integrates these three models into a unified (ASK) paradigm suitable for assessment, study, and teaching.

The platform is called FLATLAB, a Focused Learning, Assessment, and Teaching Laboratory. Each module of this virtual laboratory focuses on explicit learner outcomes for a particular course. For example, the Planar Mechanisms module, which is the main focus of this paper, is for the learning and assessment of concepts in a third-year mechanical engineering course on the kinematics and dynamics of mechanisms. The ‘Learning’ and ‘Teaching’ functions in the FLATLAB acronym reflect the student-centered and knowledge-centered components, respectively, of the ASK paradigm.

While much of the current research on virtual learning environments focuses on immersive 3D environments[2], FLATLAB takes advantage of the fact that many engineering systems have 2D representations that learners can physically interact with through a 2D visuo-haptic display. The user will interact with virtual mechanisms by driving a haptic cursor on the surface of a touch screen with the fingertip or stylus and will feel interactive forces (frictional, inertial, elastic, and constraint forces). These forces will coincide with visual and auditory feedback to create realistic interactions.

This development of this visuo-haptic interactive simulation environment involves three objectives: 1) the development of an interactive simulation environment with a touch screen interface; 2) the design, control, and testing of a haptic manipulator (Hapbot) to realize a haptic cursor; and 3) the development and experimental testing of FLATLAB e-learning modules focused on kinematics, dynamics, and control of mechanisms. FLATLAB educational and training modules will subsequently be developed for other knowledge domains.

The touch screen interface will allow learners to:

1. quickly construct and modify virtual planar mechanisms with touch gestures. Valid mechanism components include rigid links, rotary and prismatic joints, masses, springs, and friction elements. The interface should be simple and intuitive, so that learners can build any 4-joint mechanism within one minute.
2. manipulate their mechanisms in real time with drag and swipe gestures.
3. select angles and displacements to be superimposed on the virtual mechanisms in real time.
4. add impedance (mass, friction, and springs) to mechanisms and apply feedback control through simulated actuators. They can then interact with the controlled mechanisms through drag and swipe gestures.

The second objective is to develop a haptic cursor that meets the following specifications:

1. Moves on the surface of the touch screen to coincide with the visual contact point.
2. Interacts with the user’s fingertip or stylus.
3. Can apply and resist in-plane (horizontal) forces up to 20 N.
4. Provides high haptic stiffness in constrained directions.
5. Provides haptic transparency (low stiffness) in unconstrained directions.
6. Can simulate the stiffness, friction, and inertia of virtual objects and mechanisms manipulated by the user.
7. Produces minimal visual occlusion of the display.

This requires the development of a haptic manipulator to drive the cursor and is being pursued in parallel with the development of the touch screen interface.

The third objective is to develop the E-learning functionality of FLATLAB as a Focused Learning, Assessment, and Teaching Laboratory that emphasizes concepts, problem solving, critical thinking, and design. The initial focus is on a module for kinematics, dynamics and control of mechanisms. The kinematics module will be developed first, with the following main subgoals:

1. Display mathematical relations between learner-selected input variables (e.g. position, velocity, acceleration) and selected output variables, along with explicit derivations of these relations.
2. Create tools for creating dynamic lessons with embedded interactive mechanisms.
3. Create tools for creating interactive problem exercises and exams with automatic assessment.
4. Create tools for adaptive tutorials with embedded interactive mechanisms and hyperlinks to the relevant theory.

2 Literature Review

A review of interactive simulations in engineering education[3] includes several for mechanical engineering subjects: engineering graphics[4], mechanics[5], statics[6], gas turbines[7], aircraft design[8], and system dynamics[9], which emphasizes interactive Problem-Based Learning (PBL). A Matlab/Simulink based interactive learning module for control systems is described in[10].

This work aims to improve on current interactive learning environments from both a technological standpoint and in terms of educational features. The main technological innovations are the touchscreen-driven interactive simulation and the addition of force feedback through a haptic cursor. A touchscreen interface will provide a more natural and convenient method for building and manipulating mechanisms than traditional mouse-driven interfaces. Also, whereas most linkage animation programs use sliders to control the motion of input links, this interface will allow users to manipulate a mechanism directly by dragging any link.

Some of this functionality has been achieved by dynamic geometry systems.The application of such systems to the interactive analysis of mechanisms has been previously evaluated in[11], which considered the systems GEONExT, GeoGebra, SAM, OpenEuclide, C.a.R., and Cinderella. A touch-based dynamic geometry system called GeometryTouch recently introduced in[12] addresses the challenge of making precise point selections via touch. Although these dynamic geometry packages can create interactive planar mechanisms, they are not specifically designed for this. For example, four-bar linkages constructed in Cinderella must be driven by a particular link, and selecting a different driving link requires reconstructing all of the links in a different order. A relaxation approach for simulating the kinematics of mechanisms appears in[13]. In[14], it is shown that a geometric constraint engine using symbolic reasoning to satisfy each constraint incrementally yields simulations that are more robust and substantially more efficient than numerical approaches.

While the touchscreen interface will provide a natural and highly interactive interface. The addition of force (haptic) feedback to the user's fingertips (or stylus) will create more realistic interactions. A haptic force display system conceived for mechanism design applications is described in[15] and used to simulate a flight joystick using admittance control of a commercial manipulator. A similar system in[16] enables interaction with virtual mechanisms from mechanical CAD drawings. In these systems, the haptic device is separated from the visual display. A more advanced though more challenging approach is mixed reality, where the user's hand is colocated with virtual objects in the visual scene[17]. Our proposed visuo-haptic device can achieve this colocation in a simple manner for 2D environments.

A review of haptic interfaces and devices appears in[18]. A commercial haptic device commonly used in haptics research is the Phantom haptic interface[19]. A brief history of the role of such haptic devices in telepresence and virtual reality appears in[20]. A common application is the control of surgical robots and the haptic training of surgeons[21].

A major challenge in haptics is to simulate free motion and rigid constraints with a single haptic device[22]. In a orthopedic surgery application[23], hard surfaces were emulated by dynamically positioning real hard constraint between two manipulator links. A commercial admittance controlled manipulator is described in[24], where it is shown that admittance control can reduce the apparent manipulator inertia by a factor of six, while the apparent friction can be reduced to the accuracy of the force sensor. The effect on system stability of reducing the apparent inertia is analyzed from a theoretical viewpoint in[25].

One way to reduce manipulator inertia is to use a parallel manipulator design. A classification of parallel manipulators having various degrees of freedom is given in[26]. A 3-DOF planar parallel haptic interface is described in[27]. A 3-DOF wire driven planar haptic interface with low inertia is described in[28]. Another approach is to use magnetic levitation[29] to control a magnet that interacts with the user's fingertips.

3 Student-Centered Design

The system requirements are driven by educational needs, which are described here with respect to the three centers of the ASK paradigm, beginning with Student-centered learning. A challenge experienced by many engineering students is that the theoretical concepts can appear abstract and disconnected from experience. Student-centered learning accounts for the background knowledge, understanding, and skills that students bring to a course, including misconceptions[1]. It also motivates the theory in a manner that is meaningful, interesting, and relevant to the learner.

One effective approach for motivating course material is experiential learning. FLATLAB modules will engage learners in hands-on experiences by allowing them to build and test engineering systems in courses such as statics, dynamics, vibrations, mechatronics, control systems, fluid and solid mechanics, thermodynamics, and circuits. The building and manipulation of these systems is done quickly through simple touch gestures, such as tap, drag, and swipe, and multi-touch gestures. Besides building their own systems, students can experiment on preloaded examples of practical engineering systems that serve to motivate the engineering analysis and design of these systems.

The initial focus is on a module for studying planar mechanisms. The required functionality is described here in terms of a four bar linkage example. To construct the mechanism, the student taps the 'pin joint' icon and places it anywhere in the plane (which represents the fixed base link) with a tap gesture. Pinned at this joint, a new link appears, which may be sized and shaped with drag gestures. The new link is immediately active and can be cranked or spun about the joint with a drag or a swipe. The student taps the pin joint icon again to add a new joint to the new link, and then repeats the process. After placing the fourth joint, she completes the mechanism by tapping the 'merge' icon and then tapping the fourth link and the base link (to merge them into a common base link). Any link on the completed mechanism can now be dragged or swiped to drive the mechanism. This is more realistic than existing programs (such as Norton's Linkages software), where a separate slider is needed to control the motion of an input link. In this manner, the student can create any planar mechanism having any number of links and closed loops. Multi-touch gestures could be used to manipulate mechanisms having a mobility greater than two.

The real-time value of any variable of interest can be displayed at its spatial location. For example, the crank and follower angles in a four bar linkage may be labeled with their real-time values. This allows the student to determine experimentally the follower angle corresponding to a given crank angle. Hence, without any mathematics, the student is introduced experientially to a particular problem of mechanism analysis (i.e. positional kinematics) and can solve it experimentally. Since the construction and manipulation of the mechanism is simple and intuitive, these problems can be understood not only by junior engineering students, but also by junior high school students.

Besides motivating theoretical concepts, FLATLAB can foster abstract understanding by stripping away irrelevant features. For example, two links connected by a sliding joint (in the mechanism module) are represented as the two ends of a line having variable length, in contrast to a CAD representation in which a piston slides inside a cylinder or on a surface. This simple representation captures the kinematic symmetry between the two links and makes it easier for students to visualize *inversions* of the mechanism, wherein different links are held fixed. Moreover, it helps students conceptualize planar linkages as polygons in which certain lengths and angles are variable. These polygons may be reduced to triangles and solved by trigonometry.

FLATLAB differs from traditional engineering CAD packages in that its primary purpose is educational, specifically to relate theory to examples. For example, in the mechanism module, students will be able to invert mechanisms instantly using tap gestures. This will help them understand that the equations relating joint displacements in a linkage hold for all of its inversions, even though these inversions appear to yield quite different mechanisms.

The addition of the haptic cursor to the touch screen surface will further enhance the learner's experience. When she uses the haptic cursor to turn a crank, her fingertip will be constrained to move in a circle. If the crank is driving a crank-slider mechanism, she will feel the variable forces of inertia and friction transmitted to the crank as the piston reciprocates.

4 Knowledge-Centered Design

The FLATLAB platform will also embody knowledge-centered instruction, which could also be viewed as teacher-centered since the teacher specifies the knowledge (facts, understanding, and skills) that students must acquire. (The T in FLATLAB stands for Teaching). FLATLAB facilitates the process of setting learning objectives, as each type of problem can be demonstrated and solved experimentally, even before any analytical solution techniques are presented. Students can learn the organization and relevance of these problems, independently of solution methods. Tackling these problems experimentally will also provide intuition about how to solve them analytically, thus preparing students for the requisite theory.

Examples of problem classes in the planar mechanism module include: determining the mobility of planar mechanisms; determining the range of motion, time ratios, and transmission angles in single-loop planar mechanisms, solving for positions, velocities, and accelerations in single-loop planar mechanisms, designing a cam profile to yield a given follower motion, finding speed ratios of ordinary and planetary gear trains; and solving for static and dynamic forces and moments in planar mechanisms.

Lectures and textbooks can also be developed in FLATLAB. For example, students can interact dynamically with an engineering system on tablets as the instructor leads the activity and discussion. The instructor can ask students questions about the system, prompting them to find the answers experimentally.

Similarly, study problems set in FLATLAB provide hands-on interaction with the engineering systems involved. The program can detect an error in the student's solution and prompt him to attempt the problem again. If the student is stuck, FLATLAB can provide hints or solution steps. Students can use the environment to design engineering systems in addition to analyzing given ones.

5 Assessment-Centered Design

Integrating assessment with learning is a key attribute of FLATLAB. The same type of problems may be used for instruction, study, assignments, and exams, with certain features disabled during exams so that students solve them with little or no electronic assistance. This constructive alignment of learning and assessment allows students to evaluate their own competency going into exams. It also brings learner objectives into focus and begs educators to think deeply about the purpose of each course and of engineering education. If the class of problems represented by a module is too narrow and corresponds to a few easily-memorized solution procedures, then the learning will likely be superficial and memory-based. Instead, the class of problems should draw on fundamental knowledge, including basic concepts from prerequisite classes, and these concepts should provide many paths to the correct answer.

If such a platform were widely adopted as an assessment tool, it could reduce variations in learning expectations among instructors and schools. Currently, one instructor might require students in a course on mechanisms to solve any given single-loop mechanism for any kinematic

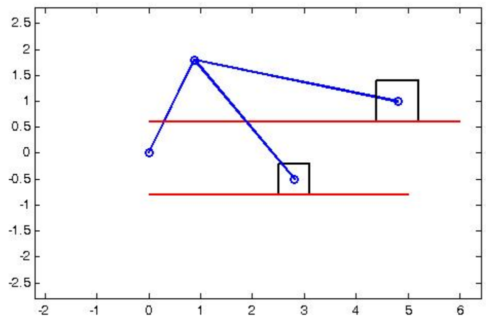

Figure 1: Animation of a Double Slider-crank

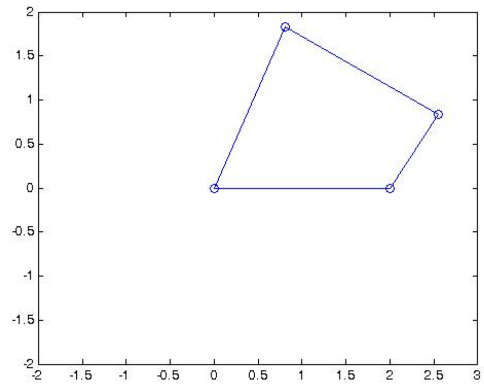

Figure 2: Animation of a Four Bar Linkage

variable, while another (even in the same school) might only require students to write (but not solve) the loop closure equation. Such differences in expectations translate into very different levels of mastery. Exams implemented in FLATLAB could be set to a quantifiable and transparent level of difficulty to ensure the quality of academic standards. This sharing of a common platform not only promotes greater consistency in academic standards, but provides a focus for discussions about curriculum design and goals of education.

Another potential benefit of using FLATLAB for assessment is fair and consistent grading. Manual grading of exams is a subjective and inherently inconsistent process that requires a great deal of judgment (as any professor who has entrusted grading to a TA can attest). This inconsistency can be overcome by well-designed computer-graded exams that feature modular problems of variable difficulty that may be solved within five minutes. Such exams can be generated and graded by FLATLAB, with part marks awarded for approximate answers. Besides analytical problems, this environment can support conceptual questions and non-computational problems that permit experimental solutions.

6 Current Computer-Based Assignments

The starting point for the development of FLATLAB was the laboratory of a third-year mechanical engineering course on the kinematics and dynamics of mechanisms taught by the author. In this computer lab, students write MATLAB programs to animate a variety of mechanisms, such as four bar linkages, slider cranks, and cams, as well as a program to compute the output speed or speed ratio of any single-stage planetary gear train. Figures 1 through 3 each show an animation frame from each of the first three lab assignments from last year.

For the lab of Figure 1, students were given a MATLAB function to animate a single slider-crank, with link dimensions, starting angle, and speed as inputs. Students modified the function to animate a double slider-crank. This introductory lab familiarized students with plotting functions and allowed them to focus on the kinematic equations.

For the lab of Figure 2, students wrote a function, taking dimensions as arguments, that animates

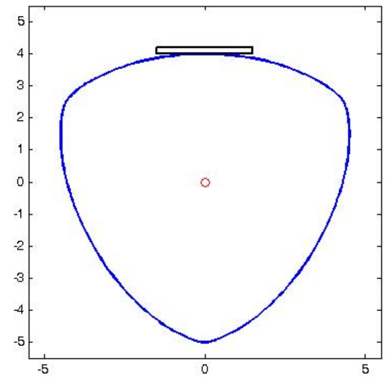

Figure 3: MATLAB Animation of a Cam

both assemblies (one shown) of a four bar linkage, assuming that the mechanism is a crank-rocker or double-crank and that a particular adjacent link is the shortest link. For bonus marks, students could generalize the function so that any link (including the coupler) may be the shortest link, thus admitting a double-rocker as well. Alternatively, the bonus marks could be earned by animating an elliptical trainer using dimensions from an open outdoor gym on campus consisting of (non-motorized) training equipment constructed from planar linkages.

For the lab of Figure 3, students wrote a function to animate a cam and follower, which follows harmonic motion for any input number of follower cycles per cam revolution (3 in the example shown). The follower width must be the minimum that maintains normal contact, and the function outputs the minimum base radius that avoid cusps in the cam profile.

These labs were found to be effective for connecting theory to practice, since students could see immediately the effect of getting the kinematic equations right or wrong. They could also see how the behavior of their mechanisms varied with dimensions. For example, reducing the cam base radius below the computed minimum value produced self-intersecting and concave segments that are mathematically valid, but mechanically impossible. Each function was coded in only a few lines by exploiting the complex number and matrix handling features of MATLAB.

Other progress supporting the development of the mechanism module includes our current use of computer-graded quizzes and exams in the corresponding engineering course. These modular five-minute problems map well into the FLATLAB environment and have demonstrated to the author their effectiveness in promoting goal-oriented learning and mastery of fundamental concepts.

7 Interactive Simulation Development

Software for the mechanism module of FLATLAB will be developed in five stages: 1) algorithms that allow the user to construct and manipulate arbitrary planar mechanisms with finger gestures; 2) algorithms for computing joint variables in real-time and superimposing them on the mechanisms; 3) algorithms for velocity, acceleration, and force analysis; 4) algorithms for creating and deploying assignments and exams on the platform; and 5) algorithms for running interactive problem-solving tutorials.

Here, we describe two technical challenges in the first stage. The first is that the system must allow the construction and manipulation of a planar mechanism with any number of links, loops, and mobility. This requires using numerical solution techniques since a general planar mechanism cannot be solved analytically. For the animations described in Section 6, analytical solutions specific to the given mechanisms were employed.

A numerical solution suggests the integration of velocity kinematics to obtain position kinematics. These velocity equations involve constraints that vary nonlinearly with mechanism position, and singularities occur when links reach their limit positions, resulting in a drop in the mechanism's mobility and the rank of its Jacobian. Also, integration of the differential equations can accumulate errors that violate the constraints. We are currently working on this problem.

A second challenge, related to the first, is to develop algorithms that allow a user to interact with a mechanism using her fingers. To model this interaction correctly, the effect of dragging a link should depend on the mobility of the link. If the mobility is zero (e.g. the mechanism is a truss), then the finger must slip on the link and produce no motion. If the mobility is 1, then the finger must slip in the constrained direction and produce link motion in the unconstrained direction. These directions vary nonlinearly with position, particularly if the driven link is a coupler. If the mobility is 2, then the fingertip position can dictate the position of the contact point while the link rotates to satisfy the mechanism constraints. If the mobility is 3, then the algorithm would maintain a constant link orientation while the user translates the link, or the user could control position and orientation simultaneously using two fingers.

For general planar mechanisms having mobility exceeding 1 (such as a front-end loader) the user should be able to drive two links simultaneously. If the mechanism reaches a limit position, the algorithm must allow additional finger slippage to reflect the loss in mobility. Regardless of the situation, the simulated interaction must reflect the behavior that would result if the user were to manipulate a real mechanism using the same finger motions.

A key paradigm that we will use to develop the algorithms that drive the interactive simulation is an approach we call *intelligent simulation*, which builds on the work of[14]. Instead of using numerical integration methods, intelligent simulation uses analytical solutions whenever possible, as a human would do. Such closed-form solutions are more efficient for real-time animation and are more robust to pathological cases (e.g. singularities where mobility is reduced)[14]. The intelligent simulation approach also supports e-learning and design functions of FLATLAB, namely to provide learners and designers with analytic solutions.

The visual display will differ from that of a conventional CAD environment in being more conceptual, emphasizing function over form. A single loop mechanism can be represented as a polygon, constructed quickly with fingertip, stylus, or mouse clicks to specify its vertices. The angle at each vertex can be specified as being either fixed or variable, representing a revolute joint. Vertices can be toggled between these states with tap gestures. Similarly, the length of line segments can be specified as being either fixed, representing rigid links, or variable, representing prismatic joints. A mechanism can be easily inverted by connecting (or disconnecting) any link to (or from) the ground.

With fingertip, stylus, or mouse, the user can operate a mechanism by dragging any point on any specified link (considered infinite in extent) to any point on its kinematic locus (i.e. consistent with the mechanism kinematics). When the mobility of the dragged point is less than 2, fingertip motions will be decomposed into unconstrained components that produce motion along the locus and constrained components that produce no motion.

Another paradigm we will employ is object-oriented programming. Point and line objects will be used to specify joints and links that interact with each other to form mechanism objects, which interact with the user. The simulation software will be prototyped using the object-oriented features of MATLAB and facilitated by its built-in functions for data handling and plotting. Real-time performance will be enhanced by translating the simulation software into Java and into Android for mobile implementations.

8 Haptic Interface Development

The development of the haptic interface requires the kinematic and electro-mechanical design of a robotic manipulator to drive the haptic cursor, as well as the design and testing of haptic control algorithms. In contrast to the large multi-DOF robotic manipulators used in[15] and[16], we require a small and economical 2-DOF haptic device that integrates easily with a touch screen display.

Competing designs for the haptic device are being evaluated theoretically and experimentally. In one design, the haptic cursor is a magnet that slides on the screen and is controlled by a grid of wires hidden under the screen. Haptic feedback is provided by Lorentz forces generated by the interaction between the magnetic field and the control current. This is similar to the planar maglev positioning system described in[29], except that full levitation is not crucial in our application, so our design requires only 2 control inputs (x and y currents) instead of the 8 control inputs used in[29] to regulate all six motion axes of the floating actuator. A technical challenge that we are currently investigating is to achieve the high control stiffness needed to simulate rigid constraints, while simultaneously providing haptic transparency (zero force) in unconstrained directions.

A second approach is to use a planar manipulator to drive the haptic cursor over the screen. A rigid-link design was chosen over a wire-driven design (such as in[28]) to reduce the number of actuators to 2. The proposed design, called Hapbot, is a novel 2-PP parallel manipulator, i.e. 2 parallel linkages between the ground and the cursor link, each having 2 prismatic joints (PP). This simple design, shown schematically in Figure 4, does not belong to the classification of parallel robots in[26]. Its rectangular workspace matches the area of the touch screen. It is designed to minimize visual occlusion of the touchscreen and to avoid contact between the manipulator links and the user's hand, except at the haptic cursor.

In Figure 4, linear (prismatic) actuators A_x and A_y are connected to ground G via force sensors F_x and F_y and drive links L_x and L_y, respectively. The haptic cursor C forms a rigid link with C_x and C_y, which slide in passive prismatic joints P_x and P_y, respectively. The cursor link (C_x, C, C_y) is made from clear flat plastic to minimize visual occlusion of the touch screen and to provide high bending stiffness in the x-y plane. Preferrably, L_x is replaced with a lead screw rotated about the

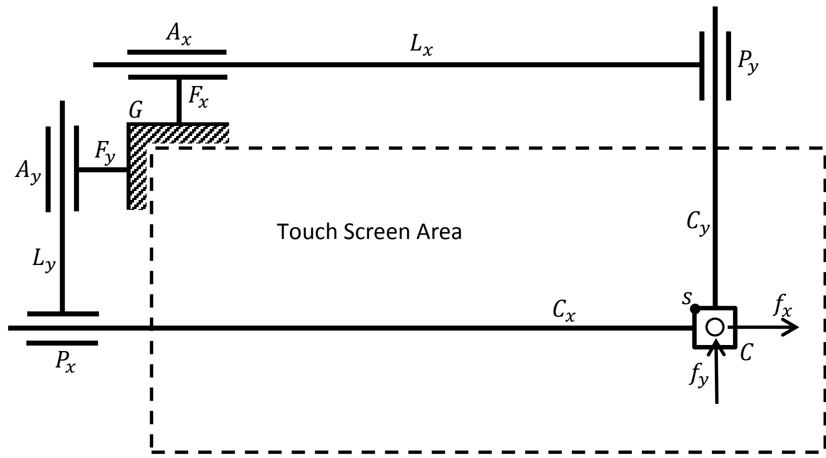

Figure 4: Conceptual Design of Hapbot Driving the Haptic Cursor (C)

x-axis by a rotary actuator A_x and extends beyong P_y to a support bearing. Joint P_y is then carried on a lead nut which translates along L_x as the latter turns. A similar modification is made to A_y, L_y, and P_x. The support of lead screws L_x and L_y at both ends makes them much stiffer, and making them thicker does not affect translational inertia, which is reduced to that of the lead nuts and the cursor link (C_x, C, C_y). A stylus nib s on the haptic cursor C makes continual contact with the touch screen to give the cursor position in world coordinates. The cursor also has a receptacle for a user's fingertip, stylus, or rotating knob to apply planar forces.

This manipulator design allows the actuators and sensors to be identical and stationary, thus eliminating moving wiring and simplifying design and assembly. The actuators will be small DC servo-motors with rotary encoders for position sensing. This position sensing is redundant but gives higher resolution than the touch screen. The force sensors F_x and F_y will use strain gauges and piezoresistors to capture static and dynamic forces over a large bandwidth. These measure the forces f_x and f_y that the user applies to the haptic cursor, minus negligible inertial forces and friction between the cursor and touch surface. Critically, this difference between cursor force and measured force is unaffected by the (considerable) friction between the lead screw and nut, which are internal to Hapbot (considered as a free body).

The haptic cursor must be able to render constrained motion at the user's fingertip. For example, a user turning a virtual crank should feel a rigid constraint in the radial direction but no resistance in the tangential direction (i.e. haptic transparency). This can be achieved to some degree by a modified admittance control using force and position measurements to control actuator current[24]. Experiments will investigate the ability of the haptic system to render simultaneously a hard constraint in one direction and haptic transparency in the perpendicular direction. To obtain repeatable quantification of stiffness, the user's fingertip will be emulated by a second Hapbot manipulator providing input motions and forces via hybrid position/force control[30]. These experiments will also investigate the effect on performance of the force sensor position (actuator-mounted versus cursor-mounted).

A third Hapbot design combines aspects of the first two designs. In this design, the manipulator of Figure 4 moves underneath the screen and is magnetically coupled to a haptic cursor that slides above the screen. The electro-magnetic coupling forces add additional degrees of freedom for controlling the virtual stiffness of the haptic cursor. The resulting macro-micro manipulator design can provide greater haptic transparency than the macro manipulator alone (i.e. the second design) because the user only interacts with the small inertia of the cursor itself (a magnet). Compared to the first design, the electro-magnetic forces need only act over a small range of motion since the macro-manipulator provides the large scale positioning of the cursor. This should permit higher forces and control stiffness. Engineering analysis and experiments will determine which design provides the greatest range of cursor stiffness.

The preferred haptic device will be integrated with the interactive simulation environment. A touch-screen tablet will dock with the Hapbot frame *G* in Figure 4. Users will drive virtual mechanisms through the haptic cursor and feel the forces of constraint, friction, and inertia computed by the interactive simulation algorithms. Experiments on human subjects will investigate the realism of the system and the effect of colocating the haptic cursor with the visual simulation versus separating the visual and haptic displays.

9 E-learning Functionality

FLATLAB will automate the main instructional function of engineering professors by providing dynamic interactive lectures, interactive exercises, tutorials, and labs, and automated assessment. Although automating such difficult functions performed by intelligent experts may appear to be an intractable problem in artificial intelligence, this challenge will be mitigated by: 1) restricting modules to focused domains of knowledge, 2) the evolution of autonomous functions from semi-autonomous ones that include a “professor in the loop”, and 3) the fact that the system will be driven by learners engaged in problem-based active learning. We hypothesize that learners will be motivated to master the interactive exercises, as these are constructively aligned with the interactive exams. In the long term, this hypothesis and the efficacy of the system as a learning tool will be investigated via longitudinal studies on learner outcomes.

The development of the e-learning algorithms and software will build on the object-oriented structure of the interactive simulation algorithms. A simple user interface will be developed for experiential learning activities, including building and manipulating mechanisms and controlling them with feedback. Interrogation functions will then be developed to allow users to query mechanisms for real-time values, such as lengths and angles, via tap gestures. These features will enhance the environment as an interactive visuo-haptic tool for mechanism analysis and control.

Object-oriented methods will also be used to develop authoring tools for interactive examples in lectures and notes and to generate interactive exercise problems and exams. Developed functions will include automatic solution generation based on kinematic methods that we have recently developed. These functions will support the semi-autonomous and autonomous assessment of exercises and exams as well as the interactive tutoring features.

The e-learning functionality of FLATLAB can serve as focal point for debating the goals and future of engineering education. For example, to what extent can or should engineering education be automated and what is the effect on learner outcomes? Such questions will be addressed in parallel with the development of FLATLAB.

10 Conclusion

This work in progress proposed a touch-based platform for focused learning, assessment, and teaching. The purpose of FLATLAB is to provide an unified environment for experiential learning, formal learning, and assessment. This will provide a dynamic environment for experiential learning and design, tools for authoring dynamic interactive notes, textbooks, tutorials, labs, exercises, and exams, automated assessment of learner outcomes. The touch-based interface will create a realistic and interactive environment, while the haptic cursor will raise this interactivity to another level with the addition of force feedback. While some technical challenges must be overcome to develop the hardware for the visuo-haptic interface, the development of the interactive touchscreen interface and the e-learning functionality is achievable in software.

Many of the challenges discussed in this paper are specific to the planar mechanism module of FLATLAB; the development of modules for other courses (statics, dynamics, vibrations, mechatronics, control systems, fluid and solid mechanics, thermodynamics, and circuits) present their own challenges. The development of FLATLAB is thus a long-term endeavour that is providing many research challenges and learning opportunities.

References

[1] J.D. Bransford, A.L. Brown, and R.R. Cocking, editors. *How people learn: brain, mind, experience, and school*. National Academy Press, Washington, D.C., 2004. pp. 23–25.

[2] J. Cecil, P. Ramanthan, and M. Mwavita. Virtual learning environments in engineering and stem education. In *Proc. Frontiers in Education Conference*, pages 502 –507, oct 2013.

[3] A.A. Despande and S.H. Huang. Simulation games in engineering education: a state-of-the-art review. *Computer Applications in Engineering Education*, 19:399–410, 2008.

[4] M. Contero and et al. Using games to teach statics calculation procedures: Application and assessment. *IEEE Computer Graphics and Applications*, 25:24–31, 2005.

[5] T.A. Philpot and et al. Games as teaching tools in engineering mechanics courses. In *Proc. ASEE Annual Conf. and Exp.*, june 2003.

[6] T.A. Philpot and et al. Using games to teach statics calculation procedures: Application and assessment. *Computer Applications in Engineering Education*, 13:222–232, 2005.

[7] J.A. Reed and A.A. Afjeh. Developing interactive educational engineering software for the world wide web with java. *Computers and Education*, 30:183–194, 1998.

[8] J.A. Reed, G.J. Follen, and A.A. Afjeh. Improving the aircraft design process using web-based modeling and simulation. *ACM Trans. Modeling and Computer Simulation*, 10:58–83, 2000.

[9] P. Mandal, K.K. Wong, and P.E. Love. Internet-supported flexible learning environment for teaching systems dynamics to engineering students. *Computer Applications in Engineering Education*, 8:1–10, 2000.

[10] N. Alaine. A matlab/simulink-based interactive module for servo systems learning. *IEEE Transactions on Education*, 53:265–271, 2010.

[11] In Juan Carlos García-Prada and Cristina Castejón, editors, *New Trends in Educational Activity in the Field of Mechanism and Machine Theory*, volume 19 of *Mechanisms and Machine Science*. 2014.

[12] In Hoon Hong and Chee Yap, editors, *Mathematical Software – ICMS 2014*, volume 8592 of *Lecture Notes in Computer Science*. 2014.

[13] Hod Lipson. A relaxation method for simulating the kinematics of compound nonlinear mechanisms. *Journal of Mechanical Design*, 128(4):719–728, 2006.

[14] Glenn A. Kramer. A geometric constraint engine. *Artificial Intelligence*, 58(1–3):327 – 360, 1992.

[15] C.L. Clover, G.R. Luecke, J.J. Troy, and W.A. McNeely. Dynamic simulation of virtual mechanisms with haptic feedback using industrial robotics equipment. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 1, pages 724–730 vol.1, Apr 1997.

[16] A. Nahvi, D.D. Nelson, J.M. Hollerbach, and D.E. Johnson. Haptic manipulation of virtual mechanisms from mechanical cad designs. In *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, volume 1, pages 375–380 vol.1, May 1998.

[17] Francesco Cosco, Carlos Garre, Fabio Bruno, Maurizio Muzzupappa, and Miguel A Otaduy. Visuo-haptic mixed reality with unobstructed tool-hand integration. *Visualization and Computer Graphics, IEEE Transactions on*, 19(1):159–172, 2013.

[18] Vincent Hayward, Oliver R Astley, Manuel Cruz-Hernandez, Danny Grant, and Gabriel Robles-De-La-Torre. Haptic interfaces and devices. *Sensor Review*, 24(1):16–29, 2004.

[19] Thomas H Massie and J Kenneth Salisbury. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME winter annual meeting, symposium on haptic interfaces for virtual environment and teleoperator systems*, volume 55, pages 295–300. IOS Press, 1994.

[20] Robert J Stone. Haptic feedback: A brief history from telepresence to virtual reality. In *Haptic Human-Computer Interaction*, pages 1–16. Springer, 2001.

[21] OAJ Van der Meijden and MP Schijven. The value of haptic feedback in conventional and robot-assisted minimal invasive surgery and virtual reality training: a current review. *Surgical endoscopy*, 23(6):1180–1190, 2009.

[22] Richard Q Van der Linde, Piet Lammertse, Erwin Frederiksen, and B Ruiter. The hapticmaster, a new high-performance haptic interface. In *Proc. Eurohaptics*, pages 1–5, 2002.

[23] N. Hungr, B. Roger, A.J. Hodgson, and C. Plaskos. Dynamic physical constraints: Emulating hard surfaces with high realism. *Haptics, IEEE Transactions on*, 5(1):48–57, Jan 2012.

[24] R.Q. van der Linde and P. Lammertse. Hapticmaster – a generic force controlled robot for human interaction. *Industrial Robot: An International Journal*, 30(6):515–524, 2003.

[25] J.J. Gil, A. Rubio, and J. Savall. Decreasing the apparent inertia of an impedance haptic device by using force feedforward. *Control Systems Technology, IEEE Transactions on*, 17(4):833–838, July 2009.

[26] Feng Gao, Weimin Li, Xianchao Zhao, Zhenlin Jin, and Hui Zhao. New kinematic structures for 2-, 3-, 4-, and 5-dof parallel manipulator designs. *Mechanism and Machine Theory*, 37(11):1395 – 1411, 2002.

[27] Mohammad R Sirouspour, Simon P Dimaio, SE Salcudean, Purang Abolmaesumi, and Colin Jones. Haptic interface control-design issues and experiments with a planar device. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 1, pages 789–794. IEEE, 2000.

[28] Paolo Gallina, Giulio Rosati, and Aldo Rossi. 3-dof wire driven planar haptic interface. *Journal of Intelligent and Robotic Systems*, 32(1):23–36, 2001.

[29] Xiaodong Lu and Irfan ur-rab Usman. 6d direct-drive technology for planar motion stages. {*CIRP*} *Annals - Manufacturing Technology*, 61(1):359 – 362, 2012.

[30] P.B. Goldsmith. Global asymptotic stability of hybrid position/force control applied to compliant unilateral constraints. *Mechanism and Machine Theory*, 34:1009–1021, 1999.