

Teaching Rigid Body Mechanics Using
Student-Created Virtual Environments

Todd D. Murphey

Electrical and Computer Engineering

University of Colorado

Boulder, CO 80309

murphey@colorado.edu

Abstract

This paper describes a recent effort to use student-created virtual environments to teach rigid body dynamics to electrical and computer engineering senior-level undergraduates and first-year graduate students. Many of these students have no background in dynamics except for their freshman year physics course. The approach described here relies on students creating their own virtual environments, takes roughly four to five weeks of instruction, and fits within the scope of other courses that may require electrical and computer engineering students to be familiar with dynamics. The background required consists of only basic linear algebra and ordinary differential equations. By the end of the pilot class, the majority of students were able to expertly model and animate high degree-of-freedom systems, such as slip-steered vehicles, helicopters, submarines, balancing unicycles, motorcycles, and puppets. A core component of the learning philosophy is that by providing students with a venue for creating their own virtual laboratories, they are able to promote their own learning. This paper summarizes the basic material presented to students, some of the virtual environments they chose to implement with only minimal guidance, and three instruments used to assess the effectiveness of the class architecture.

Keywords: Virtual reality, Robot dynamics, Engineering education, Educational technology

1 Introduction

The course “ECEN 4028/5028: Robot Dynamics and Motion Planning” was introduced into the curriculum in Spring term, 2006, at the University of Colorado in order to give senior-level undergraduate and first-year graduate students with a predominantly electrical and computer engineering background a solid background in dynamics. Most of these students have only had dynamics previously in their freshman physics sequence, yet many of them had an interest in animation, mechatronics, control, and other applications of multi-body dynamics. The required background consists of basic linear algebra and ordinary differential equations. These requirements are satisfied at any university that is accredited by the Accreditation Board for Engineering and Technology (ABET). The goal was to give these students a sufficient background (in a relatively short amount of time) to compute the dynamics of complex, realistic multi-body systems correctly and to then use these techniques in some area of application. Every student was required to develop, over the course of the term, a virtual environment for animation, dynamics, and motion planning. Developing a virtual environment helps students connect the mathematical abstraction to the real physical system they are trying to describe. Additionally, using student-created virtual environments increases student confidence in mastering difficult modeling techniques, and make those techniques more concrete.

The use of virtual laboratories in the classroom has become much more prevalent recently, including teaching electrical engineering [1, 2, 3], physics [4, 5], circuits [6], photonics [7], and instrumentation [8]. Moreover, virtual laboratories have seen more recent use in research as well [9]. The advantages of virtual laboratories are great; in particular, they allow for laboratories that can be scaled to many students without requiring many more resources and can be used for remote-learning students [7, 10]. However, these courses are typically aimed at pre-designed laboratory experiences in a manner similar to physical, in-class laboratories. Such approaches largely ignore one of the intrinsic benefits of virtual laboratories—that students can create their own virtual laboratory in a manner that reflects their talents and

interests.

The goals of traditional uses of virtual laboratories are substantively different from the goals pursued here. In particular, traditional virtual laboratories aim at improving student intuition about systems with which they do not already have experience. For example, virtual laboratories have been used with great effect in introducing students to electrical circuits [5], where the goal is not to introduce students to circuit analysis, but get students to understand the basic functionality of a circuit. The goal in the present work is related but not the same. Instead, the goal is to get students to use their *pre-existing* knowledge of a physical system to validate the mathematical tools used to analyze that physical system. Hence, the approach presented here is relying on students already having knowledge intrinsically of the dynamic world and encouraging them to compare that knowledge to the predictions of their numerical simulation. Such a comparison would be meaningless without having them generate the equations of motion themselves, rendering pre-existing virtual laboratories inapplicable.

The basic strategy employed in the class architecture presented here is to create simple virtual laboratories (a pendulum example and a rolling disk example) that students program themselves, and then allow them to create a more sophisticated virtual laboratory of their own choosing. By allowing them to choose their own virtual laboratory, students are guaranteed to be able to work with a virtual system that they already know something about, or at the very least find compelling. Developing a course that allows students to develop virtual environments of their choosing is the focus of the present paper.

One of the difficulties with a senior-level course is that student assessment in a quasi-experimental setting is difficult because of a lack of control group. The class described here had only twenty-nine students and, therefore, called for using multiple modalities of assessment. Evaluation of the teaching techniques presented here was addressed using a combination of three modalities (relevant in a quasi-experimental scenario [11]). First, an in-class survey was used to determine student perception of their learning relative to other courses they have taken. Additionally, student learning was assessed to evaluate the effec-

tiveness of the teaching technique. Thus, students were asked a question that specifically targeted their ability to generalize the concepts used in class to a more complicated situation. Third, students' final projects were assessed along a novice/expert continuum. Notably, the majority of final projects reflect a level of expert knowledge that is well beyond what was directly addressed in the classroom. The author gave relatively little input into projects, and students productively engaged in projects involving helicopters (with remarkably well-constructed system dynamics), submarines, unicycles with moment wheels for stabilization, swinging crane problems, and marionettes. The last two modalities of assessment reflect a recent focus on generalizing from concrete problems to more abstract problems, as detailed by the National Research Council [12, 13].

Last, along a more anecdotal line, nearly half the class asked about graduate level coursework they could take as electives, such as probability theory, differential geometry, abstract algebra, and advanced linear systems. The realization that this course would motivate students who do not need to take these courses to take them as electives is also a good, if informal, sign of the motivational effectiveness of the class architecture.

This paper is organized as follows. Section 2 discusses the course content, an example that illustrates the basic mathematical tools employed, and the students' creation of their own virtual environments. Section 3 discusses the various modalities of assessment of the course. Conclusions and future work (including plans to extend this course structure to teaching linear algebra and differential equations) are in Section 4. An appendix including a general description of rigid body dynamics using homogeneous transformations and sample code is also provided.

2 Course Organization and Content

Electrical and computer engineering students typically have little to no background in dynamics aside from the standard Newtonian mechanics (i.e., $F = ma$) they learn in an introductory physics course. To address this deficit, the class is organized so that mechanics

are presented in a manner that is directly parallel to the creation of a virtual environment to make the material more concrete. In particular, the class is organized so that students follow the procedure below.

1. Students learn how to define the configuration for a system.
2. Students learn how to create a virtual environment for a system using rigid body transformations.
3. Students use these same transformations to compute the kinetic and potential energies.
4. Students compute dynamic equations of motion using the Euler-Lagrange equations.
5. Students then use these models in motion-planning strategies.

The above procedure is first implemented using the rolling disk example (a disk rolling in a plane with applied torque for driving and steering, referred to simply as the Rolling Disk both here and in the literature [14, 15]), for which all students must implement a virtual environment. Students then choose a system of interest to them and create a virtual environment for it. A difficulty with allowing students to choose the systems for which they will create virtual environments is that the resulting equations of motion are difficult to control because of underactuation, nonlinearities, and the multi-input, multi-output nature of most systems. Therefore, rather than focus on control directly, the rest of the course focuses on motion planning. Focusing on motion planning allows one to create global results using only locally valid feedback controllers or even open-loop controllers.

The course consisted of 25 bi-weekly lectures at an hour and fifteen minutes each. The number of lectures is divided as follows.

1. (3 Lectures) Configuration Spaces and Topology
2. (4 Lectures) Vector Fields and Nonlinear Controllability
3. (8 Lectures) Dynamics

4. (8 Lectures) Motion Planning

Only four weeks of the course are dedicated specifically to teaching dynamics, though the discussion of both configuration spaces and vector fields lay the appropriate foundation for both dynamics and motion planning. Hence, roughly half of the course is dedicated to dynamics in one way or another. Hence, the method of presenting dynamics discussed in this paper can easily be transferred to another course where something other than motion planning is the co-emphasis with dynamics.

One of the main challenges in teaching a course with so much formal mathematics is to make sure that the course does not become a mathematics course. The author took the approach of introducing the minimum amount of material required to understand, at a computational level, the body velocity and its use in computing system dynamics and describing constraints. Hence, this class does not address the fundamental principles of mechanics, but focuses on getting dynamics right for relatively complex systems. (The course is, therefore, not a replacement for more advanced dynamics courses, but is a precursor for such courses.) In this course, the author merely comments that an advantage of Lagrangian mechanics is its ability to use generalized coordinates of the student's choosing, and that a Lagrangian approach is, therefore, not as limiting as the Euclidean nature of Newtonian mechanics.

2.1 An Example

The key connection between the dynamic analysis of rigid bodies and their animation is that mechanical modeling can be computationally considered in the same framework as animation, thus uniting the two for students. Also, the kinetic energy, potential energy, constraints, and forces can be thought of as data types that must be present to solve a given dynamics problem. (A more general description of dynamics in this computational framework is presented in Appendix A.1.) Dynamics are, therefore, presented in terms of

how these equations represent a strongly typed syntax, much like other computer languages such as C and C++. This presentation helps students see mechanics in a more systematic and general manner (compared to, for example, force balance techniques arising from $F = ma$ which typically require the space to be Euclidean and can lead to students confusing Coriolis terms with forces). Moreover, students learn to use the homogeneous representation g_{AB} (and the associated body velocity V_b) as the fundamental building block for mathematically describing, simulating, and animating any mechanical system. In the following, an example of a pendulum is used to illustrate the connection between the dynamic representation of a rigid body and its graphical representation.

The pendulum, pictured in Fig.1, is a nonlinear system that is commonly an example in first-year physics. A pendulum is discussed here to illustrate the relationship between the animation use of homogeneous coordinates (such as that used in OpenGL and other graphics languages) and the calculation of dynamic equations of motion. A more general description is provided in Appendix A.1. The pendulum was used as an ongoing example in the classroom along with the “rolling disk” example previously discussed and used in many texts [15, 14].

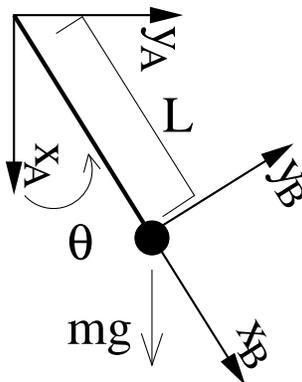


Figure 1: An example of a pendulum of length L with a point mass m at the end. Frame A is the inertially fixed frame attached to the base of the pendulum with its x axis pointed down, and Frame B is the frame with its origin at the point mass with its x axis aligned with the pendulum.

The homogeneous representation of the transformation from frame A (the inertially fixed frame with its origin at the base of the pendulum and its x axis pointing down) to frame B

(the frame with origin located at the point mass with its x axis aligned with the pendulum)

is $g_{AB}(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & L \cos(\theta) \\ \sin(\theta) & \cos(\theta) & L \sin(\theta) \\ 0 & 0 & 1 \end{bmatrix}$. The following notation is used throughout the

remainder of the text. The “dot” notation for derivative is $\dot{x} = \frac{d}{dt}(x)$, and the “hat” notation

[14] is such that if one has a vector $(u, v, \omega)^T$, then $\widehat{\begin{bmatrix} u \\ v \\ \omega \end{bmatrix}} = \begin{bmatrix} 0 & -\omega & u \\ \omega & 0 & v \\ 0 & 0 & 0 \end{bmatrix}$ (the “hat”

operation can be, of course, inverted to turn a matrix of this structure back into a vector).

Using the “hat” notation, the quantity V_b can be defined using the relation $\widehat{V}_b = g_{AB}^{-1} \dot{g}_{AB}$.

The kinetic energy of the point mass to be written as $KE = \frac{1}{2} V_b^T \mathcal{I} V_b$ where \mathcal{I} is the diagonal

matrix $\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J \end{bmatrix}$, and where m is the mass of the point mass, and J is the rotational

inertia of the point mass. (In the case of the pendulum $J = 0$ because a point mass has no rotational inertia.) The calculation of kinetic energy yields

$$\frac{1}{2} \begin{bmatrix} 0 & L\dot{\theta} & \dot{\theta} \end{bmatrix} \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ L\dot{\theta} \\ \dot{\theta} \end{bmatrix} = \frac{1}{2} m L^2 \dot{\theta}^2.$$

All first-year physics students learn that $\frac{1}{2} m L^2 \dot{\theta}^2$ is the kinetic energy of a point mass rotating about an axis of rotation. Similarly, the potential energy $V = -mgh$ (where $g = 9.8m/s^2$ is the gravitational acceleration, and h is the height of the point mass) can be computed.

The variable h is simply the x component of the origin of frame B in frame A . (Thus, the first component of the vector $g_{AB}(\theta) \cdot (0, 0, 1)^T = (L \cos(\theta), L \sin(\theta), 1)^T$ which is $L \cos(\theta)$.)

Now, with kinetic energy and potential energy in hand, one can write down the equations of motion using the Lagrangian $L = KE - V = \frac{1}{2} m L^2 \dot{\theta}^2 - (-mgL \cos(\theta))$ and the Euler-

Lagrange Equations $\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = 0$. Simplifying, one gets

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = \frac{d}{dt}(mL^2\dot{\theta}) - (-mgL \sin(\theta)) = mL^2\ddot{\theta} + mgL \sin(\theta) = 0 \quad (1)$$

The right hand side of this equation is exactly the equations obtained from Newtonian force balance, so that students will recognize the equations and can easily check them for correctness. *Mathematica* code that does these computations automatically is included in Appendix A.2.

A solution $\theta(t)$ to the differential equation in Eq. (1) (given an initial condition $\theta(0) = \theta_0$ and $\dot{\theta}(0) = \dot{\theta}_0$) can be plugged into the expression $[x(t), y(t), 1]^T = g_{AB}(\theta(t))(0, 0, 1)^T$ to give the trajectory in space of the point mass. Similarly, an animation of a polygonal object with vertices p_i would simply consist of transforming all the vertices p_i by $g_{AB}(\theta(t))$. Hence, the method for animating the pendulum uses the same tools as the method for describing the mechanics of the system mathematically. Sample code is provided in Appendix A.2.

2.2 Creation of Virtual Environments

The course is organized so that students first learn how to choose configurations for mechanical systems and then create virtual environments for them. The students are told how to make primitive graphics using vertices of polygons. Thus, for each part of a robot, they create a graphic primitive which represents that part as a collection of vertices. With homogeneous representations g of rigid body motions, they simply multiply every vertex by g to compute the set of vertices as they move along a trajectory in configuration space. Therefore, if one has a set of vertices describing a part of a mechanical system $\{p_i\}$, then $\{g(q(t)) \cdot p_i\}$ represents how the vertices move as the configuration changes with time. This representation of a rigid body is closely aligned with how three-dimensional graphics are created (e.g., using OpenGL), and how the students create their virtual environments. Example *Mathematica* code is provided in Appendix A.2 that creates a rigid body representation of

the pendulum using simple graphics primitives available within *Mathematica*. This *Mathematica* code is reasonably simple; students do not have any difficulty achieving an adequate level of proficiency during a term.

Students are able to quickly build, from scratch, mathematical models of complex mechanical systems and represent them in a virtual environment, thus giving them a result they will have intuition about. For high degree-of-freedom systems, creation of virtual environments is essential to convincing students that they have indeed mastered the requisite modeling techniques.

2.3 Other Implementations

Although implementing dynamic equations of motion using homogeneous transformations is straightforward in *Mathematica*, Maple, or any software capable of symbolic calculation, the symbolic calculation is only used for symbolic evaluation of the partial derivatives of the Lagrangian. Therefore, if an instructor is willing for students to find these derivatives in another manner (e.g., by hand), the basic methods presented in this paper can be used within any software package that can do matrix calculations and has a graphics library. Such software packages include Python, LabVIEW, Matlab/Simulink, and traditional languages such as C and C++.

One could even attempt to have students estimate these partial derivatives numerically. However, this technique is tantamount to creating a variational integrator [16] and would be inappropriate for lower-division students. Of course, introducing students to variational integrators may well be appropriate for more advanced students who study fluid mechanics, optics, or electro-magnetism, where variational integrators are reasonably common.

2.4 Final Projects: Students Create Their Own Virtual Environments

In this course, students complete a final project in which they create a virtual environment

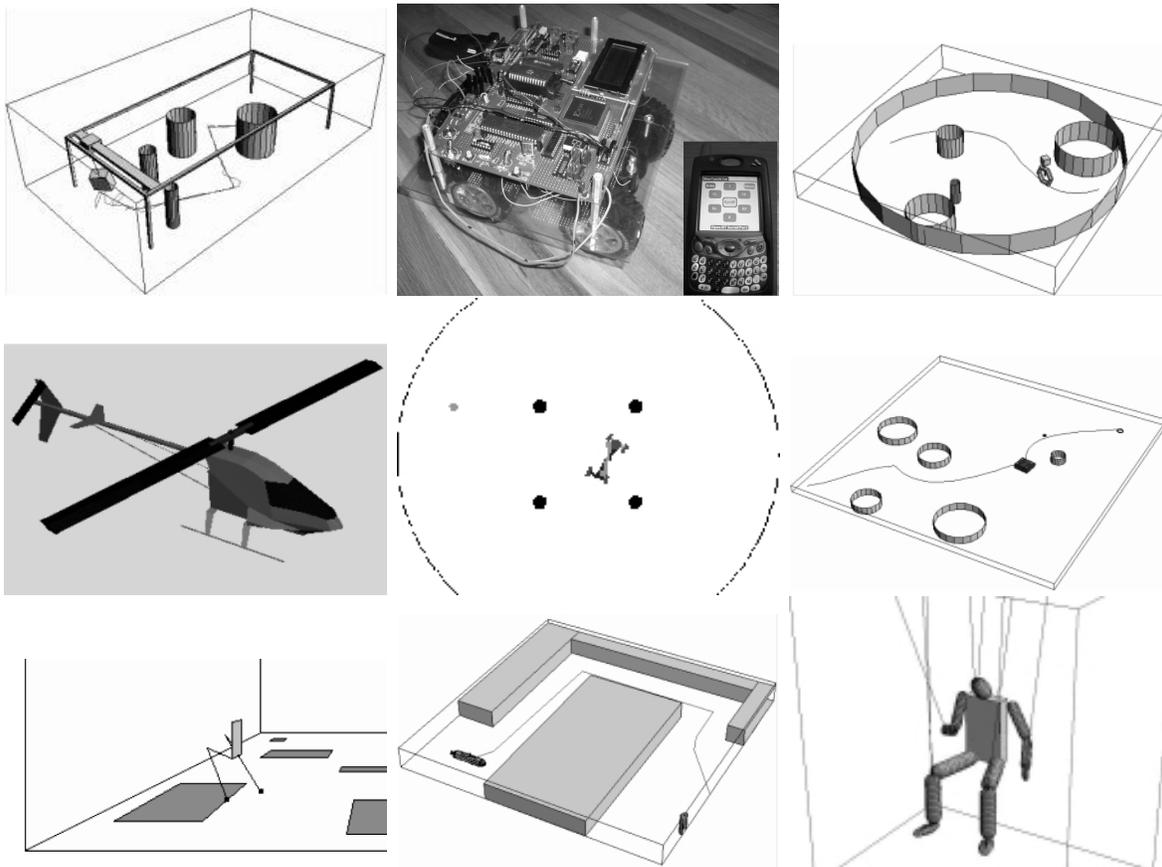


Figure 2: Final projects initiated by students included (from left to right, top to bottom) a) solving a swinging crane problem (a standard canonical problem in controls), b) creating a hardware vehicle and a simulation of the vehicle for motion planning, c) modeling an unstable unicycle with moment wheel for stabilization, d) navigating a dynamic model of a helicopter through obstacles, e) modeling of and motion planning for the snakeboard, f) modeling a slip-steered vehicle with all of its constraints properly taken into account, g) providing a kinematic model of a biped walking through a field of obstacles, h) modeling a submarine, and i) modeling a marionette. These and other projects can be found at <http://robotics.colorado.edu/~murphey/ecen4028/finalprojects/>.

for a mechanical system of their choosing. Students may decide to use the rolling disk in their final project, although only a small number of students actually chose to do so in the pilot course. The general guidelines for this project were that

1. the project must be animated;
2. the project must include kinematic constraints;
3. the project must include dynamics; and
4. the project must include motion planning.

Students generalized the rolling disk to quite difficult examples. Of the 29 students, all but five chose to do a project that was non-trivial in some significant way. Students chose a number of different projects, including a swinging crane with a package moving through a factory floor; a dynamic inverted pendulum on a unicycle moving through obstacles; a submarine that has to negotiate narrow corridors; a slip-steered vehicle with correct, overconstrained frictional dynamics and feedback control; a dynamic model of a marionette. Some of these projects are illustrated in Fig. 2.

The motion planning component of this course focused on using standard techniques in motion planning (potential-based methods, Rapidly Exploring Random Trees, Probabilistic Roadmaps [17]) to compute dynamically feasible trajectories that take a system from an initial configuration to a final configuration without hitting obstacles. For example, several systems in Fig. 2 show paths in the configuration space that were computed using various versions of these motion-planning techniques. All students in the course successfully created motion planners for their systems.

3 Evaluation of Teaching Technique

The effectiveness of the presented teaching technique was evaluated both by direct questioning of the students and by assessment of student learning in the class. Evaluation was

addressed using three separate modalities. The traditional course survey was used to find out how the students perceived their learning relative to other courses they have taken to get a sense of the perceived teaching effectiveness. Additionally, the teaching technique may be evaluated by assessment of student learning. In the course survey, students were asked to describe modeling a spacecraft in orbit, a task requiring substantial generalization from what they had learned in class. The third modality of evaluation was to assess their final projects based on the expert/novice scale (see [12, 13]) and the degree to which they substantially generalized what they had learned in class. Experts and knowledgeable students have extensive, well-structured, declarative knowledge; novices do not have such knowledge. Furthermore, experts and knowledgeable students possess nuanced understanding of how to approach complex tasks, both familiar and novel; again, novices have no such knowledge (see [12, 13]). These latter two modalities of evaluation use assessment of student learning as a measure of the effectiveness of the teaching technique. Moreover, these last two modalities reflect the recent trend in encouraging learning in terms of going from concrete problems to more abstract problems and going from specific problems to more general ones, as detailed by the National Research Council [12, 13, 11].

3.1 Student Response

Students were asked to fill out a traditional survey, where they ranked items related to how much they perceived they had learned, as well as other aspects of the class. These questions and student responses, are in Table 1. Basically students almost uniformly believed they had learned more in this class than they had in an “average” course, and similarly almost uniformly believed that the course use of a concrete, unifying example (the rolling disk) and the homeworks were helpful. In particular, the majority of students believed that they can use the material in a practical setting.

	Question	much less	a little less	about the same	more	much more
1	I have learned _____ in this class than the average class I have taken	0	2	9	10	5
		Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
2	I like that we used just a few concrete examples throughout the course	0	2	2	9	13
3	I would recommend this course to other students	0	0	3	4	19
4	I feel I can use the material we are learning in class in practical situations	0	1	3	13	10
5	Homeworks were helpful in understanding the material	1	0	5	12	8

Table 1: Table of student responses. Students were asked how much they perceived they had learned and asked how much they agreed with some statements about the effectiveness of the course.

3.2 Generalization to Satellites

Students were also asked the following question.

In a few sentences, how would you use the techniques developed in this class to model a satellite in orbit and make it point at a particular target?

This problem is a standard, difficult problem that generally requires significant expertise to solve. A correct answer would be that the students would model the rigid body components of the satellite using the Euler-Lagrange equations, would generate a motion plan to get close to the desired pointing angle, and then would use a linear feedback controller to stabilize to the desired pointing angle. Of the 26 respondents to the survey, only four did not correctly identify the basic approach to solving this problem (two of these four were among the five that did trivial final projects). Ten of the students were able to give significant detail, including specifics of modeling and what motion planner they would use. The fact that 22 out of 26 students were able to generalize so effectively what they had learned in class is one of the best indications of the success of this course architecture. Section 3.3 discusses the final component of assessment, which is also along the lines of addressing generalization.

The other two of the four students who were not able to address this problem (the two of the four that did *not* use a trivial project) were aerospace students who had already taken classes on satellite dynamics. They both commented that they had seen a similar problem before and immediately attempted to write down equations they had seen in those prior courses. Their response is, of course, exactly the opposite of what is desired—in this instance, their prior coursework was seemingly acting as an obstacle to generalization.

3.3 Novice/Expert Assessment of Final Projects

As mentioned in Section 2.4, students were asked to do a final project instead of taking a written final. The final project was designed into the curriculum partially for purposes of

evaluation of the teaching technique. In particular, the author was interested in assessing the degree to which the students could *generalize* what they had learned and the extent to which they had obtained a more *expert* level of understanding of mechanics and motion planning. Both of these types of assessment have become more common recently, as documented in [12, 13] and in [18, 19, 20, 21].

Student assessment using individualized final projects is more difficult than the assessment in Section 3.2 because the students choose the sample system instead of the assessor choosing the sample system. Indeed, in the previous mode of assessment, a satellite clearly represents a significant generalization from the rolling disk—Many universities offer entire courses solely dedicated to satellite dynamics. In assessing the degree to which students generalized the rolling disk in their final projects, six discrete modes of generalization were singled out. These modes are as follows.

1. Addition of internal dynamics or potential energy
2. Inclusion of more than two inputs
3. Analysis of systems with higher degrees of freedom
4. Inclusion of feedback control
5. Analysis of multiple, separated, mechanical systems
6. Addition of mechanics with state transitions

These six modes of generalization are certainly not unique, but represent reasonable directions of generalization that may, among other things, be required of students when they work in industry and generally require an expert’s ability to model. Each project was then (subjectively) evaluated as to whether the project had significantly generalized what had been discussed in class. Table 2 summarizes these findings.

On an average, students generalized along 2.45 modes of generalization. Also, 24 out of 29 students generalized along at least one mode of generalization. (The five students who did

Number of Modes Generalized	Number of Students
0	5
1	5
2	3
3	7
4	6
5	3
6	0

Table 2: Summary of student project generalization by number of modes generalized

not generalize were the same five who used the sample system developed in class—the rolling disk.) Some of the projects clearly generalized class material (such as the hardware robot), though not along the lines of these specific modes. Thus, assessment of generalization is somewhat conservative. Last, as part of their projects, various students learned (on their own or in discussion with the author) about differential flatness [15], overconstrained systems [22], fluid mechanics with potential flow, implicit integration of differential equations, and many other nominally off-topic areas.

3.4 Sample Student Comments

Students were encouraged to write comments regarding the class. A sample of these comments follows.

1. “I liked the emphasis on the end results (animations) rather than the approach. It allowed for many different correct answers and allows the student to come closer to fully understanding the solution. The programming afforded me a level of understanding that pure[ly] mathematical homework could not have.”
2. “This class stressed me out a lot, but the way [the professor] ran the class alleviated a lot of it. I struggled a lot with the homeworks, but sincerely felt like I understood a lot after completing them.”

3. “I think this was a great class and should be offered again in the fall. Although I don’t expect to do much work in controls, this class has provided me with a lot of insight into how to practically model a system and design reasonable animations for that system. I haven’t had any experience with Euler [angle] matrices or a true understanding of how to apply Euler-Lagrange equations to describe dynamics.”
4. “This project really made me realize how much I know and how much I lack. It is actually quite scary yet very motivating to think about future classes to come. I had a great semester.”
5. “This class is the best class I have taken since starting at the university.”

4 Conclusion

This paper presented a basic framework for introducing dynamics to students with little to no background in mechanics, using student-created virtual environments. The framework requires students to develop their own virtual environment in both homeworks (using the pendulum and rolling disk examples) and in a final project. The students create their own environments, helping them become more invested in the work and helping to make the class feasible in the absence of hardware. All of the virtual environments were made in *Mathematica*, which is widely available and used, and is relatively inexpensive for students (\$45 for six months), but can be implemented in other environments as well.

The eventual goal for using student-created virtual laboratories is to help students learn and retain methods in linear algebra and differential equations. The mechanical systems described here are particularly concrete, and none of the techniques employed are beyond a freshman student with a background in calculus. The hope is that the work presented in this paper will lead to the creation of a freshman-level course that has as its other half an introduction to differential equations and linear algebra that would then motivate the

courses students typically take later.

Acknowledgements

The author gratefully acknowledges the support for this work provided by the NSF under CAREER award CMS-0546430.

References

- [1] L. Sevgi, "Virtual tools/labs in electrical engineering education," *ELEKTRIK*, vol. 14, no. 1, pp. 113–127, 2006.
- [2] L. Sevgi and C. Uluisik, "A LabVIEW-based virtual instrument for engineering education: A numerical fourier transform tool," *ELEKTRIK*, vol. 14, no. 1, pp. 129–152, 2006.
- [3] L. Sevgi, C. Uluisik, and F. Akleman, "A matlab-based two-dimensional parabolic equation radiowave propagation package," *IEEE Antennas and Propagation Magazine*, vol. 47, no. 4, pp. 141– 145, 2005.
- [4] C. E. Wieman and K. K. Perkins, "Transforming physics education," *Physics Today*, p. 36, Nov 2005.
- [5] N. Finkelstein, W. Adams, C. Keller, P. B. Kohl, K. K. Perkins, N. Podolefsky, S. Reid, and R. LeMaster, "When learning about the real world is better done virtually: a study of substituting computer simulations for laboratory equipment," *Physical Review - PER*, 2005.
- [6] B. Butz, M. Duarte, and S. Miller, "An intelligent tutoring system for circuit analysis," *IEEE Transactions on Education*, vol. 49, pp. 216–223, May 2006.

- [7] G. Chang, Z. Yeh, H. Chang, and S. Pan, “Teaching photonics laboratory using remote-control web technologies,” *IEEE Transactions on Education*, vol. 48, pp. 642–651, Nov 2005.
- [8] J. Palop and J. Teruel, “Virtual work bench for electronic instrumentation teaching,” *IEEE Transactions on Education*, vol. 43, pp. 15–18, Feb 2000.
- [9] L. Sevgi, F. Akleman, and L. B. Felsen, “Visualizations of wave dynamics in a wedge-waveguide with non-penetrable boundaries: normal, adiabatic, and intrinsic mode representations,” *IEEE Antennas and Propagation Magazine*, vol. 49, no. 1, 2007.
- [10] T. Kikuchi, S. Fukuda, A. Fukuzaki, K. Nagaoka, K. Tanaka, T. Kenjo, and D. Harris, “DVTS-based remote laboratory across the pacific over the gigabit network,” *IEEE Transactions on Education*, vol. 47, pp. 26–32, Feb 2004.
- [11] W. Shadish, T. Cook, and D. Campbell, *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Boston: Houghton-Mifflin, 2002.
- [12] J. Pellegrino, N. Chudowsky, and R. Glaser, eds., *Knowing What Students Know: The Science and Design of Educational Assessment*. Washington, D.C.: National Academy Press, 2001.
- [13] J. Bransford, A. Brown, and R. Cocking, eds., *How People Learn: Brain, Mind, Experience, and School*. Washington, D.C.: National Academy Press, 2000.
- [14] R. Murray, Z. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.
- [15] F. Bullo and A. Lewis, *Geometric Control of Mechanical Systems*. Number 49 in Texts in Applied Mathematics, Springer-Verlag, 2004.
- [16] A. Lew, J. E. Marsden, M. Ortiz, and M. West, “Variational time integrators,” *Int. J. Numer. Methods Engrg*, vol. 60, pp. 153–212, 2004.

- [17] S. M. LaValle, *Planning Algorithms*. Cambridge Press, 2006.
- [18] R. J. Shavelson, M. A. Ruiz-Primo, and E. W. Wiley, “Windows into the mind,” *Higher Education*, vol. 49, pp. 413–430, 2005.
- [19] E. Seymour, “Student Assessment of Learning Gains (SALG) .”
<http://www.wcer.wisc.edu/salgains/instructor/SALGains.asp>.
- [20] R. Shavelson and M. Ruiz-Primo, *Assessing science understanding (J. Mintzes, J. H. Wandersee, & J. D. Novak Eds.)*, ch. On the psychometrics of assessing science understanding, pp. 304–341. Educational Psychology Series, San Diego: Academic Press, 1999.
- [21] R. Shavelson, G. Baxter, and J. Pine, “Performance assessments: Political rhetoric and measurement reality,” *Educational Researcher*, vol. 21, no. 4, pp. 22–27, 1992.
- [22] T. Murphey, “Motion planning for kinematically overconstrained vehicles using feedback primitives,” in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2006.

A Appendix

A.1 Computational Tools Required For Both Dynamics and Virtual Environment Development

To learn dynamics in the way described here, one needs only a few very simple tools to describe the mechanics of a system [14, 15]. Moreover, these tools can be immediately used for purposes of creating a virtual environment for the system of interest. Thus, an instructor should emphasize that both animation and rigid-body mechanics are quite *simple* and tightly related to each other. For this reason, only the following computational tools are introduced during the course.

1. The configuration q , which uniquely specifies all points in the mechanical system;

2. The homogeneous representation of a rigid body motion $g(q)$, as it depends on q . The homogeneous representation $g(q)$ is of the form

$$g(R, p) = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix},$$

where R is a $n \times n$ rotation matrix and p is a point in \mathbb{R}^n (where $n = 2, 3$);

3. Three particularly useful rotation matrices are the Euler angle matrices R_X, R_Y, R_Z defined as rotation about the coordinate axes, e.g.,

$$R_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix};$$

4. The body velocity V_b , where $\widehat{V}_b = g^{-1}\dot{g}$ and the “hat” operator is defined by

$$\begin{bmatrix} 0 & -\omega_3 & \omega_2 & v_x \\ \omega_3 & 0 & -\omega_1 & v_y \\ -\omega_2 & \omega_1 & 0 & v_z \\ 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \widehat{V}_b;$$

5. The kinetic energy $KE(q, \dot{q})$ (which for any component of a multi-body mechanical system can be represented as $\frac{1}{2}V_b^T \mathcal{I} V_b$ with \mathcal{I} constant and diagonal when the rigid body transformation is mapped to the center of mass and the coordinate axes of the body frame are aligned with the principle axes of inertia);

6. The potential energy $V(q)$, which for most problems is either related to the height of

each center of mass (e.g., $V = -mgh$, where h can be calculated directly from the associated g) or the distance between two points (e.g., a spring potential, also directly obtainable from g);

7. The constraints on the system $\omega(q)$, where $\omega(q)\dot{q} = 0$, which are typically calculated as $v \cdot V_b = 0$, with v a constant co-vector;
8. External forces F_i (which are typically physical inputs to the system).

With this data, one can correctly compute the equations of motion for a multiple degree of freedom system. Given a Lagrangian $L = KE(q, \dot{q}) - V(q)$, constraints $\omega(q)$, external forces F_i , the Euler-Lagrange equations are

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} &= \sum_i F_i + \omega(q)^T \lambda \\ \frac{d}{dt} (\omega(q)\dot{q}) &= 0. \end{aligned}$$

A.2 Sample *Mathematica* Code

This section provides *Mathematica* code that solves for the dynamic equations of motion for the pendulum (discussed in Section 2.1). A second piece of *Mathematica* code creates a simple animation. This code may also be obtained at <http://robotics.colorado.edu/~murphey> alongside the link to this paper. Readers are encouraged to try this code as evidence that the code is easily within reach of most students.

```

Clear[m,grav,L]
gAB[th_]:={{Cos[th],-Sin[th],0},{Sin[th],Cos[th],0},{0,0,1}}.
  {{1,0,L},{0,1,0},{0,0,1}};
UnHat[M_]:=M[[1,3]],M[[2,3]],M[[2,1]]};
VbodyAB[th_]:=UnHat[Inverse[gAB[th]].D[gAB[th],t]]//FullSimplify
L[th_]:=1/2 VbodyAB[th].DiagonalMatrix[{m,m,0}].VbodyAB[th]-
  (-m grav gAB[th].{0,0,1})[[1]];
DiffEQ=D[D[L[th[t]],th'[t]],t]-D[L[th[t]],th[t]];

```

The above code creates the matrix g_{AB} , creates the “unhat” function, calculates the body velocity V_b given θ (and therefore $\dot{\theta}$), calculates the Lagrangian L , and then calculates $\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta}$. Sample code that creates the animation from the solution to the dynamic equations follows.

```

L=1;grav=9.8;m=1;
ODEsol=NDSolve[{DiffEQ==0,th[0]==Pi/4,th'[0]==0},{th},{t,0,10}];
gorigin={{0,1,0},{-1,0,0},{0,0,1}};
Do[pend={ (gorigin.gAB[(th[t]/.ODEsol)[[1]]].{0,0,1})[[1]],
  (gorigin.gAB[(th[t]/.ODEsol)[[1]]].{0,0,1})[[2]]};
  Show[Graphics[{PointSize[0.1],Point[pend],
  Line[{0,pend}]}],PlotRange->{{-1.5,1.5},{-1.5,1.5}},
  AspectRatio->Automatic},{t,0,10,0.05}]

```

This code sets the length of the pendulum, mass, and gravitational constant. Then the code solves the ordinary differential equation for $\theta(t)$ given the initial condition $\theta(0) = \frac{\pi}{4}$, $\dot{\theta}(0) = 0$. (The matrix `gorigin` rotates the solution into a coordinate frame where “down” is in the negative y direction so that the animation will be aligned properly.) The `Do` loop creates the animation using the relationship $[x(t), y(t), 1]^T = g_{AB}(\theta(t)) \cdot (0, 0, 1)^T$. Note: if copying directly from an electronic copy of this paper, be sure all characters (e.g., “ r ”) and carriage returns copy properly.