2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
Paper ID #50711

Launching Dynamics Students On a Trajectory of Kinematics Success

Mr. Jonathan D. Aurand, Dunwoody College of Technology

Jonathan Aurand has been teaching mechanical engineering at Dunwoody College of Technology since 2016. Prior to joining the faculty at Dunwoody, Aurand practiced as an engineer in the power industry serving as a consulting engineer for Heat Recovery Steam Generators (HRSGs) used in combined cycle power plants and cogeneration facilities. He currently resides in Minneapolis, MN with his wife and four children.

Dr. Peter L. L. Walls, Dunwoody College of Technology

Peter Walls is an Assistant Professor in the Mechanical Engineering department at Dunwoody College of Technology in Minneapolis, MN. He has a broad background of industry experience in New Product Development, Mining, Manufacturing, Defense, and Biotech.

Completed Paper: Launching Dynamics Students on a Trajectory of Kinematics Success

Abstract

Students (and instructors) of Dynamics often appreciate the practical applications and demonstrations of the subject. Dunwoody College of Technology has incorporated a kinematics and kinetics group project in the Dynamics course for eight offerings. The project is scaffolded over the course of the semester so students can apply the concepts they're learning in class to the project. Several deliverables throughout the semester help keep students on track. Recent implementations include a greater focus on computer programming to reinforce those skills from prior semesters.

This paper reviews the history of the project and presents findings for student learning as a result of participating in the project. More recent offerings have shifted from a design and build approach to more analysis and application to a pre-built design. Differences between the two approaches are explored and discussed along with the presentation of relevant data relating to student perceptions of learning and confidence with projectile kinematics and kinetics.

Background and literature review

The authors reviewed relevant articles from the last 25 years. Papers can be divided into three categories: 1) student-built project with no consideration of student learning, 2) no build, but connection of the project to student learning, and 3) student built projects with analysis of perceived student learning. Our summary of the literature is presented in the categories identified above.

Those that have a build but no analysis of student learning include Reffeor[1], Johnson and Sanders[2], and Berg[3]. Reffeor summarized the motivation for a projectile motion project to meet open-ended design objectives in ABET requirements early in the 21st century [1]. Johnson et. al provide a thorough analysis of a golf ball launching ramp they designed and built as undergraduates [2]. Berg tasked his class with creating a Rube Goldberg machine with different groups designing and building a "stage" of the machine which would be connected to other's stages to form the complete machine [3]. Each group had a different Dynamics principle (conservation of energy, impulse and momentum, projectile motion, etc.) that their stage was to demonstrate. As shown in the YouTube video, the machine is successful, though the learning and grading is not directly addressed in the paper.

Those that have no build, but evaluate student learning or perceptions thereof include Self and Redfield[4], Fong and Self[5], Osta and Kadlowec[6], and Pal et. al[7]. Self and Redfield reported student motivation and perceived learning for a series of projects as well as traditional learning instruments such as homework and presentations [4]. Their results show a spread of motivation and perceived learning among the activities considered. Somewhat surprisingly, students reported learning the most from homework compared to the other activities considered. Fong and Self report their use of a pre-built catapult for analysis in both an introductory Dynamics (with hand calculations) and an intermediate Dynamics course where the students incorporate computer modeling [5]. It is interesting to note that this project used pre-built catapults and students analyzed the work and energy as well as projectile motion (and stress for the intermediate course) and made engineering decisions regarding which forces are negligible (drag and friction). Osta and Kadlowec recount a projectile motion apparatus used in a compressed Dynamics course [6]. They found a small change in student grades which was not statistically significant. Most of the areas students self-reported the project impacted were in softer skills such as communication and teamwork. Pal et. al recently published work regarding mini-projects which use lab-like activities to reinforce learning. The results by these authors show a reduction in DFW rates by Hispanic transfer students using project-based homework [7]

The final category includes projects with some student designed build and an evaluation of learning. Sridhara used the classic egg drop challenge in a mechanical engineering technology (MET) Dynamics course and reported on a MET capstone to design and build a golf ball shooter and related results to student performance on a comprehensive test[8]. Pal and Zaurin developed three project-based homework assignments in which students develop a small demonstration of a Dynamics concept [9]. Their results show a small improvement in achieving an "A" grade and a decrease in DFW rates by those in the project-based homework group.

Dunwoody has used some hands-on project in the undergraduate Dynamics class since fall 2017. The first iteration involved two projects, each for approximately half of the semester. The first of the two projects was a loop-the-loop where energy concepts were applied to design and construct a course for a ball. The second was a catapult which had some friendly competition for launching a racquetball as far as possible (within specified design constraints) and analyzing the design. Students enjoyed both projects but got caught up more in the competition of the catapult than the analysis.

Learning from this experience, the next iteration involved designing and building a spring powered launcher to hit a target at a range of both vertical and horizontal positions within a design range. The project remained the same for several years with changes to the range of positions and target locations. Some students found difficulty in fabricating the launcher which detracted from learning and applying the Dynamics concepts. The most recent semester was adjusted to be a design project with pre-built assemblies for the launchers and students analyzed the kinetics to choose appropriate springs.

The goal of the project is to help students develop a better understanding of particle kinematics and kinetics using an energy approach. Further, programming skills, teamwork, project management, and design skills are cultivated in an environment different from that in which they were originally taught. Of particular interest to the authors is the question of whether requiring a build portion improves student learning of Dynamics principles. Other aspects such as teamwork and project management were also considered in the scope of our analysis.

Design and Build Project Overview

The design and build project was presented to students in multiple deliverables and course content was scaffolded to support learning in a just-in-time approach. The project was introduced after projectile kinematics was taught in lecture and reinforced with homework and quizzes. Students were divided into teams of 2-3 and provided an overview of the project scope. This included a summary of deliverables, the due dates for each, and the points associated with each deliverable. The first deliverable was also described in greater detail as that is the material the students should begin working on. An example table of deliverables from the 2023 course offering is reproduced as Table 1 below. The handouts for Phase 1 and Phase 2 are provided in Appendix A. The authors estimate that approximately two class periods were dedicated to introducing, discussing, and advising the students on the project during the course of the semester. The bulk of the time spent on the project was outside of class time.

Table 1: Summary of deliverables for the design and build approach from Fall 2023

Deliverable	Points	Due Date
Phase 1—Preliminary calculations	20	10/2/2023
Phase 2—Finalized design	20	10/27/2023
Project Testing	5	12/8/2023
Project Presentation	20	12/11/2023
Project Report	25	12/11/2023
Total	90	

Note that several weeks were provided for the procurement of materials (at the school's expense) and build process of the launcher between the finalized design and the project testing. The finalized design grading included a design review to ensure school resources are being spent appropriately. Phase 1 included doing an example calculation (with the same variables for each group) by hand and then creating a computer code in Python to accept inputs for target range and height. The code was required to output the minimum launch angle and two plots: one for launch velocity vs. launch angle and one for time of flight vs. launch angle. A rubric for the deliverable was provided in the LMS assignment as well as on the handout in class. This is presented just after projectile motion is covered in the class lecture, but before particle kinetics is developed.

Phase 2 included the particle kinetics. The mass of the projectile was provided to the students and several deliverables were required. The phase 2 deliverables for 2023 were:

1. Sample calculations for two different springs assuming the projectile has a mass of 50 grams.

2. Spring(s) selection from McMaster Carr
3. Updated python code that includes the spring constant and returns required spring displacement for launch.
4. Bill of Materials for your launcher build
5. SolidWorks Models for specialized items. Anything you plan to 3D print, machine, etc. should be modeled. I do not expect a full assembly model, but that would certainly cover what is needed.

Again, a rubric was provided to the students in the handout and attached to the assignment within the LMS.

Project testing is when the performance of the launchers is assessed. A series of three targets are placed and each team takes turns using their code and launcher to attempt to hit the target. Each team makes three attempts to hit the target; on the first attempt the team is required to use the output of their code to determine their pull-back distance but can make adjustments to this value on subsequent attempts. A hit on the first attempt scores more points than a hit on the last attempt. A perfect testing outcome can result in a small amount of extra credit. Some years, if testing is particularly poor, results may be scaled up to allow a reasonable outcome for the project grade (e.g. if the best team scored 2/5, that might be scaled by 1.5x to reduce the impact (pun intended!) on student grades). No deliverables besides the in-class testing are prepared by the students for this portion.

The final deliverables are a short presentation and a project report. In some years we've held a poster session along with other courses in the department and that counts as the presentation. Both the presentation and the report are intended to summarize the team's design and analysis work on the project, provide the results of their testing, and reflect on their learning.

Changes for the Non-Build Project

For fall 2025, changes were made to reduce the build time for the project by eliminating the building aspect. The decision to remove the build stemmed from several years of project builds that had mixed success. The 2024 offering started with three teams but due to in-semester withdrawals ended with two teams for project testing. One of the builds was fairly reliable and could hit the target, the other build was poor and barely functioned but still managed one hit out of nine total shots. 2023 had five teams. One of these teams failed to complete a build. Of the remaining four teams, two had reasonably functional designs while two had haphazard builds that were marginally functional. Looking further back, some failed designs were the result of poor choice of materials (e.g. using drain tile pipe rather than standard sch. 40 PVC pipe), lack of accounting for losses in spring sizing (e.g. one group realized their spring was undersized the day of testing so decided to raise their firing height by creating a makeshift pedestal for their build), and inconsistent release mechanisms. Photos of representative student-built launchers are shown in Figure 1.

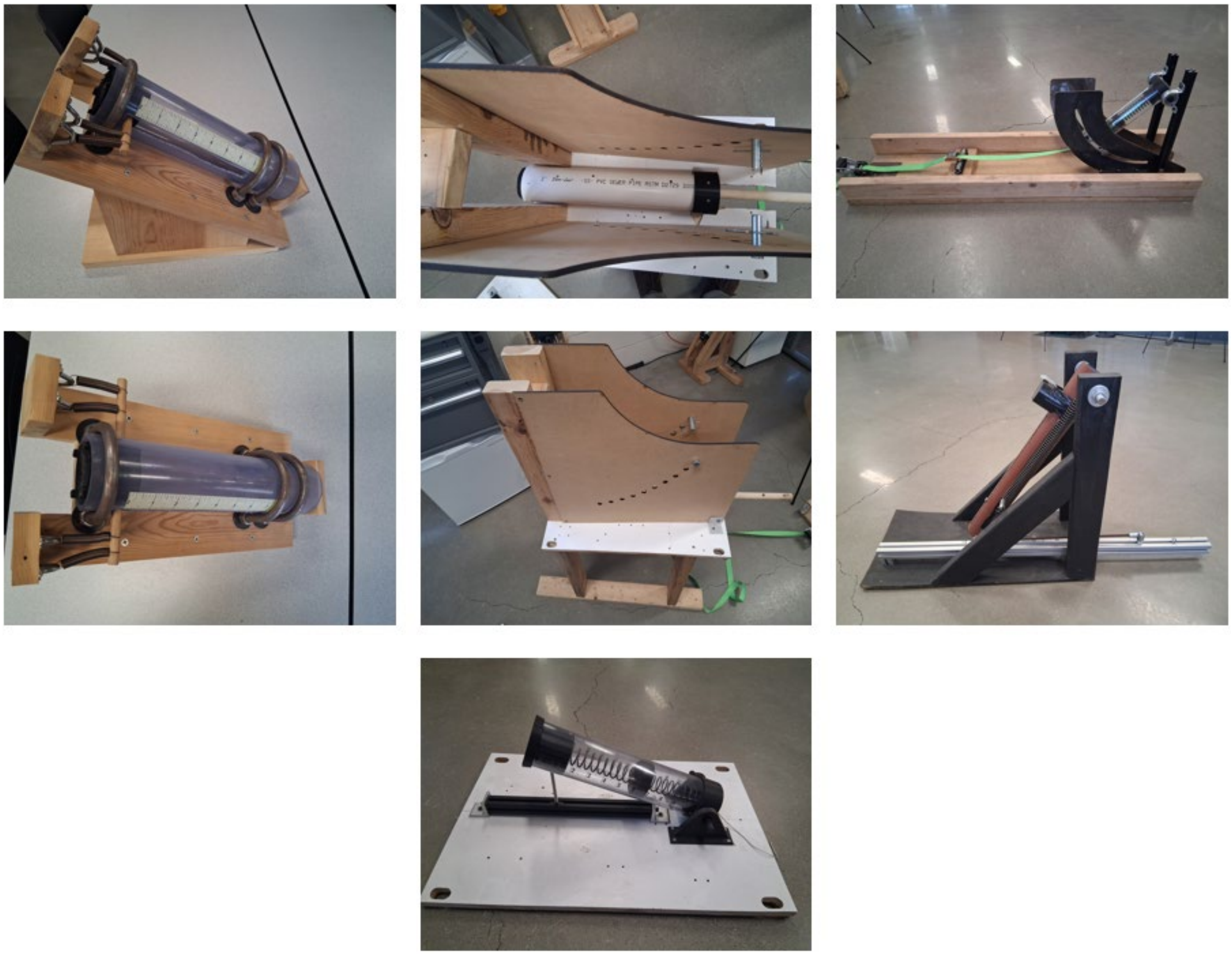

Figure 1: Representative launcher builds from students. Left column is two views of the same launcher using extension springs and a fixed launch angle. Middle column top two photos are same launcher using a compression spring and very thin PVC for a barrel with variable launch angles. Center bottom is a launcher using a clear barrel with compression springs in series and a variable launch angle. Right top is a robust design using a compression spring, variable launch angle, and a ratchet strap with release to fire while right bottom is similar but uses an extension spring and has a makeshift sight but no release mechanism.

Based on this history and students' propensity to tinker/hack builds together the instructor for the 2025 offering decided to remove the build aspect and instead require multiple spring setups. There was also some change to the programming aspect requiring a graphical user interface (GUI) in addition to the other functionality from past years. A successful student design from a past semester was replicated a few times to have several base assemblies to use for the testing aspect. Photos of the student-built inspiration and one of the replications are shown in Figure 2.

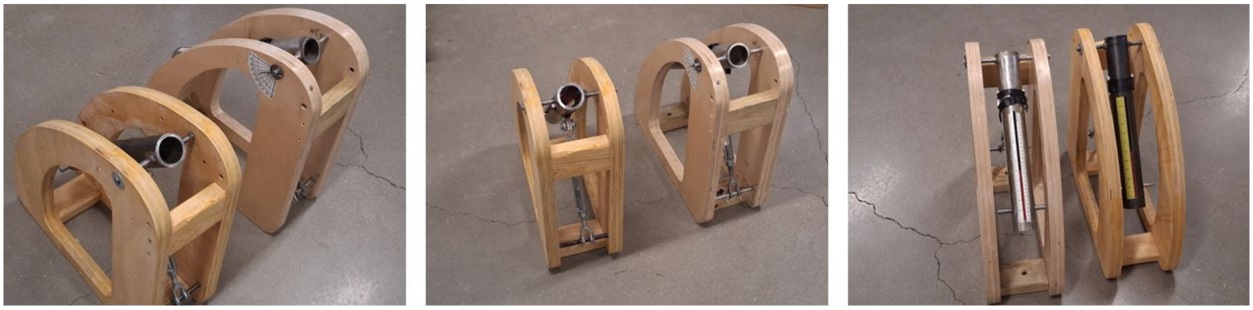

Figure 2: Student built inspiration (right build in first two photos and left in the third) and replicated design for the non-build project. This design requires extension springs but has variable launch angles and is very robust.

Additional scope was added to the analysis and sizing tasks to keep the project somewhat balanced to past offerings. Major changes to the project are summarized in Table 2. The student handouts for the changed project are also provided in Appendix A.

Table 2: Summary of changes from Design-Build to Design only

Description	Design-Build	Pre-built
Project build	Team designed and built	Use same pre-built foundation
No. spring setups	1	2
Python code	User-defined inputs/outputs	Graphical User Interface (GUI) Required

Results and discussion

Comparison of student feedback

As part of the report, students were asked to reflect on the project and determine what they would do differently if given the opportunity to repeat the project. The authors reviewed the reports from 2022-2024. Over this period there were 12 teams that completed the project. A total of 46 issues/improvements were identified by the teams. The authors categorized these into "build" issues and "non-build" issues. Build issues included things like using different materials, fabrication techniques, or types of springs. Non-build issues generally focused on teamwork, project management, testing, communication, and time management. Of the issues identified by the students, more than 65% were build-related over the period investigated. Almost none of the issues students mentioned were related to the Dynamics purposes of the project, namely teaching projectile kinematics and particle kinetics.

From the historic period feedback several themes emerged. About half of teams commented on doing test runs (or wishing they had done more test runs/calibration) and most teams identified that the assumptions made to simplify Dynamics problems (e.g. neglecting air resistance, assuming no friction, projectile being launched with no rotational kinetic energy) lead to differences between their calculated values and the results they encountered during the testing deliverable. While air resistance is negligible at the launch velocities encountered in this project, the students learned that the real world is not as neatly packaged as typical homework problems in a Dynamics text.

Student survey results

For the Fall 2025 offering a paper survey was administered to the class near the end of the semester. The participation rate was 92.8% with 13 responses. Students were asked to rank the Dynamics concepts the project helped them understand better, their programming preparedness, the biggest lesson learned, and what they would change about the project for future classes. 23% of the respondents did not follow directions in ranking (either by using checkmarks or ranking more than one item the same numeric score). In cases of multiple marked items, the average of the ordinals was assigned to each marked value (i.e. if 2 things were marked 1, each was given 1.5 as the average between 1 and 2). Results of the student feedback for the project helping them learn Dynamics concepts is presented in Figure 3 below, the horizontal line represents the median of the data, mean values are shown below in the text. Note that impulse and momentum had most responses at 4, with two outliers of 3.

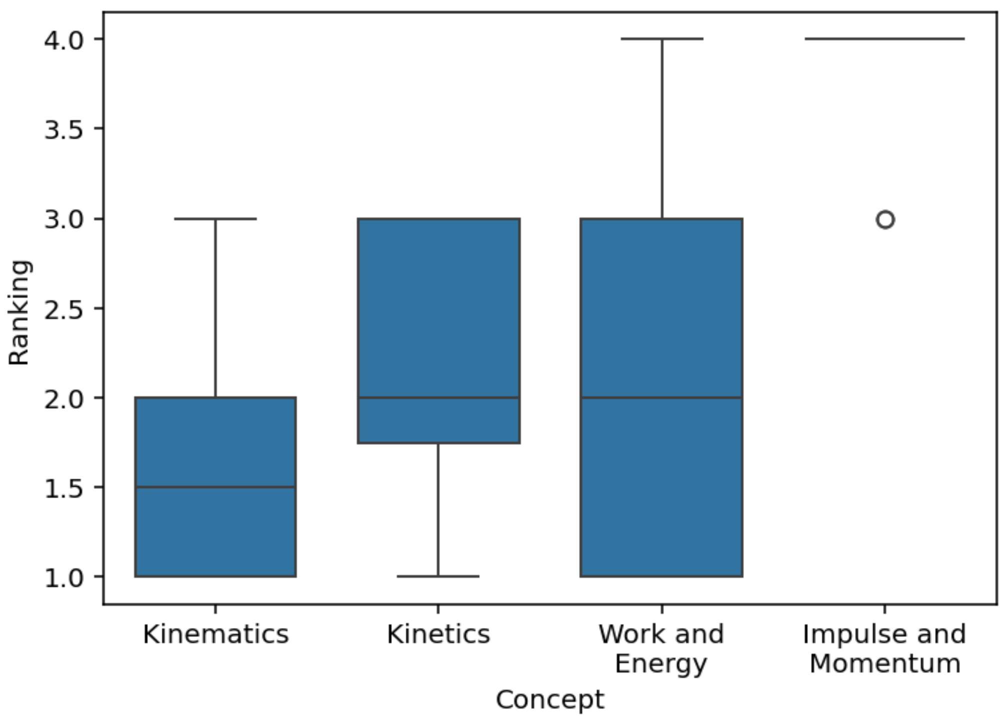

Figure 3: Responses ranking the Dynamics concept the project helped students understand better with 1 being helping the most.

Students indicated that their understanding of kinematics was most helped by the project (average ranking 1.65) followed by kinetics (2.08), work and energy (2.12), and finally impulse and momentum (3.82). It surprised the authors that work and energy was just behind kinetics as this setup is a classic work and energy problem. It was unsurprising that impulse and momentum scored so low as the projectile is in contact with the surface of the launcher throughout the launching process and the only impulse is the launching mechanism contacting the end of travel at the moment of launch. One intent of listing impulse and momentum was to serve as a control question. Only two responses ranked impulse and momentum as 3 while all other responses rated it 4. As this impulse is outside the students' scope of analysis it's expected that they didn't consider it to help their understanding. Perhaps analysis of the impact of the projectile with the target could be added to enhance impulse and momentum comprehension.

For programming preparedness, the students indicated a range from completely prepared to completely unprepared with an average of slightly less prepared than not (3.04/5 with 1/5 being completely prepared) while both the median and the mode were 3/5. The standard deviation for this item was 1.16, indicating considerable variability in perceptions of preparedness for the programming aspect. It is worth noting that the Dynamics course is sequenced one year after most students take programming. There were several comments about incorporating the GUI.

The most commonly reported lessons learned were the difference between calculations and the real world (or similar wording) with eight out of 14 comments. Three comments were related to programming and/or creating the GUI. Only one comment was related to softer skills ("time management"). This was in contrast to the historical period where multiple teams highlighted issues with time management, communication, and planning the project execution.

Responses to what students would change for future offerings fit into five themes. The responses to this prompt are shown in a histogram in Figure 4.

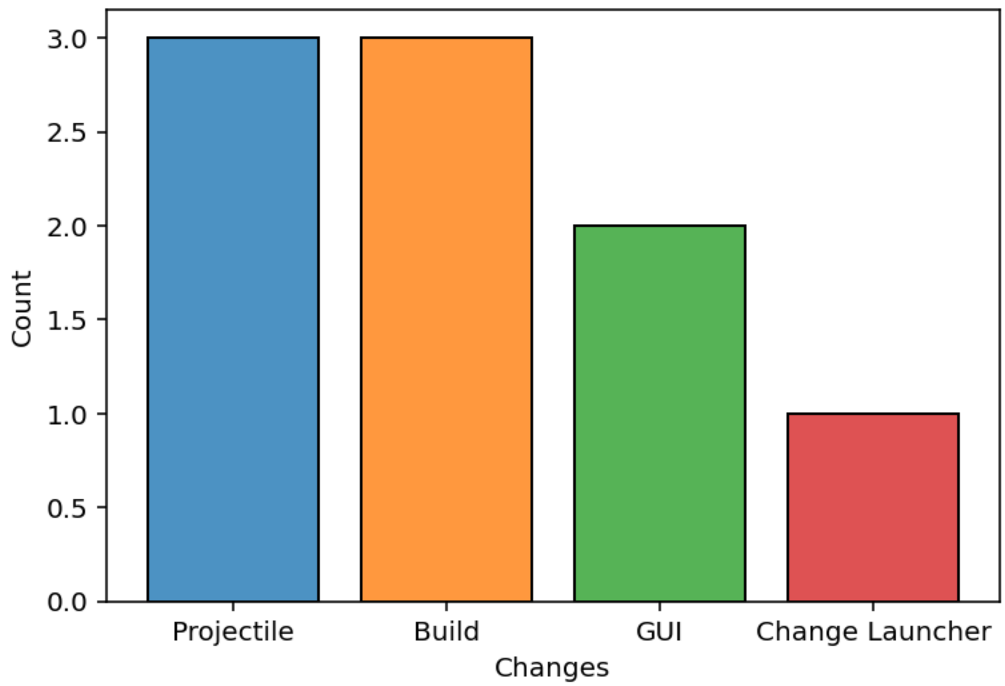

Figure 4: Histogram of changes suggested by students.

Three responses (25%) addressed the projectile itself—this year a squash ball and a rubber ball used previously with approximately twice the mass of the squash ball were the intended projectiles, but upon testing a squash ball the coefficient of friction between the ball and the launcher surface was so high the ball would jam within the launcher. The projectile was adjusted to be an end mill case with mass adjusted by adding and removing small masses within the case. While this worked well in many ways (e.g. it slides and launches with no rotational kinetic energy, the size is the same regardless of mass), student perceptions differed from the instructors. Three responses (25%) requested designing and building launchers themselves. Three responses (25%) addressed improving or designing a different release mechanism. Two responses (16.7%) suggested changes to the GUI requirement with one recommending utilizing a pre-built GUI that students would add to their codes and the other asking to remove the GUI requirement entirely. The final suggested improvement was to use a different provided launcher.

Hit rate results

One other metric we used to evaluate our change in this project was the hit rate of the design and build launchers compared to the hit rate for the pre-built launcher. Overall, the design and build hit rate averaged 25% with the best year hitting 35.6% of the attempts and the worst year hitting only 15.6% of the attempts. With the pre-built launchers in the fall of 2025 the hit rate improved to 32 of 63 attempts or 50.8%. The authors take this improvement as a step closer to the mark and conclude that much of the frustration with fabricating launching assemblies can be mitigated through this ready-made approach.

Curvilinear motion skills pass rate

Since 2023, the Dynamics course has been delivered using skills based (AKA competency based) assessment [10], [11]. Fall 2023 and fall 2024 used skills and the design and build project while fall 2025 used skills and the pre-built launcher. In all three offerings, all students passed the curvilinear motion skill, which includes projectile motion. This also supports the conclusion that removing the build portion did not adversely affect learning of the underlying physical concepts.

Conclusion

Limitations

This work was completed in a teaching environment with a small course section and without prior use of direct comparisons of student understanding of specific topics (e.g. concept inventories[12]). This reality makes it difficult to compare conceptual development between the student groups who built launchers and those who did not. Further, the extended duration of the project makes separating out the impact of the project itself from other learning activities such as homework, skills tests, etc. difficult if not impossible even if there were data from more direct measures. Nonetheless, student feedback and instructor observations still provide insight into the differences between these two populations.

Conclusion and possible future changes

Based on the indirect measures considered and student feedback, the authors conclude that removing the student build process did not hinder student learning. On the contrary, students were able to double the hit rate while still learning the Dynamics concepts at least as well as prior offerings, refreshing and expanding their computer programming skills, and working in a team.

Regarding the student complaints about wanting to build launchers, the drastic decrease in build-related lessons learned indicates that while our students would enjoy building a launcher—and many of those making the comments are already decent fabricators, the building process does not necessarily improve learning Dynamics concepts and may inhibit them as indicated by one comment from the design and build era, "Lastly, I learned to check what a project was worth before putting my all into trying and failing to complete it. Surely my grade would have been higher if I didn't build it at all so I could focus on studying for skills tests and keeping up with homework instead."

Based on student feedback and instructor observations, the change to a pre-built assembly improved students' ability to analyze and evaluate the dynamics of a system. While still present, student perceptions of softer engineering skills such as time management, project management, teamwork, and communications, were less significant than in the design and build project. Students were also more successful in actually hitting the target with the pre-built launchers.

Future improvements could involve changing the projectile shape in addition to the mass. This change would help students realize the impact of drag is negligible at the launch velocities present. Adding a student-designed and built release mechanism may also allow the students to scratch a bit of the build "itch" they have and could facilitate teaching softer skills such as project management and communication.

Finally, removing the build aspect of the project would allow for a compressed schedule to create room in the semester for a second project. Using the pre-built launchers reduces the design and procurement phase durations. This additional time could be used for a rigid body dynamics project, which is covered later in the course, or other more advanced topics such as more in-depth coverage of vibrations.

References

[1] W. Reffeor, "Incorporating Design In An Introduction To Dynamics Course," presented at the 2002 Annual Conference, Jun. 2002, p. 5. doi: 10.18260/1-2--10263.

[2] D. Johnson and J. Sanders, "A Golf Ball Launcher: An Engineering Dynamics Project," presented at the 2004 GSW, Lubbock, TX, Apr. 2004, p. 11. doi: 10.18260/1-2-620-40006.

[3] D. Berg, "Use of a Rube Goldberg Design Project for Engineering Dynamics," presented at the 2015 ASEE Annual Conference & Exposition, Seattle, Washington, Jun. 2015, p. 16. doi: 10.18260/p.24976.

[4] R. Redfield and B. Self, "New Approaches In Teaching Undergraduate Dynamics," presented at the 2001 Annual Conference, Albuquerque, New Mexico, Jun. 2001, p. 11. doi: 10.18260/1-2--9601.

[5] L. Fong and B. Self, "Modeling the Dynamics of a Small Catapult to Enhance Undergraduate Studies," presented at the Pacific Southwest Section Meeting, National University, San Diego, CA, Mar. 2009, p. 9. doi: 10.18260/1-2-1153-52380.

[6] A. Osta and J. Kadlowec, "Hands-on Project Strategy for Effective Learning and Team Performance in an Accelerated Engineering Dynamics Course," presented at the 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana, Jun. 2016. doi: 10.18260/p.25444.

[7] S. Pal, S. Outerbridge, M. Chimhrad, M. Taub, and H. J. Cho, "Enhancing Student Learning in a Blended Undergraduate Dynamics Course through Hands-on Mini-Projects," presented at the 2025 ASEE Annual Conference & Exposition, Montreal, Quebec, Canada, Jun. 2025, p. 14. doi: 10.18260/1-2--56425.

[8] B. S. Sridhara, "Course-Related Undergraduate Projects for Dynamics," presented at the 2013 ASEE Annual Conference & Exposition, Atlanta, Georgia, Jun. 2013, p. 12. doi: 10.18260/1-2--19359.

[9] S. Pal and R. Zaurin, "Work in Progress: Project-Based Homework: An Ongoing Study on Engineering Analysis-Dynamics," presented at the 2021 ASEE Virtual Annual Conference Content Access, Virtual Conference, Jul. 2021, p. 12. doi: 10.18260/1-2--38190.

[10] S. A. Atwood, M. T. Siniawski, and A. R. Carberry, "Using Standards-based Grading to Effectively Assess Project-based Design Courses," presented at the ASEE Annual Conference & Exposition, Jun. 2014.
[11] K. DeGoede, "Competency Based Assessment in Dynamics," presented at the 2018 ASEE Annual Conference & Exposition, Jun. 2018.
[12] M. Koretsky *et al.*, "the AiChE Concept warehouse: A web-Based tool to Promote Concept-Based instruction," *Adv. Eng. Educ.*, vol. 4, no. 1, Winter 2014.

Appendix A

MENG 2230 Dynamics—Project Phase 1(Original)

A common application of kinetics and kinematics is projectile motion. In this project you will be designing and building a spring powered launcher to hit targets. The targets will be located 5-15 feet away from the launcher at heights of 0-6 feet. During testing, each group will be allowed to measure the position of the target relative to the launcher. The design shall be adjustable to control two parameters:

1. The launch angle
2. The launch velocity

Each team will be made up of 2-3 students. Each team will build their own launcher. There are several deliverables/milestones in the project. The approximate timeline and points for each deliverable are listed in Table 1 below.

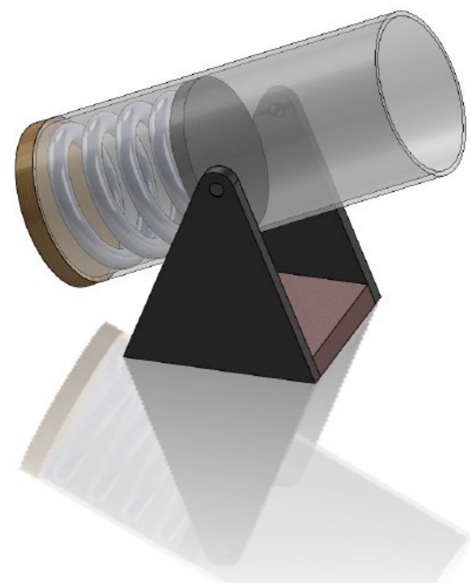

Figure 5: Example Spring Launcher

Deliverable	Points	Due Date
Phase 1—Preliminary calculations	20	10/6/2022
Phase 2—Finalized design	20	11/8/2022
Project Testing	5	12/8/2022
Project Presentation	20	12/13/2022
Project Report	25	12/13/2022
Total	90	

For Phase 1 there are two parts to be completed. Part A is solving the kinematics for an example projectile motion problem. Part B is generalizing the solutions from Part B and incorporating it into a python code to help in spring sizing. The inputs to the python code are:

1. Horizontal distance to target from a datum on the launcher.
2. Elevation of the target from the ground
3. Elevation of the projectile when the spring is undeformed (i.e., the point at which the spring will transfer energy to the projectile).

The output of the code (at this point) should be:

1. Minimum launch angle
2. Duration of the flight as a function of the launch angle
3. Required initial velocity as a function of launch angle

Deliverables:

1. Sample calculations of minimum launch angle, launch velocity, and duration of flight for a target located 12 feet away horizontally, a vertical displacement of 4 feet and a launch angle of 38°.
2. A python Jupyter notebook that takes the inputs above and returns:
    a. Minimum launch angle
    b. A graph of required initial velocity for various launch angles (plot of V-θ)
    c. A graph of time of flight for various launch angles (plot of t-θ)

The rubric for this portion of the project is provided below in Table 2 and in Canvas.

Table 3: Rubric for Phase 1

Launcher Phase 1 Rubric

Criteria	Ratings					Pts
Example Problem Example calculation neat and correct	5 pts Both correct and neat	3.75 pts Correct, but not as easy to follow	2.5 pts Incorrect, but easy to follow	1.25 pts Incorrect, and difficult to follow	0 pts No example	5 pts
Jupyter Comments/Organization Jupyter notebook documentation and organization	5 pts The Jupyter notebook is well documented and easy to read. The flow of the work is easy to follow.	3.75 pts There are some comments/markdown cells, but not as easy to use as it could be	2.5 pts The notebook functions, but is difficult to follow. Little or no comments or markdown cells.	1.25 pts Notebook does not function, but has code	0 pts Notebook is missing or severely incomplete	5 pts
Jupyter Inputs Inputs are clear and accurate	3 pts All inputs are present and clear	1.5 pts Inputs present, but incomplete or incoherent	0 pts Inputs are hardcoded or missing			3 pts
Jupyter t-θ Plot Plot of velocity versus launch angle	3.5 pts Plot is correct, has axes labels, and is easy to read/interpret	2.25 pts Plot is correct (or appears so) but is missing axes labels or difficult to read	1 pts Plot appears incorrect, but has labels	0 pts Plot is difficult to read or is missing.		3.5 pts
Jupyter v-θ Plot Plot of duration of flight versus launch angle	3.5 pts Plot is correct, has axes labels, and is easy to read/interpret	2.25 pts Plot is correct (or appears so) but is missing axes labels or difficult to read	1 pts Plot appears incorrect, but has labels	0 pts Plot is difficult to read or is missing.		3.5 pts
						Total Points: 20

MENG 2230 Dynamics—Project Phase 2 (Original)

In Phase 1, each team calculated the minimum launch angle, required launch velocity vs. launch angle, and time of flight vs. launch angle. In Phase 2, we expand upon that work to select an appropriate spring and add spring deformation calculations to the project code.

Use conservation of energy for the launcher and calculate the required stored energy for the range of initial velocity and angle of the launch. The procedure and equations are as follows:

1- The energy stored in the spring will be released in forms of potential energy (due to displacement of the ball to the location of the launch) and kinetic energy of the ball (due to its velocity). The equations and schematic figure for this transformation are shown as follows:

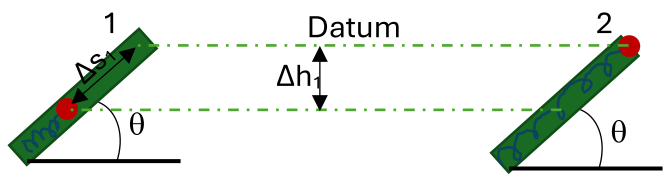

Fig 1. Schematic illustration of the launch.

$$\frac{1}{2}k\Delta {s_1}^2 - mg\Delta h_1 + \frac{1}{2}m{V_1}^2 = \frac{1}{2}k\Delta {s_2}^2 + mg\Delta h_2 + \frac{1}{2}m{V_2}^2 \qquad (1)$$

In Figure 1, a spring is compressed by the length Δs_1 from the neutral position (datum) as shown in Fig. 1.2. Since spring has momentum, it will slightly pass the datum upon release of the ball. This results in distribution of energy between the ball at launch and spring extension. To avoid this and assign all energy to the ball, a stopper can be designed to stop the launcher cup (holding the ball) from extensions beyond the datum. In this scenario both Δs_2 and Δh_2 are zero. Moreover, V_1 is zero and V_2 and θ are calculated to hit a fixed target (location of the target is measured) in Phase 1 and known.

Using equation 2 that relates displacement of the spring to vertical displacement of the ball and replacing it in equation 1, we will have:

$$\Delta h_1 = \Delta s_1 \sin(\theta) \qquad (2)$$

$$\frac{1}{2}k\Delta {s_1}^2 - mg\Delta s_1 \sin(\theta) = \frac{1}{2}m{V_2}^2 \qquad (3)$$

In equation 3 Δs_1 and k are design parameters and unknown. We can rearrange the equation for k as in equation 4 as follows:

$$k = \frac{2mg\sin(\theta)}{\Delta s_1} + m\left(\frac{V_2}{\Delta s_1}\right)^2 \qquad (4)$$

To solve equation 4 for k, the only unknown variable is the required compression of the spring from the location of the launch, Δs_1. To select Δs_1 and calculate k, several considerations are necessary that are listed below:

- When k is large, large spring energies will be stored for a unit compression of a spring while smaller energies are stored for smaller k and for the same compressed length. This means that better control and precision on the amount of the required energy and velocity of launch is available by using a spring which a smaller k.
- When k is small, Δs_1 is large, meaning that the change in the length of the compressed spring for the required amount of energy is larger, which requires a longer physical component to guide the ball.
- When k is large, the force that is required to pull the spring is also larger. For instance, assume that 400 units of energy is required for the launch. From the equation below:

$$\frac{1}{2}k\Delta s^2=400$$

Now consider 2 scenarios of:

Scenario 1: $k = 2 \rightarrow \Delta s = \sqrt{2(400)/k} = 20 \rightarrow F_1 = k.\ \Delta s = 2(20) = 40$

Scenario 2: $k = 8 \rightarrow \Delta s = \sqrt{2(400)/k} = 10 \rightarrow F_2 = k.\ \Delta s = 8(10) = 80$

It is clear that increasing k, increases the required force to compress the spring to achieve the same required energy for the launch.

For Phase 2 there are two parts to be completed. Part A is solving the kinetics for the projectile used. The goal of this is to determine maximum spring energy required to hit the target. Part B is selecting the spring and determining design details (sizes of the launch device, how you will change angle, how the projectile will be held, etc.). You will also update your python code to output the required spring displacement based on your spring selection and design details. As a reminder the inputs to the python code are:

1. Horizontal distance to target from a datum on the launcher.
2. Elevation of the target from the ground
3. Elevation of the projectile when the spring is undeformed (i.e., the point at which the spring will transfer energy to the projectile).
4. (new) Desired launch angle
5. (new) Spring constant

The output of the code (at this point) should be:

1. Minimum launch angle
2. Duration of the flight as a function of the launch angle
3. Required initial velocity as a function of launch angle
4. (new) Required spring displacement to achieve launch velocity at selected launch angle.

Deliverables:

1. Sample calculations for two different springs assuming the projectile has a mass of 50 grams.
2. Spring(s) selection from McMaster Carr
3. Updated python code that includes the spring constant and returns required spring displacement for launch.
4. Bill of Materials for your launcher build
5. SolidWorks Models for specialized items. Anything you plan to 3D print, machine, etc. should be modeled. I do not *expect* a full assembly model, but that would certainly cover what is needed.

The rubric for this portion of the project is provided below in Table 2 and in Canvas.

Table 4: Rubric for Phase 2

Launcher Phase 2 Rubric

Criteria	Ratings					Pts
Example Problem Example calculation neat and correct	5 pts Both correct and neat	3.75 pts Correct, but not as easy to follow	2.5 pts Incorrect, but easy to follow	1.25 pts Incorrect, and difficult to follow	0 pts No example	5 pts
Spring Selection Spring selection	5 pts Spring selection is clear and appropriate for the application. It is clear whether the arrangement is series or parallel.	3.75 pts A spring is indicated, but may not be appropriate or unclear on the spring arrangement.	2.5 pts There is some work to size/select a spring, but it is unclear what needs to be ordered.	1.25 pts No spring selected, but some meager work toward that end.	0 pts No spring work shown.	5 pts
Python Code Updated Code is updated to perform required operations	3 pts All inputs and outputs are provided and appear accurate.	1.5 pts Inputs and outputs are present, but code is difficult to follow and/or missing clear results.	0 pts Items are hard coded or don't update.			3 pts
Bill of Materials Bill of Materials provided and clear	3.5 pts Bill of Materials provides size, quantity, cost, and vendor for all items. It is clear and easily interpreted.	2.25 pts BOM is present, but missing some of the full points items.	1 pts BOM present, but very messy or incomplete.	0 pts No BOM provided.		3.5 pts
SolidWorks Models SolidWorks models provided for special components	3.5 pts Custom components have complete SolidWorks part models with materials, fully defined sketches, good modeling practice.	2.25 pts Custom part models are partially provided or miss some good modeling practices.	1 pts Some part provided but may not have good modeling practice and are incomplete.	0 pts No parts models provided.		3.5 pts
						Total Points: 20

MENG 2230 Dynamics—Project Phase 1 (Revised)

A common application of kinetics and kinematics is projectile motion. In this project, you will be analyzing a spring-powered launcher to hit targets (picture of launcher below in Figure 1). The targets will be located 5-16 feet away from the launcher at heights of 0-5 feet. You'll eventually need to account for 2 different projectiles and spring setups. During testing, each group will be allowed to measure the position of the target relative to the launcher to allow adjustment to hit the target. The analysis should be able to account for the following variables on the day of testing.

1. The launch angle
2. The launch velocity
3. Projectile
4. Springs Config

Each team will be made up of 2 students. Each team will be using Python to develop a numerical solver for the general projectile problem described above. There are several deliverables/milestones in the project. The timeline and points for each deliverable are listed in Table 1 below.

Figure 6: Spring Launcher

Table 5: Project Deliverables

Deliverable	Points	Due Date
Phase 1—Preliminary calculations & oral review of code	25	10/06/2025 (wk 7)
Phase 2—Finalized design – Spring selection with calculations	15	10/28/2025 (wk 10)
Proof of Concept / Prelim Validation (setup your own target/conditions)	15	11/17/2025 (wk 13)
Project Testing	10	12/3/2025 (wk 15)
Project Poster & Code	35	12/10/2025 (wk 16)
Total	**100**	

For Phase 1 there are two parts to be completed. Part A is solving the kinematics for an example projectile motion problem. Part B is generalizing the solutions from Part B and incorporating it into a python code to help in spring sizing.

The inputs to the python code are:

1. Horizontal distance to target from a datum on the launcher.
2. Elevation of the target from the ground
3. Elevation of the projectile when the spring is undeformed (i.e., the point at which the spring will transfer energy to the projectile).

The output of the code (at this point) should be:

1. Minimum launch angle
2. Duration of the flight as a function of the launch angle
3. Required initial velocity as a function of launch angle

Deliverables:

1. Sample calculations of minimum launch angle, launch velocity, and duration of flight for a target located 12 feet away horizontally, a vertical displacement of 4 feet and a launch angle of 35°.
2. A python code (Jupyter notebook, Spyder, etc.) that takes the inputs above and returns:
    a. Minimum launch angle
    b. A graph of required initial velocity for various launch angles (plot of V-θ)
    c. A graph of time of flight for various launch angles (plot of t-θ)
3. Oral review of code and calculations
    a. As a group, schedule a 15 meeting with me to review and discuss your code and work
    b. Review must occur either in the week it is due (Oct 6^{th} to Oct 10^{th}) or prior.

Make sure to review the rubric for this portion of the project in Canvas.

MENG 2230 Dynamics—Project Phase 2 (Revised)

In Phase 1, each team calculated the minimum launch angle, required launch velocity vs. launch angle, and time of flight vs. launch angle. In Phase 2, we expand upon that work to select appropriate springs and add spring deformation calculations to the project code.

Background and Overview

Use conservation of energy for the launcher and calculate the required stored energy for the range of initial velocity and angle of the launch. Some guidance is provided below:

The energy stored in the spring will be released in forms of potential energy (due to displacement of the ball to the location of the launch) and kinetic energy of the ball (due to its velocity). The equations and schematic figure for this transformation are shown as follows:

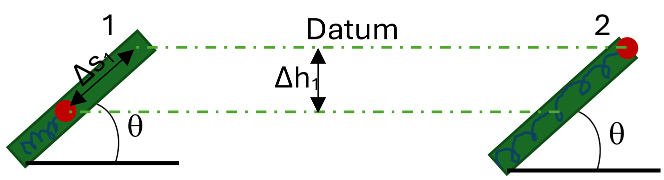

Fig 1. Schematic illustration of the launch.

$$\frac{1}{2}k\Delta {s_1}^2 - mg\Delta h_1 + \frac{1}{2}m{V_1}^2 = \frac{1}{2}k\Delta {s_2}^2 + mg\Delta h_2 + \frac{1}{2}m{V_2}^2 \qquad (1)$$

In Figure 1, a spring is compressed (or stretched as in the physical launcher) by the length Δs_1 from the neutral position (datum) as shown. Since the spring has momentum, it will slightly pass the datum upon release of the ball. This results in distribution of energy between the ball at launch and spring extension. To avoid this and assign all energy to the ball, a stopper is used to stop the launcher cup (holding the ball) from extensions beyond the datum. In this scenario both Δs_2 and Δh_2 are zero. Moreover, V_1 is zero and V_2 and θ are calculated to hit a fixed target (location of the target is measured) in Phase 1 and are known.

To solve for k, the only unknown variable is the required compression of the spring from the location of the launch, Δs_1. To select Δs_1 and calculate k, several considerations are necessary that are listed below:

- When k is large, large spring energies will be stored for the same amount of compression when compared to smaller energies stored for smaller k. The trade off is better control and precision on the amount of the required energy and velocity of launch when using a spring which has a smaller k.
- When k is small, Δs_1 is large, meaning that the change in the length of the spring for the required amount of energy is larger, which requires a longer physical component to guide the

ball. Make sure to examine the physical launcher, there are approximately 11 inches of "pull-back" available in the design.

- When k is large, the force that is required to pull the spring is also larger and harder to control and physically hold onto. For instance, assume that 400 units of energy is required for the launch. From the equation below:

$$\frac{1}{2}k\Delta s^2 = 400$$

Now consider 2 scenarios of:

Scenario 1: $k = 2 \rightarrow \Delta s = \sqrt{2(400)/k} = 20 \rightarrow F_1 = k.\ \Delta s = 2(20) = 40$

Scenario 2: $k = 8 \rightarrow \Delta s = \sqrt{2(400)/k} = 10 \rightarrow F_2 = k.\ \Delta s = 8(10) = 80$

It is clear that increasing k, increases the required force to extend the spring to achieve the same required energy for the launch.

To be completed

For Phase 2 there are two parts to be completed. Part A is solving the kinetics for the projectiles used. The goal of this is to determine maximum spring energy required to hit the target. (Remember, there are losses in the system) Part B is selecting the springs for the provided launcher (make sure you are examining the details of the launcher, how you will change angle, how the projectile will be held, etc.). You will also update your python code to output the required spring displacement based on the combination of spring selection and projectile. As you further refine your code, you'll improve the user experience. For this phase, a Graphical User Interface (GUI) is required for the user to input the values; there shouldn't be command line inputs at this phase. While you develop the code, it is fine to do so, but not for the deliverable.

As a reminder the inputs to the python code are:

1. Horizontal distance to target from a datum on the launcher
2. Elevation of the target from the ground
3. Elevation of the projectile when the spring is undeformed (i.e., the point at which the spring will transfer energy to the projectile).
4. (new) Desired launch angle
5. (new) Spring constant
6. (new) projectile mass

The output of the code (at this point) should be:

1. Minimum launch angle
2. A graph of time of flight for various launch angles (plot of t-θ)
3. A graph of required initial velocity for various launch angles (plot of V-θ)
4. (new) Required spring displacement to achieve launch velocity at selected launch angle & combination of spring constant and projectile mass

Deliverables:

1. Sample calculations for two different springs assuming the projectile has a mass of 50 grams. These sample calculations should be properly formatted and stepped through to guide the person reading them.
    a. The springs should have spring rates at least 70% different from each other (e.g. spring 2 could be 1.7 times stiffer)
    b. Keep in mind that a second projectile of approximately 25g is also going to be used, so you'll want to keep this in mind when you are selecting springs
2. Spring(s) selection from McMaster Carr (https://www.mcmaster.com/)
3. Updated python code that includes the spring constant and returns required spring displacement for launch for the provided physical launcher
4. Code must have a basic GUI for the user to input their launch parameters
    a. There are several available libraries to achieve this. In your submission, illustrate what will need to be done on my end to successfully run your code if it is beyond a built in library (e.g. pip install "library name"
5. A short writeup explaining how you used any assistance for code development, e.g. Chat GPT, Gemini, Stack Exchange, etc.
6. A single document should contain: Sample calculations, Spring(s) selection, and the short writeup on assistance.
7. Oral review of submission
    a. As a group, schedule a 15 meeting with me to review and discuss your code and work
    b. Review must occur either in the week it is due (Oct 27th to Oct 31st) or earlier if ready.
    c. During this time, we'll confirm your design and spring(s) selection from mcmcaster.

MENG 2230 Dynamics—Project Phase 3 (New)

In Phase 2, each team calculated the two set of springs they'd like to use for the remainder of the project. Additionally the code was augmented to tell the user how much they should pull the launcher back in order to hit the target. In Phase 3, your team will do some preliminary testing of your chosen setup to see if what you've predicted is accurate!

Reminder of Parameters introduced in Phase 1 for test day

- The target will be located 5-16 feet away from the launcher at heights of 0-5 feet.
- There are two projectiles that you'll need to use on final test day: a smaller 25g ball and a larger 50g ball
- You'll need to calculate, on the fly, how much to pull back the springs (and which springs) given a random set of the above options.
    - Also…. I may force you to choose a random angle as well

Testing Equipment & Suggestions

- I recommend setting up targets and precisely measuring their location relative to the launcher (put something like a water bottle to hit as a target)
- There are three "identical" launchers (one is located in the mezzanine area in CEE, the other two are located in CEE165) – Make sure to return them to where you found them when finished testing
    - Make sure you inform me ASAP if something breaks or isn't working as you think it should
- The projectiles will be in CEE165 & on the whiteboard rail in the mezzanine
    - Also make sure to return them when done or if you can't find them
- DON'T DRY FIRE THE LAUNCHER….. this severely increases the chances of breaking it (i.e. make sure it has the projectile in it)

Deliverables

- At a minimum the following tests (I suggest doing a good amount of testing)
    - Two tests for the 50g projectile with different conditions
    - Two tests for the 25g projectile with different conditions
- For both of the above tests
    - Make sure the two tests are reasonably different from each other for each mass (e.g. 10 ft vs 16 ft, 50 deg vs 30 deg). Don't do two tests that only vary by 6 inches and 5 degrees.
    - <u>Submit one video of a test of the 50g projectile and one video of the 25g projectile</u> (Make it clear in the video what the conditions are by saying them in the video)
    - Document the conditions for each test, e.g. target distance/height, projectile, calculated pullback, how close was it to the target, how much farther or less did you have to pull back the spring to hit the target?
        - Any difference between the calculated pullback and actual required (assuming the calculations are accurate!) can be attributed to "loss" in the system or assumptions breaking down (e.g. drag). Hint: You can look at incorporating a loss term in your prediction code.
- <u>Submit in a single document what you found during testing making it clear for each test.</u> Make sure to capture your thinking on each test as you perform it. It much better to take precise notes at the time, than to try and remember later! -Take pictures and include them in the document.
- If it's clear that you need to make a change, tell me sooner rather than later (i.e. before submission)