

Underwater Model Rockets: An Innovative Design Problem and Competition for Undergraduate Students in Engineering, Math and Science

Richard Layton, Joshua Holden, Tina Hudson and Laurence D. Merkle
Rose-Hulman Institute of Technology

Abstract

For a recent student conference, the authors developed a day-long design problem and competition suitable for engineering, math and science undergraduates. This paper describes the design problem, apparatus, software and tutorials for others who may be interested in replicating and improving the competition. Detailed plans for the apparatus, circuits, computer interfaces and computer programs and tutorials are made available via the Internet. The results of a personal self-evaluation (PSE) from the design competition are described.

Introduction

An annual student conference is sponsored by the Midwestern Undergraduate Private Engineering Colleges (MUPEC) group, comprising the institutions listed in Table 1. The purpose of the conference is to give undergraduate engineering, science and math students from these institutions a forum to showcase their work in oral and poster presentations. A different institution hosts the event each year.

The conference often includes a design competition in addition to the oral and poster presentations. The challenge for the conference organizers is to create a design problem suitable for students from a variety of science, math and engineering disciplines. This paper describes the design competition the authors developed for the 2004 MUPEC conference hosted by Rose-Hulman Institute of Technology. Our goal in designing the competition was to create a day-long design problem suitable for undergraduates in engineering, math and science. Our goal in presenting this work is dissemination: to describe the design problem, apparatus, software and tutorials for others who may be interested in replicating and improving the competition. Detailed plans are available via the Internet.

Table 1: MUPEC Member Institutions

Cedarville Univ.	Milwaukee School of Engng	Tri-State Univ.
Indiana Inst. of Techn.	Ohio Northern Univ.	Univ. of Evansville
Kettering Univ.	Rose-Hulman Inst. of Techn.	Valparaiso Univ.
Lawrence Techn. Univ.	St. Louis Univ.	

The Design Problem and Competition

A waterproofed model rocket is installed in the base of a vertically-mounted polycarbonate tube as shown in Fig. 1. The tube is filled with water and the rocket is launched.

The design problem. The design problem is to configure the system to maximize a figure of merit J defined as the energy ratio given by

$$J = \frac{PE_2}{KE_1}, \quad (1)$$

where KE_1 is the rocket's initial kinetic energy at launch and PE_2 is the rocket's maximum potential energy mgh , where m is the mass of the rocket and payload and h is the measured maximum height reached by the rocket while still submerged. Since the conversion of kinetic energy to potential energy is not 100% efficient, $0 < J < 1$. The larger the value of J , the more efficient is the energy conversion. The design goal is to configure the system for maximum efficiency.

Students work in teams to configure their launch. Their design determines three parameters:

1. The mass and shape of the payload, a clay nose-cone molded by the students and oven-fired.
2. The height of the vertical water column such that the rocket just breaks the surface of the water at the peak of its trajectory.
3. Rocket motor-type selection (A8, B6 or C6).

Additional design constraints include: no trial rocket launches are permitted; some of the system parameters are unknown and have to be measured or estimated; and minimum total mass requirements (motor, body and payload) of 123 g for the A8 motor, 268 g for the B6 motor, and 388 g for the C6 motor.

Students use analysis, computer simulation and appropriate computational methods to determine the combination of the three design parameters that produces the maximum value of the figure of merit. The design problem and description are not distributed to participants prior to the conference.

The competition. For the competition, a team's rocket and payload is weighed and installed in the test fixture per their design configuration. Launch is enabled by using a cryptography program to decipher a coded message to obtain a launch code (similar to a PIN) and typing the launch code into another computer window. At the same time, at another computer, a student from another team is deciphering the same coded message to prevent launch. The competition is illustrated in Fig. 2.

The cryptography exercise is included in the competition to appeal to the non-engineering participants. A math or science student having no interest in particle mechanics or fluid mechanics can still participate by attending the cryptography workshop and competing as a code-breaker.

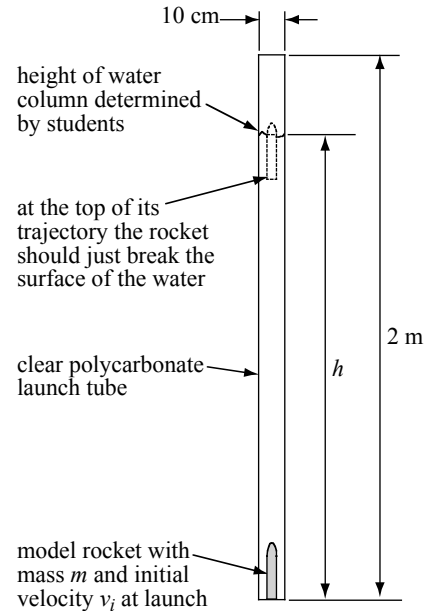


Fig. 1: Design problem schematic.

If the launch-team code-breaker succeeds first, the computer closes a firing circuit and launches the rocket. If the prevent-team code-breaker succeeds, their opponent is prevented from sending the launch code and the prevent-team wins the round. In this case, the rocket is launched manually to complete the launch-team's scoring. The height the rocket achieves while submerged is measured. The process repeats until all teams have been both the launch-team and the prevent-team.

Team scores are based on the experimental figure of merit, the prediction of the water column height, aesthetics (painted nose-cones), and the code-breaking times for both the launch attempt and the prevent-launch attempt.

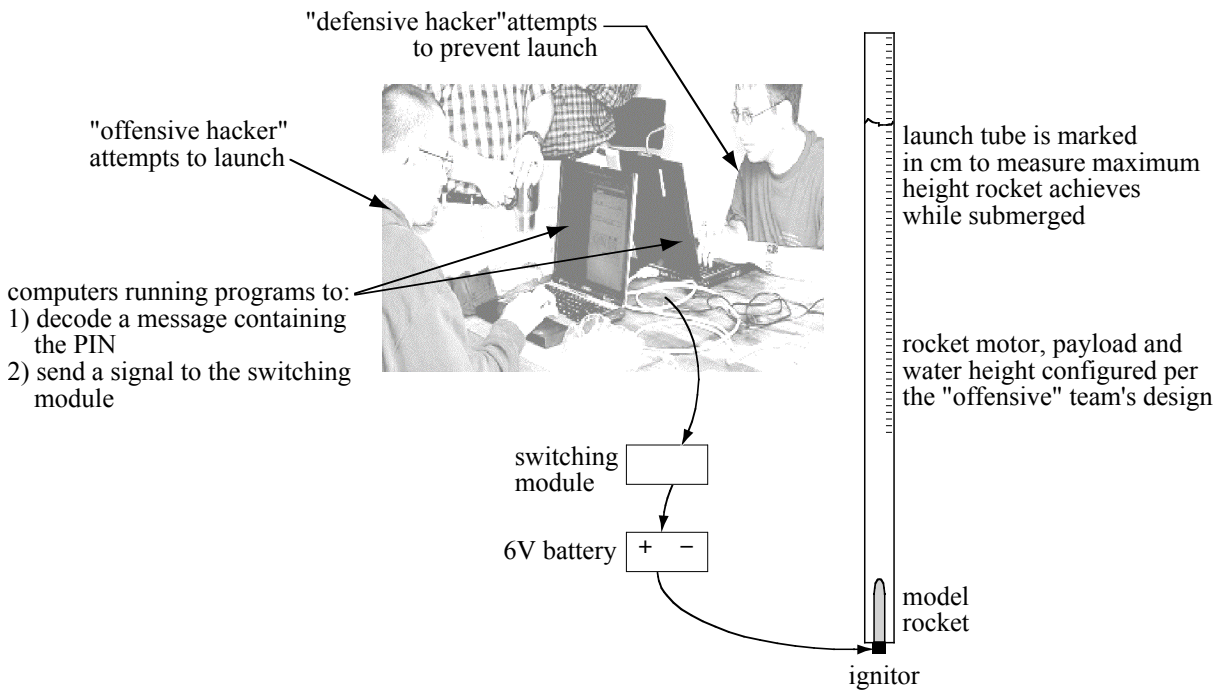


Fig. 2: Competition schematic.

Assigning students to teams. Students completed an online survey prior to the conference. The survey requests demographic information and a self-assessment of skill-levels in various skills required for the competition. The survey questions are:

1. Which one of the MUPEC member schools do you attend?
2. Your major discipline.
3. Rate your skill at setting up and numerically solving nonlinear ODEs to describe particle motion.
4. Rate your familiarity with drag forces relating to motion through a fluid.
5. Indicate your level of interest in attending a short workshop to learn computer-assisted methods to solve elementary problems in code-breaking.

At the beginning of the conference day, the thirteen student participants were assigned to teams of either three or four members using an automated team-assignment software package recently developed at Rose-Hulman [1]. The goal of the team assignments was to have students from

different disciplines and from different institutions working together and to distribute the required skills heterogeneously among the teams.

Itinerary. One of our goals for this conference was to create a design problem that would occupy a student team for most of the day. Thus we organized the day's activities to run concurrently with the design problem. Students left their design teams at designated times to attend a code-breaking workshop and to give their presentations or discuss their posters with the judges, while the rest of their team continued their work. The itinerary is summarized in Table 2.

Recall that the oral and poster presentations are not associated with the design problem; they showcase student work performed at their home institutions. Students were also invited to attend any presentation in which they had an interest.

Table 2: Conference Itinerary

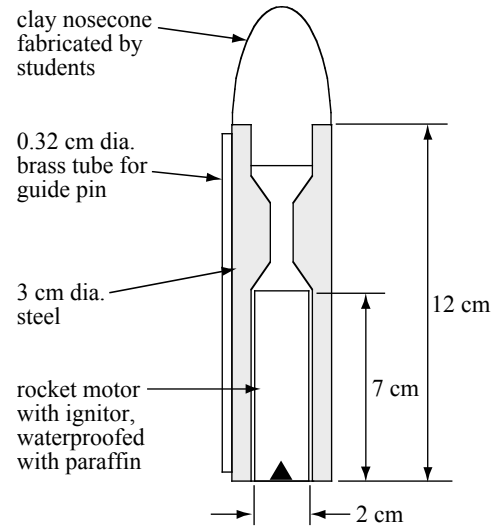
Time	Activity
8:00-9:00	Registration. Coffee, juice, muffins, and fruit. Set up posters. Last-minute team surveys.
9:00-9:50	Overview of the day's activities. PSE survey. Introduction to the design competition. Team assignments and introduce one another.
9:50-10:00	Break.
10:00-12:00	Teams brainstorm, develop a strategy and begin work on the design problem. Determine who on the team will be the code-breaker. At 10:50, code-breakers leave for workshop; others continue work.
11:00-12:00	Code-breaker workshop.
12:00-12:50	Lunch
1:00-3:30	Oral and poster presentations (concurrent with continuing design) at designated times.
	Design continues (concurrent with presentations), complete nose-cones. Code-breakers practice.
3:00-4:00	Nose-cones are due for oven-firing at 3:00, returned at 3:30. Snacks provided. Paint your nose-cones! All design and analysis documentation is finalized for judging.
4:00-5:00	Competition. <i>Underwater Hacker Missile Wars!</i>
5:00-5:30	PSE survey. Awards.

Equipment and Software

This section gives an overview of the hardware and software the authors developed for the competition. The total cost of materials for the rocket motors, launch tube supplies, and miscellaneous hardware was under \$400. Detailed plans for the apparatus, circuits, computer interfaces and computer programs and tutorials are available via the Internet [2].

Underwater rockets. The rocket bodies are fabricated from round steel stock. A typical cross-section is shown in Fig. 3. The steel is drilled-out to accept the rocket motor and the clay nosecone. The exact body dimensions differ for each of three motors to meet the minimum mass requirements listed in the design constraints.

The three rocket motors are types A8, B6 and C6, readily available at hobby stores. The three motors have a similar peak thrust of about 10 N, but differ in the total impulse delivered, as illustrated in Fig. 4. The figure shows a representative underwater thrust curve for each motor, obtained experimentally. These data are provided to the student teams in hardcopy and in Matlab and Excel format to use in their calculations and simulations. The area under each curve represents the total impulse the motor delivers to the rocket.



- Notes:
1. Dimensions shown are for motor type C6 and are approximate.
 2. Dimensions differ for motors A8 and B6.

Fig. 3: Cross-section of underwater rocket.

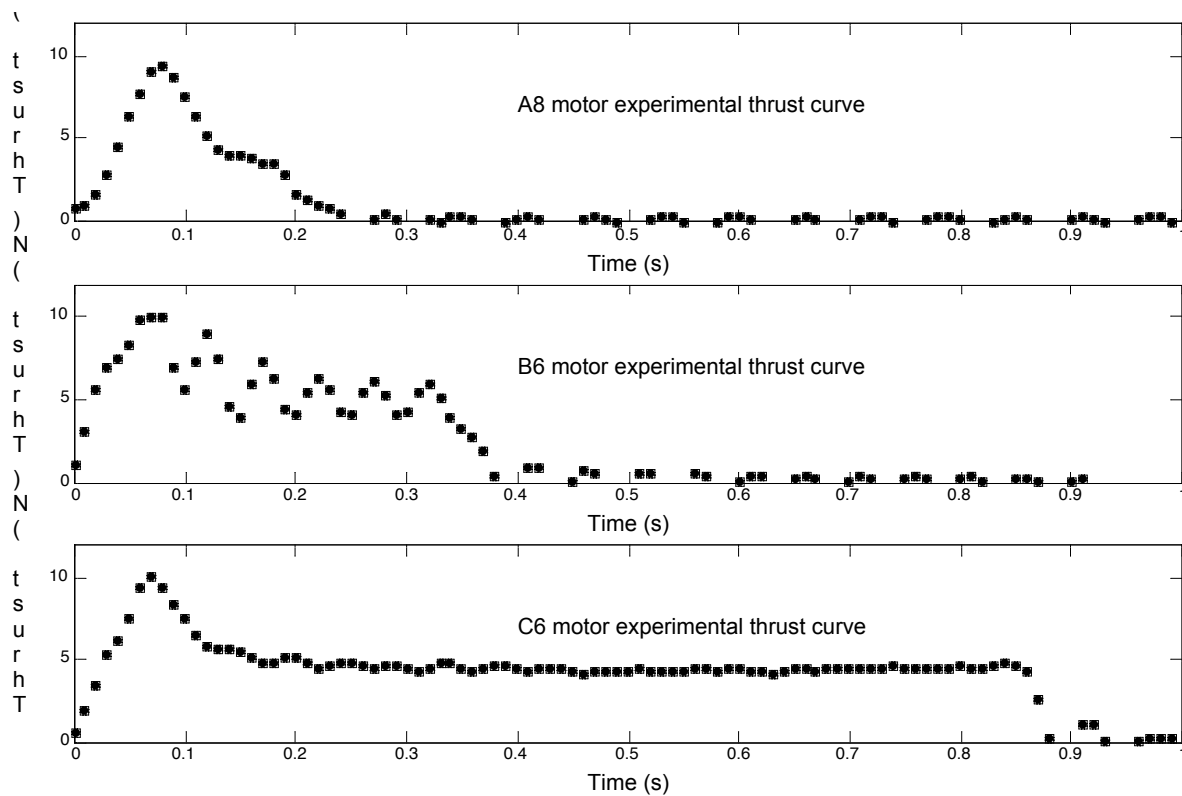


Fig. 4: Experimental thrust curves for three underwater rocket motors.

Testing apparatus. The mechanical test fixture is illustrated in Fig. 5. The polycarbonate tube sits atop a PVC base. The upper PVC tee allows access for installing the rocket on its launch guide pin. The rocket drops to the base of the guide pin located at the lower tee, which provides access for attaching the wires from the battery. A hose attachment at the base of the column is used to fill the entire column with water prior to each launch. The column is supported by a wooden frame.

The electrical switching module (not shown) consists of a board containing all of the necessary switches, electronics, and sensing devices necessary to detect which computer signals first that it has broken the code, launch or prevent the launch of the rocket, and tell the user the state of the electronics. When one of the computers breaks the code, a signal is generated and sent to the switching module. This signal controls a state machine programmed on a GAL Programming Logic Device.

The state machine has three states: (i) A *ready state*, which occurs when the testing apparatus is being set up or both contestants are trying to decipher the code. (ii) A *launch state*, which occurs when the “offensive hacker” successfully deciphers the code first. In this state, the GAL will trigger the movement of a relay switch which shorts the power supplies of a battery, launching the rocket. (iii) A *prevent-launch state*, which occurs when the “defensive hacker” successfully deciphers the code first. The state machine will remain in this state regardless of when the “offensive hacker” successfully deciphers the code, which gives the “offensive hacker” time to complete the code breaking so that the time can be recorded. The competition officials may then press an override button which will force the state machine into the *launch state*, causing the rocket to launch.

The board also contains a reset button that sends the state machine to the *ready state* from all other states. Additionally, the board contains three LED’s which tell the competition officials the state of the electronics. Further details, including Verilog code for the GAL, part numbers, resistor and capacitor values, and schematics may be obtained from the web site.

Cryptography tutorial and software. Student volunteers from each team, as well as several visiting faculty, attended a one-hour code-breaking workshop held in a computer-equipped classroom. The workshop introduces three basic types of substitution ciphers. For each type, the tutorial takes the students through a similar routine. First, the tutorial gives a brief

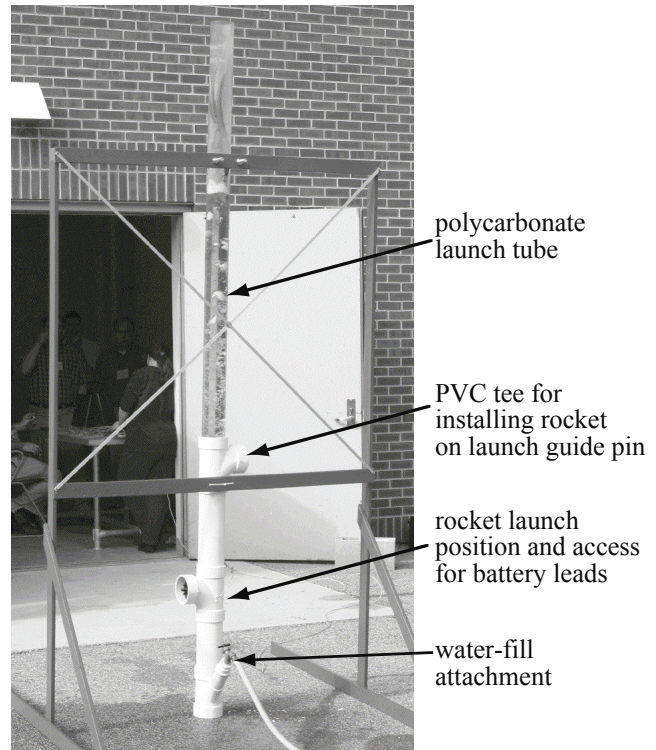


Fig. 5: Test fixture during competition.

explanation and an example. The students then encipher a given message by hand using a given key. To check their answer, students are shown how to decipher the message using the UWHMW (Under Water Hacker Missile Wars) software. After that, the tutorial discusses how to break the cipher using frequency distributions. Students are shown how to use the software to determine and test a probable key for the cipher. Finally, students practice breaking a set of sample ciphers programmed into the software.

The cryptography software used for the workshop was written by Scott Dial, a computer science major at Rose-Hulman, and is written in Java and distributed as a web-based applet. It has functions for each cipher which aid the user in the process of guessing and testing a probable key. The students are made aware that all of the messages used in the contest are recognizable (if not necessarily meaningful) English sentences when deciphered, so that it is immediately apparent if the key was correct.

In practicing using the software, students are told which messages are enciphered with which cipher. Students are also directed to pay special attention to the form of the decrypted sample messages, which are constructed in exactly the same manner as the plaintext of the messages used in the competition. Each message start with a four digit number (spelled out in words), which in the actual competition is the launch code used in the launch software. The rest of the message consists of several meaningless (but grammatical) sentences which are chosen at random by a computer program from a list.

Slides from the workshop and the UWHMW software, including sample enciphered messages, are available via the Internet at www.rose-hulman.edu/~holden/MUPEC/.

Hardware/software interface. Launch control software running on the same computers as the cryptography software allowed the students to signal when they had determined the “launch code” (i.e. the cipher key), causing the computer to signal the electrical switching module. The essential behavior of the software is described by the following pseudocode:

1. Read actual launch codes from file
2. Open the parallel port
3. While the contest is ongoing
 - a. Prompt for a scenario number
 - b. Execute the scenario:
 - i. Prompt for the launch code
 - ii. If the launch code is incorrect, go to step (i)
 - iii. Signal the electrical switching module through the parallel port
 - iv. Prompt for user acknowledgement of launch
 - v. Signal the electrical switching module through the parallel port
 - vi. Goto step 3
4. Close the parallel port

The software is written in C, and compiles under Microsoft Visual Studio 6.0. It is available via the Internet at www.rose-hulman.edu/~merkle/MUPEC.

The most challenging aspect of the hardware/software interface was overcoming the security features of Microsoft Windows XP to allow a program to access the parallel port. Briefly, when running in “protected mode” on an Intel 386 or later processor, any attempt to access an I/O port generates an exception. Earlier versions of Windows ignored these exceptions, but Windows XP does not, and the default behavior is to disallow the access attempt. A more detailed discussion of the security features and the PortTalk IO Driver that allows access to the port is available via the Internet at www.beyondlogic.org.

Solution of the Design Problem

We expect students to treat the rocket as a particle in rectilinear motion. The maximum kinetic energy, KE_1 , and therefore the figure of merit, are functions of the rocket’s maximum velocity v_{\max} . To estimate the maximum velocity, we apply the principle of impulse and momentum to obtain $v_{\max} = I/m$ where I is the impulse (the area under the thrust curve) and m is the total mass of the rocket body, motor and payload. If we assume that the rocket acquires all of this velocity at the instant it leaves its launch pad, then the figure of merit is given by

$$J = \frac{2g}{I^2} m^2 h, \quad (2)$$

where g is the acceleration due to gravity and h is the maximum height attained while submerged. (This assumption leads to underestimating the values of J , but does produce useful relative estimates for comparing the three motors.) A simulation is required to find h for a given combination of motor type and total mass. We apply Newton’s second law to obtain the following second-order nonlinear differential equation,

$$m\ddot{y} = F(t) + B - mg - \frac{1}{2}C_d A \rho \dot{y}^2, \quad (3)$$

where y is the vertical displacement of the rocket from its launch position, $F(t)$ represents the thrust data, B is the constant buoyancy force, mg is the rocket weight, and the final term represents drag. Solving this equation numerically for $y(t)$, we select the maximum value of y as the value for h for this particular combination of motor type and total mass.

Repeating this procedure for a range of values of mass for each of the three motor types produces the ideal predicted behavior of the figure of merit shown in Fig. 6. Each curve is truncated to the left due to the minimum total mass design constraint for each motor. The motor with the smallest impulse (A8) produces the most efficient launch while the motor with the largest impulse (C6) produces the least efficient launch. The maximum figure of merit ($J \approx 0.64$) is achieved with the A8 motor and with a rocket mass of 200g.

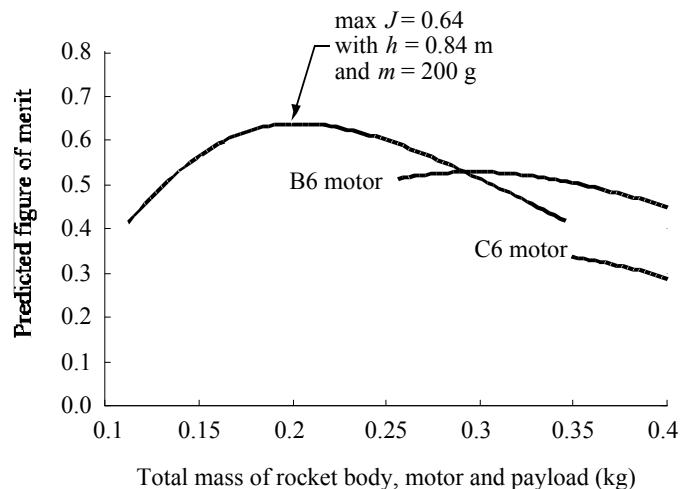


Fig. 6: Ideal predicted behavior of the figure of merit as mass increases.

Results of the Competition.

The teams obtained the experimental figures of merit and final overall scores shown in Table 3. Note that three teams obtained experimental values of the efficiency figure of merit greater than ideal maximum predicted by the model, Fig. 6. This is due to the modeling assumption that the rocket acquires all of this velocity at the instant it leaves its launch pad. Interestingly, three of the four teams correctly selected the A8 motor. Team 3, even though they had a high figure of merit, had a low overall score because of their poor code-breaking times. Team 4 was the only team that did not produce a well-thought-out analysis of the problem. Their approach was more guesswork than analysis and simulation.

Table 3: Team performance.

Team	Motor	J	Overall score
1	A8	0.89	84/100
2	A8	0.69	72/100
3	A8	0.89	71/100
4	C6	0.23	31/100

Results of the Personal Self-Evaluations (PSEs).

Personal self-evaluation forms were completed by the participants at the beginning and at the end of the conference day. For each of ten questions (listed in Table 4), students circled one of the following five possible answers:

- I could do it easily on my own.
- I could do it with difficulty on my own.
- I could do it easily with help.
- Even with help, I would have difficulty.
- I could not do it, even with help.

Table 4: Average results of the personal self-evaluations

Statement. All statements begin, "To what extent could you ..."	Before-to-after percent change
Predict the drag force on a body moving in a fluid environment?	+16
Apply Newton's second law to a small mass when the forces are not constant?	+10
Figure out an intercepted message protected by a simple code or cipher?	+8
Predict the motion of a mass subjected to known forces?	+7
Use matrix manipulation to make a secret message?	+6
Make a secret message using a simple code or cipher?	-2
Cooperate and collaborate effectively with other undergraduate students-whom you've just met-from other disciplines-from other colleges-in an open-ended design problem?	-3
Determine the kinetic energy and potential energy, at an instant, of a small mass in motion?	-4
Set up an approach to a design problem such that you could weigh the tradeoffs among alternative solution?	-6
Use computer software to set up and solve a given ordinary differential equation?	-7

The students' assessment of some of their abilities was improved by the conference experience; for some skills, however, the conference experience caused students to re-evaluate (downward) their previous self-assessment. Table 4 shows the PSE statements and the average percent change in student self assessment, where a positive number indicates the students' self-

assessment improved and a negative number indicates the self-assessment declined. The statements are ranked in order of most improved to worst decline.

Our evaluation of these results is that students on average were not very confident of their ability to apply Newton's second law or deal with drag forces in a rectilinear particle motion problem at the beginning of the day, but gained confidence after collaborating with their teammates as the day wore on. At the other end of the scale, students perhaps were overconfident in their abilities to approach an open-ended design problem, weigh tradeoffs among alternative solution, and to use computational software to solve a nonlinear differential equation as part of a larger design problem.

Conclusions

Our goal in designing the competition was to create a day-long design problem suitable for undergraduates in engineering, math and science disciplines. Success in this area is supported by noting that three of the four teams developed a successful, high-efficiency design, with team members generally from different disciplines and from different institutions. The PSE results also support this goal. The positive PSEs indicate areas in which the students gained confidence (possibly learning something new, as in the cryptography workshop). The negative PSEs indicate areas in which students thought they had some expertise even though they may have been initially overconfident. Both cases indicate that on average some level of learning has occurred for a mixed group of students from different disciplines and different institutions.

Our second goal, to describe the design problem, apparatus, software and tutorials for others who may be interested in replicating and improving the competition, has been met in general in this paper and in detail in the conference website [2].

Acknowledgments. Our thanks to so many without whom this design problem and competition simply would not have come together in time: our colleagues and coworkers Ray Bland, Patsy Brackin, Gary Burgess, Pat Carlson, Mike Fulk, and Mike McLeish, and our students Erin Bender, Scott Dial and Gerald Rea.

Bibliography

1. Cavanaugh, R., Ellis, M., Layton, R. and Ardis, M. (2004), Automating the process of assigning students to cooperative-learning teams, in proc. *2004 ASEE Annual Conf.*, Salt Lake City.
2. MUPEC 2004 conference website, www.rose-hulman.edu/MUPEC2004/

RICHARD A. LAYTON

Richard Layton received his Ph.D. from the University of Washington in 1995 and is currently an Assistant Professor of Mechanical Engineering at Rose-Hulman Institute of Technology. His interests include student team building and laboratory curriculum development. Prior to his academic career, Dr. Layton worked for twelve years in consulting engineering, culminating as a group head and a project manager.

JOSHUA HOLDEN

Joshua Holden is currently an Assistant Professor in the Mathematics Department of Rose-Hulman Institute of Technology. He received his Ph.D. from Brown University in 1998 and held postdoctoral positions at the University of Massachusetts at Amherst and Duke University. His research interests are in computational and algebraic number theory and in cryptography. His teaching interests include the use of technology in teaching and the teaching of mathematics to computer science majors, as well as the use of historically informed pedagogy.

TINA HUDSON

Tina Hudson received her Ph.D. from Georgia Institute of Technology in 2000 and is currently an Assistant Professor of Electrical Engineering at Rose-Hulman Institute of Technology. Her research interests include the development of real-time neuromuscular models using integrated circuits and MEMS devices, linear threshold circuits, and methods to intuitively teach analog and digital integrated circuit design and MEMS devices.

LAURENCE D. MERKLE

Larry Merkle received his Ph.D. from the Air Force Institute of Technology in 1996 and is currently an Assistant Professor of Computer Science and Software Engineering at Rose-Hulman Institute of Technology. Prior to joining Rose-Hulman, he served almost 15 years as an active duty officer in the United States Air Force. During that time he served as an artificial intelligence project management officer, as chief of the Plasma Theory and Computation Center, and on the faculty of the United States Air Force Academy. His interests include computer science education and the application of advanced evolutionary computation techniques to computational science and engineering problems.