Simulation Using Spreadsheets in Engineering Technology Curricula – Satisfying Multiple Learning Objectives

Zbigniew Prusak
Central Connecticut State University

Abstract

This paper describes teaching of basic simulation principles as well as underlying fundamental knowledge about a problem which is necessary for building simulation models, making simulation runs, and analyzing results. These aspects of teaching simulation are described from the perspective of using spreadsheet software – a simulation tool available at almost no-cost in all academic institution. Examples of types of exercises and projects for solving problems in fields of production control and planning of manufacturing operations, and Geometric Dimensioning and Tolerancing. Project requirements for students to fulfill learning objectives of forming important professional abilities of engineers and technologists. Learning advantages of working with spreadsheets and influence on teaching environment and difficulties encountered on different stages of simulation exercises are described.

1. Introduction

In science and engineering, the relationship between previously acquired knowledge, reasoning ability and structured problem solving techniques has been investigated extensively [1, 2, 3, 4]. Engineering thinking is not only about solving a technical problem, but also about explaining why a particular solution to a problem is the best [5]. During formation of engineers and technologists, the meaning of a studied subject, its applicability to real world situations, and creation of future situations that may employ this subject, all need to be understood by students using three basic avenues:

- conveying the knowledge to the student by the teacher
- student's own learning and reasoning built on previous experiences
- student's creativity fostered by instructor and appropriate activities

In the field of applied engineering in particular, the problem solving ability based on achievement of stated objectives, often regardless of tools and means used, needs to be based on an underlying knowledge called technical rationality [1]. Technical rationality is the traditional base of engineering knowledge and skills that is thought to remain a corner stone of all technology-related professions. In recent years, due to the general availability of high-tech learning aids, methods of teaching this engineering knowledge base have undergone substantial changes. However, conveying the knowledge to the student by the teacher will most likely

remain the primary mean of developing the technical rationality in future engineers and technologists.

The widespread availability of engineering and business software has created a tremendous opportunity for improving learning efficiency for the second and third item of the above listed learning avenues. This opportunity has created another positive learning effect by adding to the traditional instructor's role as knowledge deliverer, a role of learning mentor (an inciter and monitor of student development). It is generally accepted (as well as widely contested) that a lot of education constricts creativity. That is partially due to the fact that people having a lot of education always tend to refer to the body of knowledge possessed and ways of problem solving as they were taught. It is therefore desirable that software used for teaching engineers be very flexible and easily adaptable to a wide variety of tasks. The variety of tasks includes also applications for tasks that were not foreseen by software developers. They can be created and refined by software users based on software capabilities (calculation software, simulation software, FEA software, etc.).

2. Learning objectives and use of software

2.1. Professional abilities

Learning objectives of higher education in general [6, 7, 8, 1, 5], can be summarized as a set of important professional abilities which engineering and technology students should develop while attending university:

1. perform procedures and demonstrate understanding of techniques
2. develop plans and establish procedures
3. access and manage information
4. think critically
5. solve problems
6. verify solutions
7. conceptualize, design and create
8. communicate

With the exception of abilities number 4, 5 and 7, software can be a learning aid and tool for all the other abilities from the above list. Use of software as a tool not as a final outcome, can broaden and enhance the learning process or slow it down and narrow in scope. A narrow specialization inevitably brings about a risk of not thinking in terms of a whole system, and a difficulty in communication with specialists from other disciplines. It may lead towards disregarding a possibility of different or inventive solutions.

2.2. State-of-the-art tools or basic principles

There are strong voices in academic community calling to provide solid interdisciplinary knowledge to all graduates of university technical programs [8, 9, 10]. Such an approach to teaching is not fully compatible with immediate needs of employers, but benefits long term goals of national economy as a whole [6, 7]. Therefore, a narrowly focused technical training (presently highly popular in teaching software tools) is not a primary domain of academic education. It is

more important from university education point of view to emphasize inventiveness and good algorithms to solve problems rather than perfecting how to solve a problem using a particular engineering tool (eg. software). For the same reason that understanding is emphasized over remembering in teaching physics and mathematics, '*know-how*' must be emphasized over '*know-a-tool*' in teaching engineers and technologists. Spreadsheets are probably the best example of software that fits very well into category of versatile and widely used tools. They are flexible in building customized solutions and are accepted by almost all professions within business community. Nonetheless, due to ever progressing complexity of engineering tools, elements of specialized training also need to be included, especially in curricula of engineering technology programs, in order to provide graduates with ready to use skills [6, 11, 12].

3. Spreadsheet simulation problems

3.2 Goals

Although performing simulation is one of the secondary capabilities of spreadsheets, they can be successfully used in a wide variety of applications. This flexibility and a relatively short learning time, are viewed as the key points in this application. As compared to canned software, spreadsheets are characterized by the lack of just few right ways of solving a problem. That lack of rigidity is a great asset from an instructional point of view. The emphasis can be placed up front on the essence of a problem, approaches and solutions. Each subject of a simulation students are required to design a spreadsheet for is previously covered in a traditional way, using theory and equations. As explained later, for some projects students use canned software before they move to building their spreadsheet solutions. Various simulation exercises as well as individual and group projects were developed and used in Engineering Technology courses at Central Connecticut State University. The underlying goal was to teach students the following concepts:

- basic probability concepts
- random numbers
- randomness of input variables
- probability distribution types governing changes of input variables
- output sensitivity to variability range of input variables
- Monte Carlo simulation
- variability of results for a given length of simulation run
- necessary length of simulation run
- repetitions of simulation runs

The last three concepts are taken one step further to perform analysis of simulation results and to decide how many runs and/or repetitions are 'good enough'. This is done in two subsequent steps

- Without giving a precise guidance to the students about criteria of what is considered a 'good enough' solution (convergence results must be presented graphically)
- After submitting results of the first step, students perform simplistic risk analysis of their solutions. They are given a cost function for the deviation from what is considered a 'good solution', and a cost function for length of simulation runs. Results must be also presented graphically.

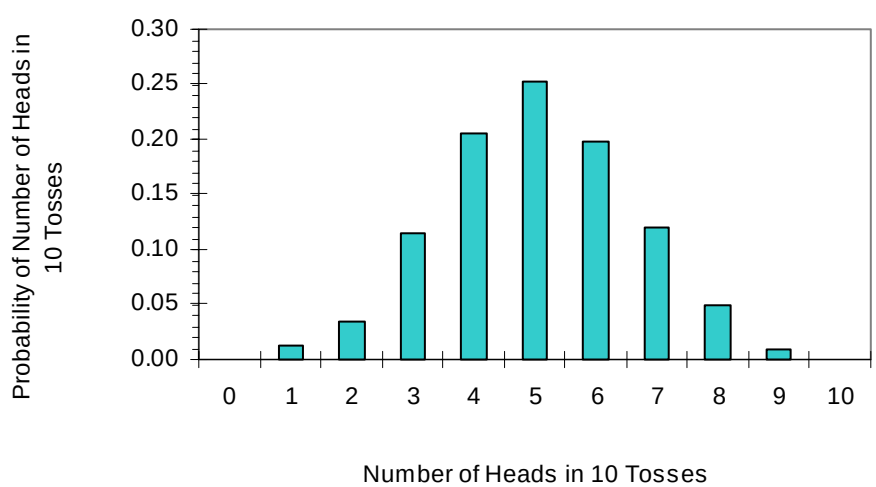
3.2 Courses and project examples

The following are the examples of simulation projects conducted during the past three years in Manufacturing, Mechanical and Industrial Systems specializations at the School of Technology at CCSU.

3.2.1 Computer-Aided Planning (junior level ET course)

- Simulation of tossing one coin and tossing multiple coins. Use of simple formulas available in math textbooks. Use of random number generator. This first simulation exercise is done with an aid of previously prepared spreadsheet. Known answers, hence ease of explaining length of simulation (example in Figure 1).
- Simulation of a queuing problem. Arrivals are assumed to be random, not Poisson-distributed. Problem is based on approaches from [14, 15]. Sometimes students can also bring similar problems from their workplaces, which are then solved in groups.
- Simulation of inventory. Basic deterministic inventory models assume constant demand and reorder lead time. Simulation project uses the demand and the reorder lead time as probabilistic variables. Solving inventory simulation problems is also based on the approach from textbook [14]. Firstly, students solve inventory simulation problem using a dedicated POM software, and plot their findings about number of runs using a spreadsheet that they design on their own. Secondly, students compare results from their simulation spreadsheets to the results from POM software and plot the findings.
- Preventive maintenance planning problem. Similar approach to the above described simulation of inventory.
- Tool replacement policy. Also similar in approach to the above described simulation of inventory.

Figure 1. Example of a result from a simulation exercise of tossing one coin and tossing multiple coins. Excel spreadsheet was used for simulation and plotting.



3.2.2 Geometric Dimensioning and Tolerancing (junior level ET course)

- Simulation of a coordinate tolerance stackup problem. Initially, students define the problem on paper through drawings and equations, and solve for worst case scenarios using upper and lower deviations of each dimension. This initial phase helps later in validating the limits of their spreadsheet solutions. The second phase consists of building a spreadsheet that includes randomness (uniform distribution) of dimensions within their tolerance zones. A little more realistic triangular distribution for the dimensions is also given as the next stage. The triangular distribution used is a simplified one, dividing tolerance zone into ten regions. The third stage of the second phase consists of using a different distribution for each dimension (for modeling purposes SQC data from a production environment is used). The third stage is yet to be completed successfully, mainly due to time constraints of the semester.

4. Learning process outcomes and difficulties encountered

Observations from the above mentioned spreadsheet simulation projects.

Learning advantages of working with spreadsheets:

- need to conceptualize, develop plan of work, look for information, create and verify
- additional practice in use of equations for solving problems with no clearly defined steps to be followed during spreadsheet building stage
- substantial reduction of computation intensive part of projects
- ease of fixing conceptual errors and gross errors of inputs
- ease of verifying what-if scenarios and sensitivity to inputs
- portability of spreadsheet results (very well suited for group work from development to refining stages)
- better understanding of probability concepts through playing with inputs and evaluating results
- development of a feel for variability and uncertainty of results
- environment fostering student learning from each other
- formal planning of activities has a real meaning to the students while working in a group

Teaching environment positives observed:

- short learning curve for spreadsheets as compared to dedicated software
- environment fostering student learning from each other
- formal planning of activities has a real meaning to the students while working in a group
- enhanced relevance of course subject to real life situations
- better attitude towards efficient spending of ones time in classrooms and labs
- challenge to the students (exceptional work can be graded above allotted maximum grade)
- personal pride and peer recognition from creating a one-of-a-kind solution or a “cool” presentation

Teaching difficulties encountered:

- different levels of spreadsheet proficiency among students (biggest impediment to this learning process)
- weak grasp of probability concepts among students
- reluctance to learn theory – students prefer practical, ready to use solutions
- time investment in planning and developing the spreadsheet solution
- discouragement often seen after few hours of work with no visible results (students usually expect instant gratification)
- students show more persistence and dexterity in tedious learning of a procedure to be followed (eg. canned software operation), than in conceptualizing and building on their own (eg. solving problems using spreadsheets)
- when compared to a dedicated software, spreadsheet solution results are presented in not as nice a form which often creates a negative perception about validity of own work.

5. Conclusions

The use of spreadsheets for simulation exercises and projects is a challenging but very valuable application of this computer tool, which is widely available and at a minimal cost. Using this versatile tool allows for maintaining the focus on solving the problem at hand and not loosing touch with the underlying theory. Students developed better feel for variability and uncertainty of results. Computation intensive parts of the projects were substantially reduced. Even employing the "let's try and see what happens" method, conceptualization, planning and design stages had to be done first. Group projects as well as individual projects fostered student learning from each other. The customized spreadsheet use resembles most work environments, where due to lack of resources for specialized software, problems must be solved using a popular and readily available tool. From the eight professional abilities necessary for an engineer and technologist, all have a possibility of development during these projects. Conceptualization, designing, thinking critically and verifying solutions are the abilities that benefit the most. The drawback of long learning process necessary for a sophisticated software was largely avoided. More time could be spent on ways of solving a technical problem and on its computational nature, treating computers as tools not the end result. Students experienced a need for formal planning of project activities, especially when working in groups.

References

1. Schon, D.: "Reflective Practitioner: How Professionals Think in Action", Basic Books, 1983.
2. Baird, W.E., Borich, G.D.: "Validity Considerations for Research on Integrated-Science Process Skills and Formal Reasoning Ability", *Science Education*, vol.71, (1987), pp.259-269.
3. Bransford, J.D., Stein, B.S.: "The Ideal Problem Solver", W.H. Freeman and Co.,New York, NY, 1984.
4. Brown, S., Walter, M.: "The Art of Problem Solving", Erlbaum, Hillsdale, NJ, 1983.
5. Robinson, J.A.: "Engineering Thinking and Rhetoric", *Journal of Engineering Education*, vol.87, no.3, (1998), pp.227-229.
6. Prusak, Z.: "Challenges to Future Engineering Professionals – How to Prepare Students to Face Them", *Proceedings of ASEE Annual Conference*, June 28- July 1, 1998, Seattle, WA.

7. Turski, W.M.: "Uczyć, nie uczyć?", *PC Kurier*, 1997, no.14, pp.133.
8. Haeck, L.V.J.: "Multi-disciplinary ou le Genie Plus", *Proceedings of ASEE Zone 1 Meeting*, Apr.25-26, 1997, West Point, NY, pp.9A3.1-9A3.9.
9. Rocoeur, P.: "L'Université Nouvelle, L'Education dans un Québec en évolution." Les Presses de l'Université Laval, 1966.
10. Jaspers, K.: "Die Idee der Universitat." 1954. Ed. Peter Owen, London, 1959.
11. Lopardo, V.J., Wu, C.: "Engineering Educations: Future Directions", *Proceedings of ASEE Annual Conference*, June 26-29, 1994, Edmonton, Canada, pp.2939-2945.
12. Marchio, D., Adnot, J., Arditi, I.: "Synergy Between On-the-job Training and Academic Education in the Case of Newly Created Part-time Engineering Curriculum", *European Journal of Engineering Education*, vol. 22, no. 4, 1997, pp.411-419.
13. Fisher, K.J.: "Spreadsheet Solutions to Applied Heat Transfer Problems", *Journal of Engineering Technology*, Spring 1995, pp.30-33.
14. Heizer, J., Render, B.: "Production and Operations Management", Prentice-Hall, Upper Saddle River, NJ, 1996.
15. Cornwall, L.W., Modianos, D.T.: "Simulation using Spreadsheets: Modeling Techniques", *Proceedings of 19th Annual Modeling and Simulation Conference*, May 5-6, 1988, Pittsburgh, PA, pp.801-805.

ZBIGNIEW PRUSAK is an Assistant Professor in the Engineering Technology Department at Central Connecticut State University. He teaches courses in Mechanical, Manufacturing and Industrial Systems programs. He has over 10 years of international industrial and research experience in the fields of precision manufacturing, mechanical systems and metrology. Dr. Prusak received M.S. Mechanical Engineering from Technical University of Krakow and his Ph.D. in Mechanical Engineering from University of Connecticut. E-mail: prusakz@ccsu.edu