On the Use of TRNSYS in a Solar Energy Technical Elective

Ian A. Gravagne
Dept. of Electrical and Computer Engineering
Baylor University, Waco, TX

Kenneth Van Treuren
Dept. of Mechanical Engineering
Baylor University, Waco, TX

Abstract

As concerns about energy infrastructure, sustainability and security become ever more important to Americans, engineering schools that have not traditionally focused on energy are moving to address the topic more formally. At Baylor University, an "energy core" of technical electives is being developed, including courses on wind energy, solar energy, power systems, turbines and combustion engines. This paper documents the authors' observations on the use of the TRNSYS simulation software package in a senior/graduate elective on solar energy. The paper gives examples of the types of projects students do using TRNSYS, how it can be used in the classroom, and some suggestions for educators considering its use in future courses. Comparisons are made between projects completed using general-purpose numerical simulation software such as MATLAB, to show how TRNSYS can help students investigate more complex systems in less time, if attention is paid to comprehension of first principles.

Background

In 2001, an ASEE paper by D. Yogi Goswami[5] reported that only about 10 U.S. universities regularly offered courses in solar energy, a dramatic drop from 150 in 1980. The 1980 number was attributed to the aftershocks of the 1970s oil crises, and the ensuing decline to the forces of energy economics throughout the next 20 years, which made solar technologies financially uncompetitive regardless of their environmental utility. Absent a more recent survey, it is not known how many universities now offer courses in solar energy today but the number is likely higher than 10 as the U.S. gradually responds to political instabilities, surging energy prices and a dawning public awareness of the environmental impact of unrestrained energy usage. In the fields of engineering, there is also broad agreement that job prospects for energy-literate engineers are up, and anecdotal evidence in states with growing populations (e.g. south and western states) suggests that employment opportunities in energy and power will remain robust in the foreseeable future.

With these factors in mind, the authors began in 2006 to organize several efforts to address energy education at Baylor both amongst upper-level engineering students and various non-engineering populations. These include the construction of an alternative energy and energy efficiency exhibit in the university's Mayborn Museum (nearly completed at the time of this writing) targeted at high school-educated adults; the initiation of a novel two-year program for freshmen of any major to study and research various aspects of energy while earning social or laboratory science credit; and the instantiation of technical elective courses in solar energy, wind energy, power systems and others soon to come. In all of the aforementioned educational initiatives, it has been the objective of the educators to keep the courses accessible to engineering majors of any stripe (e.g. electrical, mechanical, biomedical, general, etc.) possessing the appropriate prerequisites. Energy is, at its best, a naturally interdisciplinary subject, though there are pedagogical challenges inherent with teaching to such a broad audience. Another challenge is the introduction of appropriate computational tools in the courses, which is the subject of this paper.

TRNSYS in Solar Energy

TRNSYS ("Transient System Simulator," pronounced "tran-sis") was originally developed at the University of Wisconsin in the 1970s for numerical analysis of solar hot water heating systems. The UW Solar Energy Laboratory continues its development, and has authorized Thermal Energy System Specialists Inc. of Madison, WI, to sell licenses and provide support and training in the U.S. Throughout the past 30 years, TRNSYS has been steadily refined and is now widely recognized as a reliable analysis tool for complex energy systems. There are many papers comparing its predictions to field measurements or various empirical techniques[2,3,4,6,8,9,10,12,13,15]. It is beyond the scope of this paper to provide a detailed comparison of TRNSYS to other possible energy analysis and simulation tools, but TRNSYS was selected in part because it was developed specifically with solar (and other renewable) energy in mind from its inception.

Two commercial uses of TRNSYS bear mention, and supported our decision to employ it in the newly developed solar energy course. The first is its successful application to integrated building designs for LEED (Leadership in Energy and Environmental Design) certification. Most LEED certifications for new construction involve demonstrating that the proposed design will be more energy efficient than a typical design. Large multi-zone buildings are certainly not amenable to hand analysis, and examples are not hard to find where TRNSYS was successfully used to compare the proposed design to a hypothetical alternative design. The second noteworthy mention of TRNSYS is its use by the Solar Rating Certification Corporation (SRCC), the entity currently designated by the U.S. government to certify solar water heating systems for federal tax credit eligibility. Independent laboratories convey the results of ASHRAE-standardized tests on solar thermal collectors to the SRCC, which then uses TRNSYS to estimate overall system performance in a variety of climates[14,16]. Dozens of manufacturers have submitted their systems for OG-100 and OG-300 certification.

TRNSYS in the Classroom

Interestingly, despite its long history and utility, TRNSYS has seen little apparent use in formal instructional settings (two examples are in the references[7,11], though it is not known how extensively these courses use the software). It is difficult to know exactly why, but the reasons probably include the fact that solar energy is often taught as one small part of a heat transfer or solid-state electronics class in which insufficient time exists to develop expertise with specialized tools; that prior to version 16, the software was somewhat more difficult to learn and use; and that specialized software may not see enough usage across the curriculum to justify its expense. Nevertheless, after some investigation the authors elected to use it in the newly designed "Solar Engineering" course open to senior and graduate engineering students. This course was offered for the first time in fall 2007, using the text "Solar Engineering of Thermal Processes," authored by two prominent researchers in the UW Solar Energy Laboratory, John Duffie and William Beckman.

Also in support of its adoption at Baylor were the many and significant improvements appearing in version 16[1]. Among these, several stand out:

1. **User interface**. TRNSYS is a so-called procedural language. As such, most of its functionality comes from modular component models called "types," and TRNSYS "code" is simply a syntax for telling the computational engine how the inputs and outputs of various types are interconnected. Nevertheless, things can get quite complicated, and the version 16 Simulation Studio is an improvement that allows programmers to visually connect components, insert equations and specify output formatting and simulation parameters (similarly to National Instruments' LabView). The Simulation Studio undoubtedly reduces the time required to learn the software.
2. **External Program Calls and New Components**. It is now possible to call routines in MATLAB, Engineering Equation Solver or Excel. Furthermore, users can now write new components in C++ or Fortran without making any modifications to the TRNSYS kernel because all components are compiled as dynamic link libraries.
3. **Improved Building Modeling**. In Bradley and Kummert[1], several improvements to the Type 56 multi-zone building model are described. Building data is now compiled using the TRNBuild tool, a separate graphical interface for inputting the many parameters needed to simulate building environments. Buildings are not the main focus of the Solar Engineering course, but can be the subject of undergraduate independent study projects and graduate research projects. (One such project is currently underway.)

In the final analysis, software is only useful as a teaching tool if it supports and complements the students' understanding of the principles being taught. By distilling complex mathematical interactions down to icons and wires in a window, software of the complexity and flexibility of TRNSYS can easily obscure these principles. For this reason, TRNSYS was not introduced until midway through the class. After its careful introduction, however, it served its purpose by providing both in-class lecture support and a platform on which to assign homework and final projects that would have been too time-consuming to do by other means. The next section outlines examples of these uses for TRNSYS.

Lecture Support

One area where computational support can make a real difference is lecture preparation. After studying the mathematical models of various energy components (such as collectors, tanks, inverters, controllers, etc.), the next logical step is to study examples of systems of components. However, this quickly gets unwieldy from an instructional point of view. For example, a simple flat plate thermal collector with forced circulation, a counter-flow heat exchanger and a stratified storage tank represents a coupled set of nonlinear equations to solve. Doing hand examples on the board is time consuming, but yet there is value in illustrating how the system responds to changes in parameters such as pump flow rate, collector efficiency, terrestrial location and orientation, and tank size. With less than one hour of preparation, this simple system was designed in TRNSYS and the simulation was executed in class. In real time, the instructor (first author) was able to query students about possible outcomes, and then make changes to the simulation and re-run it. This was immediately satisfying for the students, but arguably required far less time to prepare than the same example programmed in a general-purpose language such as MATLAB, and certainly less time than hand-worked scenarios. Figure 1 shows the in-class thermal collector simulation, as seen in the Simulation Studio.

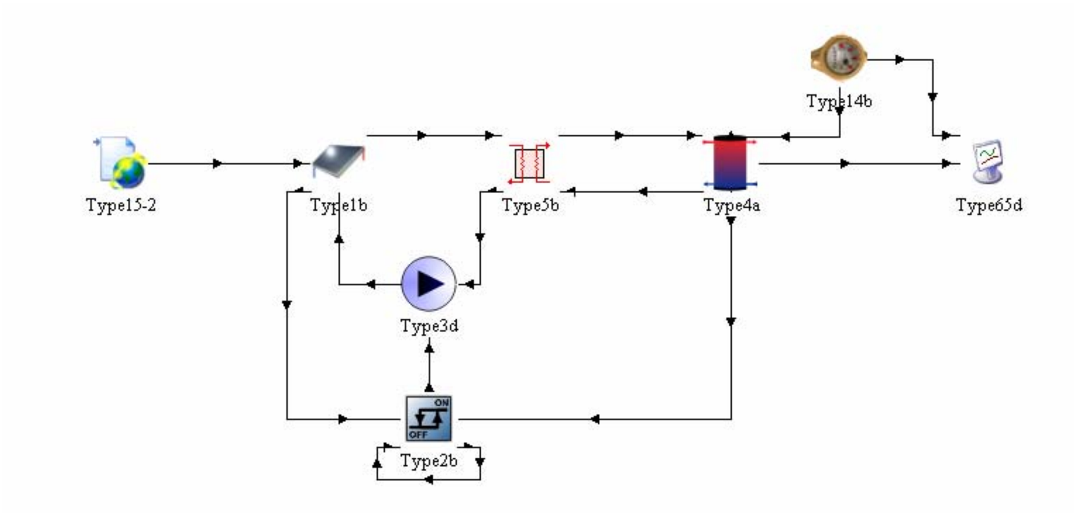

Figure 1. A double-pumped flat plate water heating system, as seen in the TRNSYS Simulation Studio. On the left, Type 15 generates weather data, and on the right , type 65 displays output.

Reinforcing Lessons from Homework

A great deal of time in any solar energy course is spent understanding the characteristics of sun-earth geometry, solar radiation models, and collectors. These topics consumed almost 50% of the class time in Solar Engineering, and the introduction of TRNSYS was purposefully delayed in order to ensure their comprehension. However, this delay permitted the students to reexamine in simulation several early homework problems that had been difficult to complete by hand.

The first problem asks students to calculate the energy gain on January 1 and June 1 of a typical meteorological year in Waco, TX, with a SunEarth Empire EP-32 collector at 35 degrees pitch and 10 degrees azimuth, under continuous forced glycol circulation with inlet temperature 20 degrees C. There are many approaches to this problem, and students face several challenges as they work it:

- Understand how to locate and read typical meteorological year 2 (TMY2) data.
- Correct for the difference between TMY2 timestamps and local solar time.
- Choose a diffuse radiation model and assume a reasonable value for ground reflectance.
- Correct the SRCC collector efficiency data for angle of incidence, flow rate and fluid heat capacity.
- Assume some mathematical behavior for the incidence angle modifier beyond its specified limit of 60 degrees.
- Remember to integrate total gain throughout the night, not just when the sun is up!

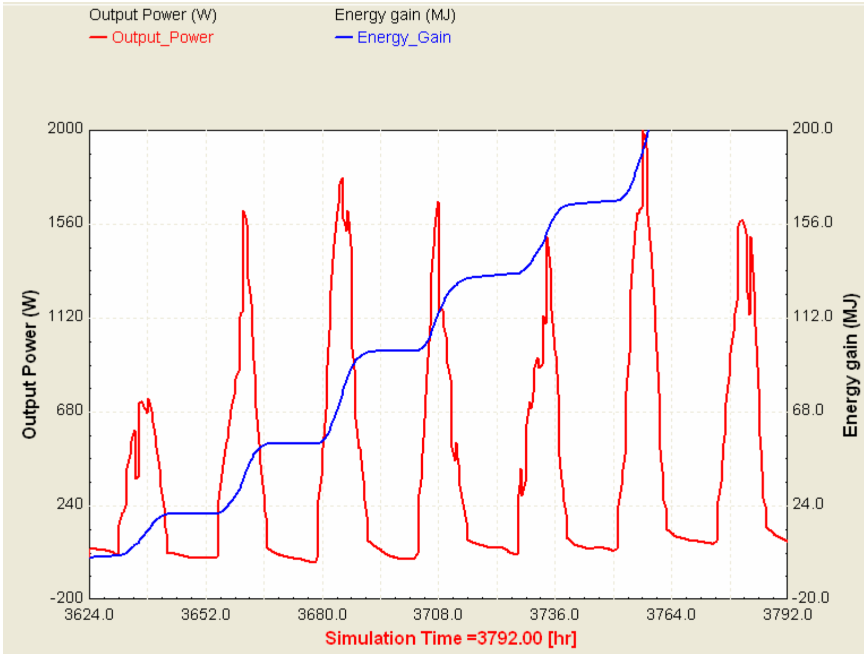

Figure 2. The output window, showing daily peaks in thermal power output and accumulated total energy gain for the week of June 1 of a typical meteorological year..

This problem produced a wide variety of solutions and prompted much discussion about the choices and assumptions that had to be made to solve it. However, underscoring the value and utility of good simulation software, the problem was reintroduced later for analysis in TRNSYS and solved in a matter of minutes. Figure 2 shows an example of the thermal power output and integrated thermal gain. .

The second problem asks students to calculate families of I-V (current vs. voltage) curves for a BP 365 and a UniSolar ES-62T photovoltaic module. Families of curves were to be generated for variations in ambient temperature and incident radiation. There are two principal challenges in this assignment:

- The I-V relationship for semiconductor-based PV cells is a transcendental equation. This sometimes makes it challenging to manipulate.
- The parameters in the model, such as the ideality factor, the effective series and shunt resistances, and the diode saturation current are not given by manufacturers, but must be computed from reference data.

In subsequent assignments, the challenge escalates because PV modules may be direct-coupled to nonlinear electrical loads, and are driven by input irradiance which itself comes from complex equations involving geometry and radiation models. The problem was first assigned and solved using MATLAB or Excel, and students expended considerable time and effort mastering the basic mathematical and physical principles. After the introduction of TRNSYS, photovoltaic simulation became much simpler and far more complex models could be tackled. The same problem – generation of I-V curves – was assigned again after mid term. While TRNSYS does not directly generate parametric curves, students understood how to adjust the inputs and parameters to Type 94 (PV array) to obtain data for curves.

Above all, however, students had a deeper understanding of what the software was doing because they had completed difficult problems without specialized simulation software first. Students were able to spot suspicious output and effectively debug their projects, because computing solutions the "hard way" helped them to develop intuition. In short, the software was at that point an enabler, empowering them to analyze systems too difficult for hand or general-purpose calculation, while not robbing them of deeper understanding.

Design Projects

Perhaps where good simulation software really shines is in the area of design. To this end, the final projects in Solar Energy were all design oriented and required the use of TRNSYS to complete. Three weeks were given to complete the projects and deliver written reports; graduate students were additionally required to present a talk about their designs. A summary of the projects follows.

- Design a theoretical flat-plate thermal collector for a hot water system located in Africa, and simulate its performance for one year. Challenges here include finding and using weather data for this location, estimating load demand based on a general description of the user population, and calculating collector efficiency parameters on the basis of first principles for radiative and convective heat transfer.
- Design a solar-assisted radiant-floor heating system for a small building in Waco that can provide a monthly average daily 150,000 BTU during nigh-time hours in January and February. Challenges include designing a system than provide an 80% solar energy fraction (SEF), selecting an effective heat exchanger, and optimizing tank capacity and collector surface area for minimum cost.
- Design a large-scale PV generating station with a sodium-sulfur battery bank for load leveling that can sustain a 50kW output power for 80% of the year. Challenges include arriving at a basic understanding of NaS battery technology, designing a system to protect the batteries from over-discharge during long periods without solar radiation, and minimizing the size of the PV array.

Figures 3 and 4 illustrate two of the student designs.

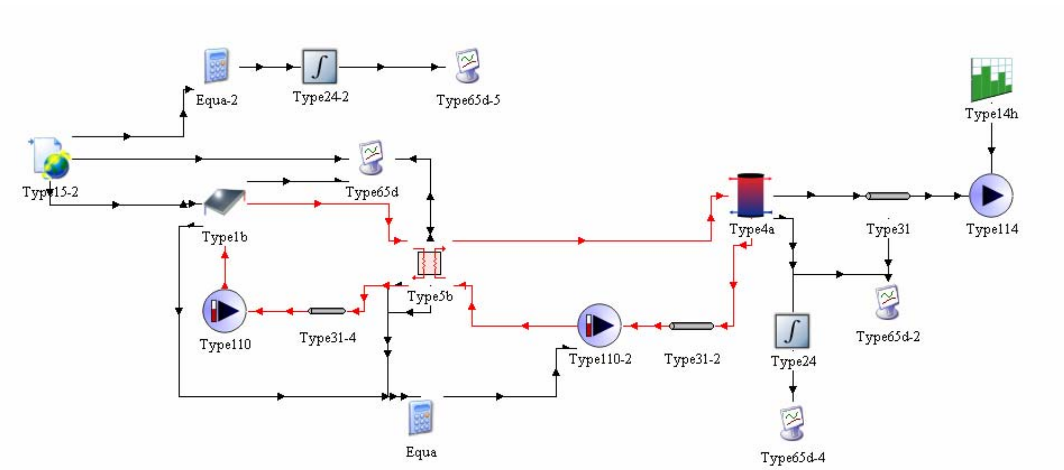

Figure 3. One student's design for a solar-assisted radiant floor heating system. At right, note the student's (incorrect) attempt to draw 150,000 BTU from the tank using a pump switched by a forcing function.

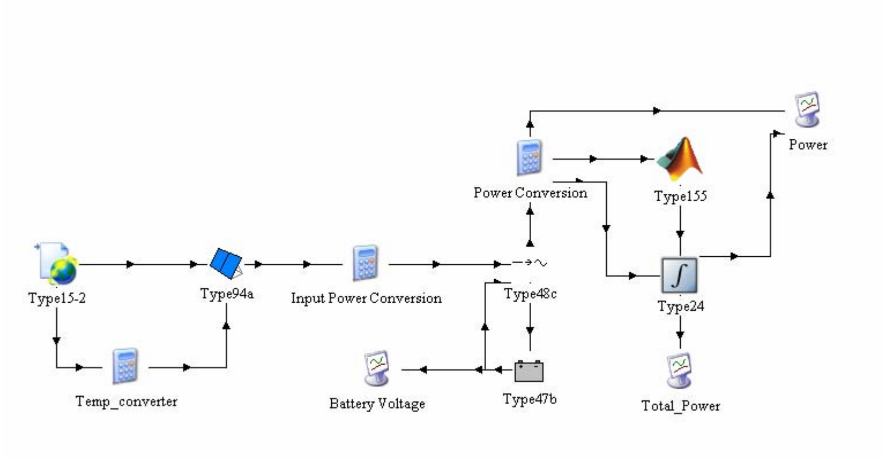

Figure 4. A design for a 50kW solar power generator. Note the student's use of the MATLAB external call (Type 155) to tally the total amount of time the system generates the rated power.

Observations and Conclusions

Before concluding, some of the drawbacks of TRNSYS bear mention. Chief among these is the decidedly mixed quality of the documentation. Though extensive in many regards, it is often difficult to find exact and thorough descriptions of the parameters, inputs and outputs for given types; occasionally one must even guess at the units. Documentation as to the behavior of the certain types is also sometimes lacking and programmers may have to look in one of several locations to find information. On the other hand, virtually all of TRNSYS' types are open-source; their underlying mathematical models and code are published. Nevertheless, lack of consistent and detailed documentation can frustrate students unless the instructor is careful to design assignments around types that are well understood.

A second minor drawback concerns units. In most cases, TRNSYS uses IP units. Users can select alternate units for fixed parameters or inputs that are hard-coded. However, component inputs that are driven by other system outputs only accept default units and programmers must occasionally deal with incompatibilities. This occurs most often when interconnecting electrical types with non-electrical types. For example, types such as PV arrays and inverters employ units of Watts and degrees Kelvin, but thermal/mechanical types usually employ units of kJ/hr and degrees C. Fortunately, unit compatibility is easily solved (there is a specific type just for this, or one may use equations) and TRNSYS almost always reports unit mismatches so they do not go undetected.

A third difficulty concerns error reporting. Upon running each simulation, the Simulation Studio compiles a "deck" file containing information about how various components are interconnected. The TRNSYS kernel reads the deck file, but knows nothing about how errors in the deck file propagated from errors in the Simulation Studio. Thus, error reports are sometimes rather cryptic and some experience is necessary in order to know what corrective action to take.

On balance, however, the authors' impressions on the use of TRNSYS 16 in the classroom are positive. Students in Solar Engineering developed a feel for the interface in a matter of days, and almost universally reported on end-of-term evaluations that they wished TRNSYS would have been introduced sooner. (Undoubtedly, they quickly realized that it would have made certain early assignments easier!) However, as argued above, the timing of its introduction in the class yielded some pedagogical advantage. Students were soon able to tackle quite sophisticated projects, far beyond what could reasonably be expected using general-purpose computational software such as MATLAB or Excel in reasonable time. Furthermore, as more and more students proceed into careers in energy, it is not unreasonable to hope that skill with energy simulation software may prove valuable. Lastly, it is worth emphasizing that TRNSYS – like any worthy energy simulation software – is not only limited to applications of solar energy. As the Baylor energy core takes shape, it should prove useful in studies of wind power, HVAC and building design, and fuel cells.

Although the qualitative evaluation of TRNSYS given here may prove helpful to educators contemplating the use of numerical simulation software in energy courses, there is still a dearth of hard evidence about its efficacy as a teaching and learning tool. We hope others in the community will also report their data and experiences as energy, and especially alternative energy, become mainstream technical electives at four-year universities.

References

1. Bradley, D.; Kummert, M.; "New Evolutions in TRNSYS – A Selection of Version 16 Features," Proc. 9th Int'l IBPSA Conference (Building Simulation 2005), August 2005, pp. 107-113
2. Mondol, J.D..; Yohanis, Y.G.; Norton, B.; "Comparison of a Measured and Predicted Long Term Performance of a Grid Connected Photovoltaic System," Energy Conversion and Management, v. 48, no. 2, 2007, pp. 1065-1080
3. Mondol, J.D.; Yohanis, Y.G.; Smyth, M.; Norton, B.; "Long Term Validated Simulation of a Building Integrated Photovoltaic System," Solar Energy, v. 78, no. 2, 2005, pp. 163-176
4. Electric Power Research Institute, "Solar Heating and Cooling (SHAC) Simulation Programs: Assessment and Evaluation – Summary Report," May 1981
5. Goswami, D.Y.; "Present Status of Solar Energy Education," Proc. 2001 ASEE Annual Conf. and Exposition, Session 1433
6. Haberl, J.S.; Cho, S.; "Literature Review of Uncertainty of Analysis Methods (F-Chart Program)," Energy Systems Laboratory report ESL-TR-08/08-04, Texas A&M University
7. Hodge, B.K.; "Alternate Energy Systems – A New Elective?" Proc. ASEE Annual Conference & Exposition, 2002, Session 2566
8. Kalogirou, S.; Papamarcou, C.; "Simple Validation Method of a TRNSYS Model for a Thermosyphon Solar Water Heating System," Proc. 10th Mediterranean Electrotechnical Conference (MELCON 2000), vol. 3, pp. 1102-1106
9. Karlsson, F., Rohdin, P., Persson, M.L.; "Measured and Predicted Energy Demand of a Low Energy Building: Important Aspects when using Building Energy Simulation," Building Service Engineering Research and Technology, v. 28, no. 3, 2007, pp. 223-235
10. Lazzarin, R. M., Castellotti, F., Busato, F.; "Experimental Measurements and Numerical Modelling of a Green Roof, Energy and Buildings, v. 37, no. 12, 2005, pp. 1260-1267
11. Lior, Noam; "MEAM 402/502: Energy Engineering," http://www.seas.upenn.edu/~meam402/
12. Loutzenhiser, P.G., et. al.; "Empirical Validation of Models to Compute Solar Irradiance on Included Surfaces for Building Energy Simulation," Solar Energy, v. 81, no. 2, 2007, pp. 254-267
13. Loutzenhiser, P.G.; et. al.; "An Empirical Validation of Modeling Solar Gain through a Glazing Unit with External and Internal Shading Screens," Applied Thermal Engineering, vol. 27, no. 2-3, 2007, pp. 528-538
14. Minnerly, B.V., Klein, S.A., Beckman, W.A.; "Rating Procedure for Solar Domestic Hot Water Systems Based on ASHRAE-95 Test Results," Solar Energy, v. 47, no. 6, 1991, pp. 403-411
15. Mitchell, J.W., Pawelski, M.J., Beckman, W.A., "Comparisons of Measured and Simulated Performance for CSU House I," ASME J. Solar Energy Engineering, v. 102, no. 3, 1980, pp. 192-195
16. SRCC; http://www.solar-rating.org/CERTIFICATION/CERTIFICATION.HTM

IAN A. GRAVAGNE
Dr. Gravagne is an assistant professor with the Electrical and Computer Engineering department at Baylor University. He teaches the Engineering Design II (“senior design”) course, as well as technical electives in solar energy, robotics and engineering mathematics. His principal research interests are the engineering applications of dynamic equations on time scales and energy education. He can be contacted at Ian_Gravagne@baylor.edu.

KENNETH VAN TREUREN
Dr. Van Treuren is a professor on the faculty in the Mechanical Engineering department at Baylor University. He teaches the capstone Mechanical Engineering Laboratory course, as well as courses in heat transfer, aerospace engineering, and wind power. His research interests include energy education and literacy and gas turbine design. He can be contacted at Kenneth_Van_Treuren@baylor.edu.