

NOVEL HANDS-ON WATER OVERFLOW SIS EXPERIMENT

in Undergraduate Process Control Laboratory

YAN FANG, M.A.K. RASEL, TRACY J. BENSON, AND PEYTON C. RICHMOND
Lamar University · Beaumont, TX 77710

This paper describes a hands-on experiment that was created in response to the ABET Program Criteria safety emphasis in the Criteria for Accrediting Engineering Programs. In 2012-2013 the Program Criteria^[1] was modified to include the phrase “including the hazards associated with these processes.” This experiment provides students with hands-on experience with a prototype Safety Instrumented System (SIS) for a tank-level control process designed to improve their fundamental understanding of ubiquitous chemical process safety systems. This “Water Overflow SIS” or “SIS” experiment is used as part of our CHEN4150 Process Control Laboratory, a senior co-requisite course with our CHEN4332 Process Control II course.

The SIS experiment involves an easily understood acrylic container water level on-off control process coupled with inexpensive off-the-shelf sensors and actuators driven by a \$20 microcontroller (Arduino) as the control system. It is modeled after liquid hydrocarbon storage tank controls found in tank farms throughout the hydrocarbon processing industry. Our pedagogical approach is similar to other microcontroller-based educational activities.^[2,3] We provide a hands-on experiment to give students the opportunity to build a bridge between theoretical safety knowledge and its practical application. [We use a similar Arduino-based “Level Measurement and Control” experiment to demonstrate measurement error, calibration, and Proportional Integral and Derivative (PID) control concepts.]

The Water Overflow SIS experiment illustrates how independent layers of protection interact to protect process operations. Students design and execute an experiment during a 3-hour lab to demonstrate how adding an SIS can reduce the overall probability of overfilling a tank. Two Arduino boards and a pair of relays are used for the experiment. One Arduino represents the Basic Process Control System (BPCS) and the other represents a dedicated SIS. Students design the

experiment based on the Probability of Failure on Demand (PFD) for the BPCS and the SIS using a Layer of Protection Analysis (LOPA). Then, the tank filling operation is started and data is collected to estimate the overall failure probability (tank overfilling) both with and without the SIS. Statistics are used to check the significance level of their experimental results. Students have full access to all of the hardware and software used for the project.

The student learning outcomes for the experiment in the laboratory class were assessed by a feedback survey and a quiz administered at the end of the semester. The quiz concerning Hazard and Operability Study (HAZOP) and LOPA was taken verbatim from a Plant Design I course exam the students completed the prior semester.

Yan Fang is a current Ph.D. student and research assistant of chemical engineering at Lamar University in Beaumont, TX. She received her B.S. (2010) in chemistry from Hebei Normal University in China and her M.E. (2013) in chemical engineering from Lamar University. Her research interests include dynamic simulation and process control. She is currently a teaching assistant of undergraduate process control laboratory.

M.A. Kader Rasel received his B.Sc. in chemical engineering from the Bangladesh University of Engineering & Technology in 2005. He received his M.S. degree in chemical engineering from Lamar University in 2010. He is currently pursuing a Ph.D. in chemical engineering at Lamar University. His research interests include dynamic simulation, instrumentation, and control for process safety and reliability.

Tracy J. Benson is an assistant professor of chemical engineering at Lamar University in Beaumont, TX. He received his B.S. and Ph.D. from Mississippi State University. His research interests include CO₂ conversion strategies and biomass conversion. He currently teaches the momentum and mass transfer classes and the unit operations laboratory.

Peyton C. Richmond is an associate professor of chemical engineering at Lamar University in Beaumont, TX. He received his B.S. from Lamar University and his Ph.D. from Texas A&M University, both in chemical engineering. His research interests include industrial process control and undergraduate education. He currently teaches the capstone plant design classes and undergraduate process control laboratory.

LITERATURE SURVEY

There is no denying that improving the safety of operating industrial plants not only protects the people and equipment within the plant but also protects people in nearby communities. Plant safety is always the most important control objective for industry. In Edgar's round table discussion, Rinard acknowledged the preeminence of safety by noting that the regulatory control system affects the size of your paycheck, but the safety control system affects whether or not you will be around to collect it.^[4] Goble noted that safety and reliability are essential parameters of automatic control systems design. It is also clear that a safe and reliable system provides a lot of benefits such as less lost production, higher quality product, reduced maintenance costs, lower risk costs, regulatory compliance, the ability to schedule maintenance, and even peace of mind and satisfaction of a job well done.^[5] Crowl and Louvar mentioned that much more recently "safety" has been replaced by "loss prevention." This term includes hazard identification, technical evaluation, and the design of new engineering features to prevent loss. They give the definition of "safety or loss prevention" as the prevention of accidents through the use of appropriate technologies to identify the hazards of a chemical plant and eliminate them before an accident occurs.^[6]

Process safety is a major focus of engineering organizations including the Center for Chemical Process Safety (CCPS) and the International Society of Automation (ISA). The ISA-84.00.01-2004 Part 1, Systems for the Process Industry Sector, which provides information for SIS safety requirements specification, SIS design and engineering, SIS installation and commissioning, SIS safety validation, SIS operation and maintenance, SIS

modification, and SIS decommissioning,^[7] is a great resource to learn more about SIS. Wayne Cohen noted that no one doubts the need for SIS in industry today. The challenge is to determine appropriate safety controls and trips and to provide them within budgetary constraints. The ISA standards for SIS has provided the industry with guidelines to ensure that engineers and plant owners follow a sensible, well planned process to design, purchase, and install the system required for each operating plant.^[8]

For undergraduate chemical engineering education, it is extremely important to emphasize the hazards and safety issues to students before they step into their first real industry position. As Feisel and Rosa noted in their work, one of the fundamental objectives of engineering instructional laboratories is to identify health, safety, and environmental issues related to technological processes and activities, and deal with them responsibly.^[9]

Our Water Overflow SIS Experiment is not only designed to improve students' knowledge of safety systems, but also to improve their conceptual understanding by providing a hands-on safety experiment for students to touch and feel. Ma and Nickerson noted that hands-on laboratories placed a strong emphasis on conceptual understanding and design skills. Professional skills were also recognized as an important mission for hands-on laboratories.^[10] Benson, *et al.* also noted that teamwork, chemical process safety, increased communication skills, and sensory awareness were the goals of hands-on experiment.^[11]

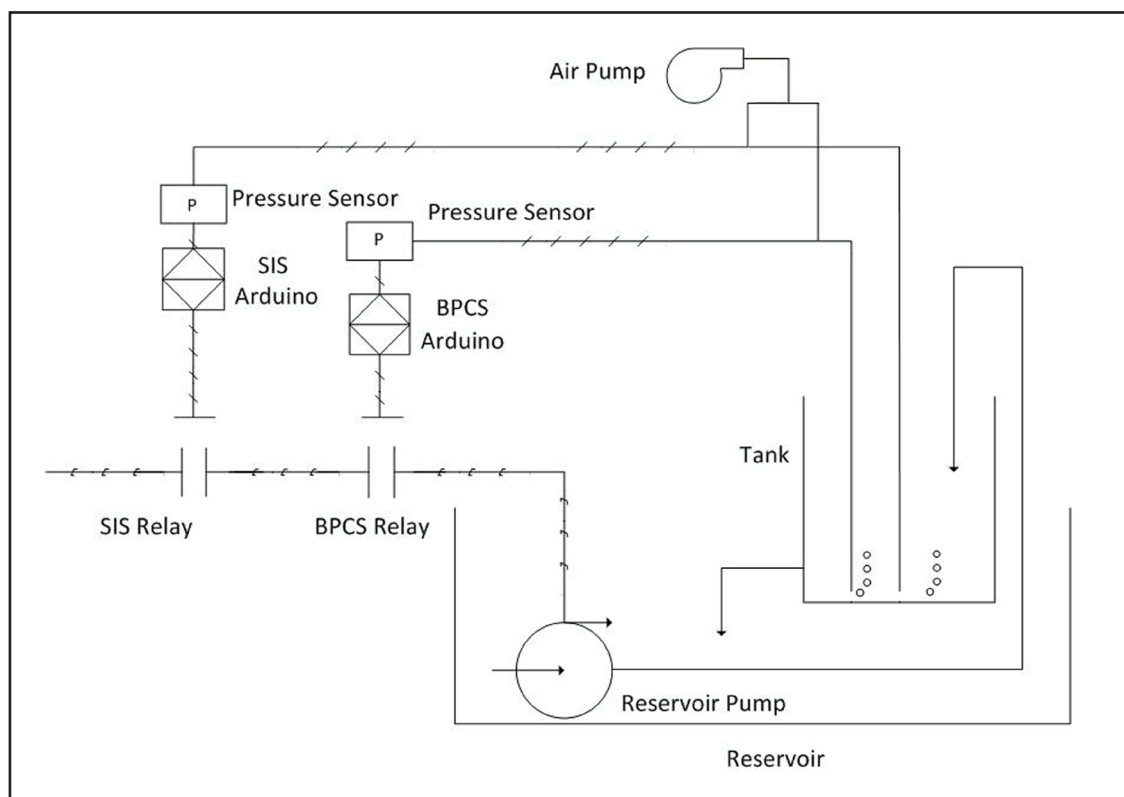


Figure 1. Water Overflow SIS experiment schematic diagram.

DESCRIPTION OF EXPERIMENT

Experiment setup

The water tank overflow SIS experiment consists of a small acrylic tank that is filled with water using an on-off submersible pump and relay-based on-off control. Figure 1 is the schematic diagram of the experiment's setup. Figure 2 is a picture illustrating the size and placement of the actual equipment. The total cost for this experiment is about \$235. A detailed breakdown of the items and their costs is shown in Appendix A, Table A-1. (As noted above, we use a similar Arduino-based experiment "Level Measurement and Control" to demonstrate measurement error, calibration, and PID control using variable speed submersible pumps.)

The pump is controlled by two relays connected in series such that both relays must be closed for the pump to run. The experiment includes two Arduino controllers to control the relays and an air pump and two independent pressure sensors to measure pressure and calculate the tank level. Students are not required to calibrate the level measurement for this particular experiment. The two Arduino programs used for the BPCS and SIS are listed in Appendix B. The level is calculated from a measured pressure sensor voltage and stored in the variable "head" as the tank water level in inches. One relay is controlled by the BPCS Arduino and the other relay is controlled by the SIS Arduino based on their independent pressure/level measurements.

The water in the tank continuously drains by gravity into the reservoir thus emptying the tank and reducing the level below the low-level setpoint when the pump is turned off. However, the pump is designed for a larger flow rate than the gravity drainage rate, so when the pump is turned on the level increases. Normally, the tank level is controlled by the BPCS Arduino between a specified high and low level by turning

the pump on when the tank reaches the low-level setpoint and then back off after it reaches the high-level setpoint. Then the control cycle repeats. During normal operation the tank never reaches the SIS high-level setpoint because it is set higher than the BPCS high-level setpoint.

A pseudo random number is used to determine if the BPCS on-off control should fail based on the PFD value entered by the student. A failure results in it not turning off the pump when the level reaches the high-level setpoint. A BPCS failure causes the tank level to continue to increase until it reaches the SIS high-level setpoint. Then, the SIS can open its relay and stop the pump in order to avoid a tank overflow event. However, there is an independent pseudo random number used to determine if the SIS will fail. Simultaneous BPCS and SIS failures will cause the tank to overflow and water to spill down the tank's side back into the reservoir until both the BPCS and SIS are automatically reset by the Arduino code for the next control cycle.

Experiment theory

The tank level is primarily controlled by the BPCS on the first Arduino board with an on-off controller. The BPCS represents the control system that process operators interact with during normal operation. The BPCS calculates the height of the liquid level from its pressure signal similar to many pressure difference based level indicators used by industry. When the level is low (*i.e.*, below 2 inches) the BPCS closes its relay to turn on the pump to fill the tank with water. When the tank level gets high (*i.e.*, above 4 inches) the BPCS opens its relay to turn off the pump in order to allow the water to drain back into the reservoir and lower the level. When the BPCS fails to turn off the pump at the random PFD frequency specified by the student in the BPCS Arduino code it causes the water level to increase to the SIS level setpoint.

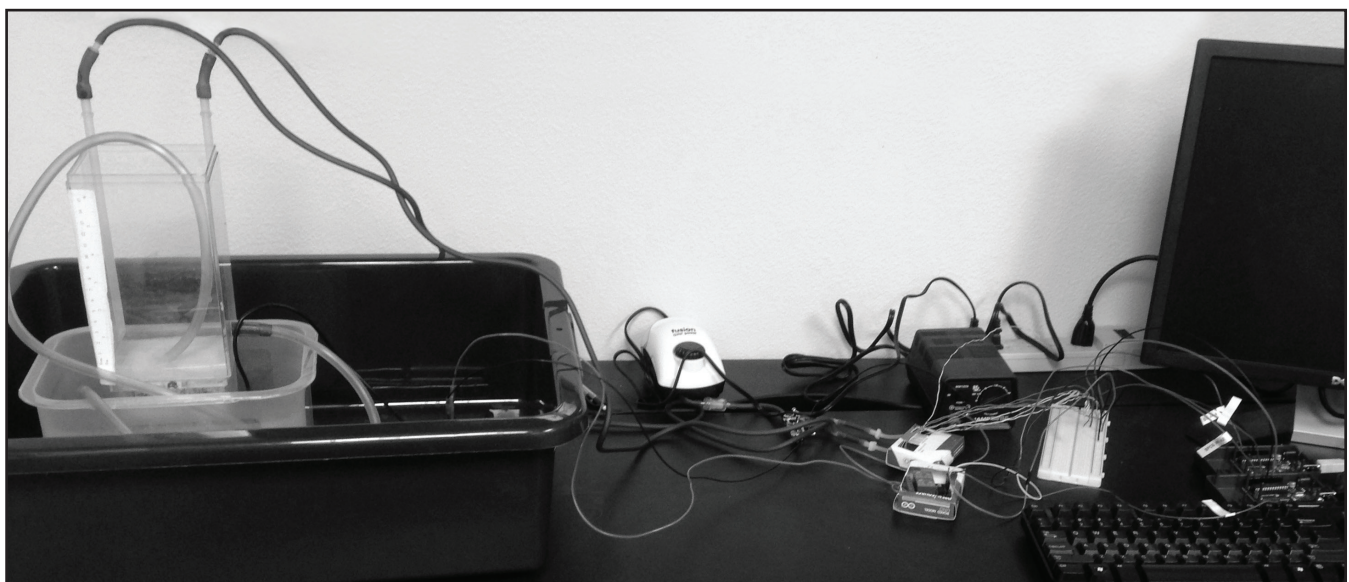


Figure 2. Water Overflow SIS experiment equipment.

The SIS represents an additional independent layer of protection and runs on a separate Arduino board and also uses an independent water level measurement to emphasize to students that these systems must be independent for the LOPA to be valid. It provides an extra layer of protection by opening the relay if the level reaches 5.5 inches. The SIS can be removed by setting the program's "SIS" variable to zero. When the SIS variable is set to zero, the SIS relay stays closed allowing the students to observe the BPCS failure rate without an SIS. However, even if the SIS is active it is programmed to fail with the PFD specified by the student, thus, there is still a finite probability that the water will overflow the tank even with an active SIS.

In practice both the BPCS and SIS failure rates must be considered as part of a LOPA. The BPCS failure rate is typically based on rule of thumb. However, an industrial SIS's failure rate must be designed using the PFD of all of its components and its testing protocol and frequency. For the purposes of this experiment the BPCS and SIS are programmed to fail at a very high rate so that students can observe the operation (and failure) of the SIS during the 3-hour lab.

Procedure and results

The students are required to complete a HAZOP table (Table 1) for overflowing the storage tank based on a BPCS failure. In this case, the node is "tank," the process parameter is "level," and the deviation (guide word) is "more."

Node	Process Parameter	Deviation (Guide Word)	Cause	Consequence	Action
Tank	Level	More			

The cause should be high flow into the tank and failure of the level control system. The consequence can be tank overflows. The action one should take is to use an SIS.

From the LOPA table (Table 2), students calculate that for a mitigated event likelihood of 0.20 (row 1, column 9), the PFD of the BPCS failure must have been 0.20 (row 1, column 3). Then, to obtain a mitigated event likelihood of 0.05 for the SIS protected system (row 2, column 9) given the BPCS value of 0.2 (row 2, column 3) the SIS PFD value should be set to 0.25 (row 2, column 7).

Project 1– without SIS

Students open the Arduino SIS_relay.ino file. In the code as shown in Figure 3, they set the SIS variable to 0, so that SIS is deactivated for this project, and upload the program. Then they open the Arduino On_Off_control.ino file. In the code as shown in Figure 4, students find the variable "pfd" which stores the BPCS PFD value and set the value to 0.2 and upload the program. Students are asked to record in a table the numbers of incidents observed during one hour of water-filling operations. With the number of total cycles and the number of failure incidents, they can estimate the probability of failure on demand of the BPCS. For example, one group recorded a total of 74 cycles including 16 failures, for an observed $P_B \cong 0.216$.

#	1	2	3	Protection Layers					9	10
				4	5	6	7	8		
	Initial Event Description	Initiating Cause	Cause Likelihood	Process Design	BPCS	Alarm	SIS	Additional mitigation (safety valves, dykes, etc)	After Mitigation event likelihood	Notes
1	Water Overflow	BPCS failure	Probability failure on Demand	1.0	1.0	1.0	1.0	1.0	0.20	
2	Water Overflow	BPCS failure	Probability failure on Demand (same as BPCS failure in 1)	1.0	1.0	1.0	Probability failure on Demand	1.0	0.05	

```

//*****
int SIS = 0; //Set value either 0 or 1; 0 turns off the SIS
//*****

```

Figure 3. Partial Arduino code of SIS_Relay.ino.

```

// probability failure on demand entered by students
//*****
float pfd = 0.2;
//*****

```

Figure 4. Partial Arduino code of BPCS On_Off_Control.ino.

Another goal for this experiment is to provide students a concept of hypothesis. In our project, we have two kinds of cycles: normal cycles and failure cycles. We have our students use the statistics for a binomial population from Ostle and Mensing to test if their failure data is consistent with the expected failure rate.^[12] Students test the following hypothesis with significance level of 1% (the probability of error, $\alpha = 0.01$). The null hypothesis, H, of this project is the PFD of BPCS equals to 0.2. The alternative hypothesis, A, is that the PFD of BPCS does not equal to 0.2. (H: $P_B = 0.2$; A: $P_B \neq 0.2$.) The following statistics are used to test this hypothesis:

$$Z = \frac{(X + 0.5) - n\rho_0}{\sqrt{n\rho_0(1-\rho_0)}}, \text{ for } X < n\rho_0 \quad (1)$$

$$Z = \frac{(X - 0.5) - n\rho_0}{\sqrt{n\rho_0(1-\rho_0)}}, \text{ for } X > n\rho_0 \quad (2)$$

In these equations, Z stands for standard normal distribution, X stands for the number of desired test (which is the number of the failure cycles in our project), n stands for the sample size (which is the number of total cycles in our project), ρ_0 stands for the assumed probability of failure on demand which is the “pfd” value students set in the code. The null hypothesis is rejected if $|Z| \geq Z_{1-\alpha/2}$. For our project, $X=16$, $n=74$, $\rho_0=0.2$,

$16 > 74 * 0.2$. So Eq. (2) is used here, $Z = [(16-0.5) - 74*0.2] / [74*0.2(1-0.2)]^{1/2}$, solved for $Z = 0.203$. From the Appendix of Ostle and Mensing,^[12] we find $Z_{0.995} = 2.58$, $Z < Z_{0.995}$. The null hypothesis is accepted. Therefore, the experiment result is verified by the hypothesis testing.

Project 2 – with SIS

Students open the SIS_relay.ino file in Arduino program, set the SIS variable to 1, PFD value to the calculated value of 0.25 as shown in Figure 5, and upload the program. Then they open the Arduino On_Off_control.ino file. The number of incidents is observed in one hour and filled in a table. For example, one group observed a total of 91 cycles with five failures. Thus, their estimate of the probability of failure on demand with an SIS, P_s , is 0.0549.

Students also use Eqs. (1) and (2) to test the hypothesis, “H: $P_s = 0.05$; A: $P_s \neq 0.05$,” with 1% significance level ($\alpha = 0.01$). For this project, $X=5$, $n=91$, $\rho_0=0.05$, $5 > 91 * 0.05$. So Eq. (2) is used here, $Z = [(5-0.05) - 91*0.05] / [91*0.05(1-0.05)]^{1/2}$, solved for $Z = 0.19$. Since $=2.58$, we find $Z < Z_{0.995}$. The null hypothesis is accepted. Therefore, the experiment result is verified by the hypothesis testing.

```

//*****
int SIS = 1; //Set value either 0 or 1; 0 turns off the SIS
//*****
...
void setup()
{ ...
  // probability failure on demand entered by students
  //*****
  float pfd = 0.25;
  //*****
  ...
}

```

Figure 5. Partial Arduino code of SIS_Relay.ino.

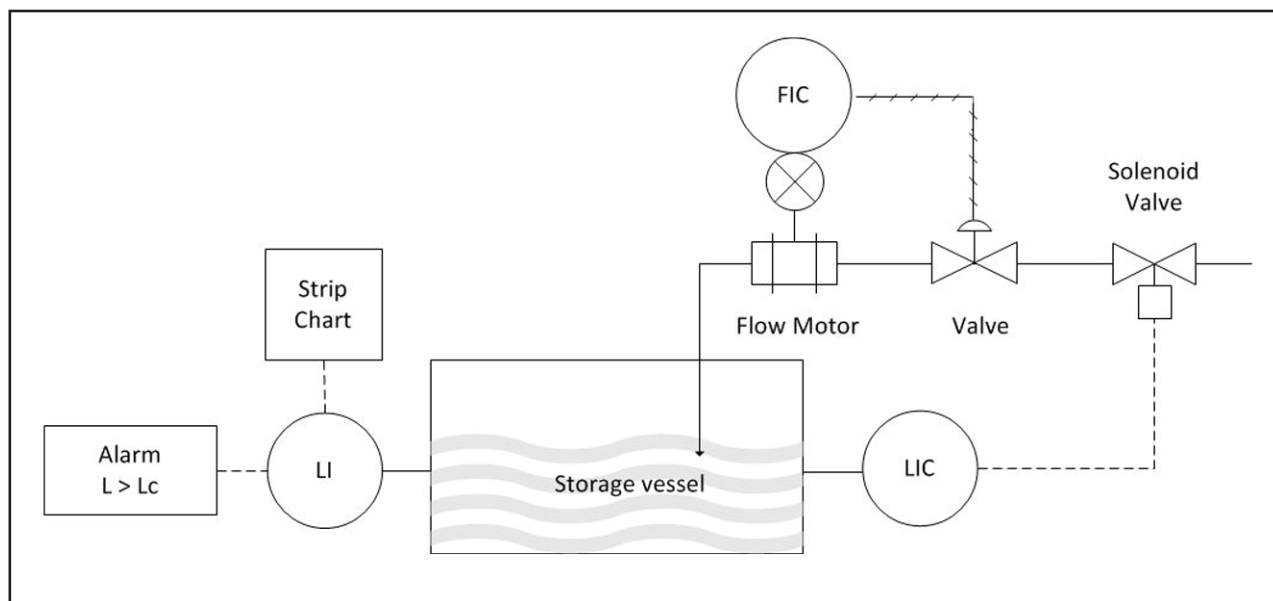


Figure 6. Level Control system with alarm. FIC stands for Flow Indicator Controller, LIC stands for Level Indicator Controller, LI stands for Level Indicator, L stands for Level, and Lc stands for alarm setpoint.

Another hypothesis, “H: $P_B = P_s$; A: $P_B \neq P_s$,” is also tested with 1% significance level ($\alpha = 0.01$) for this project. In this situation, we compare the proportions between two binomial populations. The normal approximation to the binomial distribution can be used to get an approximate test.^[12] The following test statistic equation was used:

$$Z = (\rho_1 - \rho_2) / [P(1-P)(n_1 + n_2) / n_1 n_2]^{1/2}, \text{ where } \rho_j = X_j / n_j, P = (X_1 + X_2) / (n_1 + n_2) \quad (3)$$

The null hypothesis is rejected if $|Z| \geq Z_{1-\alpha/2}$. In this project, $P = (16+5) / (74+91) = 0.127$, plug the values into Eq. (3), $Z = (0.216-0.0549) / [0.127(1-0.127)(74+91) / (74*91)]^{1/2}$, solved for $Z = 3.09$. Since $Z_{0.995} = 2.58$, we find $Z > Z_{0.995}$. The null hypothesis is rejected which means $P_B \neq P_s$. Thus, the overall probability of overfilling the tank with an SIS, 0.0549, is significantly different than the probability of overfilling the tank without an SIS, 0.216. The hypothesis test allows the students to demonstrate that the addition of an SIS significantly reduced the overall probability of overfilling the tank.

A QUIZ BEFORE AND AFTER THE LAB

The experimental setup for this laboratory is similar to but slightly different from a level control problem shown in Figure 6 that the students are given in a Plant Design I exam the prior semester following a discussion of HAZOP and LOPA.^[6]

To help gauge the extent to which this experiment improves their conceptual understanding of HAZOP and LOPA, we have the entire class retake this exam question after the control laboratory is complete. In the first problem, students are asked to determine two causes that would lead to an overfilling and complete the associated HAZOP table entries using the process parameter “level” and the guide word “more” in Figure 7.

Prior to the lab, 81% of the students determined at least one correct cause, while only 10% of the students determined both correct causes. The two causes are LIC failure and Solenoid Valve failure. Many students mistakenly think that an overfilling could be caused by an FIC failure. After the lab, 84% of the students again determined at least one correct cause, but now, 32% of the total number of students determined both correct causes. The consequences of the LIC failure and valve failure could include the vessel overfilling, vessel rupture, and pipe rupture. For the actions, most students respond that the operator should

close the valves manually when the alarm sounds. Some students suggest adding an SIS for the LIC and a second back-up alarm for the system. A few students suggest building a dike for the vessel.

The second problem is to complete the LOPA table entries and indicate if an SIS is required to protect the tank from overfilling for a target-mitigated event likelihood of 10^{-5} events/year. Students are provided with the following PFD (events per year) of the base systems: high-level alarm (LI) is 0.01, operator stops flow if alarm sounds is 0.1, flow controller (FIC) is 0.01, flow control valve is 0.001, high-level switch (LIC) is 0.01, solenoid valve is 0.001, LIC cause likelihood is 0.01, and operator stops if alarm sounds is 0.1. So the SIS is required and its PFD should be 0.01 since the mitigated event likelihood is 10^{-5} . The complete LOPA table entries are shown in Figure 8.

The students' performance on the LOPA analysis was improved after the lab but there still seemed to be some confusion about the purpose and meanings of the LOPA table entries. To help students better achieve the objectives of Water Overflow SIS experiment we plan to improve the laboratory instruction in future labs.

STUDENT FEEDBACK SURVEY

The student feedback survey was prepared in the form of comments about the experiment. The survey used a Likert Scale with the following levels: 1) Strongly Disagree; 2) Disagree; 3) Agree; and 4) Strongly Agree. The result of the 28 students'

Prior to the lab, 81% of the students determined at least one correct cause, while only 10% of the students determined both correct causes. The two causes are LIC failure and Solenoid Valve failure. Many students mistakenly think that an overflowing could be caused by an FIC failure. After the lab, 84% of the students again determined at least one correct cause, but now, 32% of the total number of students determined both correct causes.

HAZOP Table for Quiz Problem 1					
Node	Process Parameter	Deviation (Guide Word)	Cause	Consequence	Action
Storage Vessel	Level	More			
↓	↓	↓			

Figure 7. Water Overflow SIS experiment quiz problem 1. HAZOP table.

LOPA Table for Quiz Problem 2								
Initial Event Description	Initiating Cause	Cause Likelihood	Protection Layers					Mitigated event likelihood
			Process Design	BPCS	Alarm	SIS	Additional mitigation (safety valves, dykes, etc)	
High tank level leading to tank overflow	Failure of high-level switch (LIC)	0.01	1	1	0.1	0.01	1	10^{-5}
SIS Required?	YES	NO	(Circle YES or NO)					

Figure 8. Water Overflow SIS experiment quiz problem 2. LOPA tables.

responses to the Water Overflow SIS survey statements are shown in Table 3.

The survey results reflected the students' general experience with this experiment. Concluded from the assessment of the survey result, our objectives for this experiment are basically achieved. Some students mentioned in their additional comments that the experiment could be improved by giving more details in the lab description.

CONCLUSION

In this article, we describe a novel undergraduate chemical engineering safety process control laboratory experiment created in response to 2012-2013 ABET Program Criteria safety additions. The experiment's components used are very inexpensive and easy to obtain online. This experiment provides students with hands-on experience with SIS technology in an environment that allows them to apply theoretical safety analysis concepts to an easily understood real-world problem that could be seen and felt. The pre-lab quiz and student survey feedbacks generally show that the safety operation awareness of students was improved and that the hands-on safety experiment was well received.

REFERENCES

1. "2012 – 2013 Criteria for Accrediting Engineering Programs," the Engineering Accreditation Commission of ABET, <<http://www.abet.org>>, Oct. 29, 2011, <<http://abet.org/DisplayTemplates/DocsHandbook.aspx?id=3143>>

2. Teikari, P., R.P. Najjar, H. Malkki, K. Knoblauch, D. Dumortier, C. Gronfier, and H.M. Cooper, "An Inexpensive Arduino-Based LED Stimulator System For Vision Research," *J. Neuroscience Methods*, **211**(2), 227 (2012)
3. Elmore, B.B., "A Freshman Design Course Using LEGO®NXT Robotics," *Chem. Eng. Ed.*, **45**(2), 86 (2011)
4. Edgar, T.F., "Process Control Education in the Year 2000—A Round Table Discussion," *Chem. Eng. Ed.*, **24**(2), 72 (1990)
5. Goble, W.M., *Control Systems Safety Evaluation and Reliability*, 3rd Ed., International Society of Automation (2010)
6. Crowl, D.A., and J.F. Louvar, *Chemical Process Safety: Fundamentals With Applications*, 3rd Ed., Prentice Hall (2011)
7. ISA-84.00.01-2004 Part 1 (2004)
8. Cohen, W., "Application of ISA S84.01 to SIS in the Chemical and Petrochemical Industries," *Process Safety Progress*, **18**(4), 221(1999)
9. Feisel, L.D., and A.J. Rosa, "The Role of the Laboratory in Undergraduate Engineering Education," *J. Eng. Ed.*, **94**(1), 121(2005)
10. Ma, J., and J.V. Nickerson, "Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review," *ACM Computing Surveys*, **38**(3), 7-es (2006)
11. Benson, T.J., P.C. Richmond, and W. Leblanc, "Unit Operation Experiment Linking Classroom with Industrial Processing," *Chem. Eng. Ed.*, **47**(2), 91 (2013)
12. Ostle, B., and R.W. Mensing, *Statistics in Research*, 3rd Ed, Iowa State University Press (1979)

APPENDIX A

Equipment cost of water overflow SIS experiment

See Table A-1.

Statements	Strongly Disagree	Disagree	Agree	Strongly Agree
1. Process safety is part of a company's culture and the process engineer has little input toward design and implementation safety issues.	39%	43%	7%	11%
2. The Arduino Safety Instrumentation System (SIS) experiment was easy to understand and perform.	0%	0%	54%	46%
3. The Arduino SIS experiment will help me to become a more competent engineer in designing and implementing process safety systems.	0%	14%	46%	39%
4. The goal set for my group was achievable with the given amount of experimental time with the Arduino SIS experiment.	0%	0%	39%	61%
5. The large-scale applicability of the Arduino SIS experiment is readily identifiable.	0%	11%	32%	57%
6. As a process engineer, process safety is everyone's responsibility—from plant manager to operators and everyone in between.	0%	0%	4%	96%
7. There is a little value in having Basic Process Control System (BPCS) and SIS on two separate operating systems. This redundancy only adds to the cost and complexity of the process.	50%	29%	14%	7%
8. The Arduino SIS experiment improved my understanding of the Probability of Failure on Demand (PFD) as applied to controls and SISs in a chemical plant setting.	0%	0%	46%	54%
9. SIS failures can lead to costly downtimes and unsafe operations.	0%	11%	36%	54%
10. The Arduino SIS experiment, although shown as a liquid level control device, can be readily extended into other unit operations within an industry setting.	0%	0%	54%	46%

**TABLE A-1
Equipment Cost**

Item	Cost
Undivided Bus Box × 1	\$17.56
Storage box (6 quarts) × 1	\$5.50
Storage container (4''*4''*7'') × 1	\$5.99
Storage container (3.5''*3.5'' * 5'') × 1	\$6.35
Oggi acrylic straws × 2	\$4.76× 2
Arduinouno r3 × 2	\$22.01× 2
Box arduino × 2	\$13.01× 2
Fusion air pump × 1	\$14.15
Power supply regulated × 1	\$19.40
Breadboard × 1	\$5.50
Sensor pressure SMD 8-sop × 2	\$13.94× 2
4 way nickle plated valve × 1	\$6.46
Silicone tubing × 1	\$3.55
Air pump check valve × 1	\$5.13
3.6 L/min brushless submersible pump × 1	\$11.50
Sainsmart 2-channel relay module × 1	\$8.69
Kynar Barbed Reduce Connector 3/16''*1/16'' (25 pack) × 1	\$5.39
Tubing connector filters × 1	\$4.35
Permeatex silicon adhesivex 1	\$4.84
Krazy glux 1	\$3.99
TOTAL	\$235.79

APPENDIX B

ON_OFF_Control.ino programming code

```
int PwmPinR = 2;
int sensorPin = A2;
int analogVal;
float voltage = 0;
float pressure = 0;
float head = 0;
int state = HIGH;
int runN = 1;
long randNumber;
long failNumber;
int randN = 0;
int failN = 0;
// Set the inlet pump BCS relay output pin
// select the input pin for the potentiometer
// the readings from the analog input
//This is the setup function
void setup()
{
  Serial.begin(9600);
  analogReference(DEFAULT);
  pinMode(PwmPinR, OUTPUT);
  randomSeed(analogRead(0));
  // probability failure on demand entered
```

```
by students
//*****
float pfd = 0.1;
//*****
randN = 1.0/pfd;
if (randN < 1) randN = 1;
failNumber = random(1,randN);
}
void loop () {
  // read the value from the sensor:
  analogVal = analogRead(sensorPin); // 10-bit digital number 0 = 0V to 1024 = 5 V
  voltage = analogVal * (5.0 / 1023.0);
  pressure = (voltage - 1.1); // kPa (subtract the zero height voltage)
  head = pressure*4.014;
  Serial.print(head, DEC);
  Serial.print(" in H2O\r\n");

  if (head > 4 && state == LOW) {
    randNumber = random(1,randN);
    if (failNumber == randNumber){
      delay(40000);
      failNumber = random(1,randN);
      failN = failN + 1;
    }
    state = HIGH; // pump turned off
  } else if (head < 2 && state == HIGH) {
    state = LOW; // pump turned on
    runN = runN +1;
  }
  digitalWrite(PwmPinR,state);
  delay (1000); //Wait 1 second
}
```

SIS_Relay.ino programming code

```
//Safty Instrumentation System (SIS)
//*****
int SIS = 0; //Set value either 0 or 1;
0 turns off the SIS
//*****
int PwmPinR = 2;
int sensorPin = A2;
float voltage = 0;
float pressure = 0;
float head = 0;
int state = LOW;
int analogVal = 0;

int runN = 1;
long randNumber;
long failNumber;
int randN = 0;
int failN = 0;
int Nsis = 0;
void setup()
{
  // SIS relay output pin
  // select the input pin for the potentiometer
```

```

// This code will only run once, after
  each powerup or reset of the board
Serial.begin(9600);
analogReference(DEFAULT);
pinMode(PwmPinR, OUTPUT);
randomSeed(analogRead(A0));
// probability failure on demand entered
  by students
//*****
float pfd = 0.2;
//*****
randN = 1.0/pfd;
if (randN < 1) randN = 1;
failNumber = random(1,randN);
}

void loop() {
// This code will loops consecutively
if (SIS == 1) {
  analogVal = analogRead(sensorPin); //
    10-bit digital number 0 = 0V to 1024 =
    5 V
  voltage = analogVal * (5.0 / 1023.0);
  pressure = (voltage - 1.02); // kPa
  (subtract the zero height voltage
  head = pressure*4.0147;
  if (head > 5.5 ) {
    randNumber = random(1,randN);
    if (failNumber == randNumber) {
      failNumber = random(1,randN);
      delay(30000);
      failN = failN + 1;
    }
    state = HIGH;
    digitalWrite(PwmPinR,state);
    delay(27000);
    Nsis = Nsis + 1;
  } else {
    state = LOW;
  }
} else {
state = LOW; }
  digitalWrite(PwmPinR,state);
  Serial.print(state, DEC);
  Serial.print(" SIS State\r\n");
  Serial.print(Nsis);
  Serial.print(" SIS engaged \r\n");
  delay(1000);
}
}

```