Paper ID #15716

# Abstraction Thresholds in Undergraduate Electrical Engineering Curricula

**Dr. Lance C. Perez, University of Nebraska - Lincoln**

Dr. Lance C. Pérez received his B.S. in Electrical Engineering from the University of Virginia, and his M.S. and Ph.D. in Electrical Engineering from the University of Notre Dame. He is currently a Professor of Electrical Engineering at the University of Nebraska-Lincoln where he directs the Perceptual Systems Research Group. His research interests include information, video and signal processing, engineered healthcare and engineering education.

**Dr. Presentacion Rivera-Reyes, University of Nebraska - Lincoln**

Presentacion Rivera-Reyes is currently a postdoctoral research associate in the Department of Electrical & Computer Engineering at the University of Nebraska-Lincoln. He formerly held a position of teaching assistant in the Engineering Education Department at Utah State University. He also held a position as Professor of Telecommunication Engineering at Technological University of Honduras teaching courses of Transmission System to senior students. He received his B.S. in Electrical Engineering from the National Autonomous University of Honduras. He has experience in the telecommunication industry where he worked as a Project Manager developing solutions of high-speed transmission systems for internet services providers and mobile service companies. He has trained engineers and technicians through formal courses, on-the-job training, and supervising on field. His research interest includes self-regulated learning, abstraction in problem solving, and troubleshooting problem solving in laboratory environments. His long-term goals include improving laboratory hands-on activities based on how students improve their metacognitive skills.

# Abstraction Thresholds in Undergraduate Electrical Engineering Curricula

Abstract

A great deal of work has been done to study the types of problems posed to students in various disciplines and to examine the approaches used by students and experts to solve these problems. This paper describes a knowledge representation framework developed by Hahn and Chater [41] for analyzing a person's episode of reasoning while solving a problem and presents some preliminary results of the application of this framework to students taking a course in signal and systems. This course occurs in the junior year of an electrical engineering undergraduate curriculum at a larger public university. The preliminary results demonstrate that the framework can be successfully used to distinguish between different types of reasoning that students use when solving problems in this course. This study is part of a larger effort that is trying to determine if there is a specific point in a typical undergraduate electrical engineering curriculum at which the cognitive demand of the problems being posed exceeds the cognitive supply being brought to the problem by a typical student. The Hahn and Chater framework is being used to assess cognitive supply.

## 1. Introduction

According to Jonassen [1], problem solving is one of the most important cognitive activities in everyday life (p. 63), as well as in the practice of science, technology, engineering, and mathematics (STEM). Professionals in STEM areas must solve problems in order to achieve the goals of a current activity in a specific context. In the context of engineering solving a problem might involve a structured and stated process [2] or the problem may have no specific path solution and require the integration of multiple knowledge domains [3, 4]. When solving an engineering problem students must define the problem and evaluate what the problem entails. After understanding the problem, they must be ready to propose a possible solution, analyze it and decide whether to use it or find alternatives [2].

When analyzing students' solution attempts in a typical undergraduate electrical engineering curriculum, problem solving is complicated by two factors. First, many adult development theories have demonstrated that high-level reasoning and abstraction skills are not reached until one reaches the mid-20s or later [5], after a typical student has completed the undergraduate curriculum. Second, many problems in an undergraduate engineering curricula, particularly in the upper levels, require the ability to abstract a specific problem to a more generalized one, and then translate the results from the generalized problem back to the specific one. As a consequence, we hypothesize that: One, many undergraduate students do not have fully developed reasoning and abstraction abilities, i.e., their *cognitive supply* is not fully developed, and two, there exists in some undergraduate STEM curricula a point at which the disciplinary content and its associated problems have a *cognitive demand* that exceeds the students' current capacity for abstraction.

In over two decades of teaching undergraduate electrical engineering courses, we have observed that students who are proficient at solving high-level, complex problems are equipped with two abilities. First, they are able to hold multiple abstractions in their minds and manipulate them, and, second, they are able to determine which abstractions are valuable when approaching a specific problem. *Abstraction* in solving a problem is operationally defined as the ability to consult multiple similar scenarios stored in the mind, extract relevant features, and apply conclusions about the cases to the problem at hand [6]. This is often done by inventing and solving tractable problems, synthesizing solutions, and applying insights gained from them to solve the original problem. Thus, abstraction involves problem decomposition, or extraction, and rebuilding, or both. *Abstractness* is defined as the degree of abstraction present in the way in which a person is representing a problem. We define the notion of an *abstraction threshold* as a gap between the cognitive supply, i.e., the ability of a student to abstract and reason, and the cognitive demand required to solve a given problem. If such a threshold exists, we further hypothesize that it will become prominent for students in one course in a given undergraduate STEM curriculum. In order to determine if there is an abstraction threshold, it is necessary to analyze the cognitive supply brought by a student in solving a problem and the cognitive demand of a given problem.

The overall goal of our project is to determine whether or not such an abstraction threshold exists and, if so, whether it occurs most frequently in a specific course in the undergraduate electrical engineering curriculum. To do so required the assessment of the cognitive demand of problems in several electrical engineering courses and the assessment of the cognitive supply of the students who attempted to solve the problems. *Cognitive demand* may be defined as the level of thinking required of students in order to successfully engage with and solve a specific problem [7]. For example, *Bloom's Taxonomy* has been used as a tool for characterizing the cognitive demand of problems [8]. We are still refining our method for assessing the cognitive demand of problems and the focus of this paper is on the analysis of *cognitive supply* using the representation mapping framework of Hahn and Chater[41]

## 2. Types of Problems and Problem Solving

Jonassen [1] defines a problem as an unknown entity in some situation, the difference between a goal state and a current state. Finding the unknown is the process of problem solving. Woods [9] states that a problem consists of sets of initial states, goal states, and path constraints. Solving the problem is the act of finding a path that starts with some initial states, follows paths that satisfy the path constraints and ends in the goal states [9]. Much research has been in done in STEM disciplines to define and analyzes the types of problem being posed and to determine approaches or methods for solving various problems.

### 2.1 Types of Problems

*Problems According to Bloom's Taxonomy*

Blooms' taxonomy of educational objectives was originally created as a framework for classifying statements of what educators were expecting students to learn and was based on different categories in the cognitive domain [10]. The taxonomy was hierarchical and progressed from simple to complex and from concrete to abstract. These categories have been revised several times and most recently Krathwohl [10] presented a revision of Bloom's taxonomy that separated and made explicit the knowledge dimension and the cognitive process dimension [10, 11, 12, 13]. Each category represents a scheme of educational goals, objectives, and standards.

Versions of Bloom's taxonomy have been used to study types of problems in many disciplines. Clark and Ernst [11] applied the categories of the Bloom's taxonomy in the field of Geometrics and Technical Graphics Education (GTCE) with the purpose of bridging students learning, assessment, and curriculum development. Ferguson [12] conducted a study over 100 sophomore students to determine the applicability of the Bloom's taxonomy in an English as a second language course [13]. The Center for Teaching and Learning of the University of North Carolina (Charlotte) [14] has developed a series of key words to be used and the type of questions that may help in the development of statements of problems with the purpose to establish and encourage the critical thinking at different categories of the Bloom's taxonomy. These are but a few of the many examples that illustrate the utility of Bloom's taxonomy in classifying problems based on their cognitive demand.

*Well-Structured versus Ill-Structured Problems*

Jonassen [15] differentiated between well-structured and ill-structured problems and recommended different approaches for solving each type of problem. Well-structured problems require the use of a finite number of formulas and concepts and have a specific goal and a set of logical operations. They have easily comprehensible solutions where the relationship between the solution decisions and the problem statement is known [15, 16]. On the other hand, ill-structured problems tend to require the use of several content domains and includes unknown elements with any grade of confidence [16]. They may have multiple criteria for evaluating solutions [17] and they often require the learner to make judgements about the problem [18]. Frederiksen [19] described these two types of problems using the cognitive theory of instruction. He stated that well-structured problems present all the needed information with the appropriate algorithm in order to find the correct solution. On the other hand, he also noted that many that we real life problems are ill-structured. That is, they are not clearly stated, need additional information that it is not readily available, and there may be multiple solutions.

In the subsequent literature in engineering education, many researchers refer to well- and ill-structured problems in their discussions about problem solving. Kumsaikaew, Jackman and Dark [20] conducted a study to determine how expertise affects the identification of task relevant information in engineering problem solving. In their experiment, they used four problem-solving scenarios in which the structure of the problems was mixed with relevant and irrelevant information, similar to engineering problem. This posed the difficultly of challenging the subjects to identify the relevant information to solve the problem. Jonassen, Strobel and Lee [21] described the process used to identify attributes of problems in engineering workplaces. They noted that these problems are substantively different from the problems most often seen in the classroom. Workplace engineering problems are "ill-structured" and complex because they pose conflicting goals, multiple solution methods, non-engineering constraints, and multiple forms of problem representations [21].

Atman, Adams, Cardella, Turns, Mosborg and Saleem [22] mentioned the ill-structured problems in engineering design. They conducted a study comparing the design process activity of engineering students and expert engineers in which the participants designed a solution to a design problem in a laboratory-based setting [22, 23]. This kind of problem is an ill-structured problem because there is no single correct solution, only solutions that better meet the constraints and preferences of the problem statement. Moreover, Atman et al. stated that engineering students and expert engineers have to iterate through different stages rather than following a linear process to handle the ill-structured nature of design problems. Many other researchers have also conducted studies in engineering design identifying design problems as ill-structured that require judgements about the solution and high metacognitive skills [23, 24, 25]. In general, engineering problems are frequently identified as analytical problems (well-structured) or design problems (ill-structured), with the former being highly structured and amenable to algorithmic solutions and the latter requiring nonlinear and iterative solutions [26, 27, 28].

*Convergent versus Divergent Problems*

Problems have also been categorized as being convergent or divergent. In general, convergent problems are the problems of the physical world, i.e., the laws of nature govern the solutions, and these solutions lie along one converging path [29, 30]. Divergent problems are problems that have more than one solution, or a complex mix of seemingly contrary solutions, or no solutions at all. Divergent problems cannot be solved in the sense of establishing a "correct formula", and the answer must be solved at a higher level [29, 30].

Most of the problems of the textbooks are well-structured or convergent allowing a converging path in finding a unique solution. On the other hand, we can relate ill-structured problems or design problems to divergent problems because of the complex nature of these problems. The most common type of divergent problem in engineering curricula is a design type task and there has been significant work done on this in similar domains such as technology education (e.g. Kimbell and Stables 2008, Midouser 2012) and studies of experts and novices in engineering design (e.g. Jacobson 2000, Cross 2004).

## 2.2 Problem Solving Approaches

Problem solving occurs by means of mental processes such as acquiring information or defining procedures to arrive at an acceptable solution [19]. The use of an analytical step-by-step approach is typical in the process of solving well-structured problems. On the other hand, ill-structured problems require the use of a combination of knowledge, intuition and expertise to generate alternatives and choose a suitable solution. There has been a great deal of effort exerted to describe different approaches to solving problems in engineering, from quantitative to cognitive reasoning (i.e., step-by-step, engineering design, rule- and similarity-based).

Solving analytical problems in engineering may require the use of the widely accepted process developed by Polya [31] that consists of the following steps: (1) Represent the Problem, (2) Goal Setting and Planning, (3) Execute the Plan, and (4) Evaluate the Solution. In the first step, the student reads the problem statement and identifies the objective. This step depends of the students' ability to determine the structure of the problem and identify the concepts and formulas necessary to solve it. During the second step, students must develop a path to reach the solution. The Execution step is where the student carries out the procedures to solve the problem. In fourth and final step, students verify that their final solution correctly answered the given problem. A additional step that focuses on the symbol system used to understand and express elements and their relationships may be necessary to solve engineering problems[31]. Gray, Constanzo and

Plesha [32] created a five-step problem solving process to be used in mechanical engineering problems. The five steps are: (1) Road Map, (2) Modeling, (3) Governing Equations, (4) Computation, and (5) Discussion & Verification. The Road Map summarizes the given pieces of information, and an outline of the overall solution strategy. The Modeling step is a discussion of the assumptions needed to make the problem manageable. The Governing Equations are all the equations needed for the solution of the problem. These equations must be organized according the topic (e.g. Newton-Euler equations or Kinematic equations). The Computation step is the manipulation and solutions of the equations used in step 3. Finally, the fifth step is a verification that the solution is correct and a discussion of the meaning of the solution [32].

When solving engineering design problems, students must note that the process "is not linear: at any phase of the process, the engineer may need to identify and define sub-problems, then generate and evaluate solutions to the sub-problems to then integrate back into the overall process" [33]. Dym and Little [2] divided the design problem solving process into five phases: problem definition, conceptual design, preliminary design, detailed design, and design communication. Lawanto, Butler, Cartier, Santoso, Goodridge and Lawanto [34] mentioned that all these five-step design process are used to describe students' cognitive strategies when engaged in problem solving. Whether an electronic circuit or a new manufacturing system, the process of solving design problems requires applying a great deal of domain knowledge with considerable strategic knowledge [1]. Khandani [35] noted that solving design problems is often an iterative process. While implementing the solution, students must "go back to the drawing board" and modify the solution until it meets the requirements. Khandani also defined five steps for solving problems: (1) define the problem, (2) gather pertinent information, (3) generate multiple solutions, (4) analyze and select a solution, and (5) test and implement the solution. Pappas [36] stated that in order to solve engineering design problems, students require the use of creative critical thinking approaches that include: reflection, writing as thinking, visualization, unstructured brainstorming, and understanding the nature of "intentional change" in personal growth.

Despite the proliferation of definitions, frameworks, and step-by-step approaches for problem solving, there is a consensus regarding some of the important skills associated with effective problem solving. It seems that all the approaches identify that effective problem solving includes the search for a clear and concise statement of the problem along with the effective generation, selection, and implementation of alternatives [37]. Many researchers are investigating more cognitive approaches. Moreno, Reisslein, and Ozogul [38] asked how we can help college students develop problem-solving skills in engineering. They attempt to answer the question with the use of the forward-fading approach consisting in the practice of fading (feedback) in every work-out step during problem-solving practice. Sholouhi, Skalle and Aamodt [39] present an overview of different applications of case-based reasoning in petroleum engineering, focusing on solving problems related to drilling. Case-based reasoning is an approach to solving problems consisting of a cyclic process of four main steps: retrieve, reuse, revise, and retain [39]. Kamble and Tembe [40] conducted a study to investigate the effect of a concept mapping strategy in problem solving in mechanical engineering courses. Students were taught to use concept maps as a teaching-learning method. Finally, Hahn and Chater [41] propose a core distinction between rule- and similarity-based processes to describe how the representation of stored information is matched with the representation of a novel item. For the purpose of this preliminary study, the representation mapping of Hahn and Chater is used as the theoretical framework for assessing the cognitive supply that students bring to engineering problem solving.

## 2.3 Theoretical Framework: Representation Mapping of Hahn and Chater

The cognitive supply provided by students during problem solving is related to mental operations made with ideas inside their mind [42, 43]. These mental representations may be described in terms of cognitive processing models based on how knowledge is organized and accessed in the mind [41, 42, 43]. Two long-standing research traditions in exist in cognitive processing: rules and similarities [44]. Rules-based cognitive processing states that cognitive activity occurs through mental rules involving facts about the world applied to specific instances. Many theories, including Piagetian development theory, are based on this tradition [44, 45]. Similarity theories, on the other hand, state that cognitive processing occurs through comparisons of current problems with past problems that are stored in a relatively unprocessed form. Theories based on this tradition are most closely associated with behaviorist theories of learning [42, 43, 44].

Hahn and Chater [41] proposed a model based on representations of knowledge and how they are applied in the two main classes of cognitive processes (rules and similarity) and takes in consideration other types of reasoning, including prototype reasoning and memory bank reasoning. The authors chose the Hahn and Chater framework to analyze the cognitive supply of students for this study, because it captures many forms of reasoning and uses a representation mapping model that incorporates notions of abstraction. The remainder of this section describes the representation mapping model and describes its application to several students' attempts at solving problems in a junior-level electrical engineering course in signals and systems.

The representation mapping model of Hahn and Chater [41] provides a means for distinguishing between different reasoning approaches used by students when attempting to solve problems. The model differentiates between the two main classes of cognitive processes (similarity and rules), and differentiates between four types of reasoning, as shown in Figure 1. Reasoning is mapped to the quadrants in Figure 1 based on the representations of knowledge stored in the mind and the logic processes applied to that knowledge.

A representation of knowledge is a description of how a particular person organizes ideas, concepts, and methods associated with that knowledge. When solving problems, an individual has an initial representation of knowledge that they bring to the problem and a final representation of knowledge when they finish solving the problem. These representations may be the same or different depending on what happens during the problem-solving exercise. The y-axis in Figure 1 is a measure of the difference in the degrees of abstraction present in the initial and final representations of knowledge. Thus, an individual demonstrating either prototype reasoning or rules-based reasoning displays a final representation that is significantly more abstract than the initial representation. An individual demonstrating either similarity-based or memory bank reasoning has initial and final representations of the knowledge with little difference in the degrees of abstraction.

| | | |
|---|---|---|
| Difference in abstractness between stored knowledge representation and new instance representation | ***Prototype reasoning*** | ***Rules-based reasoning*** |
| No difference in abstractness between the two representations | ***Similarity-based reasoning*** | ***Memory bank reasoning*** |
| | Partial matching (Similarity processes) | Strict matching (Rules processes) |

*Figure 1.* Representation mapping model by Hahn and Chater (1998).

The core distinction between rules-based and similarity processes is accounted for by the ways in which representations are used. In particular, they differ in how the representation of a new item is integrated with existing knowledge. In rules-based processes, existing knowledge is stored as rules (if the system is linear and time invariant, then sine waves can only be scaled and delayed). If the antecedent of a rule is satisfied (the system is linear and time invariant), then the category in the consequent applies (sine waves are only scaled and delayed). In similarity processes, knowledge is stored as a set of past instances with category labels (e.g., the student is familiar with many examples of linear and time invariant systems). A new system is classified as linear and time invariant if the past instance it resembles the most has this classification. Distinction between the two processes is evident in how strictly a new item matches existing knowledge, and how specifically the new item is compared to existing knowledge. In rules processes, *matching is strict*, and the representation of the new instance is compared to a *more or less abstract* representation of the antecedent of the rule. In similarity, *matching is partial* and a matter of degree, and representations of the new instance and the existing knowledge that it is compared to are *equally specific.* This scheme explains *memory bank* reasoning as an antecedent of a rule being equally specific as the existing knowledge to which it is compared, thus providing no basis for generalization. On the other hand, partial matching occurs when the new and stored knowledge representations have different degrees of abstractness.

The representation mapping framework of Hahn and Chater has been demonstrated to be a promising method for analyzing students' solutions to chemistry problems []. Since representation mapping is not specific to a domain, it stands to reason that it should also be useful in electrical engineering.

## 3 Methodology

The purpose of this preliminary study was to determine whether the representation mapping of Hahn and Chater could be successfully applied to students' problem-solving attempts in a junior-level signals and systems course. The study consisted of interviewing students while they were solving problems from the course using a talk-out-loud protocol. Interviews were audio recorded and the students' problem solutions were recorded using a LiveScribe pen. The audio and LiveScribe recordings were then analyzed using the Hahn and Chater model [42].

### 3.1 Course Selection and Participants

The course selected for this study was entitled Signals and Systems. It is a course in the junior year of the undergraduate electrical engineering degree program at the institution where the study took place. The purpose of the course is to study the representation of signals in terms of Fourier series expansions and transforms, and the representation of linear time-invariant systems through input-output relations and convolution. The course was chosen based on its history of being a significant stumbling block for students in that its content is much more abstract than, say, a basic circuits course. Thus, it *may* be a course where the cognitive demand of the material is higher than the cognitive supply with which the students are typically equipped. Participants were recruited from a traditional, large-sized university in the Midwest United States. A total of five students enrolled in the course during the fall 2013 semester volunteered to participate in the preliminary study. They were identified as junior students, and the average age was 20. The participants were informed in detail of the purpose of this study during the recruitment process, and they signed the IRB-approved informed consent form before starting the process of data collection. Participants who completed the interview process were given a $20 electronic gift card.

### 3.2 Data Collection and Analysis

The instructor of the Signals and Systems course was asked to select and provide two test problems and one additional problem from the course as suitable indicators of the students' understanding of the material covered in the class. The instructor selected two problems from the first exam and a constructed a similar problem that had not yet been seen by the students. The problems were identified as problems #10 and #11, and the alternative problem (see Figure 2). Due to the exploratory nature of the research, data were collected through semi-structured interviews with the participants. The advantage of the interview methodology is its adaptability: it is possible to obtain information that the participant may not otherwise reveal in a more structured interview or survey. In addition, the method allows the interviewer to follow up on a participant's responses, thus obtaining more information and clarifying vague thoughts [46]. The specific interview protocol chosen for this study was developed by the authors in collaboration with a colleague who had previously used a similar protocol for studies performed with chemistry students. Interviews were conducted using a LiveScribe pen to simultaneously capture hand-written solutions and spoken explanations. The questions asked by the interviewers were designed to elicit the participants' thought processes when solving the problems. The following four questions illustrate the types of questions used by interview protocol.

- *What are the things you first noticed about the problem?*
- *What was your first approach to solving the problem?*
- *Can you walk me through and explain each of the things you did and your thoughts as you were solving this problem?*
- *What was the most difficult part of this problem?*

The course instructor was also interviewed after giving the first exam of the course. The instructor talked through the major ideas of each of the problems that were used for this study, and then asked how they thought a proficient student would solve the problems. Participants were interviewed within 2 weeks of taking the exam. They were asked to solve the problems provided by the instructor of the course, and were encouraged to verbally explain (think aloud) their thoughts while attempting to solve the problems.

After the participants finished the problems, the audio was transcribed, segmented, and coded. In order to ensure the inter-coder reliability, a postdoctoral research associate verified the data-coding process. The postdoc was trained in the IRB process, the theoretical framework of this study, and representation mapping model proposed by Hahn and Chater [42]. The postdoc and the investigator independently analyzed the first interview identifying the episodes and the type of reasoning used by the participants. Then, they met to discuss and revise the differences in coding, and any disagreements among coders were resolved for the first interview. Problems were coded and evaluated according to the following steps:

a) The cognitive supply of the participant and the instructor of the course were assessed by parsing episodes of reasoning in the individual's explanation.
b) The structure and logic of the episode were decomposed to determine the type of cognitive processes employed by the interviewee (similarity- or rules-based reasoning).
c) The structure and logic of the episodes were decomposed to determine the degree of abstraction employed by the interviewee (from no abstraction to abstraction).

Following steps (a), (b), and (c), the investigators categorized the participants' reasoning in the representation mapping quadrants (see Figure 1), as (1) memory-bank reasoning (2) rules-based reasoning, (3) similarity-based reasoning or (4) prototype reasoning.

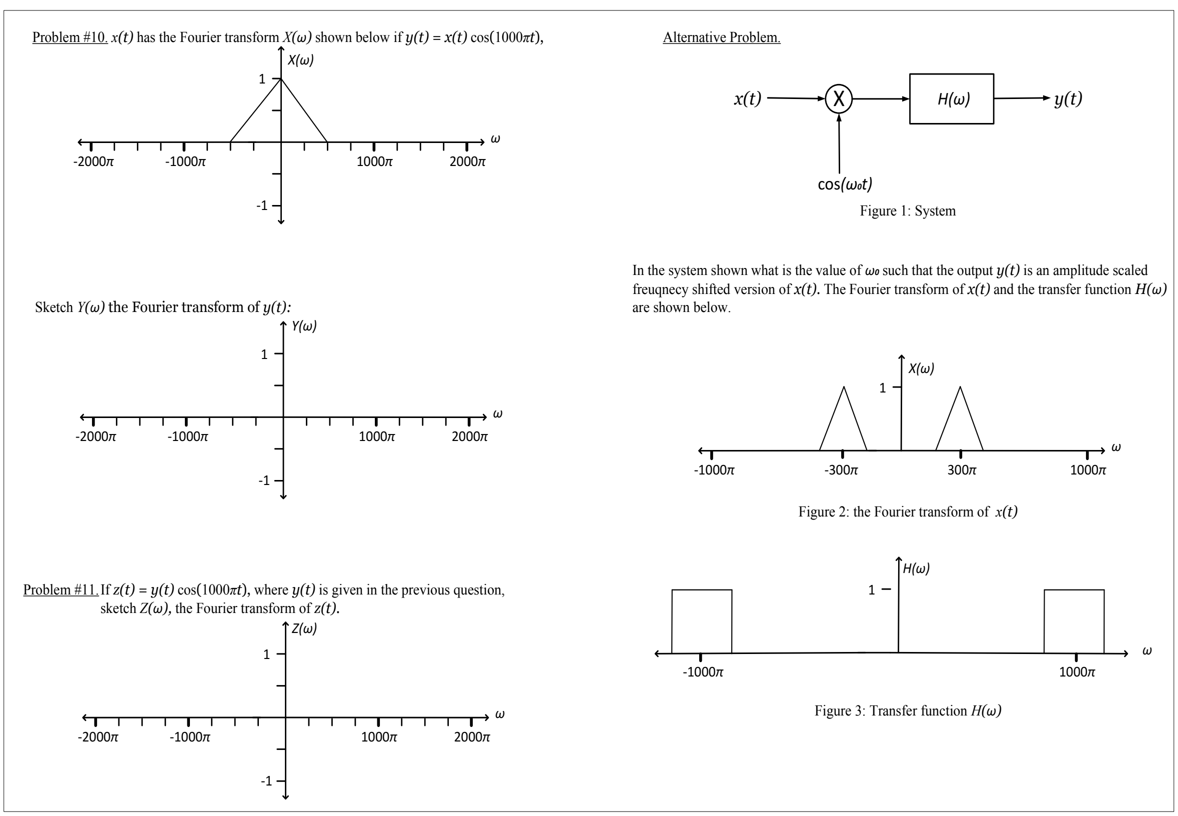

*Figure 2.* Problems from Signals and Systems that were used in this study.

## 4 Example Data Analyses

The representation mapping framework was used to identify the students' cognitive processes during problem-solving. According to the representation mapping, the differences between similarity- and rules-based processing depends on whether the matching is "partial" for similarity-based, or "strict" for rules-based. That is, the degree of abstraction, either partial or strict, determines whether or not the student falls on the left or right half plane in Figure 1. Similarly, the difference between the degree of abstractness of the student's stored knowledge representation and their new representation determines whether or not the student fall on the upper or lower half plane in Figure. If the difference in abstractness between the representations is small, then the student falls on the lower half plane. If the difference in abstractness is larger, then the student falls on the upper half plane. Thus, if a student is doing partial matching and their two representations have roughly the same degree of abstractness then they are using similarity-based reasoning.

Participants were randomly identified as students A, B, C, D, and E. For each participant, the investigator recognized characteristics that appeared to differ in terms of the type of matching being used and the degree of abstractness of the stored and new knowledge representations. Here, examples of the analysis conducted for student A are provided (problem #10 and #11). First, the researcher wrote a summary of the interview in order to identify relevant aspects to be considered during the analysis. Second, the stored knowledge, new instance representation and actions were identified. Finally, the researcher identified the type of matching being used and then the type of reasoning.

<u>Coding and analysis of the interview for Student A, problem #10 (Figure 3)</u>

Summary:
The student started by noticing that the problem states the function $x(t)$ has a Fourier transform shown as $X(\omega$. Then the student looked at the graph to find more information. The student started reading out the equation in the problem and what the problem was asking for. The student was stating explicit features of the problem. Then the student went back to the graph, thought about the cosine function and then restated the equation in the problem. After that, the student remembered that the Fourier transform of $cos(\omega_o t)= \pi[\delta(\omega-\omega_o) + \delta(\omega+\omega_o)]$ and how it can be applied to this problem. The student identified that there were shifts left and right from the original graph given and the amplitude of ½. The student started writing the equation for $Z(\omega)$ and then, sketching the graph of the $Y(\omega)$ identifying the shift phase of $-1000\pi$ and $+1000\pi$. The student knew that the peaks values would be half of the original. The student looked at the peaks and double checked that the problem was copied correctly. The student was asked why they did not use inverse Fourier transform from the beginning. The student responded that they knew that the problem was asking for $Y(\omega)$, and identified the convolution of $X(\omega)$ with $Z(\omega)$, and because $X(\omega)$ was given, the student just sketched the graph left and right *[77-93]*.

Stored knowledge:

- Identify and differentiate between the time domain and the frequency domain.

- The Fourier transform for a cosine function with frequency $\omega_o$ is two delta functions, one centered at $-\omega_o$ and one centered at $\omega_o$ both with amplitude ½.
- Multiplying a function *x(t)* by a cosine shifts the Fourier transform of the function to left and right by $\omega_o$.

New instance representation:

- Noticed explicit features and numeric values stated in the problem.

Action:

- Applied stored knowledge about Fourier transform for cosine function.
- Applied stored knowledge about sketching functions and identifying the frequency shifts.

Strict matching:

- Recognized this problem as a particular case of multiplying a function by a cosine.
- Determined specific numerical values needed to apply frequency shifting.
- No need to discard any extraneous information.

Abstractness:

- High/low, rules-based reasoning.

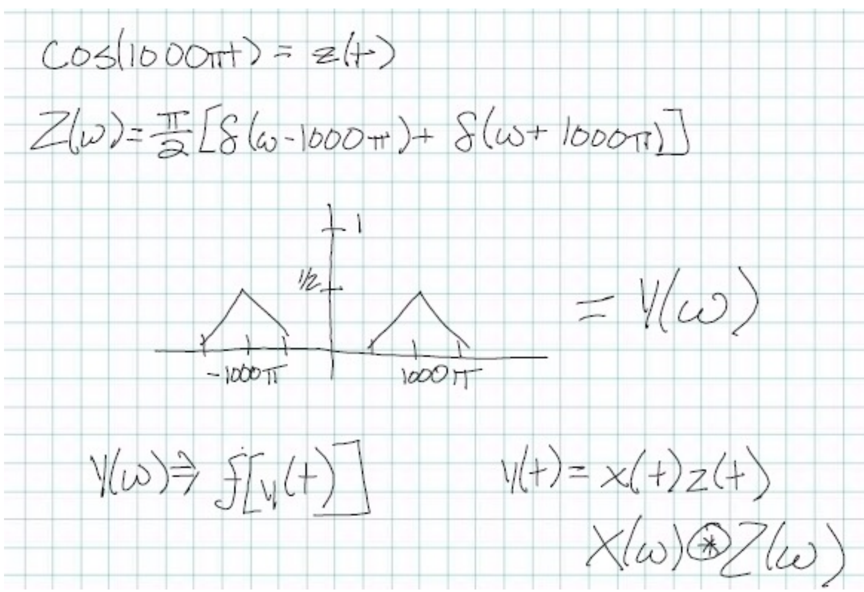


*Figure 3.* Solution of student A to problems #10.

Coding and analysis of the interview for Student A, problem #11 (Figure 4)

Summary:

The student started recalling the equations used in the previous problem. The student wrote the equation for the convolution *Y(ω)* with the cosine of the Fourier transform $cos(\omega_o t)= \pi[\delta(\omega-\omega_o) + \delta(\omega+\omega_o)]$ that it is the same of the previous problem (referring problem #10). Then, the student identified the outsides of the graph and stating that the right side was $1000\pi$ and the left side of the Y axis the same value of *1000π*. After that, the student added the two parts to get the answer of the problem. The student realized that they had to include the amplitude of ½. The student said that the most difficult part was to just identify that they could do simple Fourier transform of the cosine and then just shift to the left and right, and the easiest part was to identify that this problem is just a shift left and right the graph.

Stored knowledge:

- Multiplying a function *Y(ω)* by a cosine shifts the Fourier transform of the function to left and right by $\omega_o$.
- The Fourier transform for a cosine function with frequency $\omega_o$ is two delta functions, one centered at $-\omega_o$ and one centered at $\omega_o$ both with amplitude ½.

New instance representation:

- Noticed explicit features and numeric values stated in the problem.

Action:

- Applied stored knowledge about Fourier transform for cosine function.
- Applied stored knowledge about sketching functions and identifying the frequency shifts.

Strict matching:

- Recognized this problem as a particular case of multiplying a function by a cosine.
- Determined specific numerical values needed to apply frequency shifting.
- No need to discard any extraneous information.

Abstractness:

- High/low, rules-based reasoning.

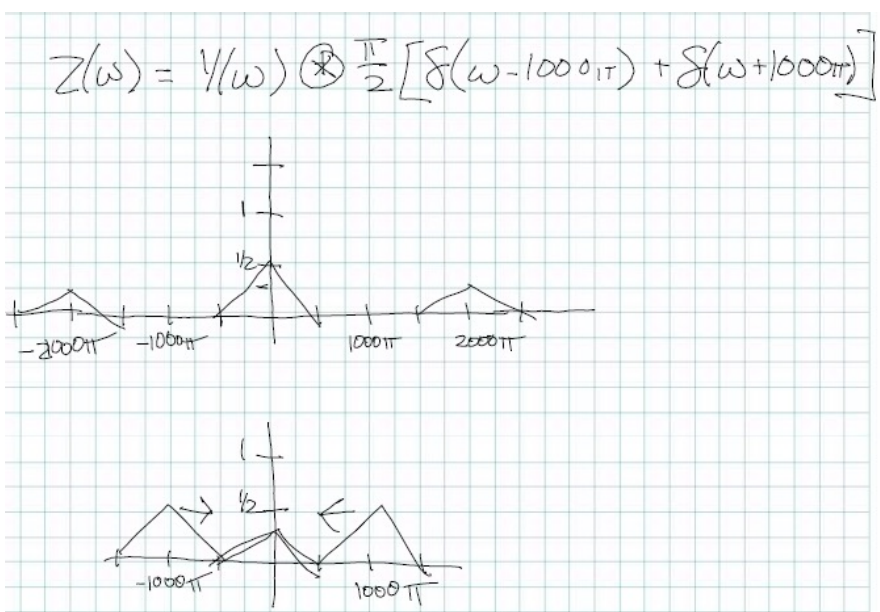

*Figure 4.* Solution of student A to problems #11.

This example of the data analysis used in the study demonstrates how the investigators were able to identify memory-bank and rules-based reasoning in one student's problem solving episodes. Similar analyses were done for all of the students in the study. The sample size that has been analyzed to this point is too small to draw conclusions, but it is worth noting that the majority of students used memory-bank reasoning to solve these problems and that prototype reasoning was never used.

## 5 Conclusions

In this paper, we demonstrated how the representation mapping framework of Hahn and Chater [41] can be used to characterize the different types of reasoning used by students as they solve problems in a junior level signal and systems course. The representation mapping framework takes into the degree of abstractness of the knowledge representations of the student and also whether the student is abstracting using a partial matching or strict matching cognitive process. The study is ongoing with data being collected in three electrical engineering courses, Circuit I, Circuits II and Signals and Systems. We believe the Hahn and Chater framework provides important insights into the problem solving of students. The next phase of the study is to attempt to determine the cognitive demand of the problems being posed in these courses using Bloom's taxonomy or some other appropriate framework. By combining the analyses of the students' problem solving attempts with the analyses of the problems cognitive demand, we hope to be able test the hypothesis about the existence of an abstraction threshold in the undergraduate electrical engineering curriculum.

## References


1. Jonassen, D.H. (2000). Toward a Design Theory of Problem Solving. *Educational Technology: Research & Development,* 48 (4), 63-85.
2. Dym, C. & Little, P. (2009). Engineering design: a project based approach. *3rd Ed, New York*: John Wiley & Sons.
3. Reitman, W. (1965). Cognition and Thought. *Wiley*, New York, NY.
4. Simon, H. (1973). The structure of ill-structure problems. *Artificial Intelligence*, 4, 145-180.
5. Inhelder B. & Piaget, J. (1958). The Growth of Logical Thinking from Childhood to Adolescence. *Basic: New York*.
6. Domin, D. & Bodner, G. (2012). Using students' representations constructed during problem solving to infer conceptual understanding. *Journal of Chemistry Education*, 89(7), 837-843.
7. Stein, M.K., Smith, M.S., Henningsen M.A. & Silver, E.A. (2000). Implementing Standards-based Mathematics Instruction: a Casebook for Professional Development. *Teacher College Press, Columbia University, New York.*
8. Bloom, B.S., Engelhart. M.D., Furst, E.J., Hill, W.H. & Krathwohl, D.R. (1956) Taxonomy of Educational Objectives: The Classification of Educational Goals; Handbook I: Cognitive Domain. *Longmans Green, New York.*
9. Woods, P.K. (1983). Inquiring systems and problems structures: implications for cognitive development. *Human Development*, 26, 249-265.
10. Krathwohl, D.R. (2002). A revision of Bloom's taxonomy: an overview. *Theory into Practice*, Vol. 41, 4. College of Education. The Ohio State University.

11. Clark, A.C. & Ernst, J.V. (2010). Engineering and technical graphics education; using the revised Bloom's taxonomy. *Journal for Geometry and Graphics*, Vol. 4, No. 2, 217-226.
12. Ferguson, C. (2002). Using the revised taxonomy to plan and deliver team-taught, integrated, thematic units. *Theory into Practice*, Vol. 41, 4. College of Education. The Ohio State University.
13. Huitt, W. (2011). Bloom et al.'s taxonomy of the cognitive domain. *Educational Psychology Interactive*. Valdosta, GA: Valdosta State University. Retrieved from http://www.edpsycinteractive.org/topics/cognition/bloom.html
14. Writing objectives using Bloom's taxonomy. The Center for Teaching and Learning, UNC-Charlotte. (April, 2016). Retrieve from: http://teaching.uncc.edu/learning-resources/articles-books/best-practice/goals-objectives/writing-objectives l
15. Jonassen, D.H. (2000). Toward a Design Theory of Problem Solving. *Educational Technology: Research & Development,* 48 (4), 63-85.
16. Wood, P.K. (1983). Inquiring systems and problem structure: implications for cognitive development. *Human Development,* 26, 249-265.
17. Kitchner, K.S. (1983). Cognition, metacognition, and epistemic cognition: a three-level model of cognitive processing. *Human Development,* 26, 222-232.
18. Meacham, J.A. & Emont, N.C. (1989). The interpersonal basis of everyday problem solving. In J.D. Sinnott (Ed.) *Everyday problem solving: theory and applications* (pp. 7-23). New York: Praeger.
19. Frederiksen, N. (1984). Implications of cognitive theory for instruction in problem solving. *Review of education research*, Vol. 54, No.3, 363-407.
20. Kumsaikaew, P., Jackman, J. and Dark, V. J. (2006), Task Relevant Information in Engineering Problem Solving. *Journal of Engineering Education*, 95: 227–239.
21. Jonassen, D., Strobel, J. & Lee, C. B. (2006). Everyday Problem Solving in Engineering: Lessons for Engineering Educators. *Journal of Engineering Education*, 95: 139–151.
22. Atman, C.J., Adams, R.S., Cardella, M.E., Turns, J., Mosborg, S, & Saleem, J. (2007). Engineering design processes: a comparison of students and expert practitioners. *Journal of Engineering Education*, vol. 96, issue 4, 359-379.
23. Gainsburg, J. (2015). Engineering Students' Epistemological Views on Mathematical Methods in Engineering. *Journal of Engineering Education*, 104: 139–166.
24. Mentzer, N., Becker, K. & Sutton, M. (2015). Engineering Design Thinking: High School Students' Performance and Knowledge. *Journal of Engineering Education*, 104: 417–432.
25. Lawanto, O. & Johnson, S.D. (2012). Metacognition in an engineering design project. *International Journal of Engineering Education*, 28(1), 92-102.
26. Wenkat, P.C. (1999). Reflective analysis of student learning in a sophomore engineering course. *Journal of Engineering Education,* 88 (2), 195-203.
27. Litzinger T.A., Van Meter P., Firetto C.M., Passmore L.J., Masters C.B., Turns S.R., Gray G.L., Costanzo F. & Zappe, S.E. (2010). A cognitive study of problem solving in statics, *Journal of Engineering Education,* 99(4), 337-352.
28. Henry, H.R. (2011). Book review: learning to solve problems: a handbook for designing problem-solving learning environments. *Interdisciplinary Journal of Problem-Based Learning*, 5(2).
29. Pearce, J. (2008). "The Education of E.F. Schumacher". God Spy. Retrieve from: http://oldarchive.godspy.com/issues/The-Education-and-Catholic-Conversion-of-E-F-Schumacher-by-Joseph-Pearce.cfm.html
30. Pepler, D.J. & Ross, H.S. (1981). The effect of play on convergent and divergent problem solving. *Child Development,* Vol.52, No.4, 1202-1210.
31. Polya, G. (1954). How to solve it: a new aspect of mathematical method. *Second edition, Princeton University Press,* Princeton, New Jersey.
32. Gray, G.L., Costanzo, F. & Plesha, M.E. (2005). Problem solving in statics and dynamics: a proposal for a structured approach. *Proceedings of the 2005 American Society of Engineering Education Annual Conference & Exposition, Portland, Oregon.*
33. Litzinger, T.A., Van Meter, P., Firetto, C.M., Passmore, L.J., Masters, C.B., Turns, S.R., Gray, G.L., Costanzo F. & Zappe, S.E. (2010). A cognitive study of problem solving in statics, *Journal of Engineering Education,* 99(4), 337-352.
34. Lawanto, O., Butler, D., Cartier, S., Santoso, H., Goodridge, W. & Lawanto, K. (2013). Pattern of task interpretation and self-regulated learning strategies of high school students and college freshmen during and engineering design project. *Journal of STEM education,* Vol. 14, issue 4.

35. Khandani, S. (2005). Engineering Design Process: Education Transfer Plan. Retrieved from http://www.saylor.org/site/wpcontent/uploads/2012/09/ME1014.1EngineeringDesignProcess.pdf
36. Pappas, E. (2002). Creative problem solving in engineering design. *Proceedings of the 2002 American Society for Engineering Education (ASEE), Southeastern section meeting, University of Florida, Gainesville, Fl.*
37. Dougherty, B. C. & Fantaske, P. (1996). Defining expectations for problem-solving skills. *New Directions for Higher Education*, 1996: 55–66.
38. Moreno, R., Reisslein, M. & Ozogul, G. (2009). Optimizing Worked-Example Instruction in Electrical Engineering: The Role of Fading and Feedback during Problem-Solving Practice. *Journal of Engineering Education*, 98: 83–92.
39. Shokouhi, S.V. (2012). An overview of case-based reasoning applications in drilling engineering. *Journal Artificial Intelligence*, Vol. 41, Issue 3, 317-329.
40. Kamble, S.K. & Tembe, B.L. (2013). The effect of use of concept maps on problem solving performance and attitude in mechanical engineering course. Social and Behavioral Science, 83, 748-754
41. Hahn, U. & Chater, N. (1998). Similarity and rules: Distinct? Exhaustive? Empirically distinguishable? *Cognition* 65, 197-203.
42. Pavelich, M.J. (1984). Integrating Piaget's principles of intellectual growth into the engineering classroom. *Proceedings 1984 ASEE Annual Conference, ASEE*, Washington, D.C., 1984, 719-722.
43. Jiang B., Xu X., Garcia A. & Lewis J.E. (2010). Comparing two tests of formal reasoning in a college chemistry context. *Journal of Chemistry Education,* 87(12), 1430-1437.
44. Cavallo, A.M.L. (1996). Meaningful learning, reasoning ability and students' understanding and problem solving of genetics topics. *Journal of Research Science Teach,* 33(6), 625-656.
45. Niaz, M. (1996). Reasoning strategies of students in solving chemistry problems as a function of development level, functional M-capacity and disembedding ability. *International Journal in Science Education,* 18(5), 525-541.
46. Gall, M. D., Gall, J. P. & Borg, W. R. (2007). Educational research, an introduction. (8th ed.) *Boston: Ally and Bacon Pearson Educational.*