Teaching-learning strategies for ill-structured problem solving in engineering

A First Annual Progress Report submitted in partial fulfillment of
the requirements for the course of

Doctor of Philosophy

by

Aditi Kothiyal
(Roll No. 13438001)

Under the guidance of
Prof. Sahana Murthy

IDP in Educational Technology
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY

August 20, 2014

Contents

List of Figures

List of Tables

Chapter 1

Overview of the past year

1.1 Introduction

One of the most important roles of a practising engineer is that of a problem solver [1]. This is corroborated by ABET which defines "an ability to identify, formulate and solve engineering problems" as one of learning outcomes required of an engineering graduate [2] and the Washington accord [3] which defines the outcome in two parts as,

1. "Define, investigate and analyze complex problems" which means to "Identify, formulate, research literature and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences."

2. "Design or develop solutions to complex problems" which means to "Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations."

Problem solving (PS) is also defined by Beyer [4] as a thinking skill and it is an essential 21st century skill and workplace competency. Thus it is important for engineering students to learn problem solving in order to be a better engineering practitioner [1].

An engineering workplace problem is ill-structured (IS) [1], i.e., it has vaguely defined or unclear goals and unstated constraints, possesses multiple solutions and solution paths and multiple criteria for evaluation [5]. These include problems like design, diagnosis-solution, decision-making, etc. It has been shown that the ability to solve the well-structured (textbook) problems does not translate to engineering problems such as engineering estimation [6]. Therefore it is important for engineering students to be explicitly trained in the skill of engineering problem solving.

With this broad goal, I did the following work over the past 16 months,

- An extensive literature survey on IS PS, with emphasis on engineering problem solving and engineering estimation problems. This led to the writing of a technical report [7] and is described in chapter 3.
- Two field studies on IS PS in engineering classrooms, with the goals of studying
  - The role of delayed guidance on student learning of IS PS - A semester long quasi-experimental study with three experiments and a follow-up grounded theory study of student perceptions described in chapter 4.
  - The role of question prompts and peer interactions on student learning of IS PS - A semester long pre-experimental study with 3 interventions and a survey of student perceptions described in chapter 5.
- Two projects that emerged from course work, one on the development of the thinking skill of representational competence and the other on the active learning strategy of think-pair-share. These are described in chapter 6.

1.2 Timeline of Work Done

The work done in the broad research area of teaching-learning strategies for ill-structured (IS) problem solving (PS) in engineering and other research projects is summarized in Table 1.1.

Table 1.1: Timeline of work done between April 2013 and August 2014

Time period	Work done
April-July 2013	▪ Presented IDP-ET TR: "Learning to estimate: Strategies for solving engineering estimation problems" ▪ Wrote a paper (ICER 2013) as part of TPS in CS101: "Effect of Think-Pair-Share in a large CS1 class: 83% sustained engagement" (with Rwitajit Majumdar, Sridhar Iyer, Sahana Murthy) ▪ Planned experimental study in EE 746 - Neuromorphic engineering: "Delayed Guidance as a teaching-learning strategy for ill-structured problem solving" (with Bipin Rajendran) ▪ Wrote a paper (T4E2013): "PULSE: A Framework for Protocol based Utility to Log Student Engagement" (with Rwitajit Majumdar)
August-October 2013	▪ Conducted two experiments in EE 746. Analysed the data and revised the study based on this. Planned and conducted third experiment. ▪ Wrote a paper (ITiCSE 2014) as part of TPS in CS101:"Think-pair-share in a large CS1 class: does learning really happen?" (with Sahana Murthy, Sridhar Iyer)
November-December 2013	▪ Did detailed statistical and qualitative analyses of experiment three data. ▪ Presented credit seminar based on this work.

January-February 2014	■ Planned grounded theory study of student perceptions of learning with Delayed Guidance. ■ Planned and began study in EE590 - Foundations of Projects as secondary researcher/mentor:"Guided problem solving and group programming" (with Abhinav Anand and Bipin Rajendran)
March-April 2014	■ Conducted interviews for the grounded theory study, began transcription and analysis ■ Concluded EE 590 study and data analysis. ■ Began project: "Development of representational competence using an enactive computer interface" (with Rwitajit Majumdar, Prajakt Pandey, Harshit Agarwal, Ajit Ranka and Sanjay Chandrasekharan)
May-July 2014	■ Submitted a paper to T4E2014: "Guided Problem Solving and Group Programming: A Technology-Enhanced Teaching-Learning Strategy for Engineering Problem Solving." (with Abhinav Anand, Bipin Rajendran and Sahana Murthy) ■ Completed interface development, conducted a pilot study and submitted a paper to T4E2014: "The enactive equation: exploring how multiple external representations are integrated, using a fully controllable interface and eye-tracking" (with Rwitajit Majumdar, Prajakt Pandey, Harshit Agarwal, Ajit Ranka, Sanjay Chandrasekharan and Sahana Murthy) ■ Began closure on project TPS in CS101: planning a journal paper (with Shitanshu Mishra, Rwitajit Majumdar, Sridhar Iyer and Sahana Murthy)

1.3 Future Work

Based on the work done in the past 16 months, I hope to situate IS PS ability as a thinking skill and develop technology enhanced learning environments for the development of this ability. My domain will be electrical engineering. However as there are several types of IS problems in electrical engineering, I am considering the following two options to scope down my problem,

- Technology enhanced learning environments for the development of estimation skills among third and fourth year electrical engineering students.
- Technology enhanced learning environments for the development of abstraction skills as the first step of IS PS in electrical engineering for third and year students.

Further literature review and learning from previous IS PS studies will help me in scoping this problem in the next 6 months. This will be followed by design and development of the learning environment.

Chapter 2

Motivation

Engineering estimation is defined as "an analysis to determine all quantities to some level of specificity" [6]. It can also be thought of as "making decisions or selecting from a multitude of options based on incomplete or unavailable details or data" [8]. An equivalent definition for measurement estimation is, "Measurement estimation can be thought of physical measurement without tools" [9]. For the purposes of this report, we similarly define engineering estimation as "the process of determining approximate values for a physical quantity without the tools of complete information and knowledge and computing resources."

Usiskin claims (as quoted by the authors in [9]) "that although estimation is considered the "weak sister" to mathematical computation, in real life it is more frequently the "stronger sister" or the "only child". Similarly in engineering analysis, often a rough estimate is not only acceptable, but the only option; it provides useful information about a problem or a design in situations when attempting to calculate accurate values via a detailed analysis is unnecessary or too complicated because of a lack of time, information and/or resources [6], [8], [10]. Methods of approximate reasoning are also well-accepted as a way to rapidly make decisions in the problem-solving and design processes [10]. Finally, as Mahajan [11] puts it, estimation is a process by which we learn to "lower our standards" and create models of the world that we can work with using the conceptual and procedural understanding that we already have. It allows us to, for example, solve complex physical problems by relaxing our demands on accuracy, creating a "coarse" model and attempting to understand a situation that we would otherwise have no intuition for. This becomes especially important at the start of the engineering design process [10].

In the light of all these reasons, we conclude that it is important for engineering students to be trained to perform estimation at the undergraduate level. Yet research [6] studying the performance of expert and novice engineers demonstrates that current engineering curricula are not teaching our students to make good estimates. In [6], Linder compared the performance of practicing engineers and MIT seniors on estimation problems and asked them to estimate quantities like drag force and energy. He found that experts always had the right order of magnitude estimate and their responses varied over few orders of magnitude only. While there were some novices who hit the right order of magnitude, most students gave estimates that varied over 8-10 orders of magnitude. This indicates that students have very little intuition for what the estimate should 'look' like. Thus there is a need for developing the ability to estimate as an explicit skill amongst our student engineers [6], [8], [10] and [11].

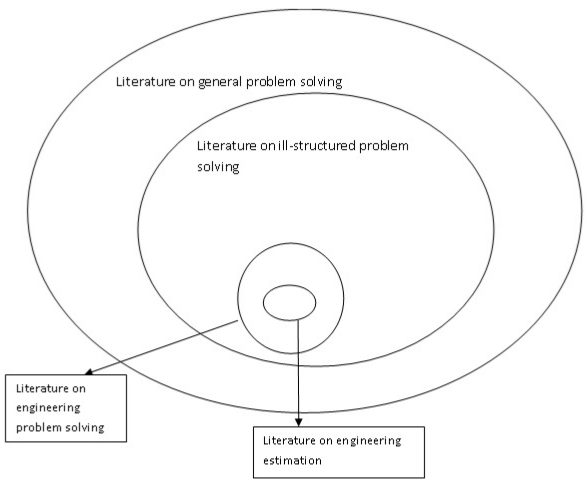
A survey of literature to identify relevant teaching-learning strategies for engineering estimation showed that while there exist several instructional strategies for measurement estimation (which is a subset of engineering estimation) and are reviewed in [9], the only formal effort to teach students estimation comes from Sanjoy Mahajan [11] at MIT who teaches a semester long course titled "Approximate methods in science and engineering" in which he teaches students several estimation tools designed by him. In addition [6], [8] and [10] offer guidelines and suggest activities to teach estimation that can be incorporated into the engineering curriculum. However these strategies are unsubstantiated by learning theories or educational research into their effectiveness for learning. This motivated me to choose the problem of designing teaching-learning tools and strategies for engineering estimation.

Chapter 3

Literature Review

In order to design teaching-learning tools and strategies for engineering estimation grounded in learning theories and evaluate its effectiveness, I reviewed literature from the larger world of engineering problems. The goal was to use results from PS research and instructional design strategies from the larger domain to design estimation PS instruction. This idea also guided my subsequent foray into IS and general PS, which have been studied extensively by researchers in the fields of education, cognitive science and psychology for the past six decades. Therefore there exist several effective teaching-learning tools and strategies for PS. My literature review endeavoured to identify potential features from these tools and strategies that can be adapted in our design of a learning environment for engineering estimation. Figure 3.1 shows the scope of my literature review.

Figure 3.1: Scope of literature review



A detailed literature review was written as an IDP-ET Technical report and is available for download at `http://www.et.iitb.ac.in/~aditi/lib/exe/fetch.php?media=problemsolvingtr.pdf`.

Chapter 4

The role of delayed guidance in ill-structured problem solving

4.1 Motivation

Following the technical report, I found literature relating to the self-learning behaviours of students as they interacted with a learning object targeting the ability of structuring an open problem for design [12]. The authors observed that

> Student A ... (post-test score = 2.5/ 12) ... In Concept Clarification Questions(CCQ) and Decision Making Task Question(DMTQ) activities, student A clicked the correct answer and proceeded to the next activity without reading feedback.

while

> Student B ... (post-test score =11/ 12) ... proceeded to next DMTQ for which she selected the wrong answer, read the feedback and attempted the question again. This time she selected the correct answer. For the third DMTQ she selected the correct answer and then read the feedback for all answers,

This showed that initial "failure" can be productive in the learning of engineering design competency of structuring an open problem, when it is followed by appropriate feedback to the student after the failure has occured. In other words, it is useful for students to not be guided or scaffolded from the start.

Delay of guidance to students has already been established as an effective strategy to improve students conceptual understanding [13], [14], [15], [16]. The premise of these instructional strategies is that allowing students to explore the domain via case studies, well or ill-structured problems or invention activities without direct instruction at the beginning, provides an opportunity to prime and differentiate their prior or implicit knowledge on the subject which can be built upon in the following form of structure/feedback/instruction. Further in the case of productive failure (PF) [14], solving IS problems also enables learners to generate, explore and refine several representations and solution methods for IS problems. This is relevant to my research problem because engineering estimation is a type of IS problem as mentioned in [7]. Therefore in my pilot study, I decided to investigate the role of PF in the learning of IS PS. Specifically I was interested in exploring if failure is useful in the first stage of IS PS, namely problem representation, as had been shown for the case of design problems in [12], which are a category of IS problems as described in [7].

4.2 Intervention Strategy

The interventions for the experimental and control group were adapted from [14] and are shown in Table 4.1.

Table 4.1: The Interventions

Experimental- DG	Control- DI
IS group PS- complex IS problem: 30 -40 minutes	Telling- providing scaffolds for IS PS process: 15-20 minutes
Class discussion of student solutions: 15-20 minutes	Practice (simple) problem- Individual : 20 minutes
Telling- providing scaffolds for IS PS process: 15-20 minutes	IS group PS- complex IS problem: 30 -40 minutes
Practice (simple) problem- Individual: 20 minutes	Discussion of student solutions: 15-20 minutes
Post test- Individual (two parts): 2.5 hours (4 days later)	Post test- Individual (two parts): 2.5 hours (4 days later)

The various stages of the interventions for the two groups were matched in content and time duration; only the order in which they occurred changed. An example of the instructional material (developed in association with the instructor for experiment 3) is shown in Appendix A. In addition to this material, the instructor also provided students with a `MATLAB` program to test and optimize their design of neural network so that fluency with `MATLAB` programming was not a factor in student performance.

4.3 Methods

4.3.1 Research Question

The broad research goal was to study the effect of delayed guidance (DG) on student IS PS skill. Our specific research questions were:

1. Does DG intervention lead to improved IS PS performance?
   (a) To what extent does DG affect the IS PS performance?
   (b) To what extent does DG affect the development of IS PS competencies?
2. What are student perceptions of learning through DG?
3. What are the differences between the strategies, heuristics etc that students who learn through DG use vs. those who learn via DI use?

4.3.2 Experiment Design

In order to answer RQ1, we chose a quasi-experimental design with two matched groups and random assignment to experimental and control groups. Our interventions were direct instruction (DI) for the control group and DG for the experimental group. RQ2 was answered via a survey. RQ3 will be answered via a grounded theory analysis of student interviews (future work).

4.3.3 Sample

Our sample was based on convenience and consisted of 25 students (16 experimental, 9 control) in the course EE746 - Neuromorphic Engineering. The students were a mix of undergraduates and graduates.

4.3.4 Procedure

- A survey was done at the start of the course to gauge students prior knowledge in the foundational concepts needed for the course, namely basic probability, their motivation levels and general PS skill. The students were then divided into experimental and control groups on the basis of their responses such that the two groups were matched for prior knowledge (t-test, $p = 0.572$), motivation (Mann-Whitney, $p = 0.392$), general problem solving skill (Mann-Whitney, $p = 0.192$), distribution of undergraduates and graduates (Mann-Whitney, $p = 0.522$) and CPI (t-test, $p = 0.253$) and remained unchanged throughout the semester. Both groups were taught by the same instructor using different interventions and at different times.
- Three experiments were conducted across the semester. In each intervention a different type of IS problem was used like hypothesis testing, design, etc. Experiments 1 and 2 were treated as training for the students in the learning strategy and the results of experiment 3 are presented here. There

was attrition across the semester and while 48 students participated in the first experiment, the number came down to 33 in the second experiment and 25 in the third (our final sample size).

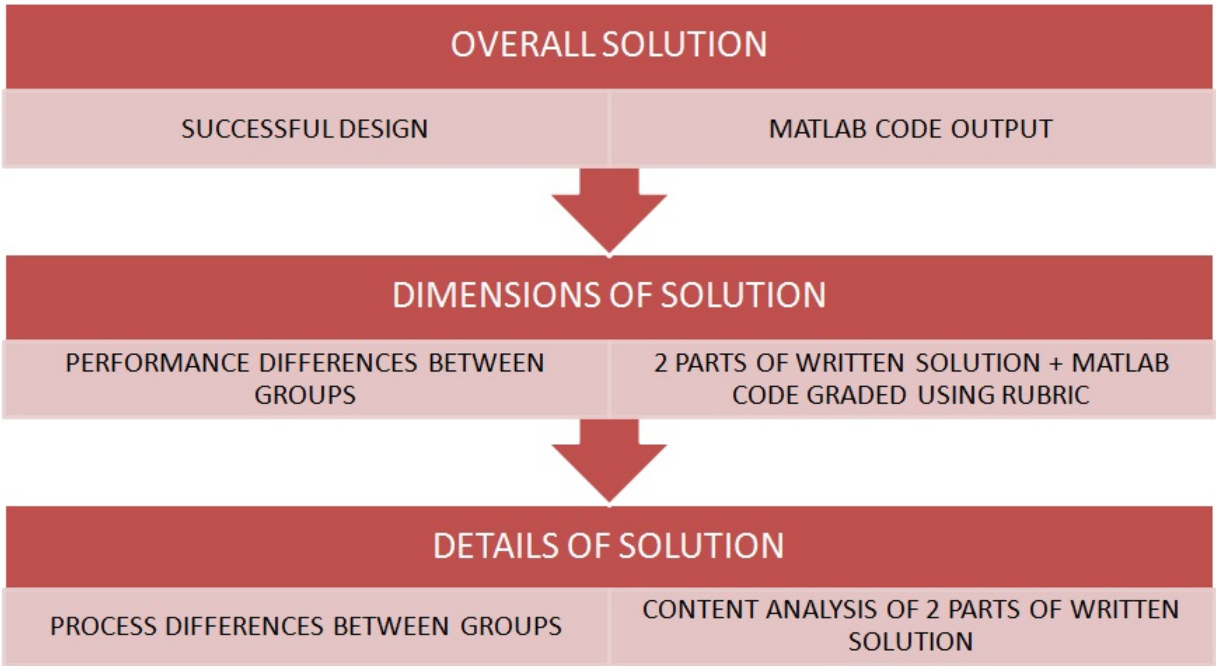
- The interventions were conducted in the tutorial sessions of the course and lasted 90 minutes for each group. The students were divided into groups of three for the intervention, which were mostly unchanged across the semester unless students were not present for the tutorial, in which case new groups were created on the spot. However care was taken to ensure that the group structure was minimally disturbed.
- After the intervention students were asked to submit their class work (both individual and group work) to the instructor via email. This was followed a post-test 4 days later which was individual and lasted 2.5 hours, but students were allowed to use their notes from class and computers for programming.

4.3.5 Data Sources and Instruments

- The survey conducted at the start of the course, testing for prior knowledge in probability, motivation to take the course and general math PS was content validated by three experts and is shown in Appendix A.
- To evaluate student learning after the intervention, a post-test was conducted consisting of an open design problem and guiding prompts, and is shown in Appendix A. The students were initially only given the open design problem to work on. After an hour their work was collected and they were given the same problem with the guiding prompts for two purposes:
  - As students have a tendency to not write much, the questions would get them to write their answer to the design problem in a way which would make it easier to evaluate them.
  - It was hypothesized, that as described in [17] and employed in [16], these guiding prompts would be a dynamic assessment which "can help reveal the hidden value of learning activities that appear inefficient, given traditional assessments of memory and problem solving."

  The students continued to work for 1.5 hours more and were allowed to change their initially proposed design solution when they responded to the guiding prompts.
- Group discussions during the interventions were audio recorded and the researchers kept a log of classroom interactions including teacher-student interactions, group dynamics and student questions.
- Student answer scripts were evaluated using a rubric (shown in Appendix A.4) consisting of 4 constructs for design process as defined by an expert (in this case the instructor) and 3 constructs for other dimensions of IS PS performance [18].
- Student perceptions regarding their own learning of IS PS were gauged using an in-class paper survey consisting of 5 questions on a 5-point Likert scale shown in Appendix A.5.

Figure 4.1: Analysis framework for student post-test scores



4.4 Analysis and Results

4.4.1 Analysis

To answer RQ1, "Does DG treatment lead to improved learning of IS PS?" we performed a three-level analysis of student solutions to the post-test as per the framework defined in Figure 4.1 and the results are described in the next section.

Criteria 1: Overall Solution

The first criteria for IS PS performance is successful design, i.e. a neural network which successfully accomplishes the task it was designed for. This was measured by testing the programs submitted by the students and if the desired sequence was detected correctly, the design was labelled a success (1) or a failure (0). The difference between the distribution of zeros and ones in the two groups was tested using a Mann-Whitney test. This result will be used to answer RQ1.

Criteria 2: Performance difference between groups

Student responses on the post test were graded using the rubric in Appendix A.4. The scores of the students in both groups were statistically compared using a Mann-Whitney test because the data is ordinal. This result will be used to answer RQ1a.

Criteria 3: Process Differences between groups

Content analysis of post-test answer scripts was done to identify the competencies used by the students and compare the two groups in terms of competencies used. This analysis will be used to answer RQ1b.

To perform the content analysis, all 25 solutions were transcribed. The expert solution (in this case the instructors' solution) and literature about IS PS competencies served as the starting point for the coding. The unit of analysis was one idea about the solution, either a piece of text, an equation or a figure or one argument. For example, a diagram of a network was coded as "Construction of representation", as was a matrix; while an argument like "this can be ensured by adjsting the spiking current threshold for 3 and using different synaptic time const for A , C" was coded as "Constructing argument". However some codes were emergent (eg, Reflection) and we had a total of 15 codes shown in Table 4.2.

Table 4.2: Codes used for content analysis.

	Code	Definition (Adapted from literature)
1	Construction of representation	Construct representations for given problem/solution
2	Use of representation	Uses representations to solve problems
3	Make assumptions	Make suitable and valid assumptions while solving problem
4	Justify assumptions	Justify assumptions made in the course of solving the problem.
5	Define problem	Problem clearly and completely stated.
6	Generate subgoals	Specific goals for problem solution is clearly stated.
7	Identify information	Known factors and constraints are identified.
8	Seek needed information	Pieces of needed information discussed.
9	Select and explain solutions	A solution is selected or developed, with explicit explanation on how the solution works. The explanation should include the interrelationship between different components, the inputs, the neurons in the network, the synapses, etc.
10	Constructing argument	Coherent and persuasive reasons are provided to support the proposed solution, and factors or constraints are discussed. Explain why particular inputs were chosen, particular time constants, etc
11	Providing evidence	Evidence to support the argument is strong and relevant. The evidence has been tested, or based on previous experience or real examples.

12	Evaluate solutions	The proposed solution is evaluated, and constraints are discussed, supported with reasoning. A statement is made about the effectiveness or benefits of the solution. The pros and cons of the solution are discussed, supported with relevant evidence (e.g. from the program), as well as how the constraints can be overcome.
13	Revise Solution	The proposed solution is revised based on information obtained from the evaluation performed. The limitations highlighted by the evaluation are overcome by modifying the parameters of the solution appropriately.
14	Assess alternative solutions	Alternative solution is stated, and the viability of the solution(s) is discussed. At least one optional solution is discussed. Reasons are given on why an option is selected over the other(s), with constraints discussed.
15	Reflection	Any comment about the problem or problem solving process

The competencies were grouped into 4 categories for analysis and the frequencies in these 4 categories were compared between the two groups using the Mann-Whitney test, and frequencies pre and post prompts within each group using the Wilcoxon-Signed Rank test.

- Problem representations (PR), Codes 1 and 2
- Construction of problem space (PS), Codes 3-8
- Generating and explaining solutions (GS), Codes 9-11
- Evaluation and monitoring (EM), Codes 12-15

Student Perceptions of learning

To answer RQ2 "What are student perceptions of learning through DG?", the frequencies of student responses to the second and third question on the perceptions survey shown in Appendix A.5 (the questions referring to learning) were determined and compared between groups.

4.4.2 Results

RQ1: Does DG intervention lead to improved IS PS performance?

By this criteria, 1/9 students in the control group were successful and 5/16 students in the experimental group were successful. While the experimental group has performed better than the control group, the result was not statistically significant ($p = 0.4$).

RQ1a: To what extent does DG affect IS PS performance?

The second criteria for IS PS performance is student scores on the post-test (both parts together) graded using the rubric shown in Appendix A.4. The scores of the students in the two groups (descriptive and inferential statistics) are shown in Table 4.3.

Table 4.3: IS PS scores of both groups

Construct (0-3)	Expt group mean (SD)	Ctrl group mean (SD)	p (Mann-Whitney)
Inputs (IP)	2 (1.2)	1.8 (1)	0.64
Time scales (TS)	2.4 (0.8)	1.9 (0.9)	0.19
Network structure (NS)	2.3 (0.8)	2.2 (0.8)	0.98
Adjusting parameters (ADJ)	2.8 (0.8)	1.9 (1.2)	$0.017^{*}(r = 0.45)$
Complete, justified proposed design (PS)	2.6 (0.6)	2 (0.9)	0.095 (r = 0.4)
Complete, working design (FS)	1.4 (1.2)	1.1 (0.8)	0.64
Quality of solution (QoS)	1.6 (1.1)	1.2 (1.1)	0.49

From these results we see that while the experimental group has performed better than the control group on all constructs, the difference reaches significance only on the dimension of "Adjusting parameters" and the effect size is moderate (0.45). The difference also approaches significance on the dimension of "Proposing a justified and complete design" and the effect size is also moderate (0.4). So we zoomed into the construct PS which reflects the competencies "Constructing Argument" and "Providing Evidence" and in order to understand its correlations with various factors affecting IS PS performance. This is shown in Table 4.4.

We observed that performance on construct PS is significantly correlated with prior knowledge, student degree program, problem solving skill and CPI, but not with motivation level of students. We also observed that performance on construct PS is a significant predictor of successful design (Spearmann's $\rho = 0.479$) and this could be the reason for the difference between the performance of the two groups on the criteria of successful design .

RQ1b: To what extent does DG lead to the development of IS PS competences?

1. We first measured the length of the solution in terms of the number of competencies demonstrated in each solution and compared the two groups on this measure. The results are shown in Table 4.5.

   These results show that the experimental group solutions are longer in terms of number of com-

Table 4.4: Predictors of performance on construct PS

Variables	Spearmann's ρ	p
PS vs pre test	0.702	0*
PS vs degree program	-0.648	0*
PS vs ps skill	0.402	0.051*
PS vs CGPA	0.403	0.051*
PS vs motivation	-0.004	0.986
PS vs group	0.388	0.056
PS vs success	0.479	0.015*

Table 4.5: Comparing lengths of solution

	Expt Mean (SD)	Ctrl Mean (SD)	p (Mann-Whitney)
Pre Prompts Mean (SD)	14.7(7.3)	10.9(4.9)	0.187
Post Prompts Mean (SD)	15.3(6.6)	9.1(6.3)	0.02*
p (Wilcoxon)	0.816	0.285	

petencies demonstrated. Further, students in the experimental group showed significantly higher ($p = 0.03$) evidence of usage of IS PS competencies with prompt sheets than students in the control group.

2. Next we determined the frequencies of the first IS PS competency category demonstrated by the students in the control and experimental group, pre and post prompts. These are shown in Table 4.6.

   We see that pre-prompts the first competency category demonstrated by the experimental group is problem space construction while for the control group it is problem representation. Post-prompts for both groups the first competency used is generating solutions.

3. The average frequency of each competency category, pre and post prompts, in both the experimental and control groups are shown in Figure 4.2. The differential and inferential statistics are presented in Table 4.7.

   These results indicate that post-prompts the experimental group shows higher frequency of usage of all competencies than the control group. Though this difference does not approach significance

Table 4.6: Frequency of first competency category

	Expt group		Ctrl group	
Competency category	Pre-prompts (%)	Post-prompts(%)	Pre-prompts (%)	Post-prompts(%)
PR	31.25	12.5	66.67	0
PS	43.75	6.25	11.11	11.11
GS	25	81.25	22.22	88.89
EM	0	0	0	0

Figure 4.2: Average frequency of each competency category

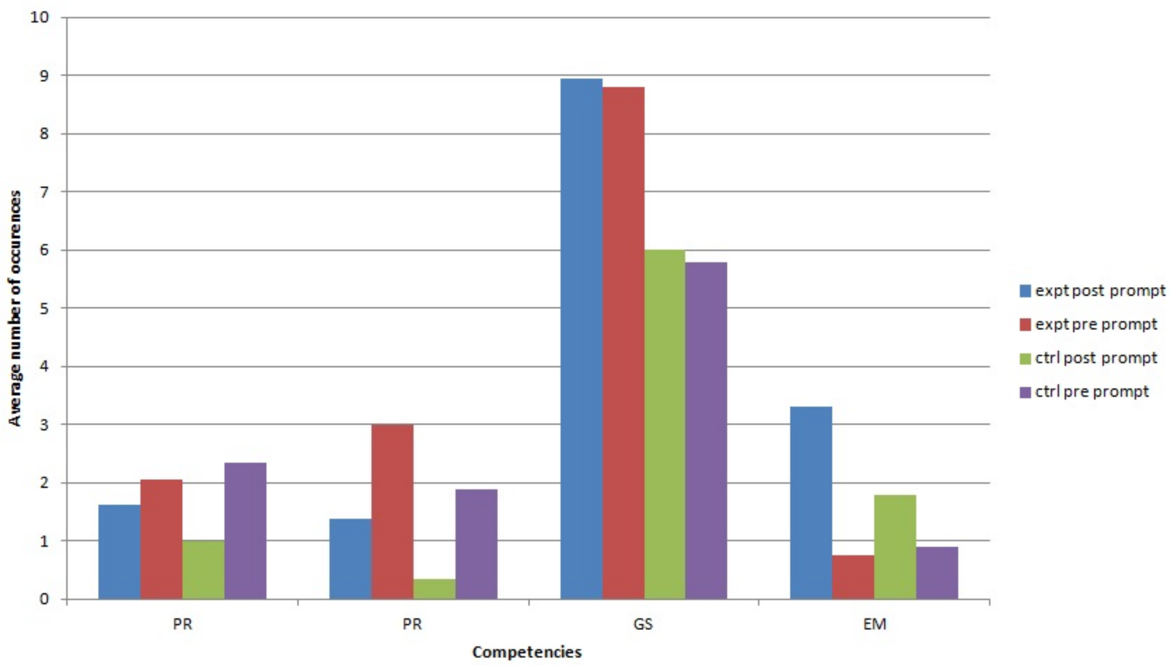
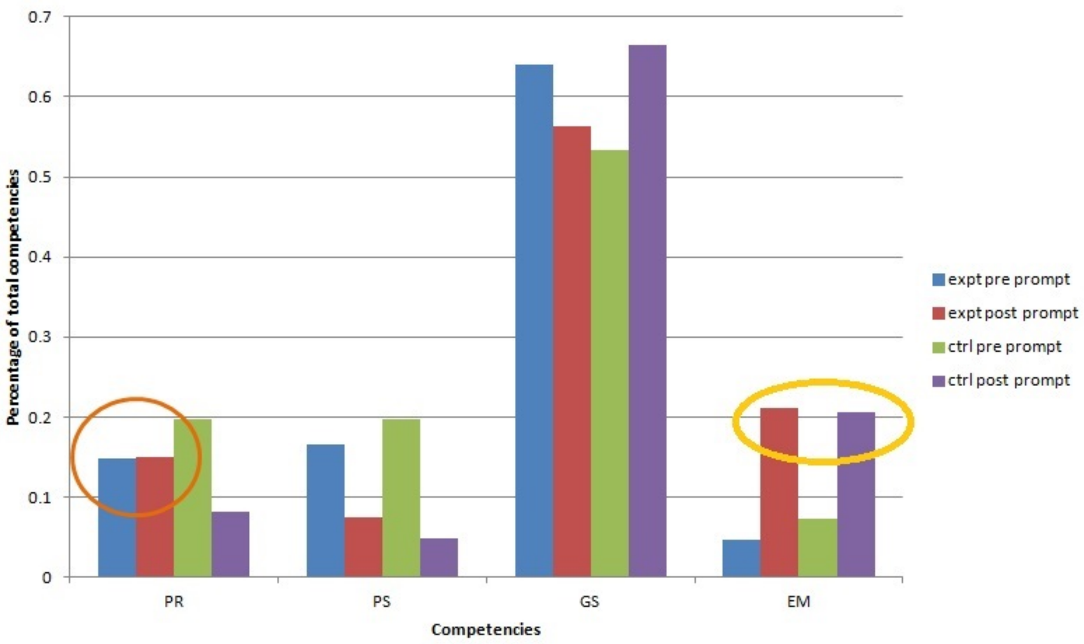


Figure 4.3: Average relative frequency of each competency category



for any category, we see marked differences in all categories. Further both groups show statistically significant improvement in the category of "Problem Space Construction" pre and post prompts, while the experimental group also shows statistically significant improvement in the "Evaluation and Monitoring" category. This indicates that the experimental group were able to use the prompts to improve their solution process and demonstration of IS PS competencies.

4. The average relative frequency of each competency category, pre and post prompts, in both the experimental and control groups are shown in Figure 4.3. The differential and inferential statistics are presented in Table 4.8.

   The data indicates that the experimental group pre and post prompts showed significantly different relative frequencies in several categories: "problem space construction" went down and "evaluation and monitoring" went up. Also, the control group pre and post prompts showed significantly different relative frequencies of "problem space construction" which went down, "problem representation" which also went down and "evaluation and monitoring" which went up. However significantly the experimental group continued to perform the same relative frequency of "problem representation" pre and post prompts.

RQ2: What are student perceptions of learning through DG?

The relative frequency of responses to questions 2 and 3 is shown in Figures 4.4 and 4.5. These show that while a majority of students agreed that the tutorials and quiz helped them learn IS PS, the spread was more for students in the experimental group than the control group. Thus a small fraction of the students in the experimental group perceived that they were not learning with the DG strategy.

Table 4.7: Frequencies of each competency category: Differential and Inferential Statistics

	Expt Mean (SD)			Ctrl Mean(SD)			p (Mann-Whitney)	
	Pre-Prompts	Post-Prompts	p (Wilcoxon)	Pre-Prompts	Post-Prompts	p (Wilcoxon)	Pre-Prompts	Post-Prompts
PR	2 (2)	1.6 (1.7)	0.492	2.3 (1.9)	1 (1.7)	0.094	0.665	0.139
PS	3 (4.1)	1.4 (2.5)	0.009*	1.9 (1.5)	0.3 (0.5)	0.047*	0.542	0.161
GS	8.8 (3.9)	8.9 (4.7)	0.89	5.8 (3.2)	6 (3.7)	0.91	0.0532	0.111
EM	0.75 (1.1)	3.3 (3.8)	0.003*	0.9 (1.2)	1.8 (1.5)	0.148	0.8262	0.4876

Table 4.8: Relative frequencies of each competency category: Differential and Inferential Statistics

	Expt Mean (SD)			Ctrl Mean(SD)			p (Mann-Whitney)	
	Pre-Prompts	Post-Prompts	p (Wilcoxon)	Pre-Prompts	Post-Prompts	p (Wilcoxon)	Pre-Prompts	Post-Prompts
PR	0.15 (0.15)	0.15 (0.24)	0.733	0.2 (0.13)	0.08 (0.11)	0.017*	0.36	0.42
PS	0.17 (0.17)	0.07 (0.11)	0.033*	0.2 (0.17)	0.05 (0.09)	0.063	0.64	0.49
GS	0.64 (0.16)	0.56 (0.23)	0.5	0.53 (0.14)	0.67 (0.16)	0.123	0.15	0.49
EM	0.047 (0.62)	0.21 (0.20)	0.004*	0.07 (0.09)	0.21 (0.1)	0.025*	0.6	0.68

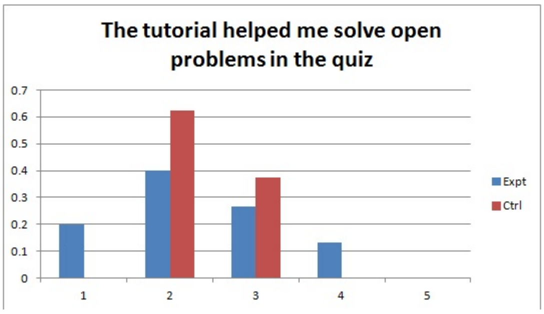


Figure 4.4: Student responses to Question 2 on the perceptions survey

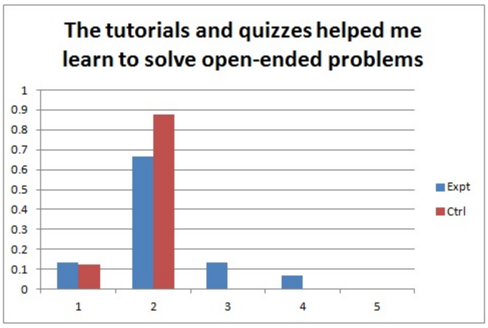


Figure 4.5: Student resonses to Question 3 on the perceptions survey

4.5 Discussion

Our results are summarized in Table 4.9.

While some of these reiterate previous results found in literature regarding the role of delayed guidance on student IS PS, an interesting result that emerged in our research was that students in the DG group began PS by structuring the problem space while those in the DI intervention began PS by constructing a representation. It is desirable that students begin by reframing the problem as mentioned in literature [7]. In this sense, DG was successful because, as hypothesized, it brings the issue of problem framing to the fore for the student.

The results are not generalizable owing to our small sample size, attrition across the semester and the confounding variables of affective abilities, metacognition and epistemic cognition which we could not bracket out. Nonetheless, this study offers several insights on how delayed guidance works in the teaching-learning of IS PS and opens several avenues for further work, some of which are listed below.

- Further analysis of post-test solutions, student observation logs, classroom interactions and in class solutions to correlate in class work to post test performance.
- See what in the intervention worked and what didn't.
- Answer RQ3, "What are the differences between the strategies, heuristics etc that students who learn through DG use vs. those who learn via DI use?"
- Use all the above to extract guidelines for design of new teaching-learning strategies for IS PS. One such tentative guideline is having students engage in authentic, real world problem solving tasks in order for them to learn problem space construction.

Table 4.9: Summary of results

Students in the experimental group:	What in the intervention *might* have caused this?
Choose parameters, propose solutions and justify them better (based on ISPS rubric).	▪ attention to critical features ▪ engage in discussions to explain and refute (corroborated by [14])
Use more ISPS competencies.	students tried multiple representation and solution methods for a given problem during the DG intervention
Started with structuring the problem space while those in control group started with problem representation.	students try to define and understand the problem by themselves during the DG intervention
Do as much problem representation post prompts as pre prompts	students tried multiple representation and solution methods for a given problem.

4.6 Contributions

- We plan to write two conference papers based on this study after which this study will be closed. The first paper consisting of results above will be submitted to LATiCE 2015 and the second paper answering RQ3 using the grounded theory study, will be sent to EpiSTEME 2015.
- This teaching-learning strategy of DG to teach IS PS is being currently employed in the course EE746 in the tutorial sessions of the course.

Chapter 5

Guided problem solving and group programming

5.1 Motivation

EE590 (Foundations of Projects) is an Electrical Engineering course at IITB meant to train senior undergraduate students hoping to do their senior theses in the area of Microelectronics in the numerical methods, techniques and programming skills necessary to carry out their research. The broad teaching-learning goals of the course are to improve students engineering (IS) PS ability and the specific goals as described by the instructor include,

1. Apply theory in real life problem to solve it, use concepts that they already know.
2. Use a known technique to set up a simulation for the real-life problem. Make decisions regarding how to do this. Make non-trivial assumptions.
3. Compare two techniques, do pros and cons analysis, make decisions.
4. Change the parameters of the simulation to improve the solutions, take decisions and optimize.
5. Improve programming efficiency.

5.2 Overview of the study

Guided by literature reviewed in [7] on the role of the dual scaffolds of question prompts and peer interactions in IS PS instruction, we developed a teaching-learning strategy for these goals in conjunction with the instructor and another research scholar at ET-IITB. The additional aspect in this strategy that differentiated it from other interventions was that the scaffolding extended into the programming part of the solution development traditionally relegated to homework. Thus this was a technology-enabled teaching-learning strategy as the availability of student laptops in the classroom made it possible to extend the scaffolds to computer programming in the classroom.

We evaluated the strategy named "Guided Problem Solving and Group programming (GPGP)" using a pre-experimental, single group pre-post test study (in the interest of fairness) and explored student perceptions. The details of this study and the results are available in a paper submitted to T4E2014 and available at `http://www.et.iitb.ac.in/~aditi/lib/exe/fetch.php?media=problemsolvinggpgp.pdf`. Briefly, results show that students IS PS ability improved and we learned that students perceived benefits of the scaffolds such as access to multiple solutions, new ideas, ability to evaluate solutions, perform optimization, select parameters and improve understanding. These benefits of the scaffolds will be considered in the future when we design instructional strategies for IS PS.

5.3 Contributions

Our contributions include,

1. Developed a research-based teaching-learning strategy for ill-structured problem solving ability for the course EE590.
2. Submitted a paper to T4E2014 based on this work.
3. Materials regarding how to conduct the GPGP strategy and activity constructor will be made available from the ET website soon.
4. Mentored a first year ET research scholar in the process.

Chapter 6

Other projects

6.1 Development of representational competence thinking skill

6.1.1 Overview

Representational competence (RC) refers to "the ability to simultaneously process and integrate multiple external representations (MERs) in that domain" [19]. While MERs are used extensively in science and engineering for concept, phenomena and situation understanding and problem solving by experts, it is well-known that students have difficulties processing and generating MERs, leading to difficulties in learning. Computer based visualizations have been used to alleviate student learning difficulties and improve their RC. However literature offers no consensus on how MERs should be presented on the interface in order for RC to develop, i.e. integration of representations to occur. The reason for this could be the emphasis on working memory load reduction in the design of computer interfaces based on information processing theories of cognition.

In this project we approached the RC development problem using emerging theories of cognition such as distributed and embodied/enactive cognition [20], [21], [22], [23] which propose that MERs have a greater role than working memory reduction alone. Computer interfaces must therefore seek to exploit all these roles effectively to allow integration of representations to occur. We designed and developed an interface based on these theories in conjunction with research scholars at ET-IITB and HBCSE, and undergraduates at IIT-Roorkee and BITS-Pilani and evaluated it using a pilot usability and learnability study. Results of these studies have been submitted to two conferences, T4E and the TELoTS workshop of ICCE and are available at `http://www.et.iitb.ac.in/~aditi/lib/exe/fetch.php?media=rct4e.pdf` and `http://www.et.iitb.ac.in/~aditi/lib/exe/fetch.php?media=rctelots.pdf`. Further design refinement will be done and further studies are planned to evaluate effectiveness for development of RC.

6.1.2 Contributions

1. Designed and developed an interface based on emerging theories of cognition.

2. Evaluated usability and learnability of the interface and developed an analysis methodology for eye and mouse tracking data, for evaluating development of RC.

3. Submitted two conference papers based on this work.

6.2 Think-pair-share in CS101

6.2.1 Overview

CS101 at IITB is a large core course with a great diversity of students in terms of exposure to computer programming. The instructor faces the challenge of keeping all the students engaged and learning in the classroom because the students with prior exposure may get distracted and tune out, while the novice learners may have difficulty with the content. The instructor needs an active learning strategy which can engage all levels of learners and is matched to the learning objectives of the course namely, conceptual understanding, students should be able to trace code, come up with programming logic and write programs. We found that the instructional strategy of think-pair-share(TPS) [24] matched the instructional goals and setting of CS101 and was employed by the instructor.

In order to evaluate the effectiveness of TPS in a large CS101 class we conducted an observational study to measure student engagement along with with research scholars from ET-IITB, a quasi-experimental study to measure student learning via TPS and explored student perceptions of TPS using a survey. The details of these studies and results are available in two papers published in computer science education conferences ICER 2013 and ITiCSE 2014 and can be downloaded from `http://www.et.iitb.ac.in/~aditi/lib/exe/fetch.php?media=tpsicer.pdf` and `http://www.et.iitb.ac.in/~aditi/lib/exe/fetch.php?media=tpsiticse.pdf`. A second quasi-experiment was conducted in the next offering of the course to strengthen the findings of the first experiment and another survey was conducted to probe into the findings of the previous year. These findings will be consolidated into a journal paper in conjunction with other research scholars in ET-IITB.

6.2.2 Contributions

1. Developed an observation protocol and quantified student engagement during TPS in a large CS101 class. Developed a methodology for converting observational data to percentage of student engagement.

2. Quantified learning gain due to TPS.

3. Wrote two papers in ICER 2013 and ITiCSE 2014.

4. This work led to the development of instructor guidelines regarding how to develop TPS activities and conduct the TPS strategy in class and these resources are available on the ET website.

Bibliography

[1] D. Jonassen, J. Strobel, and C. B. Lee, "Everyday problem solving in engineering: Lessons for engineering educators," *Journal of engineering education*, vol. 95, no. 2, pp. 139–151, 2006.

[2] M. Besterfield-Sacre, L. J. Shuman, H. Wolfe, C. J. Atman, J. McGourty, R. L. Miller, B. M. Olds, and G. M. Rogers, "Defining the outcomes: A framework for ec-2000," *Education, IEEE Transactions on*, vol. 43, no. 2, pp. 100–110, 2000.

[3] W. Accord, "Graduate attributes and professional competencies," version 2 18th june 2009, International Engineering Alliance, 2009.

[4] B. K. Beyer, "Developing a scope and sequence for thinking skills instruction.," *Educational Leadership*, vol. 45, no. 7, pp. 26–30, 1988.

[5] D. H. Jonassen, "Instructional design models for well-structured and iii-structured problem-solving learning outcomes," *Educational Technology Research and Development*, vol. 45, no. 1, pp. 65–94, 1997.

[6] B. M. Linder, *Understanding estimation and its relation to engineering education.* PhD thesis, Massachusetts Institute of Technology, 1999.

[7] A. Kothiyal, "Notes on problem solving: A literature review of problem-solving, with emphasis on ill-structured, engineering and estimation problems," technical report (TR-ET-002-2014), Indian Institute of Technology, Bombay, 2014.

[8] S. Shakerin, "The art of estimation," *International Journal of Engineering Education*, vol. 22, no. 2, p. 273, 2006.

[9] E. Joram, K. Subrahmanyam, and R. Gelman, "Measurement estimation: Learning to map the route from number to quantity and back," *Review of Educational Research*, vol. 68, no. 4, pp. 413–449, 1998.

[10] D. Dunn-rankin, "Evaluating design alternatives âĂŞ the role of simple engineering analysis and estimation," in *Proceedings of the 2001 American Society for Engineering Education Annual Conference and Exposition*, American Society for Engineering Education, 2001.

[11] S. Mahajan, *The art of approximation.* The art of approximation.

[12] M. Mavinkurve and S. Murthy, "Comparing self-learning behavior of low and high scorers with ediv,"

[13] D. L. Schwartz and J. D. Bransford, “A time for telling,” *Cognition and instruction*, vol. 16, no. 4, pp. 475–5223, 1998.

[14] M. Kapur, “Productive failure,” *Cognition and Instruction*, vol. 26, no. 3, pp. 379–424, 2008.

[15] K. VanLehn, S. Siler, C. Murray, T. Yamauchi, and W. B. Baggett, “Why do only some events cause learning during human tutoring?,” *Cognition and Instruction*, vol. 21, no. 3, pp. 209–249, 2003.

[16] D. L. Schwartz and T. Martin, “Inventing to prepare for future learning: The hidden efficiency of encouraging original student production in statistics instruction,” *Cognition and Instruction*, vol. 22, no. 2, pp. 129–184, 2004.

[17] J. D. Bransford and D. L. Schwartz, “Rethinking transfer: A simple proposal with multiple implications,” *Review of research in education*, pp. 61–100, 1999.

[18] X. Ge, *Scaffolding students' problem solving processes on an ill-structured task using question prompts and peer interactions.* PhD thesis, The Pennsylvania State University, 2001.

[19] P. Pande and S. Chandrasekaran, “A general account of representational competence.” June 2014.

[20] A. M. Glenberg, J. K. Witt, and J. Metcalfe, “From the revolution to embodiment 25 years of cognitive psychology,” *Perspectives on psychological science*, vol. 8, no. 5, pp. 573–585, 2013.

[21] D. Kirsh, “Thinking with external representations,” in *Cognition Beyond the Brain*, pp. 171–194, Springer, 2013.

[22] D. Kirsh and P. Maglio, “On distinguishing epistemic from pragmatic action,” *Cognitive science*, vol. 18, no. 4, pp. 513–549, 1994.

[23] S. Chandrasekharan, “Building to discover: a common coding model,” *Cognitive Science*, vol. 33, no. 6, pp. 1059–1086, 2009.

[24] F. Lyman, “Think-pair-share: An expanding teaching technique,” *MAA-CIE Cooperative News*, vol. 1, no. 1, pp. 1–2, 1987.

Appendix A

Instructional Material used in the DG study

A.1 The Instructional Materials used in experiment 3

A.1.1 DG group

Instructions

- Work/discuss with partners in your group.
- If you have questions, raise your hand.
- Write down your ideas, thought process, any assumptions you are making
- Draw diagrams, graphs, pseudo-code etc.
- Before coding, think of the many ways you could approach the problem, write what you expect from the simulation before doing it.
- I want to know what methods did not work.
- Grading is based not on final results, but your thought process and your ideas to solve/approach the problem (your scribble notes).
- Document (roughly) all your thoughts, as it will help us in grading.
- You have 40 mins to work on this, each group should save all the codes you have written and give it to the TA.

Coincidence detector

There are two Poisson stimulii P and Q whose arrival rates are given by $\lambda_P = 1/10$ ms and $\lambda_Q = 1/20$ ms. Design a spiking neural network that will take these stimulii as the inputs and generate a spike output if and only if the two stimulii arrive close to each other (inter-stimulus arrival time ≤ 3 ms).

Quantify the performance of your network by calculating the number of false positives and false negatives at the output of the network for a total simulation time of 10 sec.

Getting neural networks to compute

- For a given set of parameters that determine the dynamics of neurons, there is a unique relationship between input current and output spike rate and spike arrival time.

- Networks that we have worked with run at biological time scales. What will you do to accelerate the dynamics of the network? What are the principles of scaling?

- First step in a design is to specify the characteristics of the inputs and the outputs

Getting neural networks to compute

- Network design involves adjusting signal propagation paths to obtain desired objectives. This can be done by:
  - adding neurons in the information pathway
  - appropriately weighting the signal strengths in the pathways
  - modulating the time constants of neurons and/or synapses

- Think about things like:
  - What happens when you change one connection to the other?
  - How does changing the time constants of the neurons or synapses affect the output of a neural network?
  - What is the role of excitatory vs. inhibitory synapses?
  - What happens if I change the threshold voltage of the neuron?

Edge detector

An LIF neuron receives 10 independent stimulii through 10 synapses of identical strength ($w_e = 1$). The arrival rates at the 10 synapses linearly increase from 0.01/ms to 0.05/ms. A spike at time t_k results in a current flowing into the neuron given by the expression

$$I_{app,k}(t) = I_c w_e \left[e^{-(t-t_k)/\tau} - e^{-(t-t_k)/\tau_s}\right] h(t - t_k)$$

Where I_c is the minimum DC current necessary to initiate spikes in the neuron, and $\tau = 15$ms and $\tau_s = \tau/4$.

Your task is to design a neural circuit such that an output spike is generated only when there is a persistent rising edge in the total current arriving at the above input neuron.

Quantify the performance of your network by calculating the number of false positives and false negatives at the output of the neuron A for a total simulation time of 10 sec. You are free to come up with a suitable definition for the term 'persistent rising edge'.

Logistics for next class

- Submit all your scribble notes now.
- Email your individual practice problem codes written today by 9:30 PM to profb.iitb@gmail.com.
- Quiz will be conducted on Oct 15 in class.
- Open ended problem, work individually.
- Based on application of concepts covered in class/tutorial.
- Turn in all your thoughts, scribble notes, graphs, ideas, code.
- Grade based on how you convey your thoughts, not on completion of problem or final results.

A.1.2 DI group

Getting neural networks to compute

- For a given set of parameters that determine the dynamics of neurons, there is a unique relationship between input current and output spike rate and spike arrival time.

- Networks that we have worked with run at biological time scales. What will you do to accelerate the dynamics of the network? What are the principles of scaling?

- First step in a design is to specify the characteristics of the inputs and the outputs

Getting neural networks to compute

- Network design involves adjusting signal propagation paths to obtain desired objectives. This can be done by:
  - adding neurons in the information pathway
  - appropriately weighting the signal strengths in the pathways
  - modulating the time constants of neurons and/or synapses

- Think about things like:
  - What happens when you change one connection to the other?
  - How does changing the time constants of the neurons or synapses affect the output of a neural network?
  - What is the role of excitatory vs. inhibitory synapses?
  - What happens if I change the threshold voltage of the neuron?

Edge detector

An LIF neuron receives 10 independent stimulii through 10 synapses of identical strength ($w_e = 1$). The arrival rates at the 10 synapses linearly increase from 0.01/ms to 0.05/ms. A spike at time t_k results in a current flowing into the neuron given by the expression

$$I_{app,k}(t) = I_c w_e \left[e^{-(t-t_k)/\tau} - e^{-(t-t_k)/\tau_s} \right] h(t - t_k)$$

Where I_c is the minimum DC current necessary to initiate spikes in the neuron, and $\tau = 15$ms and $\tau_s = \tau/4$.

Your task is to design a neural circuit such that an output spike is generated only when there is a persistent rising edge in the total current arriving at the above input neuron.

Quantify the performance of your network by calculating the number of false positives and false negatives at the output of the neuron A for a total simulation time of 10 sec. You are free to come up with a suitable definition for the term 'persistent rising edge'.

Instructions

- Work/discuss with partners in your group.
- If you have questions, raise your hand.
- Write down your ideas, thought process, any assumptions you are making
- Draw diagrams, graphs, pseudo-code etc.
- Before coding, think of the many ways you could approach the problem, write what you expect from the simulation before doing it.
- I want to know what methods did not work.
- Grading is based not on final results, but your thought process and your ideas to solve/approach the problem (your scribble notes).
- Document (roughly) all your thoughts, as it will help us in grading.
- You have 40 mins to work on this, each group should save all the codes you have written and give it to the TA.

Coincidence detector

There are two Poisson stimulii P and Q whose arrival rates are given by $\lambda_P = 1/10$ ms and $\lambda_Q = 1/20$ ms. Design a spiking neural network that will take these stimulii as the inputs and generate a spike output if and only if the two stimulii arrive close to each other (inter-stimulus arrival time ≤ 3 ms).

Quantify the performance of your network by calculating the number of false positives and false negatives at the output of the network for a total simulation time of 10 sec.

Logistics for next class

- Submit all your scribble notes now.
- Email your individual practice problem codes written today by 8:00 PM to profb.iitb@gmail.com.
- Quiz will be conducted on Oct 15 in class.
- Open ended problem, work individually.
- Based on application of concepts covered in class/tutorial.
- Turn in all your thoughts, scribble notes, graphs, ideas, code.
- Grade based on how you convey your thoughts, not on completion of problem or final results.

A.2 The Survey

Neuromorphic Engineering **EE 746, Autumn 2013**

Background Survey

1. Your name:
2. Current program at IIT Bombay: (B.Tech/D.D/M.Tech/Ph.D)
3. Department:
4. Year you joined IIT Bombay for the above program:
5. Your score (in percentage) for PCM at +2 level:
6. Your score (in percentage) for biology (if you studied it) at +2 level:
7. Your current GPA:
8. What is your motivation for taking this class? What do you hope to achieve?

Prerequisites Survey - will not be graded. Tick all correct answers.

1. A fair die is tossed 2013 times. The mean value of the outcomes is most likely to be

   (a) 21/6 (b) 21/2013 (c) $1/e$ (d) $1/\pi$

2. 2013 fair dice are tossed one after the other and the outcomes are recorded. Their mean value is most likely to be

   (a) 21/6 (b) 21/2013 (c) $1/e$ (d) $1/\pi$

3. A biased coin (probability of head = 1/2013) is tossed 2013 times. The probability of obtaining tails in all 2013 trials is approximately

   (a) $1/2013^{2013}$ (b) 1/2013 (c) $1/e$ (d) $1/\pi$

4. There are three coins, two of which are fair (probability of tail = 1/2) and the third one is biased (probability of tail = 1/2013). A coin is picked at random and tossed thrice. The probability of obtaining three heads in a row is close to

   (a) 5/6 (b) 5/8 (c) 5/10 (d) 5/12

5. There are three coins, two of which are fair (probability of tail = 1/2) and the third one is biased (probability of tail = 1/2013). A coin is picked at random and tossed thrice, and it was observed that it landed heads each time. The probability that the chosen coin is biased is close to

   (a) 1/2 (b) 3/4 (c) 4/5 (d) 5/6

6. For $a > 0$, define a matrix

$$\begin{bmatrix} 1 & a & 0 \\ 0 & 1 & a \\ a & 0 & 1 \end{bmatrix}$$

(a) Minimum value of the determinant is 1

(b) $[1, 1, 0]^T$ is an eigenvector

(c) $(1 + a)$ is an eigenvalue

(d) $[1, 1, 1]^T$ is an eigenvector

7. A ideal n-channel MOSFET has a $V_t = 0.3$V. When $V_g = 0.24$V, $I_{ds} = 100$nA flows from the drain to the source. For same drain bias, $I_{ds} = 1$nA when $V_g = 0.12$V.

(a) Subhtreshold slope of the transistor is 60 mV/decade.

(b) Subhtreshold slope of the transistor is 120 mV/decade.

(c) When $V_g = 0$, $I_{ds} = 10$pA

(d) When $V_g = 0$, $I_{ds} = 1$pA

8. The input and output terminals of a CMOS inverter are shorted to each other. The circut

(a) Will behave as an oscillator

(b) Will consume no power in steady state

(c) Has a stable operating point

(d) Will blow up

9. For the following codes

Code A	Code B
$a = zeros(1000, 1);$	$a(1) = 0;$
for $j = 1 : 999$	for $j = 1 : 999$
$a(j + 1) = a(j) + 1;$	$a(j + 1) = a(j) + 1;$
end	end

(a) Execution time of A is faster

(b) Execution time of B is faster

(c) A takes more memory

(d) B takes more memory

10. There is only one perfect square of the form $aabb$ where a and b are non-zero integers. Which one? Show the steps used to find it?

A.3 The post-test (with guiding prompts) of experiment 3

Quiz 3 - Sequence detector EE 746

Prof. Bipin Rajendran

Three events A, B and C occurs in quick succession either in the order $A \rightarrow B \rightarrow C$ or $A \rightarrow C \rightarrow B$. Assume that there are three input neurons, each specifically tuned to respond to the presence of these three events.

Design a neural circuit with two output neurons such that the first neuron spikes only when the sequence of events is $A \rightarrow B \rightarrow C$ and the second neuron spikes only if the sequence of events is $A \rightarrow C \rightarrow B$.

You are free to chose the spike response behavior of the neurons, conduction characteristics of the synapses, and the timescales involved to simulate the behavior.

3

1. How did you encode the input signals to your network?

2. Draw the schematics of all the network structures you considered while trying to get the desired output.

3. Did you observe any pattern in the structure of your connectivity matrix?

4. List all strategies you tried in order to obtain the desired behavior.

5. What strategies did you employ to optimize the weights of the synapses in your network?

A.4 The grading rubric

The rubric used for grading student solutions (both without and with prompts) to assess their IS PS performance is shown in Table A.1.

Table A.1: Rubric for Grading Design Problem

Performance dimension	Construct	Rubric level 0	Rubric level 1	Rubric level 2	Rubric level 3
Choosing the input (IP)	Design of neural networks step 1	No mention of what the inputs are.	Inputs are defined but they are not the simplest, for eg, Poisson stimulus.	Inputs defined are simple but not complete, for eg, one for some instances.	Inputs defined are simple and complete, for eg, one for x ms.
Choosing the time constants of the neurons and synapses (TS)	Design of neural networks step 2	No mention of time scales.	Time constants of either neurons or synapses are mentioned.	Time constants of both neurons and synapses are mentioned but not justified.	Time constants of neurons and synapses are mentioned and justified.
Network structure (NS)	Design of neural networks step 3	Network structure not specified.	Network consists of >10 neurons (eg) a delay network	Network consists of 8-10 neurons.	Network consists of $<$ 8 neurons (eg) a network with interconnections.
Choosing weights and other parameters according to proposed design (ADJ)	Design of neural networks step 4	Weights and other parameters adjusted according to proposed network.	Weights and other parameters partially changed. For eg, weights changed according to own network, but VT not changed.	Weights and other parameters changed, but no justification provided for the values specified.	Weights and other parameters changed according to own design and justification provided for the values specified.
Overall solution (PS) based on written solution (network, proposed inputs, time scales, weights etc)	The ISPS competencies of "Constructing argument" and "Providing evidence"	No solution provided.	A partial solution considering inputs, time scales, network structure, connectivity matrix (weights) and other parameters proposed and no justification given.	A complete solution considering inputs, time scales, network structure, connectivity matrix (weights) and other parameters proposed but not properly justified.	A complete solution considering inputs, time scales, network structure, connectivity matrix (weights) and other parameters proposed and justified.
Overall soution (FS) based on final submitted program and parameters in it	The ISPS competencies of "Evaluating solution" and "Revising solution"	Program not submitted.	Program submitted, but does not run without errors or takes too long to run; major changes required to get program to work. (eg) inefficient inputs specified like Poisson stimulus or incorrect time scales like too short a current integration time.	Program submitted, runs but the output is only partially correct; or program runs but outputs not clear; minor changes needed to get correct output. (eg) order of inputs need to be changed.	Program submitted and runs correctly demonstrating that students design meets specifications.
Quality of Solution (QoS)	Overall quality of design-this is process independent	Poor	Weak	Good	Excellent

A.5 The Perception Survey

1. How often have you solved open problems like these before (including research)?

   a. Often b. Sometimes c. Never

2. The tutorial helped me solve the open problem in the quiz.

   a. Strongly agree b. Agree c. Neutral d. Disagree e. Strongly disagree

3. The tutorials and quizzes helped me learn to solve open ended problems.

   a. Strongly agree b. Agree c. Neutral d. Disagree e. Strongly disagree

4. I found the open problems in the tutorials and quizzes interesting.

   a. Strongly agree b. Agree c. Neutral d. Disagree e. Strongly disagree

5. I would like to solve open ended problems such as these in other courses as well.

   a. Strongly agree b. Agree c. Neutral d. Disagree e. Strongly disagree