Performance-Optimized Detection, Tracking and Modeling of Physical Phenomena in Distributed Sensing Environments

A Dissertation Presented

by

Anurag Umbarkar

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Computer Engineering

Stony Brook University

August 2014

Stony Brook University

The Graduate School

Anurag Umbarkar

We, the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Alex Doboli, Advisor
Associate Professor, Department of Electrical and Computer Engineering, Stony Brook University

Sangjin Hong, Chairperson
Professor, Department of Electrical and Computer Engineering, Stony Brook University

Emre Salman
Assistant Professor, Department of Electrical and Computer Engineering, Stony Brook University

Peter Milder
Assistant Professor, Department of Electrical and Computer Engineering, Stony Brook University

Thomas MacCarthy
Assistant Professor, Department of Applied Mathematics and Statistics, Stony Brook University

This dissertation is accepted by the Graduate School

Charles Taber
Dean of the Graduate School

Abstract of the Dissertation

Performance-Optimized Detection, Tracking and Modeling of Physical Phenomena in Distributed Sensing Environments

by

Anurag Umbarkar

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

2014

Cyber Physical Systems (CPS) are distributed systems-of-systems that perform reliable data acquisition in order to build efficient data models. Data models are mathematical expressions that describe the attributes of the observed environments. These models can be used for monitoring, tracking and predicting the dynamics of the physical phenomena. Also, data models aid in formulating decision-making procedures under resource constraints. Data model construction in CPS is challenging because the dynamics of physical environments are hard to track in real-time through a distributed sensing network with limited bandwidth and local memory. Therefore, the limited resources must be optimally utilized to boost performance metrics.

This dissertation presents a novel technique that uses distributed sensing to construct local models. A multi-level state variable lumping scheme is proposed that reduces communication traffic. A linear-programming based optimization scheme is designed to minimize the modeling error due to data losses and communication delays. Application goals help in defining the parameters of the cost function. The solution of this cost function is used to decide the resource allocation strategy. Error modeling is an important step in achieving this objective. As a part of this research, accurate models are constructed for different types of errors in the network. These include error due to data loss, communication delay, lack of synchronization and modeling errors.

An ontological approach is proposed to build centralized models using data sampled in a distributed environment. The ontological representation is used for describing relationships between model parameters. These physical models are more meaningful since the relationships are extracted from experimental data using data mining as well as causal analysis. This helps in improving the model robustness, thereby enabling the system to respond better to unexpected changes in the dynamics of the physical entities.

In the course of this research, three case studies have been explored: Detection and tracking of emergent gas clouds, Sound-based tracking for vehicular-traffic scenarios and thermal monitoring for 3-Dimensional integrated circuits. Several algorithms to detect and track emergent entities are proposed. Also, different techniques to maximize accuracy of tracking and prediction are discussed.

Similar to networks of embedded systems, we can imagine networks of human beings and their interactions. Data model and knowledge extraction can also be performed to characterize the 'creativity' of human subjects. For this purpose, we define metrics such as Novelty, Variety, Quality and Usefulness. Causal knowledge search using data from these experiments can give insight into the creative thinking process of human engineers.

To My Parents

Table of Contents

List of Figures

List of Tables

Acknowledgments

First of all, I would like to thank Dr. Alex Doboli for his continued support and encouragement throughout the course of this PhD dissertation. His keen interest and enthusiasm towards research inspired me a great deal. He has made invaluable contributions to this dissertation. I consider it a privilege to have Prof. Doboli as my advisor.

I express gratitude towards members of my Dissertation Defense Committee: Prof. Hong, Prof. Salman, Prof. Milder and Prof. MacCarthy. Many thanks also to other faculty members and staff at the Electrical and Computer Engineering department at Stony Brook University. A special thanks to Prof. Kenneth Short, Prof. Harbans Dhadwal and Scott Tierno, with whom I enjoyed working as a Teaching Assistant for many semesters.

I would also like to thank my current and prior colleagues from the Embedded Systems Laboratory, especially Varun Subramanian, Cristian Ferent, Shreyas Kodasara and Fanshu Jiao. Varun and Cristian have been good friends and have provided me with valuable advice on technical as well as non-technical matters.

My parents have always supported me in all the career choices that I have made so far. I would like to thank them for all the love and support. My sincere thanks to my sister and brother-in-law, for their encouragement and valuable guidance. Many thanks to my brother and sister-in-law for their advice and help in making important decisions.

Chapter 1

Introduction

1.1 Motivation

Modern applications, e.g., intelligent traffic systems, smart power grid, and critical infrastructure monitoring, require large-scale decision making networks that operate in tight interactions with the natural environments. Natural environments are fundamentally different than engineered systems (i.e. plants, autovehicles, and consumer goods), which have been the traditional beneficiaries of optimized control. Engineered systems have, in general, a well-defined and predictable behavior as their operating conditions and requirements are well defined. Natural environments are arguably more complex and diverse with respect to their composing elements and interactions. Many interactions and conditions of the natural world are unknown until they are produced, hence are hard to predict and characterize a-priori. Second, natural environments are continuously changing without necessarily moving towards an end state or progressing towards a final goal. The two arguments stress that the modeling and representation of natural environments must tackle dynamically changing situations that include a large variety of emerging entities and interactions.

Creating robust data models for an observed, physical environment is a critical component of Cyber-Physical Systems (CPS). Data models are mathematical expressions that describe in time and space the attributes of the observed environments, e.g., the differential equations that express energy conservation laws [23]. These models can be used to devise (or synthesize) optimized decision strategies by estimating, predicting, and identifying trends and patterns of the physical environment. Data model construction in CPS is challenging because the dynamics of physical environments is hard to track in real-time through a distributed sensing network with limited bandwidth and local memory.

Resource allocation is an important challenge for distributed data acquisition for decision making [19,55]. This is because the individual embedded nodes of the networked infrastructure, due to the cost constraints of an application, do not have sufficient resources to sample sufficient data, e.g., limited local memory, energy and communication bandwidth. Therefore, the resources of the infrastructure as a whole must be efficiently employed for the sampling task. These limitations can be significantly reduced if an entire network of embedded nodes is used for tracking. However, new challenges arise at the network level as data routing to the decision making nodes can produce significant data loss and delays. The resources of a network of embedded nodes must be allocated so that there is an optimal load balancing between sensing - processing - communication activities.

The Cyber-Physical Systems (CPS) considered in this report are distributed embedded systems that are expected to reliably acquire data from the environment to produce robust data models and perform optimized, local and global decisions [20] [19] [18]. This procedure of data acquisition, communication and processing has to be performed under timing and data loss constraints using limited availability of resources. There are three main steps in achieving the above-mentioned objectives:

- Develop procedures to detect emergent/existing physical entities, perform reliable data acquisition and store it in flexible data structures which respond to changing environments.

- Develop systematic global model to represent all components of the scene and describe relationships between those components.

- Design methods to create robust local data models that can be used to develop efficient resource allocation strategies.
- Implement algorithms to track and predict the movements of the physical entity, using insights from the local and global models.

These data acquisition procedures, tracking and prediction algorithms, robust data models and causal graphs can be used to implement decision-making procedures for specific application goals. Also, cause-effect relationships can be extracted using causal analysis and search techniques.

Similar to networks of embedded systems, we can imagine networks of human beings and their interactions. Data model and knowledge extraction can also be performed to characterize the 'creativity' of human subjects. For this purpose, we define metrics such as Novelty, Variety, Quality and Usefulness. Causal knowledge search using data from these experiments can give insight into the creative thinking process of human engineers.

1.2 CPS: Proposed Execution Platform

An overview of the proposed methodology is shown in Figure 1.1. The application goals define the nature of the sensing stage. The sensed data is then used to extract high level features. This distributed data is then aggregated by communication between the nodes in the network. The decision-making procedure uses an optimization scheme to perform resource allocation based on performance metrics. In the execution platform for this methodology, Z-language notation is used to describe the problem statement, ontological structures are used for global scene representation, symbolic expressions are derived for distributed data modeling, Kinetic data structures (KDS) are used for data management and Linear programming based optimization scheme is used for resource management. The details of this execution platform are described in this section.

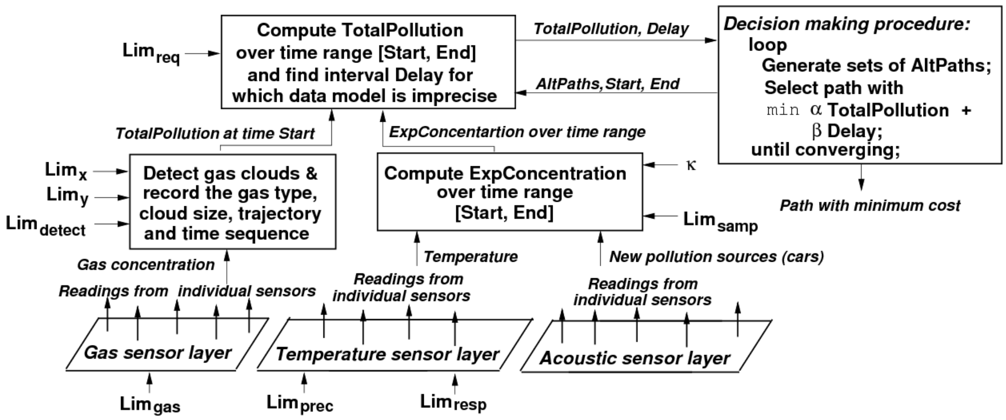


Figure 1.1: Overview of proposed methodology

1.2.1 Problem statement: Z-language notation

A goal-oriented framework to describe the CPS problem statement can be prepared using the Z language notation. An example of such a schema for gas pollution monitoring is shown in Figure 1.2. The details of this notation are described in [2]. For a specific application, the schema presents the variables, data characteristics, application goals and corresponding constraints. For example, the variables gas type, gas concentration, time are mentioned at the top of the schema while the optimization scheme using these variables is described at the bottom. A decision-making routine would use the outputs mentioned in the middle to perform necessary operations.

$\mathbf{GasType} : \{\mathbf{No_x}, \mathbf{CO}, \mathbf{CO_2}, \mathbf{SO_2}\}$
$\mathbf{ExpConcentrationType} : \mathbf{GasType} \times \mathbf{DOM_x} \times \mathbf{DOM_y} \times \mathbf{Time} \rightarrow \mathbf{R_+}$
$\mathbf{TimeI} : [\mathbf{T_L}, \mathbf{T_R}], \mathbf{T_L}, \mathbf{T_R} \in \mathbf{R_+}$ and $\mathbf{T_L} < \mathbf{T_R}$

$\mathbf{Application}[\mathbf{AltPaths}, \mathbf{Start}, \mathbf{End}]$
$\mathbf{TotalPollution!} : \mathrm{seq}\, \mathbf{ExpConcentrationType}$
$\mathbf{Delay!} : \mathrm{seq}\ \mathbf{TimeI}$
$\mathbf{ExpConcentration?} : \mathbf{ExpConcentrationType}$

$\forall \mathbf{gas} \in \mathbf{GasType}. \forall \mathbf{path} : \mathbf{AltPaths}. \forall \mathbf{t} : \mathbf{Time}. \forall (\mathbf{x}, \mathbf{y}) \in \mathbf{path} :$
$\quad \mathbf{TotalPollution!} = \mathbf{TotalPollution} \cup \mathbf{ExpConcentration?}(\mathbf{gas}, \mathbf{x}, \mathbf{y}, \mathbf{t})$
$\exists \mathbf{t} : \mathbf{Time}. \forall \mathbf{e} \in \mathbf{TotalPollution}.$
$\quad (\mathbf{t} < \mathbf{t_0}) \wedge (\mathbf{t} = \max \mathbf{e.Time}) \wedge (\mathbf{t_0} - \mathbf{t} > \mathbf{Lim_{req}}) \Rightarrow (\mathbf{Delay!} = \mathbf{Delay} \cup [\mathbf{t}, \mathbf{t_0}])$
$\exists \mathbf{path} \in \mathbf{AltPaths}. \min(\alpha \int_{\mathbf{path}} \int_{\mathbf{Start}}^{\mathbf{End}} \mathbf{TotalPollution}(\mathbf{Type}, \mathbf{l}, \mathbf{t}) \mathbf{dtdl} + \beta \sum \mathbf{Delay})$

Figure 1.2: Z-based notation describing schema for application goals

1.2.2 Global Modeling: Ontology for Scene Representation

A detailed description of this work on Ontological scene representation, including experimental evaluation, is provided in our paper [4]. An ontology describes the concepts, attributes (properties), and permanent relations among the concepts of an application class. Ontologies offer an abstract yet complete description of the possible situations that can occur in reality. Each ontology defines the concepts (components) that form a real situation, the relations according to which the concepts are linked together, and the attributes (features) of the concepts, the constraints of the attribute values (e.g., sequencing over time, impact of events, etc.). The instantiation of an ontology for a specific scenario is useful to find the mathematical models that describe the scenario. The models result as a composition of the models describing the concepts and relations identified from the ontology.

Guarino and Welty [9, 10] propose that ontologies are characterized using metrics, like unity and identity. Unity states that all instances of a concept are linked to the concept through a well defined set of properties. Rigidity indicates that properties do not change within a time window but then can change as a result of an event. A rigid property carries identity condition if it the existence of the property implies that the involved instances are equal. We propose a similar approach based on the common attributes of the instances of a concept.

Ontology description: Vehicular traffic case study

In our approach, every concept represents a group of instances that share a common set of attributes and are distinguishable from other instances and concepts by another set of attributes. Attributes are invariant features of instances. There can be various perspectives to describe the invariant character of attributes, such as invariant over time, space, population, etc.

The meaning of a traffic scene is defined in terms of a set of basic semantic elements, which cannot be defined using more basic elements and can be estimated based on the inputs coming from sensors. The basic semantic elements (BSEs) to be identified and analyzed include the following aspects:

- *Vehicle attributes*: Some of the typical vehicle attributes include kind, speed, acceleration, position, and trajectory.

- *Driver's driving profile*: A profile includes his/her preferred style of driving depending on traffic and weather conditions. For example, the driver's profile describes the likelihood of changing the speed or trajectory (e.g., switching the lanes).

- *Clusters of vehicles*: Clusters are formed by vehicles that travel while having a common set of stationary attributes, such as a constant number of vehicles in the cluster and vehicle speed variations and inter-vehicle spacing that pertain to well-defined (yet unknown) ranges.

- *Cluster attributes*: Every cluster is characterized by attributes like size (number of vehicles), speed range, trajectory, time of formation and time of dispersion. Clusters have also attributes that are different from the attributes of vehicles, e.g., spacing between cars.

- *Cluster-level, social behavior*: The way in which the drivers forming a cluster change their driving behavior based on the cluster characteristics, e.g., drivers decide to adapt to the speed of the other drivers in the cluster, or start looking for opportunities to leave the cluster.

- *Cluster dynamics*: Vehicle clusters go through modifications, such as a cluster splitting into sub-clusters and different clusters merging into a single clusters. Another kind of interaction is if two clusters automatically correlate their attributes, like speed.

- *Road conditions*: This refers to special road conditions, e.g., the position of potholes, traffic signs, and stopped vehicles.

- *Weather conditions*: This aspect relates to the nature of weather conditions, such as the position of ice and water on the road.

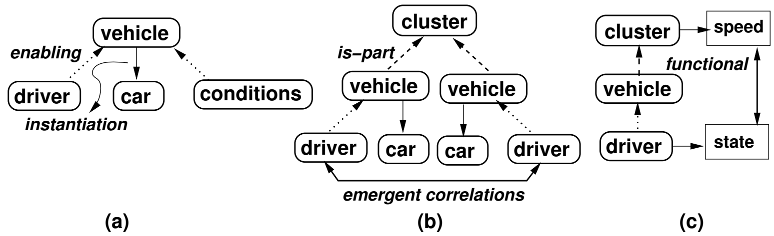


Figure 1.3: Relations in ontologies for traffic applications

The elements above define a simple ontology for traffic applications. They are the basic elements involved in traffic and are used for expressing the possible interactions and correlations in traffic scenes. Every particular traffic scene is a specific instance of the ontology. Understanding the behavior of traffic involves constructing the traffic scene corresponding to the ontology.

Figure 1.3 illustrates the nature of relations between the concepts of a traffic-related ontology. Figure 1.3(a) shows concept instantiation and enabling relations. Instantiation, indicated with solid line, defines that concept *car* is a more specific concept than concept *vehicle*. Some of the defining attributes of concept *vehicle* have a more constrained description for *car*. For example, attribute *size* is restricted to a smaller range. Still, the constrained attribute allows distinguishing the concept from other concepts instantiated based on concept *vehicle*. Enabling relations, shown with dotted lines, indicates that the characteristics of the related class are used to control (refine) more specific attribute values for the concept. For example, the attributes of concepts *driver* and *conditions* restrict the attribute values of concept *vehicle*.

The *is-part* relation in Figure 1.3(b) (shown with dotted line) defines that all attributes of the target concept depend on attributes of the originating concepts or instances. This means that for every attribute of the target concept, every participating concept has at least one attribute that influences the attribute. For example, all attributes of concept *cluster* (of vehicles) depend on the attributes of the instances (vehicles) that form the cluster. Note that the attributes of the target concept might depend also on other attributes than those of the originating concepts or instances. A concept set C_{in} is a complete description of the $is-part$ relation with concept C, if there is no other concept

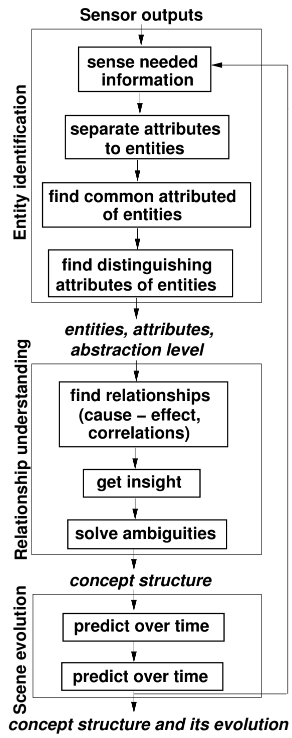


Figure 1.4: Traffic scene understanding methodology

that has a $is - part$ relation with concept C.

Constructing Traffic Scene Representations

Figure 1.4 presents the proposed methodology to construct representations of traffic scenes. The methodology has three steps: (i) identifying the entities that participate to the scene, such as individual vehicles, vehicle categories (i.e. sedan, truck, SUV, etc.), vehicle clusters, road obstacles, traffic lights, and so on; (ii) understanding the relations between the found entities, and (iii) predicting the dynamics of the scene based on analytical models for the scene as well as sensed data acquired in real time.

The first step, entity identification, finds the participating entities based on their distinguishing attributes (which separate them from other entities). Every instance, i.e. vehicle in a scene, has physical attributes (PAs), some of which can be measured directly through sensors, e.g., position, dimension, weight, temperature, time, and so on. Some of the sensor readings might be unavailable at a certain moment. Every attribute can have values from a (constrained) domain. In addition, attribute descriptions might include constraints defined over the associated attributes, including constraints between the goal of the application and attributes. Common attributes of multiple instances enable the identification of the categories to which the individual entities pertain to, like the vehicles that form a cluster.

Example: For traffic applications, possible PAs are position (of a vehicle), time, dimension and weight (of a vehicle). Speed, another attribute of a vehicle, is represented as the following tuple: $(\frac{x-x_0}{t-t_0}, (x,t), (x_0,t_0), t > t_0, t - t_0 < \epsilon)$. The speed attribute (the first component of the tuple) is defined using two other associations of PAs, (x,t) and (x_0,t_0). Besides, the constraints $t > t_0, t - t_0 < \epsilon$ must be valid to compute correctly the speed attribute. Similarly, the interspacing between two vehicles A and B is defined as $(x_A - x_B, (x_A,t), (y_B,t), x_A > x_B)$.

The second step of the methodology finds the causal relations among concepts, such as the reasons that produce certain constraints and patterns of the attributes of entities. Causes that are directly observable through sensors are utilized to formulate hypothesis on the causal relations that might originate the constraints [13, 14]. Other potential causes, which are not directly observed, are formulated based on the ontology of traffic scenes during the insight getting step. The likelihood of (observable and unobservable) causes are computed using Bayesian networks, a popular causal reasoning procedure [12].

The third step constructs the analytical models starting from the identified scene elements and the causality relations between them. The analytical models include the mathematical expressions that characterize the attributes of the elements as well as the expressions of the causal relations. These models are then used to predict the future dynamics of the represented scene. Section IV illustrates the algorithm to predict traffic scene evolution, including the behavior of the current clusters and their merging and splitting.

Deriving insights from the ontology

The goal of this work is to understand at run time the semantics of traffic scenes (environments) based on audio range inputs collected through a network of embedded nodes with sound processing features [8]. Understanding traffic scenes includes the following main challenges:

- *Finding the components of a scene*: This capability identifies the elements of a traffic scene and their defining attributes. The elements include not only physical objects (e.g., objects with attributes directly sensed through the sensors) but also more abstract elements that are used in the reasoning process, like concepts which are not directly sensed but impact the observed signals as well as abstract concepts and categories. Scene components are characterized by a set of well defined, repeatable attributes (which creates the invariant identity of a component) and a set of attributes that distinguish the concept from other concepts.

- *Understanding the relations between the components in a scene*: This capability finds the interdependencies and correlations that exist between the components in a scene, including cause - effect relations, in which a certain element causes or enables a given effect, and various kinds of correlations between elements.

  Getting insight into the cause of the existing relations is a first main requirement. In addition to the correlations that result directly from the nature of the application, other correlations are produced due to specific conditions and properties of the participating elements. For example, traffic flow can be obstructed by an obstacle on the road (direct cause) or a set of drivers with specific driving profiles that slow each other down. The second situation can be inferred from the scene characteristics even if it is not directly specified as a cause for slow traffic.

  Disambiguation is a second main requirement as multiple causes can produce similar effects. For example, group of vehicles slowing down can be either because of some conservative drivers or due to potholes present

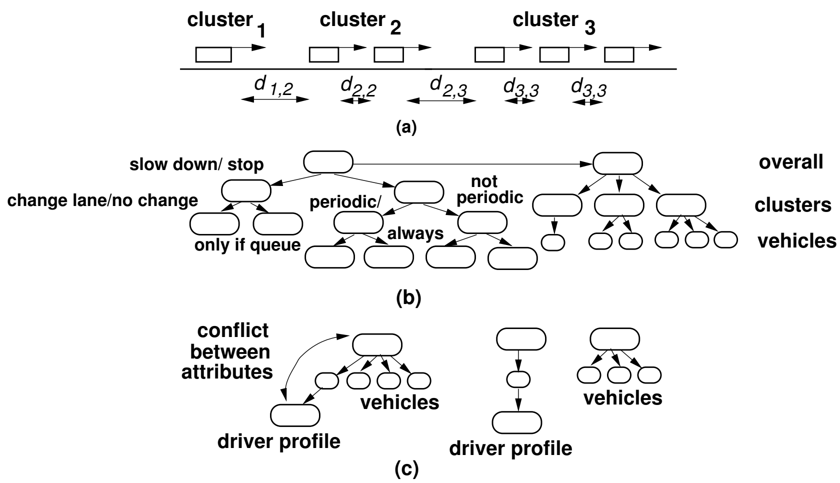


Figure 1.5: Simple traffic scene

in the road. The sensed information must be used to deduce the more likely cause that produces a situation among the two possibilities.

- *Predicting the evolution of a scene*: The capability refers to the dynamics (evolution) of a traffic scene, including the possible situations that can emerge within a future time window. Predictions are important to correct erroneous data from the sensors, to identify the necessary and sufficient data needed for scene understanding, and to preemptively adopt decisions for situations in which reactive actions are insufficient.

Example: Let's consider the simple traffic situation in Figure 1.5 to illustrate the three challenges in scene understanding. Figure 1.5(a) presents five vehicles moving on the road. Scene understanding must first identify the three vehicle clusters, where a cluster comprises of the vehicles moving according to the same pattern (e.g., similar speed and speed variations). This pattern must be different from the patterns of other clusters. If the clusters move with different speed then only speed is sufficient for cluster identification. However, if two clusters are moving at the same speed then additional attributes are needed for differentiating the clusters, such as the interspacing $d_{i,j}$ between the vehicles. A possible differentiation criterion is that interspacing is significantly larger than the average of the other interspacing.

An important aspect in concept identification (including finding concept attributes) is the identification of the necessary and sufficient information that makes the identification process possible. Moreover, inferring the information needed for scene understanding helps solving the ambiguities that can occur between different concepts with common attributes. Hence, concept identification relies not only on finding similarities between concepts but also outliers.

Another important objective of scene understanding is getting insight into the causes of the relations between concepts, and solving the ambiguities that occur during this step. These relations are not directly evident from the description of a traffic application. For example, there can be multiple causes for vehicle slow down, e.g., potholes, stopped cars, traffic lights, and flooded areas. However, these causes can be often distinguished from each other by using sufficient relevant attributes. For example, potholes force cars to mainly slow down and change lanes, while stopped vehicles cause vehicles only rarely to switch lanes or to stop. Moreover, traffic lights impose a periodic stopping of all cars, while for other periods cars movements are not affected. Finally, flooded areas cause all vehicles to stop and wait until the cars in front pass. In this case, there is no attribute that distinguishes the four cases. Instead, the ontological hierarchy in Figure 1.5(b) must be used for getting insight into the traffic scene and disambiguate the possible cause - effect relations by finding the most likely cause.

Once the concepts and their relations in a scene are understood, the information is used to understand the expected dynamics (evolution) of the scene and the emergence of new relations. For example, if the driver profile in Figure 1.5(c) does not match the speed attribute of the cluster, it is likely that that vehicle will leave the cluster in the near future.

1.2.3 Local Modeling: Distributed Symbolic Expressions

Data models can be expressed as ordinary differential equations (ODEs) in time and space and describing the attributes of the observed environment.

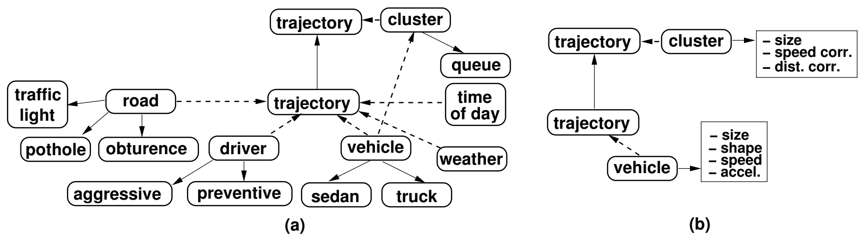


Figure 1.6: Simple ontology for traffic applications

ODEs have been used to present the energy conservation laws for a variety of physical environments and phenomena [23]. For example, the temperature and density of substances in dissipative or non-dissipative 3D volumes (e.g., the liquid inside a computer server cooling system, or the gases forming the atmosphere in a city or a room) is modeled by instantiating the flow equations in physics based on the sampled signals, energy sources, and energy flow properties of the monitored physical environment. Recent work focuses on techniques to automatically create data models for physical environments, including the causal origin of the observed variations and correlations [44, 49].

The ODEs formulated for a physical environment are solved by discretizing the equations through backward (or forward) Euler integration formula, e.g., 7-point finite difference discretizing [35, 47]. Equation discretizing is described as a network of energy injection and removal elements connected through transfer elements (i.e. resistors and capacitors) that propagate energy in time and space (see Figure 2.2(b)). Depending on the nature of the physical phenomenon, the energy injection/removal and transfer elements are described as linear, nonlinear, or stochastic expressions over the parameters of the phenomenon. In circuit design, temperature maps are examples of data models.

1.2.4 Data Management: Kinetic Data Structures

A short description of the use of KDS in the CPS framework is provided in this section. Further details and experiments related to this work can be found in the paper [3]. A decentralized strategy was proposed to produce dynamically, under timing, accurate data representing geographically-distributed physical phenomena. Our data model is based on the Kinetic Data Structure (KDS) model [17] but focuses on a distributed implementation strategy, where data is aggregated in the form of fragments distributed over the entire cloud. When the emergent cloud is detected, we compute the initial set of fragments in the form of convex hulls as shown in Figure 1.8(a). Each fragment is represented by an aggregate node that correlates its data with other aggregate nodes to make decisions that correspond to merging or splitting those fragments to represent a phenomenon. Whenever these fragments change their structure, related parameters, like area, location, density and composition, are updated.

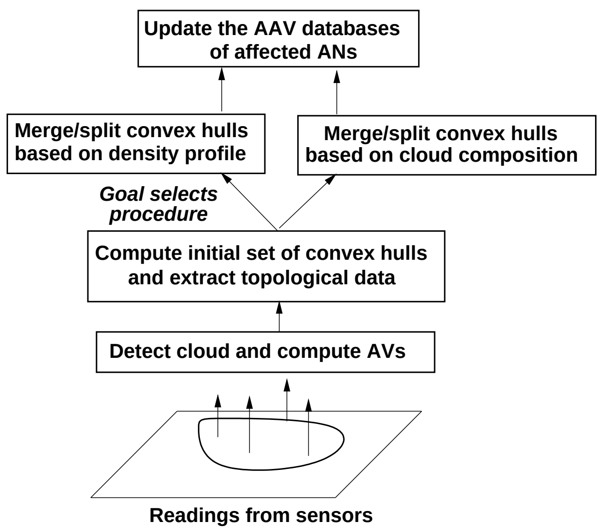


Figure 1.7: Proposed flow to construct distributed KDS

Distributed Implementation of KDS

A. Overview. The process of obtaining a meaningful representation of the cloud is summarized in Figure 1.7. At the lowest level, the sensor nodes sample emergent events and compute the attribute vectors (AV). The vectors are then used to compute the initial set of convex hulls. Data is aggregated at this level, so that each Aggregation Node (AN) contains the aggregated attribute vector (AAV) for that particular convex hull. This aggregated data can be used to compute several topographical parameters such as boundary, area and location of the convex hull.

We can obtain deeper insight into the cloud composition and density profile by performing certain additional procedures. Since this sytem follows a goal-oriented approach, the procedure is selected at runtime depending on the aim of the application. This approach is explained in detail later on. At this level, the ANs share their AAV information with the neigbours. Depending on the goal, we focus on either the density profile or the cloud composition, in order to decide which convex hulls should be split/merged. For example, this splitting process helps us gain more information in regions where the density gradient is higher which suggests variation in gas concentrations or greater variety of vehicles (sound sources). Alternately, the regions with lower gradients are combined since the information content in the region is lower. This merging and splitting results in the formation of additional AAVs at the bottom-right AN of the hull, which are then communicated to all the other ANs in the modified hull. These ANs then update their database to reflect the changes. This methodology also helps in representing changes in the structure and composition of the cloud over time.

B. KDS Parameters. In order to faithfully represent the dynamic nature of the physical cloud movement, we use KDS to characterize different aspects of the entities which form the cloud. Our cloud KDS have three main parameters:

- *Topography*: The parameter describes the geographical extent of the cloud. It can express boundary, area, and location. In order to extract these parameters, the sensing nodes generate attributes by providing either the location of the entity, or their own location if they do not have localization capabilities.

- *Composition*: The parameter contains information regarding the inherent signature characteristics of the entity being monitored. For example, the chemical composition of a gas or the frequency spectrum of a sound source. The parameter helps us to define another parameter called cloud

Algorithm 1 Algorithm for computing the initial convex hulls

for each active node $i = 1$ to n **do**
  initialize $count = 0$
  forward data towards the ANs
  if $(y_i \neq 0) and (x_i \neq N)$ **then**
    collect incoming data vectors to compute the convex hull at AN_i
    increment $count$
    if $count > 2$ **then**
      aggregate data in the format: $AAV < (X, Y)_A, H_A, D_A >$
    end if
  end if
end for

class, e.g., homogeneous or heterogeneous. The attributes required for this purpose are the signature attributes, which are generated by the nodes using their sampled data.

- *Density*: The parameter contains information regarding the density of the entity in different regions of the cloud. Combining the data from all these parameters, we can identify important derived characteristics such as the continuous/discontinuous nature of the cloud, merging/splitting of clouds, and movement of clouds over time.

C. KDS Operators. KDS include three main operators for each kind of parameter: (i) convex hull operator [17] is the mathematical model that supports aggregating data fragments into a valid result, (ii) composition operator that decides to merge fragments (applying convex hull operator) depending on the pursued goal, and (iii) splitting operator that decomposes the computing of a KDS parameter for a larger geographical area into fragments corresponding to smaller regions. The three operators are presented next.

i. Convex hulls: Suppose an emergent cloud triggered events at n nodes in a network of size $N \times N$. The n nodes sample the data and compute the low level attributes that are required to comprehensively charaterize the KDS operators. These attributes are then converted into aggregated fragments called convex hulls (CH) at the Aggregation Nodes (AN), using certain aggregation scheme as shown in Figure 1.8(a). The arrows show the communication scheme for transmitting data during the aggregation process. The procedure used to form these initial set of convex hulls is explained below. The input to this function is the attribute vector generated by the sampling nodes, represented as follows:

$$AV_i < (x, y)_i, H_i, D_i > \tag{1.1}$$

$$H_i < entity_1, entity_2, ..entity_l > \tag{1.2}$$

$$D_i < d_1, d_2, ..d_l > \tag{1.3}$$

where, (x, y) is the location of the sensor node, H is the cumulative vector which contains the list of l entities, D contains the density of each type of entity, in the same order as the names in H. Each node generates one attribute vector. So, i varies from 1 to n.

In Algorithm 1, the initial convex hulls are formed using a static algorithm where the bottom-right node in the structure is the AN for that particular convex hull. Therefore, the nodes which have y coordinate zero (leftmost column) or those which have x coordinate equal to N (topmost row), cannot be ANs. Also, the AN requires at least three active nodes to form a convex hull. The aggregated attribute vector contains $(X, Y)_A$ which is a list of (x, y) coordinates of the nodes which form the convex hull, H_A is the cumulative aggregated vector of the list of distinct entities and D_A contains the average density of the respective entities for that particular convex hull.

ii-iii. Goal-oriented composition (merging) and distribution (splitting): In the analysis, n nodes form m aggregation nodes, which in turn compute p convex hulls. Initially, the ANs transmit their AAVs to the active neighbours and receive AAVs from others. Depending on the goal, Algorithm 2 or Algorithm 3 is selected to operate on the network. Therefore, depending on the density gradient or composition of the cloud, convex hulls are split (distributed) or merged (composed), to get more insights into the cloud characteristics as shown in Figure 1.8(b).

Algorithm 2 Goal-Oriented Merging/Splitting of Convex hulls - Density profile

send AAVs to neighbouring active ANs
receive AAVs from neighbouring active ANs
if goal = Density profile **then**
  for each AN $j = 1$ to m **do**
    compare density profile D for entities
    if $\frac{dD}{d(x,y)} > threshold_{high}$ **then**
      split convex hull into smaller sections
    else
      if $\frac{dD}{d(x,y)} < threshold_{low}$ **then**
        merge convex hulls into larger sections
      end if
    end if
  end for
end if
Update convex hull information in AAV databases in the ANs

Algorithm 3 Goal-Oriented Merging/Splitting of Convex hulls - Cloud composition

send AAVs to neighbouring active ANs
receive AAVs from neighbouring active ANs
if goal = Cloud composition **then**
  for each AN $j = 1$ to m **do**
    compare list of entities H_A in the incoming AAVs
    if lists match **then**
      combine hulls
    end if
  end for
end if
Update convex hull information in AAV databases in the ANs

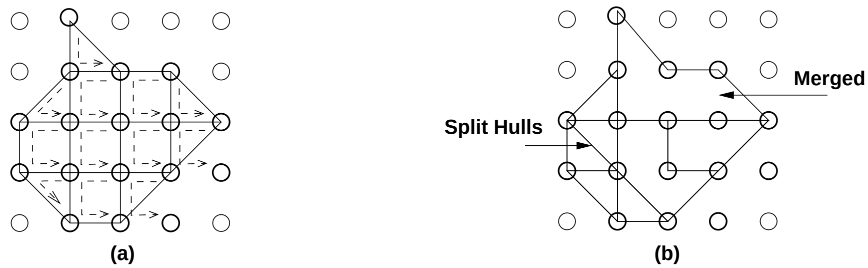


Figure 1.8: Aggregation, splitting and merging of convex hull fragments

1.2.5 Optimization: Linear Programming

Resource allocation strategies are required in order to decide the bandwidth, buffer size and communication paths used by each node in the network. A linear programming based optimization scheme was designed to perform resource allocation. The details of such a scheme applied to a thermal monitoring application is described in Chapter 2.

1.3 Going Beyond Sensed Data

1.3.1 Distributed Information in Human Networks

In certain application domains, such as human networks, there are variables that cannot be measured directly using electronic sensors. There is a need for systematic estimation and inference techniques to quantify the effects of these variables. For example, estimating the 'Creativity' of a human subject (and of a group of subjects) is a challenging problem. It can only be described as a combined effect of the novelty, quality, usefulness and variety of the solutions that the subject(s) can design. Therefore, we performed experiments [6,7] to analyze creativity in design of electronic embedded systems.

We think that the insight gained through these experimental studies can inspire the devising of new CAD tools targeted towards innovation in engineering design. Computer algorithms have been devised for automatically comparing electronic circuits based on ideas inspired by structural alignment [146], theoretical models for domain knowledge representation in circuit design based on concept categorization [147], and circuit design flows inspired by problem solving heuristics [145]. The ongoing work can continue this effort towards a complete CAD environment for innovation in electronic circuit design.

1.3.2 Causal Search, Inference and Prediction

Any complex system usually contains several variables that directly or indirectly affect each other. Some of these variables are measurable and therefore 'visible' while others are not. These 'hidden' variables cannot be directly measured either because of physical inaccessibility or technological limitations. There are several advantages of performing Causal analysis of systems where such variables might exist. Experiments for causal search were performed using data from thermal analysis of 3-dimensional integrated circuits and the creativity measurements.

Causal search can be used to extract a graphical representation of relationships between different variables in a system, directly from raw data. This gives an insight into the expected and unexpected interactions between the parameters. This graph can also be used to extract symbolic expressions that describe the behavior of the system. Also, there are algorithms specifically designed to provide information about the possible location of hidden variables in the graph. This information can be used to perform reliable prediction of dynamics of the measured entities. Decision-making routines can be much more reliable and efficient if they monitor the cause-effect relations between different entities in the system.

1.4 Thesis Outline and Contributions

This report is organized as follows: Chapters 2 and 3 discuss dicuss the work on networks of embedded systems. Chapters 4 and 5 present research on the topic of analyzing creativity in electronic design. Chapter 6 provides the conclusions and introduces 'causal search graphs' obtained from experiments in previous chapters, as future work. Here is a description of contents in each chapter and the contributions of this work.

Chapter 2 presents a multi-level state variable lumping scheme to construct robust mathematical data models from data sampled through a network of sensing devices of limited resources, like bandwidth and buffer memory. The proposed modeling scheme uses a linear programming (LP) formulation to compute the lumping level at each node, and the parameters of the networked sensing platform, i.e. best data communication paths and bandwidths. Two algorithms are described to predict the trajectories of mobile energy sources/sinks as predictions can further reduce data loss and delays during communication. Even though the procedure can be used to model a broader set of phenomena, experiments discuss the effectiveness of the method for thermal modeling of ULTRASPARC Niagara T1 architecture.

Experiments show that variable lumping reduces the overall error by up to 76.91% and delay by up to 57.62%, as compared to no lumping being used. The error is smallest if latency reduction has high priority. The attempt to minimize local error performs less lumping, however, results is larger data loss, and hence in more overall error. The attempt to reduce the overall error by minimizing the correlation error results in increased latency. As the network size increases from 25 nodes to 64 and 100 nodes, the larger communication traffic leads to further losses and delays. Therefore, accuracy-centered optimization becomes critical for performing reliable data extraction. Trajectory prediction using adaptive method (A2) reduces modeling error by up to about 10%.

Chapter 3 presents an approach for tracking and predicting the trajectory of moving entities, while simultaneously optimizing the data models. The method considers three orthogonal facets defining model precision: minimizing the sampling error of the individual embedded nodes, sampling sufficient data from distributed areas to correctly represent the phenomenon of interest, and meeting the timing delays that guarantee the timeliness of data. The three objectives are achieved by dynamically reconfiguring the architecture of the embedded nodes, and dynamically selecting the data transfer paths to the decision making nodes.

Sound based trajectory tracking is used as a case study for the proposed approach. For this application, the accuracy of data sensing is improved by about 28.5%, data loss is zero in most situations, and delay reductions are more than 20% in most cases.

Chapter 4 presents a study on the role of precedents in illuminating creative ideas during iterative design for solving open-ended problems in electronic embedded systems. Through an experimental study grounded in cognitive psychology, this work examined the influence of precedents on the novelty, variety, quality, and utility of design solutions devised through an iterative design process involving groups of participants. Another tested hypothesis was whether incremental changes of requirements improve novelty. Results show that precedents did not increase solution novelty and quality, but improved utility. Precedents reduced design feature variety as solutions converged towards a few dominant designs. Incremental modification of requirements did not increase novelty.

Chapter 5 discusses the nature of concept combinations in modular design of electronic embedded systems as well as the relation between combination characteristics and novelty, quality, and usefulness of the produced solutions. Through two experimental studies, this work explored the frequency of relation-based and property-based combinations in embedded design solutions, and how the specifics of the given building blocks, i.e. salience, relatedness and number, influenced the produced combinations. The impact of popular aids, like titles and short descriptions (briefs), in improving novelty, quality, and usefulness of the designs was also analyzed. Design solutions include mostly relation-based combinations. Design novelty correlates mainly to the purpose and context of the produced combinations. Novelty is aided by titles but not by briefs.

Chapter 6 ends this report with a summary of the conclusions and describes a topic that can be explored in the future, namely 'causal knowledge extraction' in distributed data. It is well known that correlation of values between two variables does not necessarily imply causation. We need specialized algorithms to identify such causal relations. By observing the nature of relations between the same set of variables, at different nodes in the sensor network, we can try to gain an insight into the behavior of the physical entity being monitored. Similarly, in a group of human beings, we can analyze individual responses as well as their interaction with others in the group using causal analysis. Therefore, we explore two application goals: Creativity in electronic design (explained in chapters 4 and 5) and Thermal modeling of microprocessor (from chapter 2). For each goal, the aim is to extract causal graphs from distributed data and analyze the causal relationships between different variables under varying circumstances.

Chapter 2

Robust Data Modeling in Distributed Sensing Environments

[1]

Creating accurate data models describing the dynamics of physical phenomena in time and space is important in optimized control and decision making. Models highlight various trends and patterns. However, producing accurate models is challenging as different errors are introduced by sampling platforms with limited resources, e.g., insufficient sampling rates, data loss due to buffer overwriting, reduced communication bandwidth, and long communication delays. Furthermore, the dynamics of the environment, like mobile energy sources and sinks, might further increase errors as resources must be shared between the sampling and communication activities. This chapter presents a procedure to systematically construct robust data models using samples acquired through a grid network of embedded sensing devices with limited resources, like bandwidth and buffer memory. Models are in the form

[1] Note: This chapter is based on the work published in [1]. Preliminary work on this topic was performed in collaboration with a colleague and included in [2]. Since then, several enhancements to this work were conducted by me including the following: The error models have been re-evaluated and cost function is updated. A completely new set of experiments for thermal modeling have been included. Several new figures and tables are also included.

of ordinary differential equations (ODEs). The procedure constructs local data models by lumping state variables. Local models are then collected centrally to produce global models. The proposed modeling scheme uses a linear programming (LP) formulation to compute the lumping level at each node, and the parameters of the networked sensing platform, i.e. best data communication paths and bandwidths. Two algorithms are described to predict the trajectories of mobile energy sources/sinks as predictions can further reduce data loss and delays during communication. The computed parameters and trajectory predictions are used to configure the local decision making routines of the networked sampling nodes. Even though the procedure can be used to model a broader set of phenomena, experiments discuss the effectiveness of the method for thermal modeling of ULTRASPARC Niagara T1 architecture. Experiments show that the presented method reduces the overall error between 58.29% and 76.91% with an average of 68.87%, and communication delay between -11.49% and 57.62% with an average of 21.85%.

2.1 Introduction

Cyber-Physical Systems (CPS) are expected to integrate data acquisition, networking, and control in an effort to produce effective decisions while operating in complex physical environments [39]. The acquired data is utilized to build data models, which are then used to devise (or synthesize) optimized control (decision) strategies by estimating, predicting, and identifying trends and patterns of the physical environment as expressed by data models. Data model accuracy is critical as unaccounted modeling errors might produce unpredicted decisions, hence reduce the robustness of the systems.

This chapter focuses on data models expressed as ordinary differential equations (ODEs) in time and space and describing the attributes of the observed environment. ODEs have been used to present the energy conservation laws for a variety of physical environments and phenomena [23]. For example, the temperature and density of substances in dissipative or non-dissipative 3D volumes (e.g., the liquid inside a computer server cooling system, or the gases forming the atmosphere in a city or a room) is modeled by instantiating the flow equations in physics based on the sampled signals, energy sources, and energy flow properties of the monitored physical environment. Recent work focuses on techniques to automatically create data models for physical environments, including the causal origin of the observed variations and correlations [44, 49].

The ODEs formulated for a physical environment are solved by discretizing the equations through backward (or forward) Euler integration formula, e.g., 7-point finite difference discretizing [35, 47]. Equation discretizing is described as a network of energy injection and removal elements connected through transfer elements (i.e. resistors and capacitors) that propagate energy in time and space (see Figure 2.2(b)). Depending on the nature of the physical phenomenon, the energy injection/removal and transfer elements are described as linear, nonlinear, or stochastic expressions over the parameters of the phenomenon. In circuit design, temperature maps are examples of data models. A multi-grid thermal modeling method is proposed in [35] to address the heterogeneous areas of 3D ICs. A similar approach, based on adaptive discretizing grids, is discussed in [29].

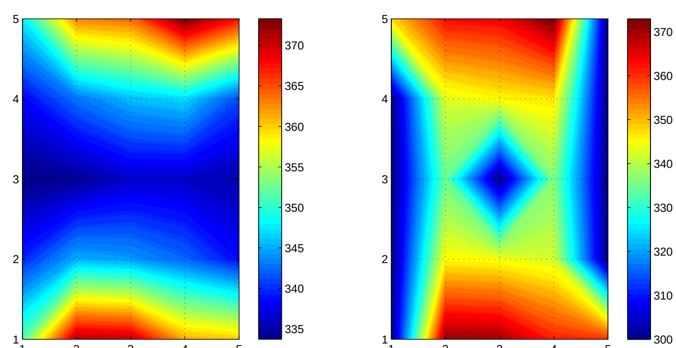


Figure 2.1: (left) Complete thermal map and (right) thermal map for distributed sensing with data loss; the map is significantly distorted due to data loss

Data model construction in CPS applications is challenging because the dynamic of physical environments is hard to track accurately and in real-time through a distributed sensing network with limited resources, like bandwidth and memory. Continuously streaming large volumes of data samples can result in important data losses due to low bandwidth and many data overwrites in the memory buffers. Also, low bandwidths and long communication paths increase the time delays of data samples, thus adding further errors to time-sensitive data models. Figure 2.1 shows that, even for a smaller, 25-node network, there is significant difference between the real thermal map (left) and the map (right) based on distributed sensing. As the network size grows, the streaming volume increases and consequently the difference in the thermal maps further increases. The modeling error reduces the optimality of the devised control decisions of CPS applications.

The impact of data loss on model accuracy can be mitigated by building local data models, and then streaming the local models instead of the samples used to create the models. For example, for Wireless Sensor Networks (WSNs), data aggregation has been proposed to improve the network performance, like throughput, bandwidth, and energy consumption [31, 32, 36]. A regression-based framework for modeling sensor data is presented in [30, 33]. Data communications are reduced by exploiting the correlations between data sensed by neighboring nodes [52] or by fitting generic functions that approximate the transmitted values [22].

Building accurate, global data models from streamed local models raises two main issues. First, the decision on which local models to produce (e.g., which state variables to express) is made only based on local information even though the decision impacts the accuracy of the global model. The embedded sensing nodes lack global knowledge about the environment's dynamics, including moving energy sources and sinks, and changing sensitivities of its state variables. This issue introduces significant differences from methods like [29, 35, 40], which assume statically known environments in which all state variables are available to construct the model. Second, due to data loss, model accuracy depends on the resource characteristics of the distributed sensing platform. Resource utilization must be optimized such that the resulting data loss and time delays generate a minimum modeling error. This issue has been less explored by related work.

This chapter presents a procedure to systematically construct robust data models based on samples acquired through a grid network of embedded sensing devices with limited resources, like bandwidth and buffer memory. The data models are in the form of ordinary differential equations (ODEs). The procedure constructs local data models by lumping state variables, so that the errors due to data loss and time delays are minimized. The model error is captured by mathematical bounds for four types of errors: due to data loss, time delays and clock non-synchronization, loss of correlation information, and variable lumping (loss of state variables). These errors describe the types of inaccuracies that occur during data model construction using a distributed sensing network. The proposed bounds are then used to compute using linear programming (LP) the lumping level of the local models (e.g., number of lumped variables) and the parameters of the networked sensing platform, including data communication paths and bandwidth rates. The computed parameters are utilized to set threshold values used by local schemes (at the embedded sensing nodes) to decide the specific modeling actions (see Figure 2.10).

The novelty of the work is in that it emphasizes the robustness of the created data models (ODEs) by tackling the connection between modeling error and model characteristics, physical environment dynamics, and sensing platform resources. Traditional aggregation methods [28, 33, 52] focus on reducing communication traffic to lower power and energy consumption. Feedback-based adaptation addresses the changing traffic conditions and time requirements of data aggregates sent by parent nodes to their children over a tree network [31]. Aggregation also avoids transmission of redundant data [32]. The method in [54] defines a centralized threshold-OR fusing rule for combining sensor samples under normally distributed, independent additive noise conditions. Xue et al. [51] propose a locally weighted fusion function for improving model accuracy. A cluster-based technique is proposed in [38] to perform structural health monitoring. Data aggregation is at the cluster-level by filtering spatial and semantic correlations to improve energy efficiency and reduce data storage. Particle Swarm Optimization (PSO) is used in [45] for data aggregation to minimize network cost and communication delay. The correlations between data sensed by neighboring nodes have been used to reduce the amount of data transmitted in a network [52]. The work in this paper complements these results by adding model accuracy as a main requirement of distributed model construction.

The experimental section uses the proposed data modeling procedure for thermal modeling of ULTRASPARC Niagara T1 architecture [34]. However, the method is not limited to thermal modeling and can be used for modeling other physical phenomena, like the characteristics of ocean water, e.g., salinity, temperature, and pH. There have been extensive studies on thermal modeling in three-dimensional Integrated Circuits (3D-ICs). Current techniques [37, 47] observe the thermal behavior of 3D-IC by simulating the chip as a part of the pre-fabrication validation stage. Once this behavior is characterized, thermal management through voltage and frequency scaling and task scheduling is performed [25–27]. Although sensors are distributed over the entire area of the chip, most thermal management policies use a centralized repository of the real-time thermal data. However, policies must include models for data losses and delays associated with the real data collection networks in order to address inevitable modeling errors. The proposed work presents a distributed approach to physical phenomena modeling while minimizing the effect of various model error sources.

This chapter has the following structure. Section 2.2 presents an overview of physical phenomena modeling, its associated modeling errors, and the proposed modeling methodology. Section III discusses the model error bounds, trajectory prediction, and the related optimization formulation. Section Section 2.4 details the experiments. Conclusions end the chapter.

2.2 Data Modeling and Error Minimization

This work considers data models that are *ordinary differential equations (ODEs)* over time and space for the parameters of physical environments. Parameters are strongly coupled with each other, e.g., pressure and temperature, wave propagation speed and gas density, and many more [23]. In addition to environmental characteristics, the dynamic of physical parameters is decided by mobile energy sources and sinks with variable properties. Some energy sources and sinks move along unknown trajectories. Figure 2.2(a) illustrates the concept. This section details the characterization of physical phenomena, the types of modeling errors, and the proposed modeling procedure to reduce modeling error.

2.2.1 Characterizing Physical Phenomena

Let's assume that $Y_i(t)$ are the state variables of the monitored physical environment, such as state variable Y_i corresponds to the nodes of the discretization grid used in modeling. Note that only some $Y_i(t)$ are observed (e.g., sampled) through sensors. The well-known mathematical expressions of state variables Y are as follows [35]:

$$S_i\dot{Y}_i(t) = \dot{E}_i(t) + \sum_{k\in K}\frac{Y_k(t) - Y_i(t)}{Gr_{k,i}} + \sum_{k\in K}\frac{Y_i(t) - Y_k(t)}{Gr_{i,k}} \tag{2.1}$$

K is the set of neighbors of node i. $Gr_{i,k}$ and $Gr_{k,i}$ are the gradient coefficients that define the in- and outgoing energy flows between two neighboring points i and k. The coefficients S_i represent the energy stored at node i, and the terms $\dot{E}_i$ are the added or removed energy at node i. The model parameters are dynamic because physical parameters, like temperature, density, pressure, and humidity, change in space and time [23]. Figure 2.2(b) illustrates the above mathematical expressions.

Example: Section IV present a case study for thermal modeling of integrated circuits (ICs). State variable Y is temperature (T) while coefficients S and Gr represent heat storage and heat transfer coefficients, respec-

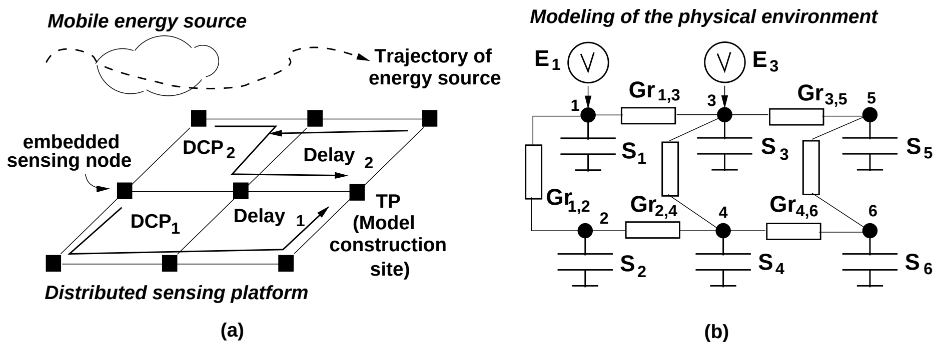


Figure 2.2: (a) Distributed sensing and (b) data model structure

tively [23, 53]. The energy injection due to power dissipation of the IC cores is denoted by variable E. Depending on the workload of the cores, the power dissipation varies causing variation in the values of variable E over time and space. Then, equation (2.1) corresponds to the following heat transfer equation:

$$S_i \dot{T}_i(t) = \dot{E}_i(t) + \sum_{k \in K} \frac{T_k(t) - T_i(t)}{Gr_{k,i}} + \sum_{k \in K} \frac{T_i(t) - T_k(t)}{Gr_{i,k}} \tag{2.2}$$

The left-hand side of the equation denotes the change in heat energy at node i in time dt. The right side indicates the two causes of this change, the energy injection (E) due to power dissipation and the energy transferred from neighboring nodes k to node i. Coefficients S vary according to the heat capacity, and coefficients Gr indicate the thermal conductivity of the material.

The energy sources and sinks in equation (2.1) can be static or mobile. Then, energy source (sink) E_i is the sum of all energy sources (sinks) that are located at node i at time moment t. The expression of E_i is as follows:

$$E_i(t) = \sum_{p \in ES} E_p(t) \delta_i(\int_t v_p(\tau) d\tau) \tag{2.3}$$

Set ES is the set of all energy sources (sinks) and v_p is the speed of source p. Function $\delta_i(x)$ is one, if $x = i$ at time moment t, otherwise it is zero.

For example, in ICs, energy sources and sinks correspond to thermal energy injection (due to power dissipation) and removal (via liquid-cooling microchannels or heat sinks), respectively. Depending on the variable workload of memories, CPU cores, buses, and other devices, these spots are observed in time at different locations on the chip and are therefore considered “mobile”.

Finding physical models means, conceptually, that the set of equations (2.1) and (2.3) formulated at each node i of the discretization grid is solved symbolically. For example, if all energy sources and sinks are static ($v_p(t) = 0$), the identified data model are the following solutions of the equation set:

$$Y_i(t) = \sum_k \alpha_{i,k} e^{\lambda_k t} + \sum_l \beta_{i,l} t^l \tag{2.4}$$

Parameters $\alpha_{i,k}$, λ_k, and $\beta_{i,l}$ depend on the parameters S_i, $Gr_{i,k}$, $Gr_{k,i}$, and $\dot{E}_{i,p}$ of the modeled process (equations (2.1) and (2.3)). The solutions indicate explicitly how cause variables influence the parameters of the model as well as the importance (sensitivities) of the variables in deciding the model.

Note that solving symbolically the differential equations (2.1) and (2.3) is difficult. For example, in the general case, the expressions of $\alpha_{i,k}$, λ_k, and $\beta_{i,l}$ are hard to compute. There are closed form solutions only for specific situations. Also, the expressions of the energy sources E_i at points i are usually unknown and their values are not directly sampled by the sensing devices. Also, the expressions of terms S_i, $Gr_{k,i}$, and $Gr_{i,k}$ might be unknown. Hence, the expressions and behavior of these unknowns must be found during model construction, e.g., through profiling and/or identification.

Another difficulty in precise data modeling stems from the correlations between the trajectories of energy sources (sinks) (e.g., sound, heat, etc.) and the communication paths used to transmit the sampled data. As shown in Figure 2.2(a), data acquired by the sensor nodes is sent along various data communication paths (DCPs) with minimum loss and within the needed timing constraints. DCPs are defined from the sensing nodes to Target Points (TPs), where the streamed data is saved. The selected DCPs influence the modeling error as they determine the experienced data loss and delays while

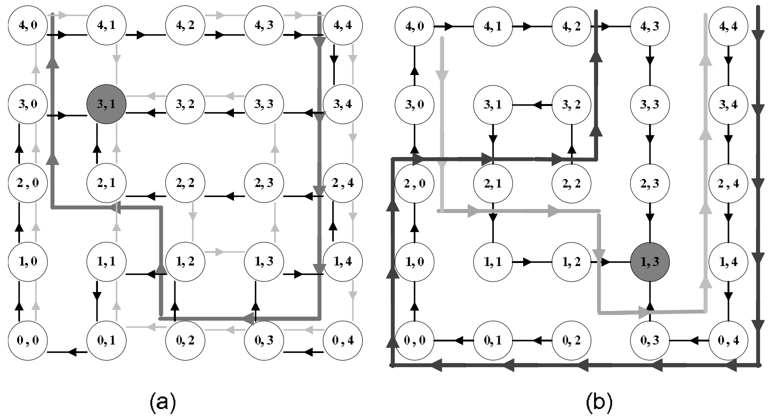


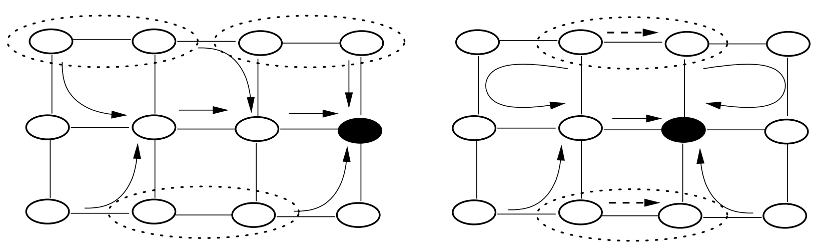
Figure 2.3: Data path configurations used in tracking trajectories

forwarding data to TPs. Data loss occurs when data stored in local buffers is overwritten before it is forwarded either because of an ongoing data sampling or data reception. This situation also increases the delay of transmitting data to TPs.

Example: Figure 2.3(a) shows two different DCPs, and the same mobile energy source trajectory (highlighted in bold) moves through the network for those configurations. The TP is the black bubble. The experimental model for the networked nodes is based on PSoC processor [21]. The first DCP does not experience any data loss for the considered trajectory. The average delay for the nodes is 1434.62 msec and the maximum and minimum delays are 2040 msec and 1010 msec, respectively. Six nodes in the second DCP experience data loss for the same trajectory. The average delay for the nodes that did not experience any data loss is 1962.86 msec and the maximum and minimum delays are 5660 msec and 1010 msec, respectively. Hence, nodes have an average delay of 36.82% more than path configuration one. Different DCPs can yield different levels of performance for the same trajectory. Moreover, Figure 2.3(b) shows two different trajectories running through the network, which uses the same path configurations. Trajectory one is shown with black line and trajectory two with grey line. Trajectory one has data loss at seven nodes compared to trajectory two which has no data loss. The nodes in trajectory one experience an average delay of 73.43% higher than trajectory two. Hence, the same DCP can yield different levels of performance for different trajectories.

2.2.2 Errors during Distributed Data Modeling

Main challenges in constructing precise data models based on samples from a distributed sensing network include minimizing the errors introduced during sensing, estimation and communication of the parameters in equations (2.1) and (2.3). The four types of errors express the inaccuracies that occur when computing locally (at each network node) the parameters S, Gr, and E in equation (2.1) based on the values sampled by a node and the data received using DCPs. Errors are grouped into the following four categories:



(a) (b)

Figure 2.4: Correlation errors

- *Errors due to data losses*: Data losses are due to buffer overwriting when streaming from sensors to the collection site for model construction (e.g., TPs). Data losses can be reduced by increasing the buffer sizes but this requires more resources. Such resources are not available for basic sensing nodes. Data loss also occurs when sensor data cannot be acquired due to hardware constraints of the sensing frontends, e.g., insufficient sampling frequency or inaccurate mixed-signal frontends [24].

- *Errors due to time delays and clock non-synchronization:* The accuracy of the models depends on the delays with which the sampled data becomes available at the TPs (see Figure 2.2(a)). The communication delays are incorporated to some degree into the data models, hence further reducing their accuracy. Transmitting more data samples to construct better models increases the delay of the communication paths. Also, there is no common clock signal available to the distributed nodes, hence there is "jitter" noise added to the data samples because of non-synchronization of the local clocks at each sensing node.

- *Path-induced errors:* If the path configuration shown in Figure 2.4(a) is used for data collection, certain correlations are missed as the nodes on a DCP do not have access to the samples sent on another DCP. Missed correlations are represented in the figure by dotted lines. The error introduced due to missed correlations is called *correlation error*. It can be minimized by selecting a path configuration such as in Figure 2.4(b).

Data modeling can use state variable lumping to reduce data loss and delays. Variable lumping eliminates the less significant state variables and simplifies the structure of the discrete representation of the state equations. Variable lumping might eliminate one or several consecutive variables. Lumping introduces the following modeling error:

- *Errors due to lumping:* During lumping, some intermediate state variables are removed. At the TP, the original data needs to be extracted from this lumped information. The inaccuracies observed in the extracted data represent the error due to lumping of state variables while forming the local models.

Assuming an additive error model, the local modeling error at node i is equal to:

$$Err_i = Err_i^{(Loss)} + Err_i^{(Delay)} + Err_i^{(Corr)} + Err_i^{(Lump)} \tag{2.5}$$

The total modeling error over all nodes i ($\sum_i Err_i$) should be minimized by an optimized modeling scheme.

2.2.3 Methodology for Creating Distributed Models

Every sensing node of the network makes local decisions in an attempt to minimize the local error described by equation (2.5). The decisions include selection of a certain DCP, the communication rate of the DCP, and performing variable lumping or not. Let's assume that the following probabilities express these decisions: $prob^{(DCP_p)}$ is the probability of using path p (from the set of available paths), $prob^{(r_{j,p})}$ is the probability of using rate j (from the set of available communication rates of DCP p), and $prob^{(lump)}$ is the node's probability to perform variable lumping. Then, finding the optimized implementation scheme requires computing the probabilities for each sensing node in the network. At run time, every sensing node uses its specific probabilities to dynamically select the most effective DCP depending on the trajectories of the energy sources (sink), configures the parameters of the DCP (e.g., baud rate), and decides whether to lump or not.

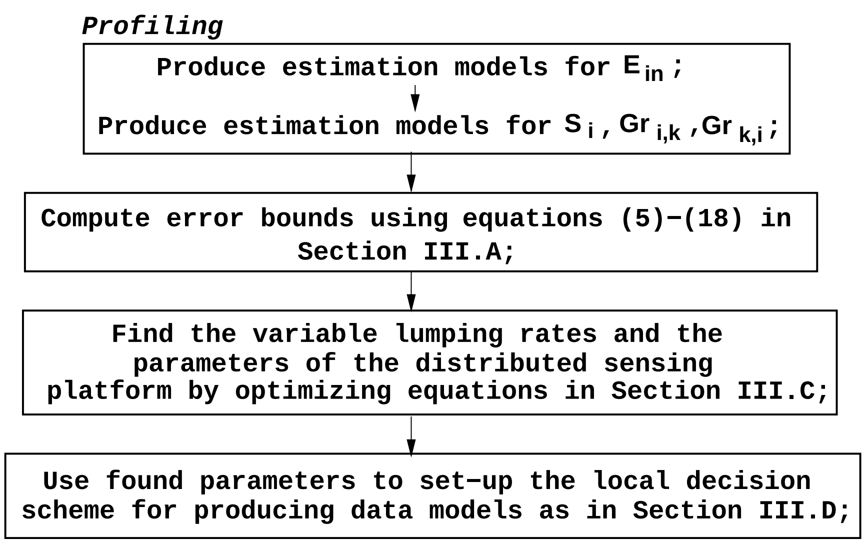


Figure 2.5: Data modeling methodology

Figure 2.5 presents the methodology to construct mathematical data models for distributed sensing platforms. The first step, profiling, uses raw data samples of the observed state variables Y (a sub-set of all state vari-

Table 2.1: Symbols used in the model and their definitions

Symbol	Definition	Symbol	Definition	Symbol	Definition	Symbol	Definition	Symbol	Definition
Y (C)	State variable	E (C)	Energy source/sink	Gr (C)	Transfer coef.	S (C)	Storage coef.	Err (O)	Total error
$Err^{(Loss)}$ (O)	Err.loss	$E^{(Delay)}$ (O)	Err.delay	$E^{(Corr)}$ (O)	Path-ind.err.	$Err^{(Lump)}$ (O)	Err.lump.	$Err^{(Buff)}$ (O)	Err.buff.loss
$Err^{(Coll)}$ (O)	Coll.err.	$error_{(k)}$ (O)	Err.discard.data	γ (P)	Coef.bounds	κ (P)	Coef.bounds	BW (R)	Comm. bandwidth
DCP (R)	Data Comm. Path	t (T)	Time	δ (R)	Time disc.step	p (O)	Lumping level	q_{sens} (R)	Sampling rate of sensors
α (O)	Rate disc.sensor data	β (O)	Rate sel.bandw.	λ (O)	Rate sel.lump.	buff (R)	buff.size	$DATA^{OUT}$ (R)	pack.size
lump_level$_j$ (R)	Lump.level	Loss (O)	Data loss	In_rate (R)	Req.resol.	NET^{IN} (O)	Input data	NET^{OUT} (O)	out. data
$Delay_p$ (O)	Delay DCP	$prob^{(DCP)}$(O)	Prob.sel.DCP	$prob^{(r)}$(O)	Prob.sel.BW	$prob^{(Lump)}$(O)	Prob.lump	x (y)	Cartesian coord.
$N_{alt\ traj}$	Number altern. traj.	l	Length traj.	V	Velocity agent	Grad	Grad. traj.	θ	Angle traj.

ables Y) to produce the initial estimates of parameters E_i, S_i, $Gr_{k,i}$, and $Gr_{i,k}$ in equations (2.1) and (3.3). The parameters are found for a large set of scenarios for the dynamics of the monitored physical entities. Table 2.1 summarizes the model variables estimated during profiling (labeled as (P)). The second step uses the profiling information to find the characteristics of the dynamics, e.g., the bounds for first and second order derivatives of Y and energy sources E_i. This insight is then utilized to find the maximum error bounds that intervene during distributed data sensing. The error types are discussed in Section III.A and captured in equations (2.5)-(2.18). The next step finds the variable lumping scheme and the parameters of the distributed sampling architecture that minimizes the errors of the data models, hence maximizes the model robustness. The parameters are computed by solving the optimization model described by the equations in Section III.C. Finally, the computed parameters are used to set-up the local data modeling decision making scheme of each embedded node. Section III.D presents the local routines.

2.3 Optimization of Distributed Data Modeling

This section introduces the algorithms and error models that are part of the data modeling methodology in Figure 2.5. Section III.A details error modeling. Section III.B discusses trajectory prediction for mobile energy sources (sinks). Section III.C describes the computation of the probabilities used in local decision making: probability $prob^{(DCP_p)}$ of using path p (from the set of available paths), probability $prob^{(r_{j,p})}$ of using rate j (from the set of available communication rates of DCP p), and probability $prob^{(lump)}$ to perform variable lumping. Table 2.1 summarizes the variables of the modeling and their meaning.

2.3.1 Detailed Error Modeling

The four error types introduced in Section II.A are modeled as follows.

i. Errors due to data loss. Data loss error is the sum of the errors due to buffer loss and collection loss. Buffer loss $Err^{(Buff)}$ represents the error due to buffer overwriting and collection loss $Err^{(Coll)}$ is the error due to the limited sensing (sampling) capabilities of a node.

$$Err_i^{(Loss)} = Err_i^{(Buff)} + Err_i^{(Coll)} \tag{2.6}$$

Lemma 2: The loss of n consecutive data values due to buffer overwriting at node i is described by the following expression:

$$Err_i^{(Buff)} \leq \frac{n}{BW}||\dot{Y}_{max}| - |Y_{n+1} - Y_0|| + \frac{n}{BW^2}||\ddot{Y}_{max}| - |\dot{Y}_{n+1} - \dot{Y}_0|| \tag{2.7}$$

BW is the data communication bandwidth at node i's input.

Proof: Using the first three terms of the Taylor series (Y_0 is the starting value), $Y(t) \approx Y_0 + \dot{Y}t + \ddot{Y}t^2$. Losing n consecutive data results in estimating the first two derivatives as $\dot{Y} \approx BW\frac{Y_{n+1}-Y_0}{n}$ and $\ddot{Y} \approx BW\frac{\dot{Y}_{n+1}-\dot{Y}_0}{n}$. The total error due to the miss-prediction of the first derivative is $\frac{1}{BW}\sum_{i=0}^{n}||Y_{i+1} - Y_i| - |Y_{n+1} - Y_0|| \leq \frac{n}{BW}||\dot{Y}_{max}| - |Y_{n+1} - Y_0||$. Similarly, the error due to the miss-prediction of the second derivative is $\frac{1}{BW^2}\sum_{i=0}^{n}||\dot{Y}_{i+1} - \dot{Y}_i| - |\dot{Y}_{n+1} - \dot{Y}_0|| \leq \frac{n}{BW^2}||\ddot{Y}_{max}| - |\dot{Y}_{n+1} - \dot{Y}_0||$.

Collection errors are introduced if data cannot be acquired fast enough due to the hardware constraints of the sensing frontends. For example, the analog-to-digital converters are too slow, or not all sampled values in the input buffers can be processed [24]. Let's assume that error Err_{sens_j} is due to discarding one physical value of sensor $sens_j$. The total error introduced by all discarded data is as follows:

$$Err_i^{(Coll)} = \sum_{\forall sens_j} \alpha_j \; q_{sens_j} \; Err_{sens_j} = \sum_{\forall sens_j} (1sec)\alpha_j \; q_{sens_j} (\sum_{\forall k} error_{(k)}) \quad (2.8)$$

q_{sens_j} is the sampling rate of the sensor (number of samples in one second). α_j is the rate of discarding values at sensor $sens_j$ ($\alpha_j < q_{sens_j}$). $error_{(k)}$ is the error introduced for tuple k by the discarded data. Term 1 sec was added to correctly represent the unit of the error.

In equation (2.6), buffer and collection errors represent conceptually the same type of errors, which are due to data packets loss (data is overwritten during communication or not sampled due to low sensor sampling rates). The unit for the errors in equation (2.6) is BU, the basic unit of the state-variable Y, e.g., degree Kelvin if variable Y is temperature.

ii. Errors due to time delays and clock non-synchronizations. These errors are introduced by the delays at which the sampled data reaches the target point (TP). Errors change equations (2.1) to $S_i\dot{Y}_i(t) = \sum_p \dot{E}_{i,p} + \sum_k \frac{Y_k(t-T_1^{Delay})-Y_i(t-T_2^{Delay})}{Gr_{k,i}}$, where T_i^{Delay} is the delay of the related data communication paths between nodes i and k. The errors change the fundamental matrix (ϕ) of the equation set (2.1), which changes the symbolic expressions of the state variables Y (equation (3.3)).

As already explained, it is difficult to compute closed form expressions for state variables Y and then estimate the error based on the differences in the fundamental matrix due to time related errors. Instead, the local errors introduced by time delays are characterized by the differences in the time delays of any data communication paths(DCPs) s and t that converge at node i (see Figure 2.2(a)):

$$Err_i^{(Delay)} \sim \sum_{\forall s,t \in DCP_i} |T_s^{Delay} - T_t^{Delay}| \quad (2.9)$$

DCP_i is the set of all communication paths converging at node i.

iii. Path-induced errors. These errors are introduced by the missed data correlations due to the sampled data being sent using different DCPs.

The error is estimated as follows:

$$Err_{b,path} = \sum_{\forall b} \sum_{\forall p}^{NL} \sum_{c \,\in\, p} error_{(p)} \tag{2.10}$$

b is a pair of neighboring nodes i and k that belong to different paths. $error_{(p)}$ is the error introduced due to the missing tuples c at all lumping levels p of node k (assuming that data is transmitted from node i to node k).

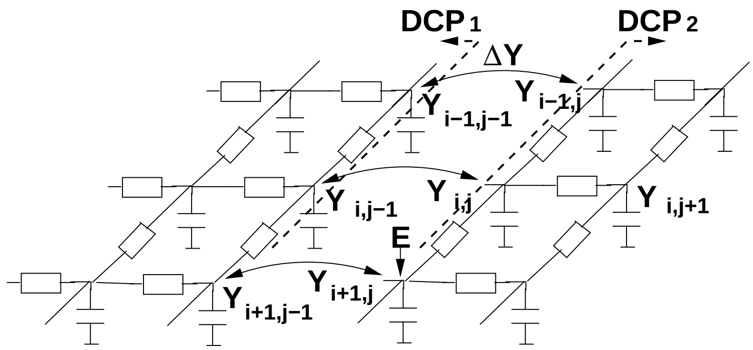


Figure 2.6: Path-induced error estimation

The path configuration used for data collection eliminates the coupling gradient coefficients for some of the nodes in the network. Figure 2.6 shows a network of transfer and storage elements modeling state variable Y expressed by equation (2.1) (the resistive elements represent parameters Gr and the capacitive elements correspond to parameters S). The Y values sampled along column $j-1$ are communicated using DCP_1, while the values sampled along column j are sent using DCP_2. As the values sent using DCP_2 are not available to the nodes on DCP_1, any partial models built during data communication do not include the correlations between the neighboring state variables, e.g., values $Y_{i,j-1}$ and $Y_{i,j}$. Hence, any decision made using the partial models does not consider the coupling gradients between columns j and $j-1$. For example, if there is a large energy source at node $Y_{i+1,j}$, without the correlation information, the partial models would suggest that there is an energy source at node $Y_{i+1,j-1}$, therefore some action (e.g., cooling) should be targeted at that point. As shown in Figure 2.6, correlation errors are modeled by disconnecting columns j and $j-1$ in the network model.

Lemma 3: The expression for path-induced error at lumping level 2 in Figure 2.7 is given by the following expression:

$$Err_{i,corr} = \frac{1}{\gamma\ G_{r_{a,i}}}\ [expr_{x,corr\ 1} - expr_{x,corr\ 2}] \tag{2.11}$$

$$expr_{i,corr\ 1} = \gamma\ S_i^{(equiv)}\ G_{r_{a,i}}\ \dot{Y}_i(t) + (\gamma+1)\ Y_i(t) \tag{2.12}$$

$$expr_{i,corr\ 2} = \gamma\ Y_A(t) + Y_B(t) + \gamma\ G_{r_{a,i}}\ E_i^{(equiv)}(t) \tag{2.13}$$

γ is the ratio of the coefficients Gr from neighboring nodes.

Proof: After decoupling the network along the nodes on the communication path, the equivalent coefficient parameters G_i^{equiv} and S_i^{equiv} are measured from the decoupled node to the actuator. The error associated with decoupling the columns $j-1$ and j is proportional to the flux through the removed gradient variables, which is computed by applying Kirchhoff's law at node i. The derivation leads to equation (2.11), where $Y_A(t)$ and $Y_B(t)$ are state variables at nodes A and B respectively, $\dot{Y}_i(t)$ is first order derivative at node i, and $E_i^{(equiv)}(t)$ is the external input energy source incident at the node.

Lemma 4: The path-induced error at state variable Y_i over W samples is bounded by the following expression, where $bound_{W,corr} = Err_{x,corr}(t) - Err_{x,corr}(t - W\ \delta)$:

$$bound_{W,corr} \leq \frac{W\ \delta}{\gamma\ G_{r_{a,i}}}\ [bound_{W,corr\ 1} - bound_{W,corr\ 2}] \tag{2.14}$$

$$bound_{W,corr\ 1} = \gamma\ S_i^{(equiv)}\ G_{r_{a,i}}\ \ddot{Y}_{i,MAX} \tag{2.15}$$

$$bound_{W,corr\ 2} = \gamma\ \dot{Y}_{A-i_{MIN}} + \dot{Y}_{B-i_{MIN}} + \gamma\ G_{r_{a,i}}\ \dot{E}(equiv)_{i,MIN} \tag{2.16}$$

$\ddot{Y}_i^{MAX}$ is the maximum of the second derivative of the state variable Y, $\dot{Y}_{A-i_{MIN}}$ and $\dot{Y}_{B-i_{MIN}}$ are the minimum first order derivative of the difference in sensor readings at i with respect to nodes A and B. These values are estimated using profiling.

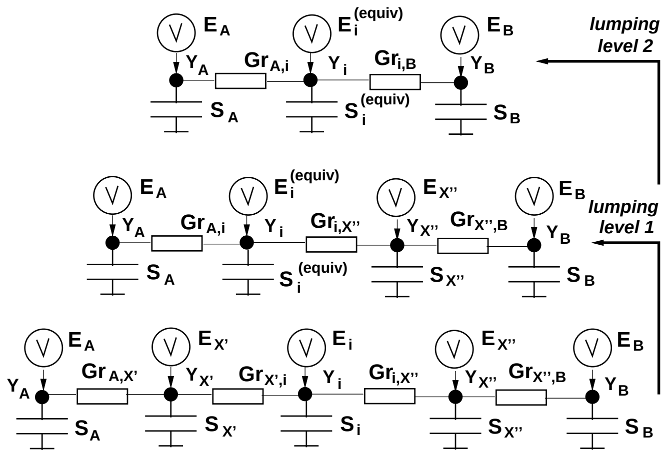


Figure 2.7: Multi-level variable lumping

iv. Errors due to state variable lumping. These errors are generated by removing state-variables during lumping. The bottom part of Figure 2.7 shows the variables sampled at five nodes Y_A, Y_X', Y_i, Y_X'', and Y_B and the corresponding model for equation 2.1, including transfer elements (G_r), storage elements (S), and energy injection elements (E). During lumping, at level 1, variable Y_X' is removed and the equivalent parameters $Gr_{A,i}$, $S_i^{(equiv)}$, and $E_i^{(equiv)}$ are computed and connected to node Y_i as shown in the figure. Similarly, at level 2, the procedure is repeated to remove variable Y_X''. The model on top approximates the original model shown at the bottom of the figure. Assuming that data is sent from node Y_A to node Y_B, the described lumping procedure is performed at node Y_B.

Lemma 5: The lumping error at node i is expressed as follows:

$$Err_i^{(Lump)} = \kappa|\gamma Y_A(t) + Y_B(t) - (\gamma+1)Y_i(t-\delta) - (\gamma+1)\frac{\dot{E}^{(equiv)}(t)}{S_i}| \quad (2.17)$$

where $\kappa = \frac{\gamma S_i Gr_{AB}}{(\gamma+1)^2\delta + S_i Gr_{AB}}$. $E^{(equiv)}$ models the equivalent external input energy. δ is the time discretizing step. γ is the ratio of $Gr_{i,B}$ and $Gr_{A,i}$.

Proof: The expression for lumping error at a node i is derived by subtracting the expressions for the discretized ODEs of $Y_i(t)$ before lumping from the expression after lumping ($Err_i^{(Lump)} = |\Delta Y_i(t)|$).

The following bounds exist for the lumping errors in equation (2.17).

Lemma 6: The maximum increase in lumping errors for lumping state variable Y over M samples is as follows:

$$Err^{(Lump)} \leq \kappa M\delta(bound_1^{(Lump)} - bound_2^{(Lump)}) \quad (2.18)$$

where, $bound_1^{(Lump)} = (\gamma+1)\delta\ddot{Y}_i^{MAX} + \gamma\dot{Y}_{A-i}^{MAX} + \dot{Y}_{B-i}^{MAX}$ and $bound_2^{(Lump)} = \frac{(\gamma+1)\delta}{S_i}\dot{E}_i^{(equiv),MIN}$.

$\ddot{Y}_i^{MAX}$ is the maximum of the second derivative of state variable Y, $\dot{Y}_{A-i}^{MAX}$ and $\dot{Y}_{B-i}^{MAX}$ are the maximum first order derivative of the difference in sensor readings at i with respect to nodes A and B. ($Y_{A-i}(t) = Y_A(t) - Y_i(t)$, and $Y_{B-i}(t) = Y_B(t) - Y_i(t)$) $bound_M^{(Lump)} = error_{Y_i}(t) - error_{Y_i}(t - M\delta)$. The values for $\dot{Y}^{MAX}$, $\ddot{Y}^{MAX}$, and $\dot{E}_i^{equiv,MIN}$ are found through profiling.

The computing of the error bounds due to buffer loss, path-induced errors, and lumping errors assumed that the dynamics of state-variable Y in equation (2.1) can be expressed with a reasonable accuracy using the first three terms of their Taylor series expansion. Taylor series expansion has been a popular method to approximate systems with weak nonlinearities, like analog circuits [42] and mechanical systems [50]. Other popular techniques for describing parameter dynamics include Volterra series [43] and nonlinear model order reduction [41]. Figure 2.12 plots the lumping and correlation error bounds for four different scenarios presented in Section IV.B. The bounds change in time (e.g., iteration) corresponding to the repositioning of energy sources, i.e. moving heat sources in the case of data sets 2 and 4. Moreover, Figure 2.13 shows that there is a small error between the actual thermal data from the 3D-ICE simulator [46] and the data map constructed using the method based on the proposed approximation model of the bounds. However, if the modeled parameters have stronger nonlinearities then methods based on Volterra series can be used to compute the error bounds.

2.3.2 Mobile Energy Source/Sink Trajectory Prediction

As explained in Section II, model errors also occur due to the correlations between the trajectories of mobile energy sources (sinks) and the data communication paths (DCPs). Trajectory prediction algorithms help in reducing errors by minimizing data loss and delays of data communications through optimized selection of data communication paths (DCPs). In addition to our previous method [48], this work presents a new algorithm for predicting the trajectory of energy sources (sinks). The two algorithms differ depending on their assumptions on the trajectory characteristics. The first algorithm is for trajectories that pass stochastically through a set of bounded regions. This algorithm was also presented in [48]. The second method assumes trajectories with both quasi-static parts and parts in bounded regions.

i. Stochastically Bounded Trajectories. The algorithm uses the concepts of bounded trajectory and stochastically-bounded trajectory. As explained in [48], a *bounded trajectory* refers to a mobile energy source's physical trajectory that is located inside a region defined by minimum and maximum gradients. The region is called Trajectory Approximating Region (TAR). Then, the trajectory is approximated as sequences of convex - concave fragments with bounded gradients of known ranges. The point separating each successive convex - concave fragments is called inflexion point. The average time distance ΔT between inflexion points is known.

Example: Figure 2.8(a) illustrates a trajectory expressed in this way. The gradient of the convex fragment is in range $[Grad_{Min}^{convex}, Grad_{Max}^{convex}]$. At time T_1, there is a break of the two dashed lines as the trajectory switches to the concave part. The gradients of this part are in range $[Grad_{Min}^{concave}, Grad_{Max}^{concave}]$. The trajectory in the figure is well approximated by the corresponding TAR.

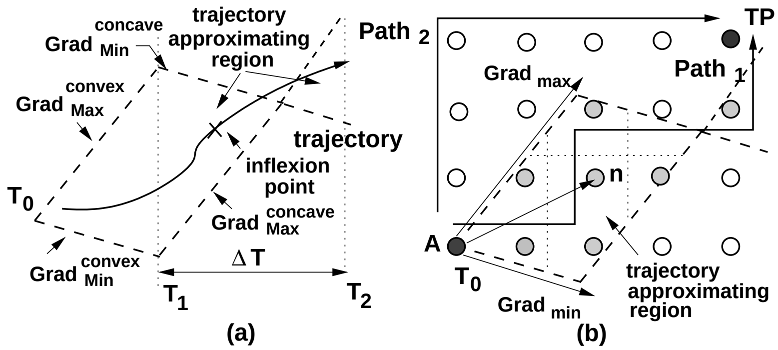


Figure 2.8: Trajectory description using bounded trajectory model

Probability p_n that node n, inside TAR, samples the agent's trajectory is estimated as follows. Let's consider a discretization of TAR into sub-regions, as shown with dotted line in Figure 2.8(b). Probability p_n depends of the unknown length $l_{trajectory}$ of the trajectory inside the sub-region containing the node, the area $Area_{sub-region}$ of the sub-region, and the number $N_{alt\ traj}$ of alternative trajectories that are inside the sub-region and pass through the node [48]:

$$p_n \propto \frac{l_{trajectory}}{Area_{sub-region}}\ N_{alt\ traj} \approx \frac{\int_{\Delta T}\sqrt{x(t)^2+y(t)^2}dt}{Area_{sub-region}}\ N_{alt\ traj} \qquad (2.19)$$

or

$$p_n \propto \frac{\int_{\Delta T}\sqrt{x_0^2+y_0^2+2(Grad_x+Grad_y)t}\ dt}{Area_{sub-region}}\ N_{alt\ traj} \qquad (2.20)$$

$x(t)$ and $y(t)$ are the unknown equations describing the agent's trajectory. x_0 and y_0 are the coordinates of the agent at time T_0, and $Grad_x$ and $Grad_y$ are the gradients at time T_0 of the trajectory along the two axes. The number of alternative trajectories $N_{alt\ traj}$ can be estimated based on (i) the angle defined by vector $\overrightarrow{(A,n)}$ from node A (the node currently considered) to node n and the vector corresponding to $Grad_{min}$, and (ii) the angle defined by vector $\overrightarrow{(A,n)}$ and the vector corresponding to $Grad_{max}$. The larger the product of the two angles, the higher is $N_{alt\ traj}$:

$$N_{alt\ traj} \propto \widehat{\overrightarrow{(A,n)}, \overrightarrow{Grad_{min}}} \times \widehat{\overrightarrow{(A,n)}, \overrightarrow{Grad_{max}}} \tag{2.21}$$

Two approximations were introduced for expressions (2.20) and (2.21) to reduce the effort of computing probability p_n by the sensing nodes. First, we considered that all nodes of a TAR are equally likely to be part of the trajectory. Therefore, expression (2.20) has the same value for all nodes, and is computed using the area value for the entire TAR without any discretization. Then, N_{alt} is the same for all nodes and can be eliminated from the expression. Second, we assumed that an agent's trajectory can have any gradient inside TAR, hence, in the worst case, it coincides with the data communication path (DCP) inside the region. Then, probability p_n can be estimated as follows:

$$p_n \propto \frac{1}{length_{n \in DP_i}} \tag{2.22}$$

DP_i is the DCP containing node n. $length_{n \in DP_i}$ is the length of DP_i inside TAR.

Example: In Figure 2.8(b), DCP $Path_2$ has no nodes inside the regions, hence the probability of its nodes sampling the trajectory is lesser than for the nodes of $Path_1$.

Counting the number of path nodes inside a TAR requires a low computing effort. For windows of size one, expression (2.20) is proportional to the cosine of the angle between the trajectory and the DCP at node A.

Moreover, *stochastically-bounded trajectory* are bounded trajectories in which the change to another fragment follows a stochastic rule. Figure 2.9(b) shows a trajectory that is described by four fragments, each being characterized by specific gradient ranges and average time ΔT of switching to another fragment. The switching between different fragments is modeled as a Markovian process. Let's denote x_i the steady-state probability of fragment i, and $t_{i,j}$ the transition rates between the fragments. The values of $t_{i,j}$ are found through observing various trajectories through the same region. The steady-

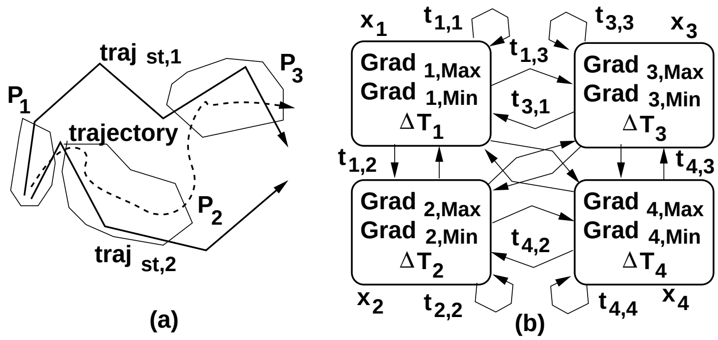


Figure 2.9: Trajectory description using stochastically bounded trajectory model

state probabilities can be computed by solving the following equations:

$$t_{i,i}x_i - \sum_{j \neq i} t_{j,i}x_j = 0 \tag{2.23}$$

and

$$\sum_{\forall k} x_k = 1 \tag{2.24}$$

Using the same reasoning as for bounded trajectories, probability p_n in expression (2.20) is updated as follows:

$$p_n \propto x_i \ \frac{\int_{\Delta T} \sqrt{x_0^2 + y_0^2 + 2(Grad_x + Grad_y)t} \ dt}{Area_{sub-region}} \ N_{alt\ traj} \tag{2.25}$$

or the approximation in equation (2.22) is changed to expression:

$$p_n \propto x_i \ \frac{1}{length_{n \in DP_i}} \tag{2.26}$$

ii. Adaptive trajectory prediction. The second method adapts its prediction based on the current state and the rate of change of trajectory parameters. If the rate of change of velocity or angle changes beyond a range, the trajectory has entered another state. For example, a higher rate of change of angle causes the trajectory to loop. A lower rate of change of angle causes an almost linear trajectory.

Using the above definition of a trajectory state, the prediction method implements the following steps. If the rate of change of angle and velocity over time has been constant or within a small range (e.g., to support noise margins), the trajectory is assumed to remain in the same state. A new state is defined by changes of velocity ΔV and angle $\Delta\theta$ over time. As shown in equation (2.27), the probability p_{state} for which a trajectory remains in the same state can be computed based on the time for which it has been in a given state. That depends on the number $n_{traj_{state}}$ of previous events for which the trajectory assumed the same state and the number of deviations of current values of velocity $n_{\delta(\Delta V)}$ and angle $n_{\delta(\Delta\theta)}$ with respect to the current state. Hence,

$$p_{state} \propto \frac{n_{traj_{state}}}{n_{\delta(\Delta V)} + n_{\delta(\Delta\theta)}} \tag{2.27}$$

For higher probabilities, predictions correspond to longer future trajectory segments, and define time $t_{predict}$ over which the trajectory can be predicted accurately:

$$t_{predict} \propto p_{state} \tag{2.28}$$

$t_{predict}$ depends on the current state of the trajectory e.g., rate of change of velocity and angle, and the current absolute values of angle and velocity.

2.3.3 Computing the Parameters of Decision Making Policies

The procedure computes the amount of sensing and lumping at every sensing node and the parameters of the networked sensing platform, so that a cost function including the overall error and propagation delay is minimized. The computed parameters for the networked sensing platform include the path utilization rates and bandwidths of the data communication paths (DCPs), and are used to set-up the local decision making schemes of the nodes (Figure 2.10). An earlier version of the method was presented in [28].

Table 2.1 summarizes all the parameters of the linear programming description, including their symbols and description. Also for each symbol we indicated the way in which it is found: (P) indicates that the parameter is found during profiling, (C) means that the parameter is computed by the local decision scheme (Subsection III.D), (R) means that the parameter has a known value (e.g., performance requirement or resource size), and (O) indicates that the parameter is computed by solving the LP formulation of the optimization scheme. Otherwise, the parameters are part of the mathematical analysis used to compute error bounds or trajectory prediction.

The details of the LP formulation are presented next.

i. Cost function. The LP equations are solved along with the cost function to compute the utilization rates: α for discarding sensed values (equation (2.8)), λ for different lumping levels, β for bandwidth values, and $Path$ for DCPs.

$$\min \sum^{\forall p \in S} \{\zeta Err_p^{(Loss)} + \xi Err_p^{(Delay)} + \eta Err_p^{(Corr)} + \theta Err_p^{(Lump)} + \mu Delay_{av_p}\} \tag{2.29}$$

where, S is the set of all DCPs. Parameters ζ, ξ, θ, μ and η are user-defined weights that reflect different importance assigned to the various modeling error types and delay.

The following equations describe the LP optimization model. The equations are specific to data communication path DCP_j with different path segments ($p_x \in DCP_j$).

ii. Errors due to data loss ($Err^{(Loss)}$). Data loss due to buffer overwriting ($Err^{(Buff)}$) occurs at node x when the input rate is higher than the output rate and the difference is more than the available buffer size $buff$. The data loss is equal to:

$$Loss_x = |(In\_rate_x + NET_x^{IN}) - NET_x^{OUT} - buff| \tag{2.30}$$

$buff$ is the local buffer size. The average sensing rate of node x (In_rate_x) is equal to the required sampling resolution. NET_x^{IN} is the amount of data input at node x from other nodes in the path. The amount of data output from node x is NET_x^{OUT}.

$$NET_x^{IN} = \sum_{j \in Pred} NET_j^{OUT} \tag{2.31}$$

$Pred$ is the set of all nodes that precede immediately node x on the used data communication paths (DCPs).

NET_x^{OUT} depends on the amount of lumping at node x and all previous nodes pv in path segment p_x of the DCPs that lead to node x:

$$NET_x^{OUT} = prob_x^{(Lump)} DATA_x^{OUT} + \sum_{\forall pv \in p_x} prob_{pv}^{(Lump)} DATA_{pv}^{OUT} \tag{2.32}$$

$DATA_x^{OUT}$ is the size of the packet sent out by node x for the locally lumped data model.

The error due to buffer loss $Err^{(Buff)}$ is expressed using equation (2.7) in which n is replaced by variable $Loss_x$. Collection errors ($E^{(Coll)}$) are described as in equation (2.8). Equation (2.6) describes the total error $E^{(Loss)}$ due to data loss at node x.

iii. Path delay ($Delay_{av}$) and errors due to time delays ($Err^{(Delay)}$). The delay of path p_x is the sum of the average execution time of all nodes x along the path plus the average time for transmitting the output data of each node:

$$Delay_{p_x} = \sum_{\forall x \in p_x} (Exec_x + Delay_x^{out}) \tag{2.33}$$

$Exec_x$ is the execution time of the primitives at node x.

The average time for transmitting the output data NET_x^{OUT} of node x

is expressed as follows:

$$Delay_x^{out} = NET_{x,MAX}^{OUT} \sum_{\forall j} \beta_j \frac{1}{BW_j} \tag{2.34}$$

β_j is the rate of using bandwidth BW_j for the output link and $\sum_{\forall j} \beta_j = 1$. $NET_{x,MAX}^{OUT}$ *is the maximum* NET_x^{OUT} *at node* x *found during profiling. The worst-case value for* NET^{OUT} *had to be considered in order to keep the model linear.*

The average delay ($Delay_{av}$) is the average of the delays $Delay_{p_x}$ for all paths p_x used for communicating the sensed data to the data construction site.

The error due to time delay $Err^{(Delay)}$ is described using expression (2.9) in which T^{Delay} is replaced by variables $Delay_{p_x}$.

iv. Correlation Error ($Err^{(Corr)}$). The correlation error is given by the equation below:

$$Err^{(Corr)} = \sum_{\forall i \in DCP} Path_i \; Err_i^{(Corr)} \tag{2.35}$$

where $Err_x^{(Corr)}$ are computed using equation (2.14).

v. Lumping Error ($Err^{(Lump)}$). The lumping error is given by the equation below:

$$Err^{(Lump)} = \sum_{\forall x \in DCP} (prob_x^{(Lump)} Err_x^{(Lump)}) \tag{2.36}$$

where $Err_x^{(Lump)}$ are computed using equation (2.18). $prob_x^{(Lump)}$ is the probability that node x is lumped, and is given by the following equation:

$$prob_x^{(Lump)} = \sum_{\forall j} \lambda_j lump\_level_j \tag{2.37}$$

where $lump\_level_j$ corresponds to the lumping level, which is defined by the percentage of lumped nodes in p_x. λ_j is to the utilization rates of using $lump\_level_j$. $\sum_{\forall j} \lambda_j = 1$.

```
make_decision(Err_total, W, λj, βm , Path k ){
  while (true) {
    if (equation (38) is true) {
      select lumping at level j with rate λ j;
      update total error up to current time;
    }
    estimate mobile source trajectory using
      method in Section III.B;
    for(DCP which overlap less with the predicted trajectory){
      select DCP k with rate Path k ;
      select communication BW m with rate βm ;
    }
  }
}
```

Figure 2.10: Local decision making routine

2.3.4 Local Decision Making Routine

The local decision making routine at a node is shown in Figure 2.10. It uses as input parameters the computed values for the rates $Path_k$ of using DCP k, β_m of using different bandwidths BW_m for DCP, rate λ_k of employing lumping hierarchy level k at a node i, and the estimated minimum error Err_{total}. Equation (2.38) is used to decide locally, if node i performs lumping or not. This decision is based on the following lemma:

Lemma 1: Node i performs variable lumping at current time $T^{(curr)}$, if the following constraint is met:

$$\frac{Err_i^{(prev)} + Err_i^{(k)}}{\mathbf{E}[Err_i]} < \frac{W - T^{(curr)}}{W} \tag{2.38}$$

Otherwise, there is no lumping at time $T^{(curr)}$. $\mathbf{E}[Err_i]$ is an estimation of the minimum total error at sensing node i over time window W. The estimate is computed by the optimization in Section III.C. $Err_i^{(prev)}$ is the error due to previous lumping. $Err_i^{(curr)}$ is the error introduced by the current state lumping.

Proof: $\mathbf{E}[Err_i]$ represents the lower bound of the error that can be achieved by the sensing node (according to the optimization in Section III.C).

Assuming a uniform distribution of the lumping error, the total error in the remaining time of the current time window W (i.e. $W - T^{curr}$) is $\mathbf{E}[Err_i]\frac{W-T^{curr}}{W}$. Hence, the decision making procedure should select to lump or not, so that the difference between the bound $\mathbf{E}[Err_i]$ and the sum $Err^{(prev)} + Err_i(k) + \mathbf{E}[Err_i]\frac{W-T^{curr}}{W}$ is minimized. This proves the lemma.

Conceptually, the local decision making procedure at node i tracks the lower bound $\mathbf{E}[Err_i]$.

Next the decision making routine estimates the trajectory of mobile energy sources or sinks using one of the algorithms presented in Section III.B. The DCPs with least overlapping with the trajectories are selected as they reduce the data loss along the DCPs (see Section II.A and Figure 2.3). From the selected DCPs, the actual DCP k used in communication is chosen based on the computed rates $Path_k$. The bandwidth of the chosen DCP is set to value BW_m based on the rates β_m.

The LP formulation is solved offline, as shown in Figure 2.5, while trajectory prediction and lumping level selection are executed online. The energy consumption and time delay due to these computations is small compared to the cost associated with data communication. Apart from generating and sending its own data packets, each node receives data from other nodes, and forwards it towards the target point along the selected DCP. Due to state-variable lumping, data traffic and communication is reduced leading also to reduction in average energy costs. In a sampling-processing-communication iteration for unoptimized case, up to 70.84% of the time is spent on communication by a node in a network of size 25. In the optimized case, the average reduction in traffic compared to unoptimized case up to 25.69%.

2.4 Experiments and Results

Experiments were performed to verify the accuracy and efficiency of the discussed modeling scheme. An overview of the simulation framework is shown in Figure 2.11. A temperature-sensing network was modeled in SystemC to describe accurately the timing, resolution, bandwidth, and local memory of

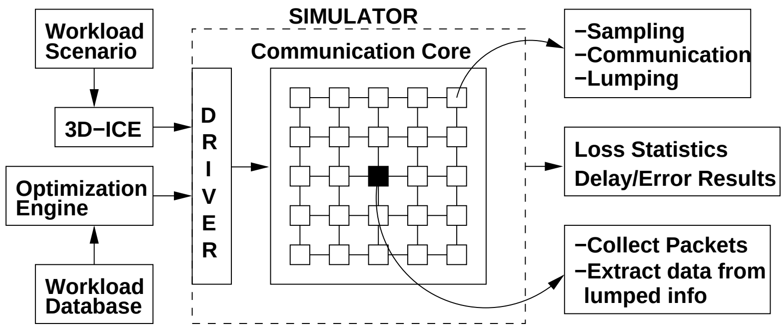


Figure 2.11: Experimental framework

reconfigurable mixed-signal PSoC processor [21]. PSoC processor includes an 8-bit processor operating at 24 MHz, 2k SRAM and 64k EPROM on-chip memory, programmable clocks, and serial data communication (SPI and UART) with programmable bandwidths. This Simulator has a communication core, which consists of a grid network of embedded nodes that perform various tasks, such as sampling, communication, and generation of local models. One of the nodes in the network, is designated as the target point (TP), and all other nodes are connected to this TP via predefined path configurations. The TP receives data packets, extracts temperature information from model parameters, and builds the resultant thermal map. Loss, delay and error statistics are then extracted from the information received by the TP.

2.4.1 Trajectory Prediction and Optimization

The two trajectory prediction algorithms were studied using six trajectories defined such that they cover the entire data communication network. The trajectories are presented in [48], and cover different trajectory directions, velocity values, angle ranges, and rates of change of velocity and angle. The data communication paths (DCPs) were defined such that there are three unique path configurations for each of the four target points. Hence, the experiments used a total of twelve path configurations.

Table 2.2: Summary of the two trajectory prediction algorithms

		Average Data Loss Improv.		Average Delay Improv.(%)	
Trajectory	Events	A1	A2	A1	A2
1	20	0.75	0.75	-8.50	26.00
2	21	1.00	1.00	0.75	20.00
3	25	1.00	1.00	2.50	21.75
4	30	4.00	4.50	3.75	18.50
5	26	1.25	1.25	-8.00	18.75
6	74	6.25	7.75	-16.75	12.50

Both algorithms, stochastically bounded trajectories method (A1) and adaptive method (A2), offer an improvement in data loss as compared to the best-case static path configurations, in which all nodes in the network are static. For the adaptive algorithm (A2), in most cases, the data loss is zero and it is close to zero for A1. The maximum data loss was four samples for A1, two samples for A2, and 33 out of 74 events for static path configuration. Both algorithms provide improvements in average delay as compared to static path configurations. Improvements are as high as 38% for A1 and 40% for A2, but there are instances in which the resulting delay is higher for the two methods as compared to static paths.

Table 2.2 summarizes the improvement and percentage improvement in average data loss and average delay, respectively for the six trajectories compared to the best-case static path configurations. Trajectories 4 and 6 are the longest trajectories in terms of number of events generated. For both trajectories, algorithms A1 and A2 produce significant reduction in data loss, but only method A2 results in reduced delay. For trajectories 1, 2, 3 and 5, both algorithms achieve small improvement in data loss over the best-case static configuration results. Algorithm A2 causes reduction in average delay for all these trajectories, while A1 manages to do that only for trajectories 2 and 3. The results suggest that algorithm $A2$ is superior to algorithm $A1$.

2.4.2 Data Modeling

Experimental set-up: The improvements in data loss, modeling error, and latency were recorded and compared with the unoptimized case. The scalability of the methods was tested by running simulation using three network sizes: 25 nodes, 64 nodes, and 100 nodes. Four different data communication path (DCP) configurations between nodes were used for each of the 3 network sizes.

The simulator selects an appropriate configuration for each iteration based on results of the optimization engine. For the unoptimized case, the path utilization rate was set as 25% for each path configuration. So, each configuration had an equal probability of being selected. The output of the optimization engine was also used to select the lumping thresholds and the communication bandwidth. For the unoptimized case (used as a reference), the bandwidth utilization rate was fixed as 33.33% for each bandwidth value. For the optimized case, the results of the minimum error estimation step were used to set probabilities β_j of the three bandwidth values.

In the optimization engine, weights θ, μ and η in the cost function were set to reflect different importance assigned to modeling error, delay and correlation error: OptRatio1 $(\theta/\mu/\eta) = 0.6/0.2/0.2$, OptRatio2 $= 0.2/0.6/0.2$, and OptRatio3 $= 0.2/0.2/0.6$. Depending on which ratio is being used, the probability of using a particular communication rate changes. Weights ζ and ξ were set to 0. For example, OptRatio2 assigns more importance to reducing latency, hence increasing the probability of using the fastest communication rate.

Thermal modeling was used a case study for distributed data modeling, even though the technique can be also used for other data modeling situations. The thermal data was generated for the ULTRASPARC Niagara T1 architecture [34]. Experiments were performed using eight datasets, which represent different workload scenarios resulting in stationary, fluctuating and moving hotspots. The temperature behavior was found for the datasets using the temperature simulator 3D-ICE [46].

Model validation. The bounds for lumping error and correlation er-

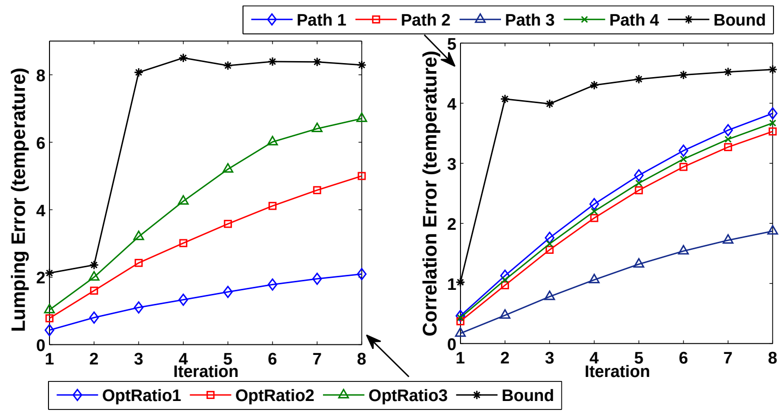


Figure 2.12: Bounds on lumping (left) and correlation errors (right)

ror were estimated using equations (2.18) and (2.14), respectively. The thermal data was generated through profiling of various workload scenarios. The bounds were tested for each of the datasets used in the experiments for all DCPs and OptRatios. The value of correlation error depends on the selected communication path and the dataset. While the magnitude of the bound depends on the dataset, the shape of the curve (over time) depends on the configuration of the DCP. Among all the datasets, Dataset 6 has the highest average correlation error. Figure 2.12(right) shows the correlation error for Dataset 6 for all DCPs along with the bounds. The bounds were recomputed at every iteration. Low values of the bounds correspond to state variables with small values, while large bounds reflect large state variables. The slope of the bound plots describes the rate of change of the state variables. Path 1 has the highest error compared to the other DCPs because it has longest path segments. The two main factors that affect the value of lumping error are dataset and optimization ratio (OptRatio). Dataset 3 has the highest lumping error compared to other datasets. The lumping error for all OptRatios for Dataset 3 along with the bounds is shown in Figure 2.12(left). The shape of the curve for the bound depends on the lumping level while the magnitude depends on the dataset. The error is highest for OptRatio3 since it performs the highest amount of lumping. Note that if the number of iterations increases, the bounds would adjust accordingly by moving up. Hence, the actual errors values would not exceed the bounds.

The validation of S and Gr coefficients was performed for the unoptimized case as well as optimized case for all OptRatios and all DCPs using a large data set for simulated workloads. First, the sensed temperature data is converted to equivalent storage (S) and gradient coefficients (Gr) at each node. At the target point, the original thermal information is extracted back from these coefficients. For example, for Dataset 8, the average error associated with these conversions is 2.5% for the un-optimized case and 3.0% for the optimized case. Figure 2.13 shows a comparison between the actual thermal map sensed by the sensor nodes versus the thermal map extracted from the S and Gr coefficients. The two maps are very similar.

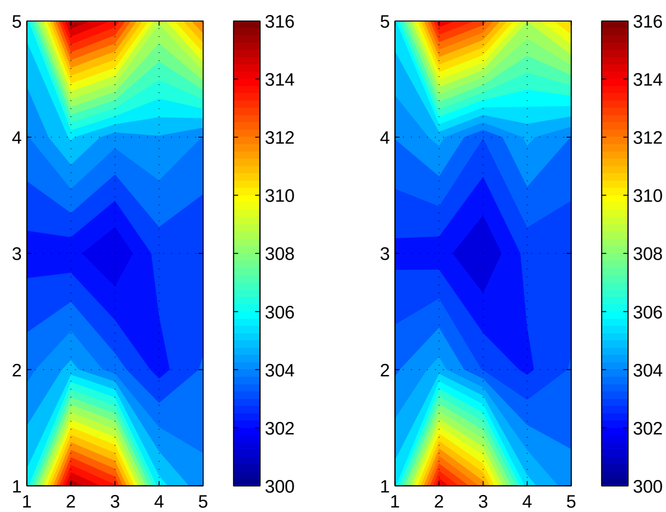


Figure 2.13: Complete thermal map (left) and thermal map constructed using the data model (right)

Results and analysis: The results for Dataset 2 (two moving heat sources) and Dataset 6 (six stationary heat sources) for different network sizes are detailed. Then an overview of results for all datasets and network sizes is summarized.

The following parameters were extracted from the simulation results: (i) *data not sampled* gives the number of instances when the node failed to sense the temperature data because it was busy forwarding packets from previous nodes on the path; (ii) *buffer loss* is the loss of data because the packet got overwritten in the buffer due to incoming packets; (iii) *lumped data* gives the loss of packets because the node was lumped; (iv) *data received* is the number of packets received at the target point; (v) *avg. delay* is the average delay associated with packets that were received at the model construction node; and (v) *avg. error* is the average value of lumping error for the packets that could not reach the model construction node.

Dataset 2. The experimental results for Dataset 2 are as shown in Figures 2.14 and 2.15. The x axis in Figure 2.14 gives the network size. The y axis shows the number of packets in terms of percentages. The 'data not sampled' and 'buffer loss' represent undesirable losses as they cause uncontrollable errors for the data models. 'Lumped data' represents controllable loss as its impact is factored in the final error of the data models.

For the 25 node network, the unoptimized case has the highest collection loss and buffer loss as compared to any of the optimized cases. Also, the delay and error is much larger for the unoptimized case. The amount of lumping increases if the cost function assigns higher priority on reducing latency and lower importance to modeling error and correlation error. The increase in lumped data leads to lower communication traffic but causes a higher modeling error. Also, faster bandwidth rates are used. The average delay increases as the network size increases. Since a larger volume of packets is generated in the larger networks, but the buffer size of the nodes remains fixed, there is higher collection loss and buffer loss. The values of average error are comparable only because there is higher loss leading to fewer packets reaching the TP.

A couple of anomalies were noted in the results for network sizes 64 and 100 nodes. Firstly, the requirement to increase lumping to improve communication (at the penalty of higher error) was less than if the objective was to reduce error by allowing less local lumping. Less lumping combined with slower communication rates causes increase in delay and losses due to buffer

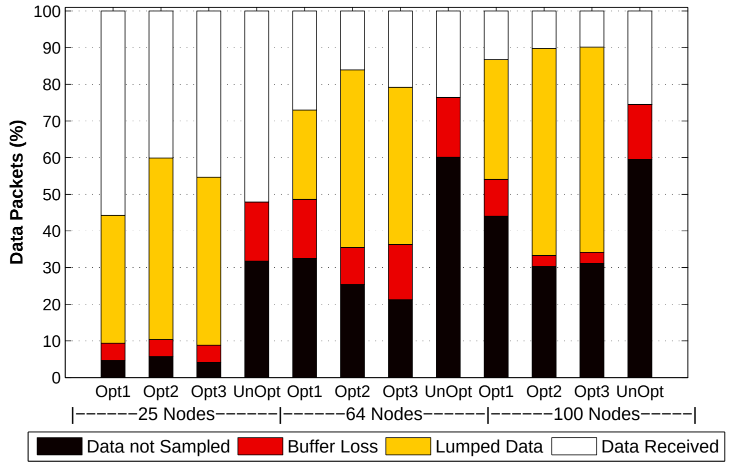


Figure 2.14: Dataset 2: data loss for different network sizes

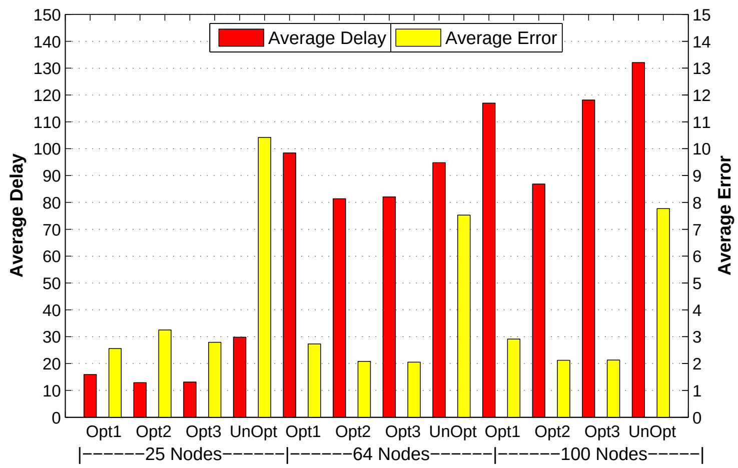


Figure 2.15: Dataset 2: delay and error for different network sizes

overflow. Therefore, fewer packets reach the TP, leading to increase in error. Secondly, especially for 64 node network, the delay for unoptimized case is less than the delay for the case that minimizes lumping. This is because delay can only be estimated for packets that actually reach the TP. However, since very few packets actually make it to the TP in the unoptimized case (hence the large error) and since most of these packets are from nodes close to the TP, the average delay is minimized. Therefore, it is important to combine the loss and the delay/error statistics to get a more accurate understanding of the results.

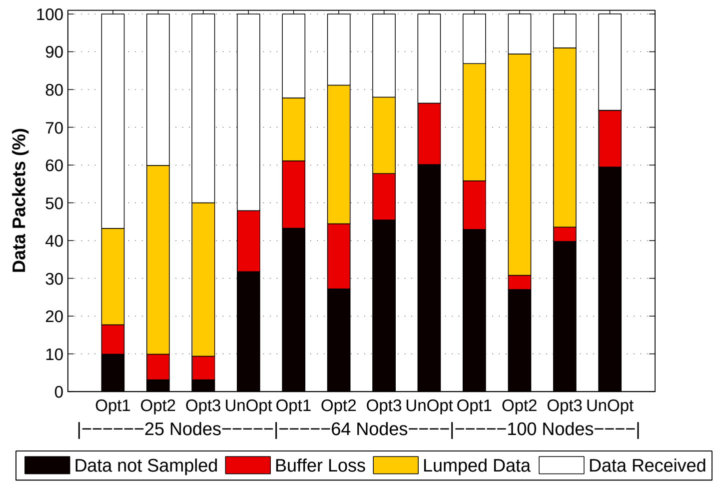


Figure 2.16: Dataset 6: Data loss for different network sizes

Dataset 6. The experimental results for Dataset 6 are as shown in Figures 2.16 and 2.17. Most of the observations noted for the results for Dataset 2 are also valid for Dataset 6. An important difference is in the amount of lumped data as there is less lumping for Dataset 6. The basic purpose of lumping is to reduce communication traffic by lumping some of the less important data. Since Dataset 6 has more thermal activity (hotspots), there are fewer regions where data can be lumped. This causes an increase in communication traffic leading to higher delays and buffer loss.

Summary of results. Table 2.3 summarizes the percentage improvement in loss, model error, and delay for the optimized cases vs. the unoptimized case. Columns 4, 6, 8, and 10 in the table present the amount of performance improvement due to trajectory prediction (TrPr) using algorithm A2. The results are averaged over the 8 datasets. The collection loss (data not sampled)

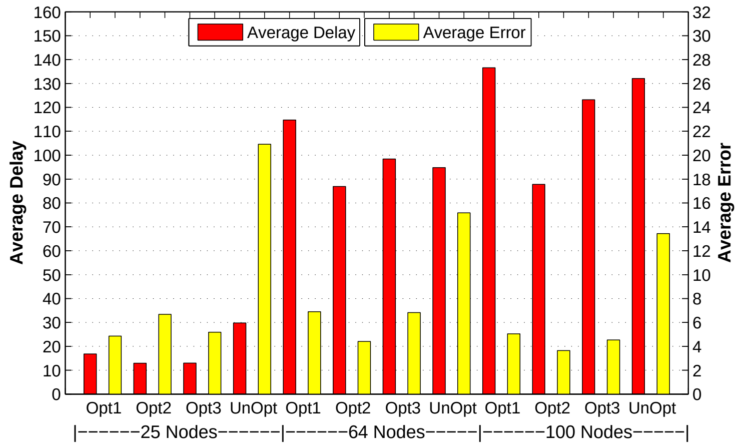


Figure 2.17: Dataset 6: delay and error for different network sizes

is reduced by up to 84.43% for 25 nodes, up to 57.86% for 64 nodes, and up to 50.50% for the 100 node network. Similarly, for buffer loss, there is significant reduction of up to 67.34% for 25 nodes and up to 76.16% for 100 node network. The improvement is smaller for 64 nodes up to 16.31%. This is because, for the unoptimized case, more than 60% of the data is lost as collection loss. So, there is already less traffic in the network leading to less buffer loss. This causes the misperception that the buffer loss is comparable for the optimized and unoptimized cases. It is also interesting to note that the percentage improvement for both, collection and buffer loss, is higher for OptRatios 2 and 3 as compared to OptRatio 1. The reduction in overall error is up to 76.91% for 25 nodes, between 58.29 to 73.49% for 64 nodes and up to 72.99% for 100 nodes. The highest improvement is shown either by OptRatio 1 or 2, for any network size. There is a significant reduction in the average delay for 25 nodes (up to 57.62%), especially for OptRatio2 which tries to improve latency. For the other network sizes as well, the improvement in latency is highest for OptRatio2 but the percentage is smaller for 64 and 100 nodes. Again, the low values for 64 nodes are caused by the fact that more than 60% of the packets in the unoptimized case are not generated at all (collection loss)

Table 2.3: Summary of Results

Net Size	Opt Ratio	% Improvement vs UnOpt Case							
		Coll Loss	TrPr Coll Loss	Buff Loss	TrPr Buff Loss	Error	TrPr Error	Delay	TrPr Delay
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
25	1	79.10	9.43	62.50	14.52	76.91	1.05	45.69	2.16
	2	83.61	6.35	67.34	10.48	68.21	1.89	57.62	1.54
	3	84.43	13.93	69.35	18.95	72.67	2.82	56.57	0.80
64	1	36.26	14.27	-14.02	4.54	58.29	9.35	-11.49	0.81
	2	57.96	19.27	16.31	4.88	73.49	5.21	5.83	7.04
	3	44.80	20.30	13.11	9.97	65.97	9.74	-2.14	6.96
100	1	25.72	8.07	24.89	5.04	62.66	3.47	0.40	1.24
	2	50.50	14.97	76.16	1.37	72.99	3.56	34.95	3.60
	3	37.02	12.37	76.79	0.63	68.68	3.91	9.25	1.29

leading to less traffic and lower delays for the remaining packets. Hence, it is important to combine the information over the four columns to get useful insights into the results of these experiments.

For the 25 node network, trajectory prediction provides up to 13.93% reduction in collection loss and up to 18.95% reduction in buffer loss. It also contributes to a small improvement in overall error and delay. For the 64 node network, the prediction results in up to 9.74% reduction in error and up to 7.04% reduction in delay. The improvement is loss is also high, up to 20.30% for collection loss and up to 24.54% for buffer loss. Similarly for the network size of 100 nodes, there is a small contribution to reduction in loss, delay and up to 12.37% reduction in collection loss.

In a platform with constant bandwidth, $j = 1$ and variable β_j is one in equation (2.34). Optimization of the selection of DCPs and lumping levels is still performed to minimize data loss and delay. Table 2.4 compares the optimization results when three bandwidths BW_j are used versus constant bandwidth. When fixed BW_1 (slowest) is used, as expected, the improvements in loss, error, and delay are less than for the fully optimized case (multiple BW_j) for all optimization ratios. For fixed BW_3 (highest), the results for collection loss and error are similar to the fully optimized case (multiple BW_j).

Table 2.4: Optimization Results for Single versus Multiple Bandwidths

BW	OptRatio	% Improvement vs UnOpt Case			
		Coll Loss	Buff Loss	Error	Delay
Multiple BW_j	1	79.10	62.50	76.91	45.69
	2	83.61	67.34	68.21	57.62
	3	84.43	69.35	72.67	56.57
Fixed BW_1	1	52.66	52.78	63.30	-18.76
	2	66.22	72.69	59.09	0.06
	3	61.17	66.67	61.67	-6.42
Fixed BW_3	1	87.90	52.78	79.81	67.65
	2	86.16	59.72	74.59	70.74
	3	87.23	56.94	76.68	69.89

The improvement in delay is higher and in buffer loss is less. But, operating only at maximum bandwidth increases power consumption.

2.5 Conclusion

This chapter presented a procedure to construct robust data models using samples acquired through a grid network of embedded sensing devices with limited resources, like bandwidth and buffer memory. The procedure constructs local data models by lumping state variables, and then collects centrally the local models to produce global models. The modeling procedure uses a linear programming formulation to compute the lumping level at each node, and the parameters of the networked sensing platform, like data communication paths and bandwidths. Two algorithms are described to predict the trajectories of mobile energy sources/sinks as predictions can further reduce data loss and delays during communication. The computed parameters and trajectory predictions are used to set-up the local decision making routines of the networked sampling nodes. Experiments discuss the method's efficiency for thermal modeling of ULTRASPARC Niagara T1 architecture.

Experiments show that variable lumping reduces the overall error by up to 76.91% and delay by up to 57.62%, as compared to no lumping being used. The error is smallest if latency reduction has high priority. The attempt to minimize local error performs less lumping, however, results is larger data loss, and hence in more overall error. The attempt to reduce the overall error by minimizing the correlation error results in increased latency. As the network size increases from 25 nodes to 64 and 100 nodes, the larger communication traffic leads to further losses and delays. Therefore, accuracy-centered optimization becomes critical for performing reliable data extraction. Trajectory prediction using adaptive method (A2) reduces modeling error by up to about 10%.

Chapter 3

Optimizing the Accuracy of Sound Based Tracking

[1]

This chapter presents an approach for optimizing the accuracy of data models produced based on data sampled through a network of embedded sensors. The method considers three orthogonal facets defining model precision: minimizing the sampling error of the individual embedded nodes, sampling sufficient data from distributed areas to correctly represent the phenomenon of interest, and meeting the timing delays that guarantee the timeliness of data. The three objectives are achieved by dynamically reconfiguring the architecture of the embedded nodes, and dynamically selecting the data transfer paths to the decision making nodes. Sound based trajectory tracking is used as a case study for the proposed approach.

[1]Note: This chapter is based on the work published in the paper [16]. This research was conducted in collaboration with the co-authors of the paper and my contributions are as follows: The theory and experiments related to sound-based localization and node-level optimization. Also provided assistance in the development and implementation of the trajectory prediction algorithms.

3.1 Introduction

Resource allocation is a main challenge for distributed data acquisition for decision making [19,55]. This is because the individual embedded nodes of the networked infrastructure, due to the cost constraints of an application, do not have sufficient resources to sample sufficient data, e.g., limited local memory, energy and communication bandwidth. Therefore, the resources of the infrastructure as a whole must be efficiently employed for the sampling task. For example, a sensing node running a sound-based tracking algorithm on an 8-bit microcontroller can track only vehicles moving less than 10 km/hour [66]. In addition, many sensor readings might be incorrect, if the background noise is very high, or for certain angles between the node and the sound source. These limitations can be significantly reduced if instead a network of embedded nodes is used for tracking. However, new challenges arise at the network level as data routing to the decision making nodes can produce significant data loss and delays.

The resources of a network of embedded nodes must be allocated so that there is an optimal load balancing between sensing - processing - communication activities. Zhao et al. [19] explain that this problem needs meticulous investigation due to its importance for distributed information processing systems. Several important resource allocation methods have been presented in the literature mainly to optimize energy and power consumption. Munir et al. [64] propose a technique using Markov Decision Processes to tune the parameters of the node architecture (i.e. processor voltage and frequency, and sensing frequency) to reduce energy consumption while operating in changing environments. Software reconfiguration for WSN is discussed in [60]. Lu et al. [62] discuss energy efficient node cluster formation using data correlations and spatial properties of the application. A decentralized adaptive resource allocation approach is discussed in [63]. Nodes use a bidding scheme to allocate resources in which they optimize their return (i.e. the utility of their actions) while minimizing their payments (e.g., consumed energy). Other resource allocation methods for distributed networked systems are discussed in [58, 61, 67, 68]. While minimizing energy consumption is important to prolong the functioning of a network, acquiring sufficient data is equally important for effective decision making. Performing resource allocation to optimize the quality and quantity of the data acquired through a network has not been studied yet.

This chapter presents a new approach for optimizing the accuracy of distributed data acquisition for dynamic phenomena through a network of embedded sensors. The method considers three orthogonal facets defining model precision: minimizing the sampling error of the individual embedded nodes, sampling sufficient data from distributed areas to correctly represent the phenomenon of interest, and minimizing the timing delays that guarantee the timeliness of data. The first objective is achieved by dynamically reconfiguring the frontend architecture of the embedded nodes. The information about the signal level and bandwidth is used to adapt the parameters and topology of the amplifier and filter blocks of the frontend. Dynamic data transfer path selection minimizes data losses and delays by using a probabilistic model for the likelihood of a node to run out of resources, either because of it being on the sampled trajectory or being part of the forwarding paths of other nodes. Three models, called quasi-static, bounded and stochastically bounded models, are presented for likelihood prediction.

The chapter has the following structure. Motivational examples for this work are discussed in Section 2. Section 3 presents the proposed adaptation methods. Section 4 discusses experimental results. The chapter ends with conclusions.

3.2 Motivation

The signals and data sampled through the network of embedded nodes is utilized to create models (of the observed physical phenomena) used in decision making. A model is defined by the following quadruple:

$$Model =< Function, Error, Utility, Cost > \tag{3.1}$$

Function is the mathematical expression of the model. *Error* is the prediction error of the model. *Utility* is the utility of using the model in decision making, (e.g., improving the quality of decisions). *Cost* is the cost of producing the model, including the related time, and used hardware resources and energy. This chapter is mainly concerned with minimizing the error of the model.

The modeling error depends on three main factors: (i) the accuracy of the signal sampling at the individual nodes, (ii) the data loss during communication to the decision making node (called target point in this chapter), and (iii) the delay with which data is received at the target point, as excessively delayed data might become obsolete. The next two examples explain how resource allocation impacts the three factors.

A. Signal sensing accuracy. Sound-based tracking algorithms were implemented on PSoC embedded processors [21], and then the effect of ambient noise on the localization accuracy was studied. The different Signal to Noise ratio (SNR) values were 0dB, 6dB, 20dB and 40dB. Readings were taken for values of Direction of Arrival (DoA) ranging from -90 to +90 in steps of 15 degrees. A reading was considered abnormal if its value deviates from the expected value by more than five degrees. The experiment counted the number of instances of abnormality out of twenty readings for each angle for each value of SNR. The rms error was calculated only for values which were not abnormal. Figure 3.1 plots the percentage abnormality for localization for low noise and high noise conditions. Similarly, Figure 3.2 plots the rms error for localization for low noise and high noise.

These experiments detail the fact that the probability of getting an abnormal DoA estimate increases with increase in the ambient noise, close to 100% in certain conditions. The accuracy of the localization data, expressed based on error depends to a large degree on the noise conditions. The temperature dependency of the speed of sound in air introduces additional abnormality in the results.

B. Accuracy at the network level. The correlations between the trajectory of the tracked sound source and the selected data communication paths influences both the experienced data loss and delays while forwarding data to

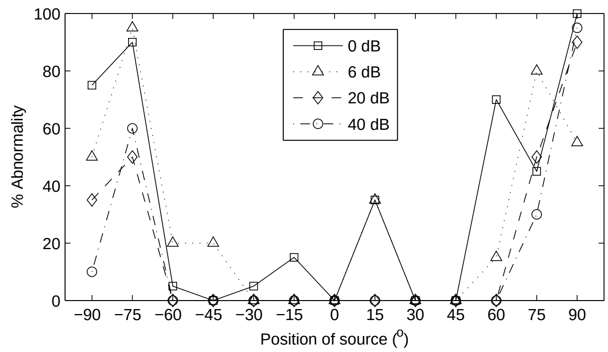


Figure 3.1: Percentage abnormality for high and low noise conditions

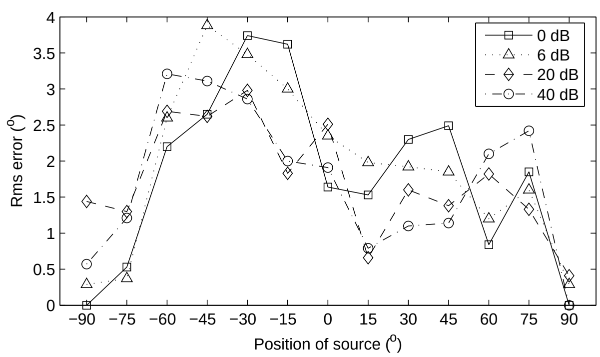


Figure 3.2: Rms error for high and low noise conditions

the target point.

It has been experimentally observed that, due to the limited resources of a node, data can remain in the receive buffer while the node is sampling. Data loss occurs when data stored in the buffer is overwritten before it is forwarded. This situation also increases the delay of transmitting data to the target node as described by Figure 2.3 in section 2.1 of Chapter 2.

In conclusion, resource allocation, e.g., the frontend resources used for localizing the sound source by a node and the selected data communication paths, determine the three main factors defining the error of the models used in decision making. The quality of resource allocation schemes depends to a high degree on the characteristics of the environment, e.g., ambient noise, and monitored phenomena, i.e. trajectory.

3.3 Two Methods to Improve Model Accuracy

This section presents two resource allocation methods to improve data acquisition accuracy at node and network levels.

3.3.1 Improving Accuracy at the Node Level

The capability of PSoC to dynamically reconfigure its mixed-signal frontend can be used to improve the accuracy of the data (e.g., sound source location) sampled by each embedded node. This optimization dimension has not been explored before by other sound-localization implementations which use FPGAs or ASICs [56].

The adaptation procedure operates as follows: after detecting a sound source, the sound characteristics, e.g., level, frequency bandwidth, and noise, are found and used to dynamically customize the frontend by modifying the topology and parameters of the building blocks at run time. The frontend has one amplifier, one filter and one ADC. (i) The level of the signal is used to adjust the gain of the amplifier until a predefined threshold for the signal amplitude is achieved. (ii) The frequency bandwidth of the tracked signal decides the bandwidth requirement of the filter, which is then used to select the topology and filter parameters that match closest the requirement. The block parameters of the alternative filter solutions are stored locally by every node. (iii) The reconfigurable input MUXs is used to integrate temperature sensing. The hardware of the analog frontend used for sound localization is modified at runtime and reused for temperature sensing. Temperature sensing helps further improving the localization accuracy.

A shown in Section IV, the three frontend reconfiguration steps improve the node-level accuracy of data acquisition without increasing the utilized hardware resources and energy.

3.3.2 Improving Accuracy at the Network Level

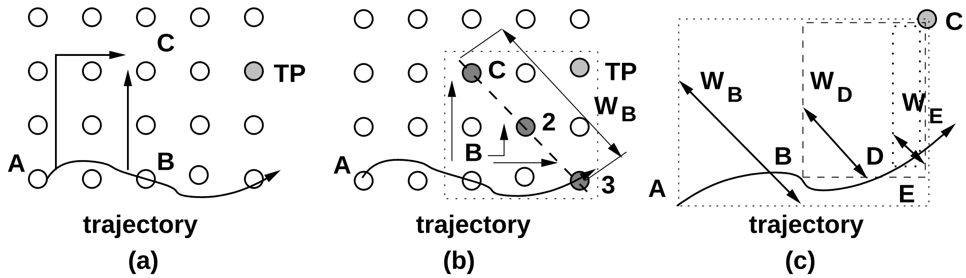


Figure 3.3: Data loss prediction

At the network level, the proposed resource allocation method decides the used data communication paths and bandwidths, so that the overall data losses and delays are within the acceptable errors of the model (equation (2.5)).

Each node sampling or forwarding data to the target point makes communication-related decisions without relying on global state information as its update to every node in the network would require very large overhead. For example, in Figure 3.3(a), node A decides to forward the sampled data using the minimum length path going through node C. Simultaneously, node C also receives the data sampled by node B. In this case, data is lost because the requirements exceed the communication resources of node C. If any of the nodes A and B selects a different path then it experiences a longer delay but there is no data loss. This resource allocation problem is an instance of the Prisoner's Dilemma problem in game theory [65] as the local decisions of a node interact strongly with the decisions of another node without the nodes knowing each others decisions.

The data loss at any node n of a path p_i is as follows:

$$loss_{n,p_i} = p_n K + N_n^{in} - N_n^{out} \tag{3.2}$$

p_n is the likelihood of having data samples at node n (e.g., the trajectory passes through the vicinity of the node). K is the number of data tokens (bytes) of each sample. N_n^{in} is the total input data of node n, and N^{out} is the total output data of the node. N_n^{in} includes the data transferred to the node along all paths converging to the node. Similarly, N_n^{out} is the data sent out along all output paths available at node n.

The total loss of the nodes of path p_i is:

$$loss_{p_i} = \sum_{\forall n \in p_i} loss_{n,p_i} \tag{3.3}$$

Every node must select the path p_i and its bandwidth (communication speed) such that the loss and delay is minimized:

$$\arg_{p_i} \min(\alpha\ loss_{p_i} + \beta\ Delay_{p_i}) \tag{3.4}$$

α and β are weights describing the importance of data loss and delay for deciding the error of the decision making model.

Equations (2.1) and (3.3) incorporate several probabilistic variables. The likelihood p_n of sampling data at a node depends on the tracked trajectory, which is known only during run time. Variable N_n^{in} depends on the trajectory and the decision of other nodes in the network to use a communication path that brings data at the input of node n. However, the nature of the decisions made by other nodes is unavailable to the node. Instead, the node must decide path p_i using estimations of the probabilistic variables. The time window ΔT considered for estimation is a trade-off between the preciseness of estimation and the related processing overhead at the node. The prediction procedures for the two variables are discussed next.

Prediction procedure 1: trajectory prediction. This chapter presents three ways to predict a trajectory based on the nature of the tracked source.

1. Quasi-static trajectories. We define an unknown trajectory to be quasi-static, if the trajectory can be approximated well (at run time) as a collection of fragments pertaining to a set of statically defined trajectories. This is a good approximation for phenomena described by differential equations with parameters in known ranges of values, e.g., the trajectories of unpowered, floating beacons in a pond. For each of the static trajectories in the static set, the data path offering minimum loss and delay can be found using the method described in [67].

Each node stores the static trajectories and the best data communication path for the trajectory. The data path selection procedure dynamically estimates at every node the most likely static trajectory to which the current node (and its neighboring nodes) might belong to. Then, the optimal data path is dynamically selected for that node. The probability of the current node n being approximated by $traj_{st,i}$ of the static set is estimated using Bayesian inference [57]:
$p(traj_{st,i}|n) \propto$

$$p(n|traj_{st,i}) \sum_{traj_{st,k}} p(traj_{st,i}|traj_{st,k}) p(traj_{st,k}|n_{prev}) \tag{3.5}$$

$p(traj_{st,i}|n)$ is the probability of having trajectory $traj_{st,i}$ given that node n sampled the current trajectory. $p(n|traj_{st,i})$ is the probability of node n sampling data given the static trajectory $traj_{st,i}$. $p(traj_{st,i}|traj_{st,k}$ is the probability of having a transition to $traj_{st,i}$ given $traj_{st,k}$. $p(traj_{st,k}|n_{prev})$ is the probability of having $traj_{st,k}$ given that the previous node n_{prev} sampled the trajectory.

The static path having the highest value $p(traj_{st,i}|n)$ is selected as being the one to which the current nodes belongs to, and its data path is used for forwarding data.

2. Bounded trajectories. We define a trajectory as a bounded trajectories if the trajectories that can be approximated as a sequence of convex - concave fragments, where each curve segment has bounded gradients with known ranges. The point separating each successive convex - concave fragments is called inflexion point. The average time distance ΔT between inflexion points is also known. The prediction model estimates that the real trajectory is located inside the region defined by the minimum and maximum gradients. The region is called trajectory approximating region. The mathematical expressions are described in detail in section 2.2 of Chapter 2.

3. Stochastically bounded trajectories. We define a trajectory as stochastically bounded trajectories if the trajectory is approximated by bounded fragments in which the change to another fragment follows a stochastic rule, instead of alternating between convex and concave regions like for bounded trajectories. Similar to bounded trajectories, stochastically bounded trajectories are composed of fragments with gradients limited to a known range [$Grad^i_{Min}$, $Grad^i_{Max}$]. The mathematical expressions are described in detail in section 2.2 of Chapter 2.

Prediction procedure 2: data flow prediction. The reasoning for estimating variable N^{in} in equation (2.1) is discussed based on Figure 3.3(b). Node A must decide if it should forward data to target point TP through node C. However, the data might get lost due to the data sampled by node B. The decision of node B to use node C as a forwarding node to TP depends on the number of shortest-path alternatives available to node B. All shortest-path alternatives are included within the bounding box shown with dashed lines. Hence, instead of using node C, node B could use all forwarding nodes situated at the same distance as node C, such as nodes 2 and 3. If the nodes are equally spaced then the likelihood of node B selecting node C depends on the number of alternatives:

$$likelihood_{B,C} \propto \frac{1}{W_B} \approx \frac{1}{\sqrt{2} \times dist(B, C)} \tag{3.6}$$

W_B is the number of nodes equally spaced to node B as node C. $dist(b, C)$ is the communication delay between nodes B and C.

As shown in Figure 3.3(c), the likelihood of node C receiving additional samples forwarded from other nodes than node A is then as follows:

$$likelihood_C \propto \sum_{\forall B \in Traj} \frac{1}{W_B} p_B \tag{3.7}$$

where p_B is the likelihood of node B sampling the trajectory. Or,

$$likelihood_C \propto \sum_{\forall B \in Traj} \frac{1}{\sqrt{2} \times dist(B, C)} p_B \tag{3.8}$$

Table 3.1: Frontend reconfiguration results

RMS noise LPF	**RMS noise BPF**	**Percent Improvement**
105.74	32.03	69.70
105.20	29.39	72.05
98.39	22.73	76.89
89.51	25.85	71.12
85.40	26.17	69.34
82.63	26.80	67.56
104.35	25.91	75.16
98.05	37.71	61.53
115.29	31.49	72.68
85.53	22.75	73.39

Estimating transmission delay. The total transmission delay in equation (3.4) for path p_i is:

$$Delay_{p_i} = Time_{proc} + \sum_{n=1}^{k_1} r_n \times Time_{comm} + k_2 \times Time_{wait} \tag{3.9}$$

$Time_{proc}$ is the processing time of the sound localization algorithm executed by a node. $Time_{comm}$ is the data communication time between two successive nodes. k_1 is the number of nodes to which data is forwarded including the target point. r_n is the number of re-transmissions for the next node due to the wireless connection. $Time_{wait}$ is the average waiting time in the input buffer of a node. k_2 is the number of nodes in which data is buffered before reaching target point. r_n and k_2 are Gaussian variables with mean and variance set by the similar characteristics of the wireless communication module and buffering at the network nodes.

3.4 Experiments

3.4.1 Improving Accuracy at the Node Level

A SNR of 0dB (high noise conditions) was used to perform experiments to quantify the percentage improvement in the rms value of noise due to fil-

Table 3.2: Direction of Arrival Results

AE for SNR 0dB	**AE for SNR 9dB**	**AE for SNR 20dB**	**PI for 0-9dB**	**PI for 9-20dB**
3	2	2	33.33	0.00
3	2	2	33.33	0.00
2	2	2	0.00	0.00
4	3	3	25.00	0.00
4	2	2	50.00	0.00
4	3	2	25.00	33.33
4	3	2	25.00	33.33
5	2	2	60.00	0.00
3	2	2	33.33	0.00
2	2	2	0.00	0.00

ter reconfiguration. The results are as shown in Table 3.1. The percentage improvement in rms noise ranges from 61.53% to 76.89% with an average improvement of 70.98%. This reduction in rms noise translates into improvement in SNR.

Due to frontend reconfiguration, SNR of 0dB changes to 10.75dB, SNR of 3dB to 16.75dB, SNR of 9dB to 19.75dB, SNR of 20dB to 30.75dB, and SNR of 40dB to 50.75dB. This improvement in SNR decreases the probability of obtaining abnormal results as shown in Figure 3.1. The SNR value closest to 10.75dB and whose data is available is 9dB. Using this data and by simulating a probability model using random numbers, experiments were conducted to compare the number of Abnormal Estimates (AE) obtained over for both these SNR values and then identify the Percentage Improvement (PI) in results. The actual estimates were assigned to the 13 nodes on the trajectory ranging from -90 to +90 degrees in steps of 15 degrees. Similar experiments were performed for 9dB and 20dB. The results are summarized in Table 3.2.

We observe that when SNR changes from 0dB to 9dB, the percentage increase in correct estimates is between 0% to 60% with an average increase of 28.5%. Also, when SNR changes from 9dB to 20dB, the percentage increase in correct estimates is between 0% to 33.33% with an average increase of 6.67%.

3.4.2 Improving Accuracy at the Network Level

The following path selection algorithms discussed in Section III were experimented for simulated PSoC networks of sizes 25, 64 and 100 nodes. $Algo1$ refers to path selection based on the minimum cosine between the trajectory and the data communication path (approximation for bounded trajectory model). $Algo2$ uses path selection based on a qualitative expression of the gradient ranges and number of inflexion points of the bounded trajectory model (bounded model). $Algo3$ refers to the quasi-static trajectory model. $Algo4$ implements equation (2.22) for path selection. The equation is computed for the node over a window of k forwarding nodes. $Algo5$ implements the stochastically bounded trajectory model.

The simulation models were calibrated according to the timing delays and data waiting times in buffers measured for real nodes. The number of re-transmissions r_n in equation (3.9) for path delays was also calibrated based on measurements of physical wireless networks. For every network, experiments considered four different target points, and three different data path configurations. Various trajectories were considered.

Table 3.3 presents data loss values for the network with 100 nodes. Column one shows the target point which was used. Column 2 indicates the path configurations used for that target point. Column 3 presents data loss for the trajectory for that particular path configuration. The heading for Column 3 has data loss (out of n), where n is the length of the trajectory. Columns 4, 5, 6, 7 and 8 give (data loss for that algorithm) / (data loss for the static path configuration) for performance comparison of the algorithm with the static DAPs.

The table shows that all data path selection methods reduce the data loss compared to static data paths. For example, the losses of the static paths can be as high as 8 data values out of 19 values (50%). For this case $Algo1$ and $Algo4$ experience no loss, and $Algo2$ looses only one value. Similar percentage data loss reductions were observed also for the network with 64 nodes. For this network, $Algo4$ has a slightly smaller data loss than $Algo1$. Increasing the window size k for $Algo4$ does not produce any improvement meaning that the

Table 3.3: Data loss for network with 100 nodes

Target Point	**Path Config**	**Data Loss (out of 19)**	Data Loss for Algos				
			Algo1	**Algo2**	**Algo3**	**Algo4** (k=2)	**Algo4** (k=4)
1	1	3	0/3	1/3	3/3	0/3	0/3
	2	3	0/3	1/3	3/3	0/3	0/3
	3	1	0/1	1/1	3/1	0/1	0/1
2	1	2	0/2	2/2	1/2	0/2	0/2
	2	7	0/7	2/7	1/7	0/7	0/7
	3	1	0/1	2/1	1/1	0/1	0/1
3	1	8	0/8	1/8	4/8	0/8	0/8
	2	3	0/3	1/3	4/3	0/3	0/3
	3	3	0/3	1/3	4/3	0/3	0/3
4	1	6	2/6	0/6	1/6	2/6	2/6
	2	0	2/0	0/0	1/0	2/0	2/0
	3	1	2/1	0/1	1/1	2/1	2/1

related trajectory model should be used only for short-term predictions.

Table 3.4 presents the average delay for networks with 100 nodes. Column 1 shows the used target point. Column 2 indicates the path configurations used for that target point. Column 3 shows average delay for that particular path configuration. Columns 4, 5, 6, 7 and 8 show percentage improvement of average delay compared to static paths.

For 8 out of 12 cases, $Algo1$ and $Algo4$ reduce the average delay by more than 20%, up to 45%. For 6 cases, $Algo2$ offers delay improvements of more than 15%, up to 32%. The window size k does not affect the delay saving of $Algo4$. Note that in 11 out of 12 instances, one of the four algorithms reduces delay compared to any of the static paths. This suggests that identifying the best approximation model for a trajectory is very important, and generic trajectory models are insufficient. $Algo3$ performs worst compared to the other methods with respect to both loss and delays. Only for this method, we regularly see loss and delays higher than for static paths. That is because the performance depends on how well the real trajectory is approximated by the static trajectories.

$Algo5$ was experimented for trajectories that turn back. These are

Table 3.4: Average delay for network with 100 nodes

Target Point	**Path Config**	**Average Delay**	Percentage Improvement in Avg Delay				
			Algo1	**Algo2**	**Algo3**	**Algo4** (k=2)	**Algo4** (k=4)
1	1	1946.67	32.13	32.04	5.891	30.85	30.08
	2	1928.67	31.50	31.41	5.012	30.21	29.43
	3	1288.24	-2.55	-2.69	-42.21	-4.49	-5.66
2	1	1731.76	21.13	6.25	15.95	20.65	19.70
	2	2118.33	35.53	23.36	31.29	35.13	34.36
	3	1455.56	6.17	-11.54	0.00	5.59	4.47
3	1	2250.91	45.54	32.97	15.38	45.54	45.54
	2	1819.38	32.63	17.07	-4.69	32.63	32.63
	3	1445.63	15.21	-4.38	-31.75	15.21	15.21
4	1	1901.54	27.80	16.22	26.38	26.87	21.83
	2	1722.63	20.30	7.52	18.73	19.28	13.71
	3	1400.00	1.93	-13.80	0.00	0.67	-6.18

difficult to handle by the other algorithms, including static paths. There was no data loss in all but one instance for the network with 100 nodes, and no data losses in all cases for networks with 25 nodes. The maximum loss was 2 data values out of 15 for networks with 64 nodes.

3.5 Conclusions

This chapter presents a new approach for optimizing the accuracy of distributed data acquisition for tracking dynamic phenomena through a network of embedded sensors. The method considers three orthogonal facets: minimizing the sampling error of the individual embedded nodes through frontend reconfiguration, and selecting dynamically data paths to avoid data loss and delays due to embedded nodes running out of resources. The likelihood of a node to run out of resources, either due to it sampling the trajectory or being part of other forwarding paths, is estimated using three prediction models. The accuracy of data sensing is improved by about 28.5%, data loss is zero in most situations, and delay reductions are more than 20% in most cases.

Chapter 4

The Role of Precedents in Increasing Creativity during Iterative Design of Electronic Embedded Systems

[1]

This chapter presents a study on the role of precedents in illuminating creative ideas during iterative design for solving open-ended problems in electronic embedded systems. Through an experimental study grounded in cognitive psychology, this work examined the influence of precedents on the novelty, variety, quality, and utility of design solutions devised through an iterative design process involving groups of participants. Another tested hypothesis was whether incremental changes of requirements improve novelty. Results show that precedents did not increase solution novelty and quality, but improved utility. Precedents reduced design feature variety as solutions converged towards a few dominant designs. Incremental modification of requirements did not increase novelty.

[1]Note: This chapter is based on the work published in [6].

4.1 Introduction

Design problems in electronic embedded systems are often open-ended as they address new core applications and functionalities in emergent yet important domains, like intelligent infrastructure, robotics, healthcare, automotive industry, and many more. Open-ended design problems are incompletely specified, or represent needs based on organizational or personal perspectives, judgments, and predictions [78, 80]. Other kinds of design problems include fully-specified, well-defined problems and fully-specified, infeasible problems [91, 109, 116]. Fully-specified, well-defined problems are described as complete sets of requirements solved through optimization or transformation methods. Tackling fully-specified, infeasible design problems involves finding the contradicting requirements and then solving these through specific resolution rules [70, 74, 84].

Creativity is important in solving open-ended design problems because new goals and functions, novel design solutions, and original resolution rules must be found. There are many definitions of creativity [73]. Creativity is often characterized by referring to the novelty (e.g., solutions have less frequent features) and utility (i.e. solutions satisfy precise needs) of the solutions [71, 102, 120, 122]. Creativity in physics is described by inventiveness and orderliness, and creativity in art is represented by imagination and originality [119]. In engineering, design creativity is characterized by the level of meeting goals [112]. [93] suggests that the cognition factors involved in creativity are fundaments (e.g., words), classes (categories), relations, patterns, problems, and implications. Innovation originates through a process that includes both convergent and divergent thinking. Convergent thinking refers to naming objects, classifying objects into categories, finding correlations, identifying patterns, finding changes, and providing unique conclusions. Divergent thinking involves finding different words with the same properties, producing alternative relations with the same meaning, and changing a structure's meaning for a new purpose. A broad set of theories and models on creativity are discussed in the research literature in cognitive psychology [71,92,97,102,117,120,125,126].

The relation between creativity theories and models in cognitive psychology and innovation in engineering is subtle. Many creativity tests in cognitive psychology neglect the comprehensive nature of engineering design and ignore domain specific information. For example, tests on divergent thinking are weak when used to measure and predict creativity in real-world situations [101, 129]. Other creativity models focus on concrete innovation situations while considering more specific engineering problem details. [74] define innovation as contradiction solving. Altshuller (1988), the creator of TRIZ method, proposes a set of rules to resolve a broad variety of contradictions. Dubois, Eltzer and De Guio [84] extend TRIZ into Generalized Contradiction model, in which combinations of contradictions are identified for given engineering problems. Other approaches to design innovation suggest problem-solution co-evolution [82, 115]. Poon and Maher [105] propose co-evolving problem descriptions and solutions by combining genes with modified behavior from interacting populations of problems and solutions. Kryssanov, Tamaki and Kitamura [98] describe a model for the dynamics and non-determinism of design.

In spite of the breadth of existing work on creativity and design innovation, there are still few design creativity studies that capture the specifics of electronic embedded system design problems while being grounded in models devised in cognitive psychology. Electronic embedded systems have several characteristics that distinguish them from other general-purpose or engineering problems:

- Embedded systems are arguably more complex than other engineered systems therefore design conceptualization is harder [80].
- Modularity is intrinsic to design [79, 105], not a subsequent step, like in architecture or mechanical engineering [81, 121].
- Embedded systems are programmable. This helps achieving a higher utility through customization to broader needs and contexts but increases the hardness of conceptualization as more options are possible.
- Design procedures are iterative [76]. Iterative design includes successive iterations that continuously use previous solutions as starting points to

create designs with new goals, extra functions, and sub-structures inspired by previous designs [95, 108].

The implications of these characteristics on design creativity are not well understood.

This chapter presents a study on the role of precedents in illuminating new, creative ideas during iterative design procedures for open-ended problems in electronic embedded systems. We defined design precedents as solutions and solution features that were available to participants to solve a problem. Specifically, the set of available solutions included the designs developed during the iterative solving process by the participants in a group. Each group had six members, and each member had access to the solutions of any other member of the group. Solution features included the related purposes (e.g., needs addressed by designs), functionalities, and implementation details of designs. This definition is similar to that by Eilouti [87] as it refers to available solutions. However, Eilouti considers mature, well-accepted solutions, while our precedents are designs devised during the iterative process. Mature solutions are less common in emerging application domains, such as for embedded system design. Section 1 also summarizes other precedent definitions.

The study verifies five hypotheses. The first two hypotheses state that precedents enhance creativity by stimulating new ideas during early design stages, but then creativity decreases during later design iterations. Hypotheses three and four state that precedents improve design quality and utility throughout all iterations. These outcomes are expected because most designs converge towards a common solution as more of the effective features are adopted while the less efficient features are dropped. Hence, precedents in iterative design are likely to produce lesser solution feature variety than if precedents are not used. Finally, the fifth tested hypothesis is that creativity increases again if new requirements are added to the original problem specification. New requirements are expected to support more divergent thinking and new ideas. Our discussions present the insight gained from the experiments.

The study is based on the experimental procedure in cognitive psychology proposed by Mobley, Doares and Mumford [102]. Even though the experiment by Mobley et al. is a simplification of the design process, it is more analogous to the common design activities than other creativity-oriented experiments in cognitive psychology. Section 1 offers more details on the similarity.

The presented study is important for getting insight on the conditions that influence creativity in iterative design procedures. Majority of design flows in electronic embedded systems are iterative, hence originate dynamic innovation contexts depending on the amount of available, problem-related knowledge. Having more precedents should help innovation, but precedents can also produce fixation. Also, it is important to understand whether innovation occurs mainly during early design iterations, or if similar levels of innovation should be expected throughout the entire process. If opportunities for innovation are objectively less during later iterations, then design methodologies should first emphasize exploring divergent approaches followed by an emphasis on improving implementation quality during later stages. Moreover, understanding how changing problem requirements affect the innovation level is a main issue in open-ended problems [108] as design specifications often change during the design process. Finally, the obtained insight is useful in creating novel, innovation-oriented CAD tools for embedded system design. Robertson and Radcliffe [107] argue that design tools in engineering often lag behind state-of-the-art design methods, and might negatively impact innovation through bounded ideation, premature fixation, and circumscribed thinking.

The chapter has the following structure. Section one describes the procedure used in this study. Section two presents the experimental results. Section three discusses the results and the obtained insight. Finally, conclusions are offered.

4.2 Experimental Procedure

This study observed the impact of precedents on design novelty, variety, quality, and utility. It was based on the experimental procedure in cognitive psychology developed by Mobley, Doares and Mumford [102]. We think that this procedure matches better the specifics of electronic embedded system design than other creativity tests, like divergent thinking tests. Zheng, Proctor and Salvendy [129] explain that traditional tests in cognitive psychology, like structure of the intellect divergent production test, Walchan-Kogan test, Getzels-Jackson test, and Torrance Test of Creative Thinking, address ideation but fail to capture other important design aspects, like problem framing, solution evaluation, and detailed implementation.

The procedure by Mobley, Doares and Mumford involves three integrated steps performed on a given set of exemplars. Exemplars are verbs and nouns grouped into three categories. Exemplars vary depending on their kind (e.g., artifacts), typicality, and degree of relatedness of the words in a group. For example, exemplar groups include seat, tire, brakes, wheel (group 1), glove, baseball, baseball bat, football (group 2), and bicycling, running, swimming, lifting weights (group 3) [102]. Exemplars are (in concept) similar to basic building blocks in design of modular embedded systems. The first step in the procedure by Mobley, Doares and Mumford requires finding a category label for the exemplars. This step is analogous to problem framing in open-ended problem solving, and combines analysis, ideation, and evaluation. The second step in the procedure by Mobley, Doares and Mumford creates more exemplars for the selected category label. The third step requires writing a story that uses the category label and exemplars. Steps two and three are conceptually similar to detailed implementation in embedded system design, including identification of additional building blocks needed for implementation and the actual creation of a design solution by relating the building blocks to each other. Therefore, we argue that the procedure by Mobley, Doares and Mumford offers a good basis to capture more accurately the specifics of embedded system design. For this study, we extended the procedure by Mobley, Doares and Mumford [102] in three directions:

1. The design exercises included broad design goals as embedded system applications are rarely without a-priori purposes. Goals are important in general-purpose creative tasks [74] as well as in domain-specific inno-

vation [129]. Problem-framing was part of the solutions.

2. The procedure was performed iteratively for four consecutive iterations to study the role of precedents.

3. The experiment was modified for group settings to replicate more accurately the characteristics of design environments in real-life. Participants worked individually, but at the end of each iteration and before starting the next iteration, the design solution of each participant was communicated to the group. These design ideas represented the precedents for the next iteration.

The study tested the following five hypotheses: **Hypothesis 1:** Design precedents in iterative electronic embedded system design produce more new features than if no precedents are used. More novel features are generated during early iterations. The number of new features drops during the later iterations as design solutions converge. **Hypothesis 2:** Design precedents produce design solutions of lesser variety than if no precedents are used. Hence, precedents create more converging solutions while having no precedents encourages more divergent solutions. **Hypothesis 3:** Design precedents produce design solutions of higher quality (performance) than if no precedents are used. **Hypothesis 4:** Design precedents improve the overall utility of solutions during all iterations of the iterative design procedure.

The literature discusses several kinds of design precedents. For example, precedents are past solutions that are reused to solve new problems [87]. Alternatively, precedents are guidelines modified according to new requirements [83]. In feature alignment for analogical reasoning, precedents establish correspondences with the current problem, followed by transfer of structural features from the analogy to the solution being devised [125]. In pattern-based design, precedents are parameterized templates describing an entire class of solution. Precedents are important in exposing hard-to-anticipate, global properties, e.g., the density and occupancy of urban environments [111]. The repertoire of precedents must be numerous, detailed, and consistent because precedents must offer a comprehensive sampling of the possible features of solution spaces [69, 111]. Precedents help finding new ideas, hence do not limit creativity in architecture [87], avoid rediscovery of well understood solutions [69], help deeper understanding and better interpretation of results [87, 111], and increase the number of solutions in mechanical engineering [123].

In our study, we defined precedents as the solutions and solution features that were available to the participants to solve the problem. The available solutions included the designs developed during the iterative solving process by the participants in a group. Each member had access to the solutions of any other member of his/her group. Solution features included the related purposes (e.g., addressed needs), functionalities, and implementation details of the designs.

We expected that high quality precedents eliminate alternatives that are less effective, thus reduce the variety of solutions. Also, more precedents will increase the number of associations to similar problems and solutions, thus improve solution novelty. These hypotheses are consistent with other studies that show that larger pools of solutions increase the number of combinations inspired by the pool [117]. The number of ideas is correlated to the number and commonality of input stimuli as they stimulate more associations [85]. Nijstad, Stroebe and Lodewijkx [103] suggest that more variety of ideas boosts creativity as broader sets of concepts are accessed. In contrast, Chua and Iyengar [75] explain that more options do not always improve creativity as people simplify and eliminate to cope with higher complexity. The ability to handle more design options increases with experience. Heylighen, Deisz and Verstijnen [94] report that students in architecture are less creative, if they devise more solutions. Thus, more precedents can lower creativity too. High flexibility in design tasks, including high degree of uncertainty in problem specification, also reduces creativity. Thus, defining design goals is likely to improve innovation. However, other studies suggest that previous solutions can reduce creativity due to fixation [96, 118].

Our design study corresponded to component-based design style, in which building blocks are connected to form a solution. In contrast to other component-based design styles, i.e. in architecture [87], embedded system design emphasizes the importance of causal relations among blocks, such as the output of one block generates inputs for its following blocks. Problem framing is more flexible than in other domains as there are more applications, contexts, goals, and functions that can be identified. For example, adding intelligent behavior based on programmable embedded systems is a useful capability for most current engineering applications. Finally, the existing constraints relate mainly to technological aspects, like nature of electrical signals, execution time, resolution, power consumption, and less to esthetic, cultural, or context-induced constraints. In addition, embedded systems must offer active yet intuitive interaction with humans, like through voice, touch, image, and so on. These differences are expected to result in a different impact of precedents on design novelty, quality, and utility than that the impact observed in other design domains.

Hypotheses 5: After several iterations, adding new functional requirements to the problem specification increases the number of novel features in the design solutions. This hypothesis is based on the observation that many successful engineering designs introduce functional modifications to current solutions, or follow established design patterns. Goldenberg, Mazursky and Solomon [89] suggest five common templates in product innovation: introducing dependencies between previously unrelated variables, producing control dependencies between unrelated components, eliminating one component of the design with and without changing the main functionality, and splitting one component into subcomponents that jointly have the same functionality as the original component. Maimon and Horowitz [100] explain that innovative designs change the nature of relations between design variables without modifying the design logic for the considered technology and without adding new types of components. Dzbor and Zdrahal [86] propose transformation rules for problem framing, like restricting solution acceptability, identifying and solving contradictions, reinterpreting solutions in different framing contexts, and incremental changing of design principles. Design transformations are formally expressed as graph-based descriptions [88, 113, 114]. Hypothesis 5 was suggested by the idea that new functional requirements will introduce new relations between the design blocks of a solution. Section 1.1 presents a detailed description of the experiments used to test the five hypotheses, and Section 1.2 describes the measurement procedure of the results. Section 2 offers a statistical analysis of the results. Finally, Section 3 presents a discussion of the experimental data in practical terms.

4.2.1 Detailed Description

The description of the experimental procedure is as follows. Participant had to solve the following design exercise. The title of the application was "intelligent remote controller". Participants had to devise an innovative solution for a remote controller that meets the following specification:

- Performs its functions using voice/sound-based commands.
- Has intelligent yet easy-to-use functions, and uses maximum number of sensor of different kinds.

Participants were told that it was important that the solution incorporates novel functional capabilities. The following list of sensors, functionalities, and intelligent responses were available to be incorporated into design solutions. Other blocks could be used too.

Sensors Functions Responses to user Altitude Select among multiple choices No response Microphone Set parameters Indicate performed function Fluid flow for gases and liquids Define sequences of selected choices Offer short demo of results Force (linear and torque) Set time limits Warning about impossible problems Pressure Associate responses to choices Suggest other related functions Proximity Set priorities for multiple choices Different options of communication (audio, video, text, vibration) Stress and strain Temperature Vibration Wind speed

Participants had to submit the following outputs as part of their solution:

- To describe alternative design solutions for an intelligent remote controller by explaining the devised functionality, the used sensors, and the response type offered to users.

- To explain the used design principle, e.g., block diagram with inputs and outputs, and expected solution performance, like size, weight, speed, and power consumption, and expected beneficiaries, including possible applications.

Participants were told that the design evaluation follows the next rule: 25% for satisfying the design requirements, 25% for solution performance, and 50% for uniqueness (novelty) of the solution. The experiments involved a control group (CG) and an experimental group (EG):

- **EG:** Participants had to privately design their solutions but collectively see and discuss their solutions in a group setting. Subjects were asked to refine their design three times. Between successive iterations, the EG subjects discussed their solutions. After three iterations, the functional requirements were extended (while leaving the core functionality unchanged), and one more design refinement was performed. The added requirement was that the remote controller had to be used to identify frequent habits of the user. All iterations were performed immediately after each other. After each iteration, each participant briefly stated to the other group members the functionality of his/her solution, the used sensors, and the responses offered to the user.

- **CG:** Participants worked for the same total amount of time but without following the iterative design scheme. CG conducted their designs in private. They were not allowed to see any other solutions during the entire process.

In each of the iterations, participants had the liberty to continue adding new ideas to their existing design or start designing a new solution. The measurement procedure for the experimental study is presented next.

4.2.2 Measurement Procedure

All design solutions were assessed using the method proposed by Shah, Vargas-Hernandez and Smith [112]. A similar technique is also suggested by

Schunn, Lovell and Wang [110]. According to the method, the novelty of any design can be estimated by identifying the frequency of its features appearing in other designs too. The novelty rating for each design was estimated using the following formula [112]:

$$MN_i = sum_{j=1}^{n} fj \times Sj \tag{4.1}$$

where, n is the number of features of the design, fj is the weight of feature j, and Sj is the novelty index of feature j.

The novelty index of a feature is computed according to the following expression (Shah, Vargas-Hernandez & Smith, 2003):

$$Sj = \frac{(Tj - Cj)}{Tj} \times R \tag{4.2}$$

Tj is the total number of designs from the investigated set having feature j present, and Cj is the number of designs from the set using the same feature's implementation as the currently evaluated design. Term R normalizes the index value to the desired range which in our case is range [0,10]. Specifically, the following attributes were considered to compute the novelty metric:

- *Nature of sensing:* sense the human body, environment, or both.
- *Type of inputs:* Inputs were classified as voice (V) only, touch (T) only, V and T, brain wave and gesture (Ges), V and Ges, and other/none.
- *Number of sensors:* The groups were 1-2 (small), 3-4 (average), 5-6 (large), and 7-8 (very large).
- *Type of outputs:* control signal (CS); display (D); CS and D; CS and alarm; CS, voice and D; CS and alarm and D; and vibration and D. Traditionally, simpler systems use CS and D, while the more advanced designs utilize voice and other means, like vibration.

These attributes characterize the main components of a design solution, which depend on the application domain, specific problem, and solution structure. The variety metric is calculated for a set of designs based on the diversity of the features that the designs offer [112]. It is computed using a hierarchical representation called genealogy tree, which is specific to the application domain. For this study, the genealogy tree had four levels as shown in Figure 1. The different features for each attribute can be represented as nodes on that level. For each of the nodes, we counted the number of designs that shared the specific feature.

The four levels define a top-down description of an electronic embedded system:

- *Level 1* describes the nature of sensing, such as if the system operates based on data collected from the physical environment (like temperature, humidity, pressure, etc.), from observing physiological signals of the human body, or in other ways. The nature of sensing is decided to a large degree by the goal of the design and induces further constraints that shape the nature of design solutions.

- *Level 2* expresses the type of inputs, like sensing voice, touch, brain waves, and so on. The kind of inputs limits to a certain degree the nature of computing that can be realized by the system.

- *Level 3* indicates the complexity of the system. Having more sensors also imposes more complex functionality to process the sensor inputs.

- *Level 4* presents how the computed information is presented to users, e.g., through control signals, triggering alarms, using displays, and so on.

The dotted path in Figure 1 from levels 1 to 4 represents a design that performs human body sensing, has voice and touch inputs, uses 7-8 sensors, and gives a display output. Similarly, genealogy trees can be constructed for any other set of features, or other types of electronic circuits as discussed in (removed due to blind review).

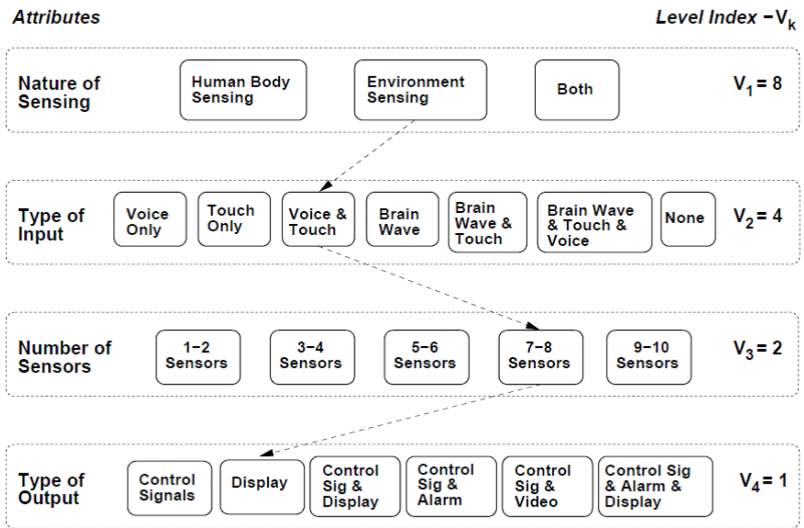


Figure 4.1: Genealogy tree for embedded system design

Following are the arguments that support our statement that the representation in Figure 1 is a genealogy tree as proposed by Shah, Vargas-Hernandez and Smith (2003). Genealogy trees are hierarchical structures in which four consecutive abstraction levels represent the physical principles, working principles, embodiments, and details of a group of designs [112]. Physical principle represents the highest level of abstraction and detail level the lowest. For embedded systems, the nature of signal sensing (input) represents the most basic way of differentiating designs. For example, sensing signals of the physical environment (e.g., temperature, pressure, humidity, etc.) implies devising applications and algorithms that perform control and interfacing while obeying universal laws of physics and chemistry (removed due to blind review). In contrast, sensing signals of the human body (i.e. voice, brain waves, touch, etc.) involves conceptually different purposes and processing activities that must consider the subjective, individual-dependent meaning of signals (Yang, 2006). Hence, the nature of signal sensing represents the most abstract way of distinguishing embedded systems. Moreover, the type of inputs (Level 2) represents the next abstraction level as it decides to a large degree the specific inputs and processing algorithms of a system. For example, an embedded system that monitors a user's level of attention can utilize perspiration monitoring by sensing skin conductance, voice analysis based on audio signal sensing using microphones, or gesture and facial analysis using video cameras. Different interfacing and processing techniques are needed in each case. The third abstraction level corresponds to the number of used sensors as it decides whether local or distributed sensing is implemented and if signals are homogeneous or heterogeneous. The lowest abstraction level is represented by the type of output (Level 4) as it expresses the details of presenting the produced output (e.g., display, alarm, vibration, etc) but without changing the meaning of the output and the algorithmic ways in which it is computed.

A typical design flow includes four stages: problem framing, design ideation, detailed implementation, and evaluation [128]. For the representation in Figure 1, Level 1 covers problem framing, levels 2 and 4 together correspond to design ideation and evaluation, and Level 3 is for detailed implementation. The nature of sensing (Level 1) decides the application domain and the types of functions offered by solutions, which both relate to problem framing. The type of input (Level 2) corresponds to various ideas for realizing the framed goal and function (ideation). The type of output (Level 4) completes a design, therefore enabling the evaluation of a solution. Finally, the number of sensors (Level 3) defines the implementation details of sensing and processing in a solution. These associations were not discussed for the original genealogy tree representation in [112].

Using the genealogy tree in Figure 2, the variety metric is defined as follows [112]:

$$MV_i = \sum_{j=1}^{m}(f_j \times \frac{\sum_{k=1}^{n}(V_k \times b_k)}{MAX_V}) \times R \tag{4.3}$$

where, n is the number of levels in the genealogy tree, and bk indicates the number of branches at level k. m is the total number of features and fj is the feature's weight in the overall variety score. Similar to equation (1), the weights are selected depending on the significance of a feature in meeting the design requirements. The value R normalizes the variety score to a desired range which in our case is [0,10].

Term MAXV is the set's maximum achievable variety, defined as follows [112]:

$$MAX_V = N \times \sum_{k=1}^{n}(V_k) \tag{4.4}$$

where, N is the total number of circuits of the current set and Vk, called the variety index, is a weight associated to each level of the genealogy tree.

The quality ratings of designs were estimated by considering the following performance parameters: precision, cost, power, size, and ease of interface (EoI). First, a pool of input and output devices was created by observing a set of designs. Actual data from an electronic device catalog was used to rate these devices, using the five parameters mentioned above. The devices with highest precision were selected while creating the pool in order to make sure that the functionality is faithfully implemented. In case of a tie, the cheapest of the options was selected. Therefore, these parameters were weighted while calculating quality. Also, since the number of devices and sensors used in each design was different, average ratings of the parameters were calculated before calculating quality. The overall quality rating was estimated using the following equation:

$$MQ_i = 2^0 \times Precision_{Avg} + 2^1 \times Cost_{Avg} + 2^2 \times (Power_{Avg} + Size_{Avg} + EOI_{Avg}) \tag{4.5}$$

Design feasibility was estimated by analyzing the following aspects of the solutions:

1. Complexity of the different parts of the design.

2. Interactions and interfacing between these different parts.

3. Nature of inputs and outputs.

4. Availability of technology for implementing the solution.

The following factors were considered to estimate the utility (usefulness) of a design:

1. Does the design satisfy a need?
2. Does it follow the design constraint?
3. Is there any similar design solution already available?
4. Is the proposed design better than the similar options?
5. Can the solution be used to build other things?

The next section offers a statistical analysis of the results, and Section 3 presents a discussion of the experimental data in practical terms.

4.3 Experimental Results

Participants were senior undergraduate and first-year graduate students in the Department of Electrical and Computer Engineering. The control group (CG) designed voice-based remote controllers in isolation, and spent the entire time on design, refinement, and testing. The experimental group (EG) worked in isolation but examined and discussed in a group setting. CG had 10 participants. EG had 24 participants grouped into six, equal-size sub-groups (DSGs). Each iteration was 15 minutes long, followed by 5 minutes to discuss the solutions. Participants in the EG were told the number of times they had refine the design. The total experiment time was 120 minutes for both groups.

Undergraduate participants were mainly Electrical and Computer Engineering (ECE) senior students in their last semester and graduate participants were mostly first year graduate students in ECE. Hence, we don't estimate that there was a significant difference in their background and experience in embedded system design. Undergraduates were mostly domestic students and graduates were mainly international students. This could produce some differences in their motivation and ability to work in groups. Section A presents our main observations on this issue.

Table 4.1: Novelty Compare: Control Group Vs. Experimental Group

ANOVA Results			
	F	p-value	Fcrit
CG Itr1 Vs. EG Itr1	3.75	0.06	4.15
H0		FTR	ANH
CG Itr2 Vs. EG Itr2	0.44	0.51	4.15
H0		FTR	ANH
CG Itr3 Vs. EG Itr3	1.73	0.20	4.15
H0		FTR	ANH
CG Itr4 Vs. EG Itr4	1.34	0.26	4.15
H0		FTR	ANH

4.3.1 Design novelty

The analysis of novelty ratings considered the variation over four iterations. Figure 2 illustrates the average novelty ratings of the designs in EG and CG for the four iterations. Table 1 shows the average novelty ratings for CG compared to those for EG. H0 represents Null Hypothesis, which always states that there is no significant difference between the means of the two analyzed datasets. F-test was used to determine if Null Hypothesis is accepted or not. F-test computes two values: p-value and F/Fcrit value. p-value indicates if the test "fails to reject" (FTR) or "fails to accept" (FTA) the Null Hypothesis. This indication has to be confirmed using F/Fcrit value. The result of this confirmation is either "accept Null Hypothesis" (ANH) or "reject Null Hypothesis" (RNH). Differences are statistically insignificant, if Null Hypothesis was accepted (ANH) for the comparison, and statistically significant, if Null Hypothesis was rejected (RNH).

In order to compare CG and EG ratings, average values were calculated for each DSG. The average novelty ratings for CG and EG are not statistically different for all of four iterations (Null Hypothesis is accepted in all cases). Thus, the average amount of design novelty produced by CG and EG is similar across all iterations. This rejects Hypothesis 1 indicating that design precedents do not produce more novel designs than if no precedents are used.

The percentage of designs which either retained or improved their ratings for the other iterations is shown in Table 2. For CG, although all but one

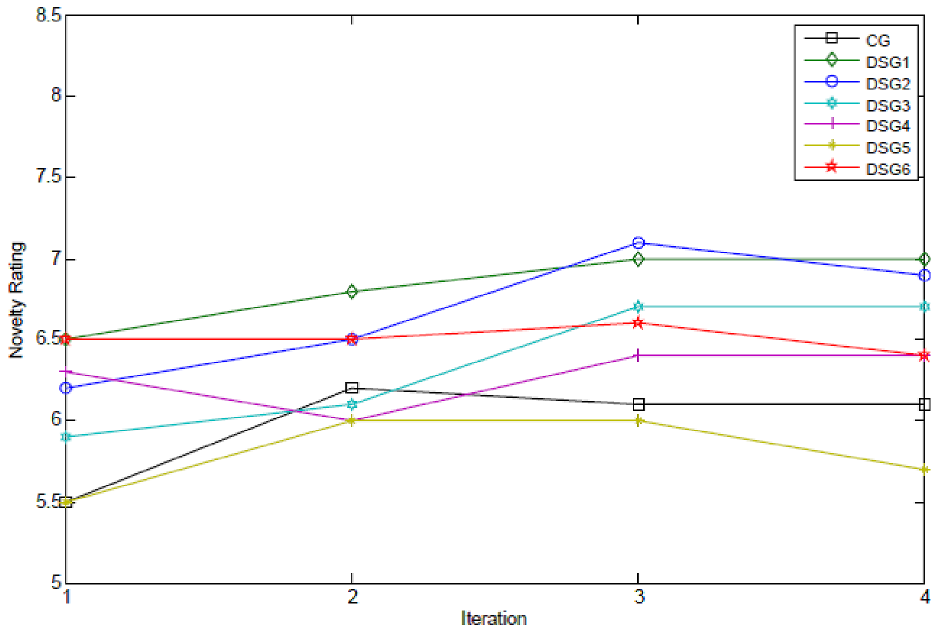


Figure 4.2: Novelty Comparison: Control Group (CG) and Experimental Group (EG)

Table 4.2: Percentage of designs which retained/improved their Novelty ratings

	Itr1 to Itr2	Itr2 to Itr3	Itr3 to Itr4
CG	100	50	50
EG	66.67	62.5	58.3

participant improved their rating from Iteration 1 to Iteration 2, only five of the ten subjects improved from Iteration 2 to Iteration 3, and only three out of the ten subjects improved from Iteration 3 to Iteration 4. For EG, 13 participants increased their novelty from Iteration 1 to Iteration 2, 3 participants had the same novelty, and 8 participants had a decrease in the novelty of their designs. For Iteration 2 to Iteration 3, 13 participants got higher novelty, 2 participants had the same novelty, and 9 participants experienced a lower novelty. Finally, for Iteration 3 to Iteration 4, 8 participants had higher design novelty, 6 participants had the same novelty, and 11 participants produced designs of lesser novelty.

Table 4.3: Percentage Improvement ANOVA results

ANOVA Results			
	F	p-value	Fcrit
CG Vs. EG H0	0.06	0.82 FTR	7.71 ANH

Table 4.4: Novelty Compare: Iteration 3 Vs. Iteration 4

ANOVA Results			
	F	p-value	Fcrit
CG Itr3 Vs. CG Itr4 H0	0.00	1.00 FTR	4.41 ANH
EG Itr3 Vs. DG Itr4 H0	0.04	0.83 FTR	4.05 ANH

Table 3 contains the one-way ANOVA analysis for the data in Table 2. This test also indicates that the amount of innovation remained statistically similar for CG and EG groups for all iterations, which supports the conclusion that Hypothesis 1 is rejected.

Following is a brief description of the main design features used in the different sub-groups. For DSG 1, two designs perform human body (HB) sensing and two perform environment sensing. For DSG 2, from Iteration 2 onwards, one design has HB sensing, two use environment sensing, and one both. For DSG 3, iterations 1 and 2, two designs are based on HB sensing and two designs on environment sensing. For iterations 3 and 4, one design uses HB sensing, two designs rely on environment sensing, and one design on both. For DSG 4, one design considered HB sensing and three designs used environment sensing. For DSG 5, all four designs use the most common features in almost all iterations: inputs are for environment sensing or voice/none, and outputs are control signals or display. For DSG 6, two designs implement HB sensing, and two use environment sensing. Subgroup DSG 5 has the lowest average novelty.

Table 4 compares the novelty ratings for Iteration 3 vs. the ratings for Iteration 4, for CG as well as EG. The results indicate that there is no

Table 4.5: Variety Ratings: Control Group (CG) and Experimental Group (EG)

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
CG	3.9	4.5	4.1	4.4
EG	2.2	2.6	2.9	2.8

significant difference in the means for the two iterations. The Null Hypothesis is accepted, thus Hypothesis 5 is rejected. This suggests that adding new functional requirements after several iterations does not increase the novelty of designs.

We further analyzed the impact on novelty due to ideas borrowed by certain designs from other designs in their subgroup (DSG). A total of 16 ideas were borrowed by the designs in the six subgroups. Most ideas were borrowed after Iteration 1 (62.5%) and Iteration 3(25%). The novelty ratings improved in 8 situations (between 1.3% and 21.8%). In most cases, the borrowed ideas were added as features in the current designs, hence did not generate significant novelty increases. However, there was one case where an entirely new design emerged based on a borrowed idea: In Iteration 1, Design 1 used heart-rate monitoring as one of the features in the design "Control iPod music while jogging". Design 3 borrowed this idea and proposed a new design ("Intelligent heart monitor for patients") based completely on this new concept. This case represents developing a conceptually new solution using an idea inspired by design precedents. In another three cases, the borrowed ideas related to the type of input (Level 2) and increased novelty ratings between 10.5% and 12.7%. Finally, some adopted ideas corresponded to number of sensors (Level 3) and output type (Level 4) to produce novelty increases between 1.3% and 5.5%. The novelty ratings of the other 8 instances of borrowed ideas dropped (between -1.7% and -15.8%) or remained constant. These designs simply adopted the borrowed ideas without adding any new features.

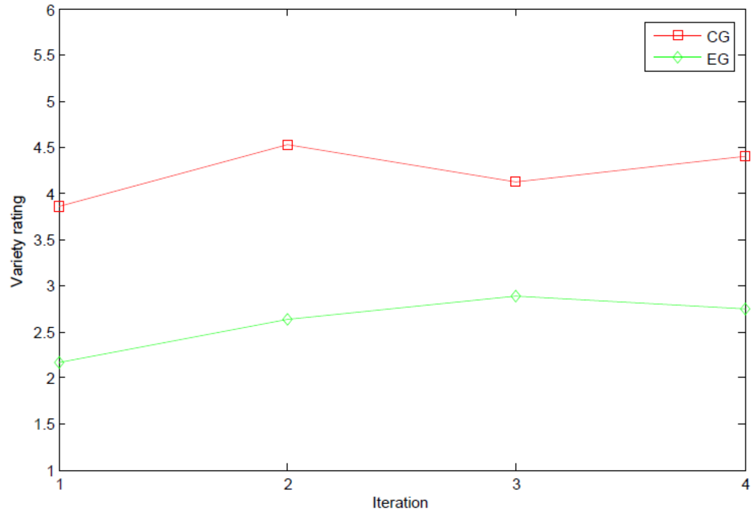


Figure 4.3: Variety Comparison: Control Group (CG) and Experimental Group (EG)

Table 4.6: Variety Compare: Control Group Vs. Experimental Group

ANOVA Results			
	F	p-value	Fcrit
CG Vs. EG H0	56.14	0.00 FTA	5.99 RNH

4.3.2 Design Variety

The variety ratings for CG and EG are given in Table 5. CG variety rating is higher than EG rating for all iterations. This result indicates that CG participants used a broader range of features for their designs. However, since EG participants interacted after each iteration, they had an option of either borrowing ideas from others or intentionally avoiding commonly used features. Since the genealogy tree has weights associated with each level, if features from a higher level were added to a design, the variety rating improved more than when feature from lower level were added. This explains the fluctuations in the plots shown in Figure 3.

We compared the variety ratings over the four iterations. Table 6 shows the result of the ANOVA test. We concluded that there is significant difference in the means between the ratings for CG and EG. The Null Hypothesis is rejected. Hence, Hypothesis 2 is accepted indicating that design precedents produce design solutions of lesser variety (converging solutions) than if no precedents are used (divergent solutions). The total number of features (by combining all levels of the genealogy tree) used by CG designs was sixteen features. The twenty four designs in EG utilize a combined total of twenty design features. All sixteen features in CG are in the EG too. In addition, the four new features (inputs: brain wave, BW & T, BW & T & V; output: CS & V) are found only in three designs in EG. The remaining twenty one designs use the same feature space as CG. Even though CG has a slightly smaller feature set than EG, the features of CG are combined in a larger set of alternatives. This explains the higher variety rating for the CG population for all iterations.

As expected, there is a correlation between design novelty ratings and variety ratings. For example, eight out of the ten designs in CG had similar features. The remaining two designs contributed features which were very different (sensing: environmental and HB sensing; output: display, CS & A & D) from the feature-set present in the other eight designs. For example, in Iteration 1, Design 18 is the only one using three out of the thirteen features. Therefore, the novelty rating for the two designs is high (between 7.5 and 9) while the others have low novelty (between 4 and 6.5), which caused the average to drop. In EG, since there was exchange of ideas, there is no such subset of designs which outperformed the others. Therefore, the novelty ratings for all designs in EG are in the same range, between 5 and 8. Solutions in EG are more convergent while solutions in CG are more divergent, including instances with both high and low novelty ratings.

The analysis of the novelty and variety ratings for undergraduate vs. graduate students was also conducted (removed due to blind review). Following are the main observations: 1) average novelty ratings are highest for CG graduate students and EG undergraduate students suggesting that graduate students work better alone while undergraduates are more creative in groups. 2) For both CG and EG, novelty increases for graduate students for each consecutive iteration and decreases for undergraduates. This suggests that graduates were more motivated in creatively improving their design during the entire experiment. 3) Average variety ratings for CG graduate students are highest followed by CG undergraduates, EG undergraduates, and EG graduates (in that order). However, we could not conclude that undergraduates or graduates were more successful in producing designs of higher variety. The observed differences between the design novelty ratings for undergraduates and graduates might be a result of different levels of motivation as well as different educational experiences as undergraduates were mainly domestic students and graduates were mostly international students. Understanding the implications of these differences between the two groups requires more detailed studies, which were beyond the scope of this work.

4.3.3 Design Quality

Since the implementation level contains maximum details about the design, the most reliable way to estimate the performance ratings is by including information at this level. Hence, equation (5) for the quality metric was instantiated based on the features of implementation level.

The average quality ratings of the designs in EG and CG for the four iterations are shown in Figure 4. The quality ratings for CG were compared with those for EG using one-way ANOVA test for each iteration, as shown in Table 7. The Null Hypothesis was accepted for all the iterations suggesting an equivalence of means between quality ratings for CG vs. EG. This conclusion rejects Hypothesis 3 as there is no difference between the average qualities of CG and EG designs.

The lowest quality rating was 4.18 and the highest was 9.46. For CG, the quality rating is highest ($Q = 9.37$) for Design 10 in iterations 1, 3, and 4. Design 6 has the highest score for Iteration 2 ($Q = 9.31$) and the second-highest ratings for all other iterations. However, it is interesting to note that although both these designs have consistently scored high in quality, their novelty ratings were low. Also, Design 6 is identified to be infeasible. On the other hand, Design 9 has the lowest quality rating for all four iterations ($Q =

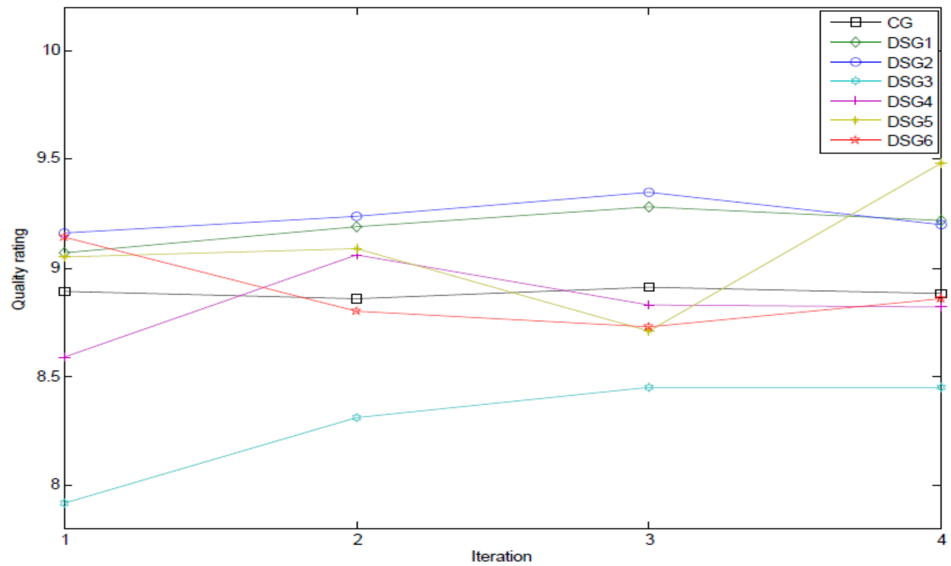


Figure 4.4: Quality comparison for Control Group (CG) vs. Experimental Group (EG)

7.41) but it has a medium novelty rating. Although these observations suggest an inverse relationship between the novelty and quality ratings, it is important to note that Designs 2 and 7 are feasible and have high novelty ratings and high quality ratings for all iterations.

For EG, Design 8 has the highest quality (Q = 9.46) for iterations 1, 2, and 3, and the second-highest rating (Q = 9.46) for Iteration 4. However, a few other designs also share this rating (Q = 9.46) for some of the iterations:

Table 4.7: Quality Compare: Control Group Vs. Experimental Group

ANOVA Results			
	F	p-value	Fcrit
CG Itr1 Vs. EG Itr1 H0	0.03	0.86 FTR	4.15 ANH
CG Itr2 Vs. EG Itr2 H0	0.21	0.65 FTR	4.15 ANH
CG Itr3 Vs. EG Itr3 H0	0.00	0.93 FTR	4.15 ANH
CG Itr4 Vs. EG Itr4 H0	0.41	0.53 FTR	4.15 ANH

Design 13 in Iteration 1, Design 16 in Iteration 2, and Designs 5 and 6 in Iteration 3. Design 18 has the highest quality in Iteration 4 (Q = 9.90). Again, although Design 8 has high quality in all iterations, it has low novelty. On the other hand, Design 11 is infeasible, has low quality in all the iterations but it has high novelty. Also, Design 14 has high novelty, medium quality, and is infeasible. An exception to this trend would be Design 5 which has high novelty and high quality ratings for all the iterations.

Quality ratings were also calculated separately for each design activity, e.g., problem framing (Level 1 in Figure 1), ideation (Level 2), detailed implementation (Level 3), and evaluation (Level 4). Quality was computed for problem framing using the following parameters.

For designs that sense the environment:

- Sensor location: indoors or outdoors.
- Discrete measurements or continuous measurements (sensing).
- Monitor parameters or control parameters.
- Safety: potentially damaging to itself or other devices.

For designs that performed human-body sensing:

- Sensor locations: internal or external.
- Discrete measurements or continuous measurements.
- Monitor parameters or control parameters.
- Invasive or non-invasive.

For the ideation and evaluation levels, the parameters used to compute quality ratings were the same as used for implementation level. However, the designs were treated as black-boxes and sensor details were not taken into consideration because these details relate to the implementation level.

Table 8 presents the comparison results for the following three cases: problem framing (Level 1) vs. implementation (Level 3), problem framing

Table 4.8: Quality Comparison

ANOVA Results	F	p-value	Fcrit	ANOVA Results	F	p-value	Fcrit
Level 1 Vs. Level 3				Level 2 and 4 Vs. Level 3			
Itr 1 Vs. Itr 1 H0	9.23	0.00 FTA	3.99 RNH	Itr 1 Vs. Itr 1 H0	56.41	0.00 FTA	3.99 RNH
Itr 2 Vs. Itr 2 H0	17.33	0.00 FTA	3.99 RNH	Itr 2 Vs. Itr 2 H0	89.87	0.00 FTA	3.99 RNH
Itr 3 Vs. Itr 3 H0	22.09	0.00 FTR	3.99 RNH	Itr 3 Vs. Itr 3 H0	79.09	0.00 FTA	3.99 RNH
Itr 4 Vs. Itr 4 H0	21.56	0.00 FTA	3.99 RNH	Itr 4 Vs. Itr 4 H0	112.61	0.00 FTA	3.99 RNH
Level 1 Vs. Level 2 and 4							
Itr 1 Vs. Itr 1 H0	14.31	0.00 FTA	3.99 RNH				
Itr 2 Vs. Itr 2 H0	21.83	0.00 FTA	3.99 RNH				
Itr 3 Vs. Itr 3 H0	16.79	0.00 FTA	3.99 RNH				
Itr 4 Vs. Itr 4 H0	25.91	0.00 FTA	3.99 RNH				

(Level 1) vs. ideation and evaluation (Level 2 and 4), and ideation and evaluation (Level 2 and 4) vs. detailed implementation (Level 3). The Null Hypothesis is rejected in all iterations, for all the cases. This indicates that there are statistically significant differences between the quality ratings of the four design activities.

Most variations of the quality ratings over the four iterations were at the ideation and implementation levels, and less for problem framing. This is because most design additions were at the ideation and implementation levels. The ratings at Level 1 (problem framing) change only if the concept of the design is modified either by selecting a new application or significantly modifying the functionality.

4.3.4 Utility ratings

Table 9 presents the comparison of the utility ratings for CG and EG. Utility rating considered the level of desirability for CG and EG designs, e.g., the offered functions are attractive to users. The Null Hypothesis was accepted

Table 4.9: Utility comparison for Control Group vs. Experimental Group

ANOVA Results			
	F	p-value	Fcrit
CG Itr1 Vs. EG Itr1 H0	0.16	0.69 FTR	4.15 ANH
CG Itr2 Vs. EG Itr2 H0	0.61	0.44 FTR	4.15 ANH
CG Itr3 Vs. EG Itr3 H0	0.04	0.84 FTR	4.15 ANH
CG Itr4 Vs. EG Itr4 H0	0.13	0.73 FTR	4.15 ANH

for all iterations. However, CG had an average utility of 5.8 in Iteration 1 and an average utility of 6.2 in Iteration 4. EG had an average utility of 5.6 in Iteration 1 and an average utility of 6.4 in Iteration 4. The utility increase for EG is statistically higher than CG. Hence, Hypothesis 4 is accepted indicating that design precedents cause increase in utility over iterations. Note that only 1 out of 10 designs in CG had a utility rating higher than 8.0, while 6 out of 24 designs in EG had a utility greater than 8.0.

For CG, the utility results ranged from 3.5 to 7.3 in Iteration 1 while in Iteration 4, the range was 4.6 to 9.2. Six out of 10 designs saw an increase in utility over the four iterations. Design 5 had the highest utility in Iteration 4. The design is for an 'intelligent' refrigerator that keeps track of the food items (quantity, nutritional value), informs the user via email and text when certain items need to be restocked and makes suggestions on the closest place to purchase those items.

For EG, the results in Iteration 1 ranged from 3.5 to 8.8. The range for Iteration 4 was from 5.0 to 10. Fifteen out of 24 designs saw an increase in utility from Iteration 1 to 4. Design 11 had the highest utility in Iteration 4 because the design uses a remote control to encode brainwaves (neural activity) and transmits it to other users as a means for 'telepathic' communication. In the first three iterations, the design uses the remote controller to control different devices using brainwave mapping (commands).

4.4 Discussion

The novelty ratings of the control (CG) and experimental groups (EG) are similar during iterative design. This result rejects Hypotheses 1 which states that the use of design precedents results in more novel designs. The variety of CG designs is larger than the variety of EG solutions suggesting that each EG converged to a set of ideas that were considered to be best in terms of feasibility and utility. The Hypotheses 2, which states that precedents lead to more converging solutions and lesser variety, is accepted. It seems that EG participants spent some of their time on incorporating borrowed ideas into their own solutions rather than identifying new features.

Every design in CG introduced new features in Iteration 2, suggesting that the initial iteration had a positive role in helping ideation in problem understanding, framing, and detailed solving. Perttula and Liikanen [104] report a similar observation according to which design novelty increases over time even if designers work independently. Statistically, there was a difference in the increase of average utility ratings between CG and EG designs, which accepts Hypothesis 4. However, most participants did not make (in Iterations 2, 3, and 4) major changes to their designs that would impact significantly the overall utility. Most designs only added a couple of sensors to include additional but minor functionality. Only a few subjects made major improvements or proposed entirely new designs in the latter iterations. Hence, group setting and discussions in iterative design help design utility evaluation and aid converging to a smaller set of solutions, but are less useful in increasing novelty or improving implementation quality (Hypothesis 3).

Our conclusions about the impact of precedents on design novelty differ from those of studies in other design domain, like architecture. Reports show that precedents have a beneficial role for creativity, like they improve novelty [87], avoid existing designs [69], and spawn new ideas. One reason for these differences could be the nature of precedents. Previously used precedents are well documented, fully specified, and widely accepted solutions. Our precedents are intermediate designs, which are more typical for emerging application domains. In such situations, having well established solutions is unlikely. Moreover, our precedents did not serve deeper understanding of the problem as in most cases participants borrowed ideas on certain functions but abstained from reframing their problem, from modifying goals, or from identifying new contradictions in their solutions. Precedents did not expose hard-to-anticipate, global properties like in Senbel, Girling, White, Kellett and Chan (2013). Another reason could be the difficulty of the open-ended problem. Participants got fixated on their previous solutions or some of the communicated designs as they might have considered the problem as too challenging. The main benefit of precedents is in improving design utility as the iterative, group-based process helps reinforcing design features considered as valuable, discarding less important functions, and adding more details, which also enhances the desirability of solutions in real situations.

Hypothesis 5 is rejected as adding new requirements later in the design process did not increase novelty. One reason could be the fact that the main problem requirements remained unchanged. The new solutions had incremental changes that included the new requirements, however, without modifying the core of the solutions. Rarely, a borrowed idea triggered an emergent design, such as by placing a novel conceptual idea into a new problem framing. This is in contrast to the findings reported in Pertulla and Liikanen [104], which indicate that initial designs act only as temporary constraints.

Our experiments show that most borrowed ideas are very specific, such as precise functions. Producing emergent features seems to represent rare events (but not necessarily accidental events), which is hard to describe as statistical models for creativity because emergence was not correlated to the commonality or dissimilarity of designs. This suggests that producing emergent features probably requires borrowing more abstract concepts which are then instantiated for the specific solution under development. Using metaphors or analogies in problem framing and development of new solutions could improve design novelty. Similarly, Lawson [99] reports that experts use more compressed knowledge as precedents, such as symbolic references, while novices rely on more concrete, geometric descriptions.

Experiments suggest that group formation is an important factor in deciding the final novelty, quality, and utility of designs. If group members have a tendency to generate similar designs, it is likely that the feedback during discussions acts mainly to reinforce the selection of the same features, thus produce less novel ideas. An important corollary is that the initial pool of design ideas (e.g., after Iteration 1) must be sufficiently diverse, and can be used as a predictor of the expected design novelty after iterative design. However, team diversity can also result in lesser innovation because of social categorization [124]. Similar to Coskun, Paulus, Brown and Sherwood [77], we also found that more new ideas are created initially, such as in Iteration 2. However, we believe that this is not only due to fixation and lesser motivation, but also due to the group settings, which acted like filters to eliminate less promising ideas. The filtering effect is expected to be stronger than remote association accessing as the variety of CG is lesser than for EG. Experiments showed for many designs an inverse correlation between novelty and quality. Designs of high novelty were more likely to have lower quality. The main differences in quality are due to (the estimated) EoI, cost, and power consumption. This suggests that coming up with ideas for new embedded system designs might increase the cost and power consumption of the solutions while the interfaces with users are not well optimized. More importance to design quality should therefore be set during ideation and evaluation. This observation is similar to the much broader conclusions in Goldenberg, Lehmann and Mazursky [90]. They suggest that major technological changes might produce market failures either because of their high cost or low product quality.

We think that problem framing is the most important step in deciding the overall novelty of electronic embedded system designs as it uncovers new business niches and applications for which the existing building blocks and domain knowledge can be utilized to create novel solutions for existing needs. For example, a new concept is the idea of sensing physiological signals of human bodies to understand their ongoing activity. This is a new way in controlling an embedded system, based on implicit, passive participation of users, in contrast to traditional methods in which they explicitly and actively indicate their intentions. Once the principle is selected, various solving strategies, based on optimization, similarities, reuse, and design patterns, can be employed to produce an efficient design. Careful problem framing is also important as it is likely to improve the quality of final designs [72].

4.5 Conclusions

This chapter presented a study about the role of precedents in illuminating creative ideas during iterative design procedures for open-ended problems in electronic embedded systems. Precedents were defined as the solutions and solution features developed during the iterative solving process by the participants in a group. Through an experiment grounded in cognitive psychology, this work explored the influence of precedents on novelty, variety, quality, and utility of design solutions devised during four consecutive iterations. Another tested hypothesis was that incrementally changing problem requirements improves design novelty. The experimental procedure was based on the method proposed by Mobley, Doares and Mumford (1992) as this procedure captures more accurately the nature of embedded system design than other standard creativity tests.

Experiments showed that precedents did not increase design novelty as compared to the group that did not utilize precedents. Most novel ideas were proposed in Iteration 2 independently of precedents being considered or not. Novelty rate slightly decreases as more iterations are performed. Precedents in iterative design flows reduced the variety of design solutions as the group setting seemed to filter out ideas that were perceived as less promising. There was no statistical difference between average novelty and quality ratings of the designs created while precedents were communicated or not, but utility was higher for the designs created in a group setting that shared the developed solutions. The only design with emerging features originated in a group setting in which one design idea was conceptualized by another participant to create a very innovative solution. This suggests that devising truly novel design ideas seems to be more of a rare event than a systematic process described through steady-state, statistical equations. Most of the borrowed precedents were simply added to an existing solution without changing the main characteristics of the solution. Modifying the design specification after three iterations did not produce significant novelty increases. Precedents and the iterative, group-based process improved utility by reinforcing features considered to be valuable, discarding less important functions, and adding more details to the design implementations. Many solutions had an inverse relation between novelty and quality.

In our experiments, precedents did not improve creativity as compared to the participants that did not use them. This suggests that, likely, precedents did not help deeper understanding, expose hard-to-anticipate, global properties, or reduce fixation. This is in contrast to the findings of other studies on the impact of precedents in domains like architecture. However, these precedents were well documented, well accepted designs while the precedents in our case were intermediate designs, which were subject to further development. Intermediate designs are typical for emerging technological domains in which accepted, mature solutions are not available yet.

Solving open-ended design problems in electronic embedded systems remains a difficult task. More effective design methods and CAD tools are needed. Existing approaches for co-evolving problem formulations and solutions [82, 105] can be a starting point but the specifics of embedded system design must be also addressed. Moreover, group settings seem not to help in producing design alternatives with a high variety of features. Instead, designers should work individually and then use group settings to evaluate the utility of their solutions. Also, group interactions should involve descriptions at a more conceptual level rather than express mainly details on functions and implementations. Finally, design creativity results are correlated to the participants' motivation to effectively solve the exercises, especially during the later design iterations. The levels of motivation must be continuously monitored as decrease in motivation can reduce creativity. We plan to tackle these issues in our future studies.

Chapter 5

Two Experimental Studies on Creative Concept Combinations in Modular Design of Electronic Embedded Systems

[1]

This chapter discusses the nature of concept combinations in modular design of electronic embedded systems as well as the relation between combination characteristics and novelty, quality, and usefulness of the produced solutions. Through two experimental studies, this work explored the frequency of relation-based and property-based combinations in embedded design solutions, and how the specifics of the given building blocks, i.e. salience, relatedness and number, influenced the produced combinations. The impact of popular aids, like titles and short descriptions (briefs), in improving novelty, quality, and usefulness of the designs was also analyzed. Design solutions include mostly relation-based combinations. Design novelty correlates mainly to the purpose and context of the produced combinations. Novelty is aided by titles but not by briefs.

[1] Note: This chapter is based on the work published in [7].

5.1 Introduction

Design methodologies for electronic embedded systems stress the importance of modularity. Modular design solutions are created by connecting basic building blocks with well defined functionality, interfaces, and performance, e.g., components, library circuits, or intellectual property (IP) blocks. Modular design reduces design cost and effort by reusing building blocks, and enhances design correctness as repeated testing and verification of blocks eliminate most of their errors [76, 153]. New blocks are rarely created. Thus, designing original electronic embedded systems mainly involves finding new ways to relate blocks. This explains the significance of finding novel and useful combinations among building blocks.

The importance of concept combinations in creativity has been intensely studied by research in cognitive psychology [117, 144, 167]. Concept combinations are of three kinds. Property-based combinations transfer features from one concept, called modifier, to another concept, called head concept [174]. For example, Lagne [157] explains that in “zebra clam”, property “stripes” of the modifier concept “zebr” is transferred to the head concept “clam”. Relation-based combinations introduce new relations between concepts [157, 176]. For instance, “mountain stream” is a relation-based combination that defines a location-based connection between concepts “mountain” and “stream” [157]. Hybrid combinations are a mixture of relation and property-based combinations, such as combination “musician painter” [175]. Various conditions influence the kind of produced combinations, like the salience of concept features [149], and the similarity and abstraction level of combined concepts. More similar concepts originate more property-based combinations, while dissimilar, yet easy- to-relate concepts create more relation-based combinations [175]. Abstract concepts favor relation-based combinations, and basic concepts help property-based combinations [161]. Property-based combinations are harder to create [157] but enable new features beyond those of the initial concepts [175], even though other studies challenge these findings [173, 176]. Creativity is higher for concepts with less typical, less salient features [148], combinations of dissimilar features

[150, 161], abstract concept combinations [126], and anomalous combinations [135, 173].

Enhancing creativity in electronic embedded system design based on the insight gained from studies in cognitive psychology is not straightforward. Studies in psychology rarely capture the specificity and complexity of embedded system design problems. Problems in embedded system design are often wicked (ill-defined). Wicked problems express loosely or incompletely specified requirements, or present needs based on organizational or personal perspectives, judgments, predictions, or beliefs [78, 80]. Modularity is intrinsic to the design process [79, 81, 105] while in other domains, like architecture or mechanical engineering, modularity is less common or identifying modules is a step that follows design and implementation [121]. Also, embedded systems are programmable, which enhances their capability to be customized to specific needs. Thus, designs can achieve higher utility [105]. Finally, electronic systems are more complex than other engineering systems, e.g., in mechanical engineering, therefore conceptualization is harder [80]. However, the complexity of electronic systems can be effectively tackled through top-down design methodologies, in which design activities are performed separately at consecutive levels of abstraction, including behavioral level, logic level (e.g., gate netlist, schematic), and physical level (i.e. layout). In spite of these differences, devising new experimental studies based on work in cognitive psychology is instrumental in understanding the connections between creativity in electronic embedded systems design, the characteristics of the produced concept combinations, and the specifics of the domain-specific design knowledge, e.g., the nature of the existing building blocks. Gaining insight on these connections is important to devise more effective methodologies and CAD tools for design innovation.

This chapter presents two experimental studies on the nature and characteristics of concept combinations in modular, electronic embedded system design and the implications of these characteristics on the creativity of the produced designs: " The first experimental study explored the nature of concept combinations in design solutions (e.g., property-based, relation-based, and hybrid combinations), and how the attributes of the given building blocks, i.e. salience, relatedness and number, influence the novelty, quality, and usefulness of the combinations. " The second experimental study addressed the importance of using design aids, like titles and short descriptions (briefs), in improving the novelty, quality, and usefulness of final design solutions. This analysis was motivated by similar studies in domains like construction and architecture. Studies show that various aids can increase creativity, e.g., design briefs [165], word graphs expressing semantic associations between annotations [166], and video-story creation [163]. However, other aids, like sketching, are found to be less critical in early conceptual design in architecture [136].

Our two experimental studies are based on the work by Mobley, Doares and Mumford [102]. Their experimental procedure on concept combinations is an elegant and simple representation of the main elements in general-purpose, creative activities, e.g., (i) the involved domain knowledge (described by a given set of exemplars), (ii) information structuring and aggregation (emulated through category labels), (iii) gaining insight into a problem description (mimicked through the step of enumerating more exemplars), and (iv) producing solutions (captured through story writing). Similar to the sets of exemplars in Mobley, Doares and Mumford (1992), design flows for electronic embedded systems assume the existence of domain knowledge in the form of libraries of building blocks (modules) with well defined functions, interfaces, and properties.

The insight gained through the two experimental studies on concept combinations is important to develop new design methodologies and CAD tools for innovation in electronic embedded system designs. For example, if experiments show a biasing towards relation-based combinations then new strategies are needed to produce more property-based or hybrid combinations. Focusing also on property-based and hybrid combinations would produce more alternative representations for concepts and their combinations, which is likely to improve solution creativity [152, 164]. Also, the insight on the efficiency of using the two design aids is useful as designers often lose sight of the pursued goals or get fixated on previous solutions. But, as titles and briefs are regularly used in design, understanding their impact on concept combinations and design creativity can help devising more effective ways to utilize these aids in a focused pursuing of design goals, quicker defixation, and more robust divergent thinking.

The chapter has the following structure. Section one presents the specific innovation issues in electronic embedded system design studied in this work. Section two presents the experimental studies. Section three discusses the results. Section four offers conclusions.

5.2 Experimental Procedure

5.2.1 Concept Combinations in Modular Embedded System Design

Our work devised the following two experimental studies on design innovation in modular, electronic embedded system design.

Study 1: What is the nature of concept combinations in modular design of electronic embedded systems? What are the novelty, quality, and usefulness of combinations in such designs?

The following five hypotheses were defined as part of the first study.

Hypothesis 1: Relation-based combinations are more common in modular, embedded system design than property-based combinations.

Hypothesis 2: The ratio of relation-based combinations and property-based combinations is not affected by the similarity of the given basic blocks.

The two hypotheses state that modular electronic, embedded system designs include mainly relation-based combinations and few property-based combinations. The hypotheses are supported by the observation that property-based combinations are rare in everyday life [143]. Also, the last resort hypothesis [174] indicates that, during problem solving, humans first consider creating relation-based combinations, and if they fail then they analyze property-based combinations. In addition, the nature of electronic basic blocks and the specifics of current design methodologies suggest that relation-based combinations are dominant. Blocks in embedded system designs have mostly orthogonal features, which are functionally different and satisfy distinct needs. The orthogonal features of blocks are tied together such that the outputs of a block act as inputs to another block. New solutions are built by iteratively adding, deleting, and replacing modules. Utilizing blocks with orthogonal features is justified as having multiple, functionally-similar blocks as parts of the same design does not offer new functional capabilities but increases complexity and cost.

The study also aimed to understand the kind and amount of features in creative design solutions produced through concept combinations. Creative solutions must be novel, feasible, and useful [133]. Novelty refers to the number of features that are not present in the given blocks or in other similar solutions. Feasibility states that a produced concept combination and its features are correct with respect to the domain-specific knowledge, including physical and engineering laws. Usefulness represents the capability of the design solution to satisfy the requirements of the problem. The next three hypotheses express the relations between the nature of combined concepts and the novelty, quality, and usefulness of resulting combinations. Independent variables were the characteristics of building blocks, i.e. their salience, relatedness, and number.

Hypothesis 3: The novelty of the design solutions produced through relation-based combinations is higher if the combined building blocks are less related, and lower if the blocks are more related.

Hypothesis 4: The quality of the design solutions produced through relation-based combinations is lower if the combined building blocks are less related, and higher if the blocks are more related.

Hypothesis 5: The usefulness of the design solutions produced through relation-based combinations is lower if the combined building blocks are less related, and higher if the blocks are more related.

Hypotheses 4 and 5 suggest that related building blocks are likely to be correctly and effectively used together again in new solutions as they have been already utilized together in a larger variety of conditions and solutions, thus their features (capabilities) are likely to be orthogonal and enhancing of each other for a broad set of conditions. In contrast, exemplars that have not been often used together are more likely to induce incorrect relations as their capabilities are not fully understood, or because some of their capabilities are conflicting. While the literature indicates that unexplored situations have the potential of producing new solutions [144], the risk of having incorrect or inefficient relations is also higher. Finally, cause-effect relations, like input - output connections between blocks, are very natural to human reasoning as inputs express an event that triggers a reaction (response) represented by the output [139]. In contrast, less common relations, like circular relations between concepts, represent in day-to-day life inconsistent formulations, like paradoxes. However, circular relations are useful in circuit design as they induce features beyond those of the building blocks, even though the risk of incorrect operation remains high.

Engineering design has studied extensively the connection between design creativity and dissimilarity of source and target domains in analogical reasoning [125, 170]. Analogical reasoning, a popular method in engineering design innovation [137, 138, 141], transfers features of the source (e.g., an existing design) to the target solution. Analogical distance quantifies the degree of dissimilarity between concepts [138, 172]. Both across-domain (hence, more distant) and within-domain (thus, closer) analogies are used in design [138]. Distant analogies improve creativity [141] but close analogies based on superficial features are easier to identify.

Study 2: How important are design aids, like titles and design aids, in concept combinations for modular, electronic embedded system design?

The second study explored the importance of aids, like titles and short descriptions (briefs), in improving the novelty, quality, and usefulness of concept combinations in electronic embedded systems. Briefs have been shown to be helpful in enhancing creativity in developing novel product advertisements [154]. In embedded system design, briefs are used for design specification and communication between the involved parts but are less utilized as tools to enhance novelty and improve the transformative potential of design solutions.

The next two hypotheses were considered.

Hypothesis 6: Using titles to summarize a design solution improves the novelty of the solutions.

Hypothesis 7: Using short descriptions (briefs) to describe a design solution improves the novelty of the solutions.

Our hypothesis states that "title-like descriptions" as well as short descriptions (briefs) are expected to offer more insight into the uniqueness and usefulness of a solution by contrasting its features to the pursued goal. Titles and briefs act as aggregators of solution features, hence, indicate their purpose and distinguishing attributes. According to our hypothesis, contrasting a title to the purpose of a design solution followed by repeating the design exercise is expected to enhance creativity, as the contrast highlights how well the purpose matches the problem requirement (usefulness aspect) and how unique features are (novelty). Subsequent iterations that integrate the uniqueness attributes to the requirements and then repeat the design process for the new purpose are likely to strengthen creativity due to the explicit focus on what is perceived to be original. Also, producing new relations is aided by explicitly expressing in a short description (brief) the purpose of a solution [154, 155]. Solution purposes (goals) set directions along which new relations are created. The preciseness of purpose impacts the relations [140]: tight purposes make the search difficult as there might be few feasible combinations. Loose purposes create only a weak guiding of the search. There is a trade-off between the number of feasible solutions and the directionality along which solutions must be searched.

5.2.2 Experimental Studies

This section details the two experimental studies meant to verify the seven hypotheses of the two studies. The studies are based on the work on concept combinations by Mobley, Doares and Mumford [102]. In their original experiment, participants are asked to find a label that encompasses three groups of exemplars, use the label to propose other similar exemplars, and then develop a story that uses the exemplars. The novelty and quality of the labels, exemplars, and stories are rated. Similarly, our experiment gave participants a group of exemplars (e.g., basic building blocks used in electronic embedded systems) that had to be utilized to develop an innovative design, including its purpose (similar to the labeling step in Mobley, Doares and Mumford's experiment) and implementation diagram (similar to story writing in the original experiment).

Description of the experimental studies

Study 1: The five hypotheses of the first study were tested using the following experiments. The first group of participants used exemplars (building blocks) that are common in electronic embedded system problems but their functionality is orthogonal, thus well distinguished from each other. The second group considered unrelated exemplars, such as the exemplars are rarely used together in solutions. For example, GPS are mainly utilized for mobile applications, while cooking stoves and hair driers are static devices. The third group utilized exemplars that are often used together. For example, camera, sonar, GPS, accelerometer, speakers, alarm signals are typical components of mobile devices, like smart phones and tablets. The fourth group used a larger set of common exemplars. The subjects in all four groups worked individually. Table 1 lists the devices given to each group of participants.

Table 1: The four sets of building blocks used in the two experimental studies

Each group utilized the following problem description: "Using the above devices, develop a novel electronic embedded system that you feel is useful in your household. Use as many devices from the list as possible. Provide a description of the system that presents how the devices relate (interact) with each other. The solution will be rated based on its level of novelty (uniqueness compared to the solutions of your colleagues and designs discussed in textbooks, media, web), its usefulness to solve the problem, and its correctness (feasibility to be implemented). The experiment time is 10 minutes."

The problem description for Group 4 was developed to explicitly test Hypothesis 2 and as an additional experiment to check Hypothesis 1. Markman

Table 5.1: The four sets of building blocks used in the two experimental studies

<table>
<tr><th colspan="3">Group 1</th><th colspan="3">Group 2</th></tr>
<tr><td>gas sensor
GPS
temperature sensor
accelerometer
microphone</td><td>wireless link
OpAmp
resistors/ capacitors
processor
memory</td><td>on/off switch
light
motor
fan</td><td>on/off switch
GPS
voice recorder
pressure sensor
capacitive sensor</td><td>processor
memory
wireless link
OpAmp
resistors/ capacitors</td><td>cooking stove
robotic arm
hair drier
latch</td></tr>
<tr><th colspan="3">Group 3</th><th colspan="3">Group 4</th></tr>
<tr><td>GPS
camera
accelerometer
sonar
touch screen</td><td>wireless link
Ethernet link
OpAmp
memory
processor</td><td>on/off switch
speaker
alarm signal
color display</td><td>gas sensor
humidity sensor
GPS
sonar
temperature sensor
accelerometer
movement detection sensor
microphone
voice recorder
pressure sensor
touch screen
capacitive sensor</td><td>wireless link
Ethernet link
OpAmp
analog frontend
resistors/ capacitors
processor
FPGA
memory
flash memory</td><td>on/off switch
light
robotic arm
motor
fan</td></tr>
</table>

Table 5.2: Nature of concept combinations

Group	Nr. of designs	Relation-based Combinations (correct only)	Property-based Combinations (correct only)	Average number of relation-based combinations
1	21	70 (70)	5 (0)	3.33
2	17	42 (42)	3 (0)	2.47
3	18	38 (38)	0 (0)	2.11
4	12	36 (36)	0 (0)	3.00
Total	68	186 (186)	8 (0)	

and Wisniewski [161] suggest that having more similar exemplars encourages property-based combinations through property migration from one concept to another. However, if most combinations are still relation-based then having more exemplars (and likely more similar features too) should not affect the amount of property-based combinations as these are rare. Otherwise, the amount of property-based combinations should be higher than for Group 1.

Study 2: The second experimental study verified Hypotheses 6 and 7. Participants were given the same building blocks as those for Group 4 in Table 1. The subjects in all three groups worked individually

The first group (control) used the same problem description as the groups in Study 1. The second and third groups (experimental) followed the next descriptions: Using the above devices, develop a novel electronic embedded system that is useful at your home. Use as many devices as possible. Propose a title that summarizes best the features that you think make the solution unique and valuable. (Group 3 used the following sentence instead: Then, provide a description of the system that highlights the novelty, quality and feasibility of your solution.) Then, refine the solution to improve the novelty, usefulness, and correctness of your solution. Then, propose a second title that best summarizes the features that make your system unique and valuable. Compare the initial and the second title and indicate what makes your system unique. (Group 3 used the following two sentences instead of the last two sentences: Develop a brief description of the change. Summarize how your design changed because of the short description.)

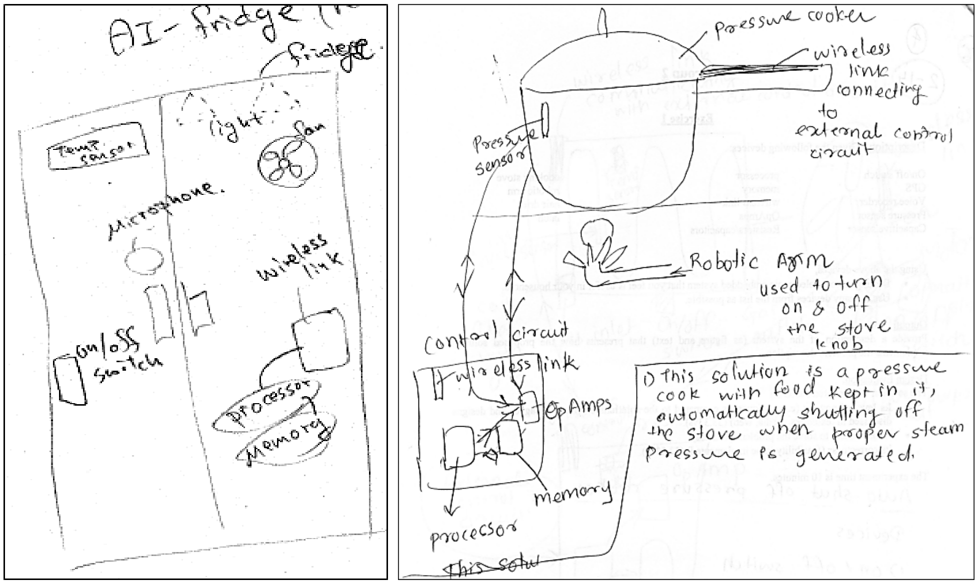


Figure 5.1: Two design samples developed in Study 1

5.3 Experimental Results

The experimental results for the two studies are presented next. The subjects were graduate students in the Department of Electrical and Computer Engineering (ECE). Most students were in their first year of graduate study majoring in either Electrical Engineering or Computer Engineering with a moderate level of expertise in embedded system design. The study was conducted on a voluntary basis. Participants did not receive any reward for participating at the study but were told about the importance of the research. Subjects were provided with a printed paper that provided the problem description and a few sheets of blank paper to write their responses in pencil. The proctoring was done by three proctors.

5.3.1 Study 1

Experiments were conducted separately by four groups of students: 25 students were in Group 1, 17 students in Group 2, 20 students in Group 3, and 14 students in Group 4, for a total number of 76 students. 9 responses were discarded as they did not include any relevant information (the corresponding responses are marked as N/A in Appendix A). Participants worked individually to produce design solutions. Each solution was assessed by two independent raters (senior graduate students with extensive design background in embedded systems). The raters were initially trained by the faculty on the assessment of solution novelty, quality, and usefulness. Inconsistencies between their ratings were solved by the two raters with the help of the faculty.

Figure 1 presents two design samples. The left figure shows a design called "AI-fridge" (Design 13 in Group 1). It automatically regulates the temperature inside a fridge, and accepts voice commands. It informs the user about items that need to be purchased based on history of previous usages, detects the user's location, and provides the directions to nearest store where that item can be bought. It has the highest novelty rating in Group 1. The right figure illustrates a design called "Auto Shit-off Pressure Cooker" (Design 14 in Group 2). It detects steam pressure and automatically shuts off the stove using a robot arm. Its novelty rating is highest in Group 2.

Nature of concept combinations: Table 2 presents the number of relation-based combinations (RBCs) and property-based combinations (PBCs) for the four groups. Column two indicates the number of designs in each group. The number of PBCs is very small compared to number of RBCs. Hence, Hypothesis 1 is correct.

Next, we observed the effect of exemplar similarity on the number of RBCs. The average number of RBCs is highest (3.33 per design) for Group 1 (common, well distinguished exemplars) and lowest (2.11 per design) for Group 3 (often used together exemplars). The average is high (3.00 per design) for Group 4 (more exemplars) as well. The average is low (2.47 per design) for Group 2 (rarely used together). Hypothesis 2 is correct as the variation of the ratio of RBCs is small for the four groups of exemplars.

The identified PBCs are either infeasible or inefficient. For example, the property of a tachometer is given to an accelerometer. This is likely to be feasible to be realized but is less efficient than using traditional tachometer sensors. A similar conclusion was noted for other situations, like giving to a microphone the property of a speaker, and to the GPS that of a communication transceiver or cellphone/pager. Other property transfers were deemed to be infeasible, such as a motor that acts as a pressure sensor to control the opening and closing of a door. However, as suggested by Estes and Ward [144], infeasible combinations can be the starting points in developing novel solutions.

Novelty assessment: As a first step in novelty assessment, we devised a new equation that robustly characterizes embedded system design solution novelty. We used initially the novelty index and variety metric proposed by Shah, Vargas-Hernandez and Smith [112], however this characterization sometimes failed to correctly capture the novelty, especially if designs utilized traditional design ideas but tackled unique purposes (goals) or operated in special environments and conditions. Human raters considered such solutions to be very creative but the two metrics gave low ratings. To address this issue, we devised a new equation that considers the nine main factors specific to embedded system designs. Each factor was weighted such that the error between the novelty ratings of human raters and the novelty estimations predicted by the equation is minimized. The novelty estimation of each design was verified manually by two raters to eliminate inconsistencies. The following formula was used in assessing solution novelty:

$$\begin{aligned} Nov_{overall} = \{&[w^2 \times (Nov_{factor9} + Nov_{factor8})] + [w^1 \times (Nov_{factor7} + Nov_{factor5})] \\ &+ [w^0 \times (Nov_{factor6} + Nov_{factor4} + Nov_{factor3} + Nov_{factor2} + Nov_{factor1})]\} \times R \end{aligned} \tag{5.1}$$

Where factor 1 refers to the number of exemplars used in a solution, factor 2 represents the types of relations connecting the exemplars, factor 3 relates to the topology of the combinations, factor 4 indicates the capabilities added by the combinations, factor 5 describes the properties added to the system, factor 6 presents the system topology, factor 7 indicates the type of the system, factor 8 defines the purpose of the design, and factor 9 presents the place of use. Term R is a normalization constant such that the novelty range is [0, 10]. Weights w reflect the importance of the nine factors.

Weight w was computed by minimizing the error between the novelty

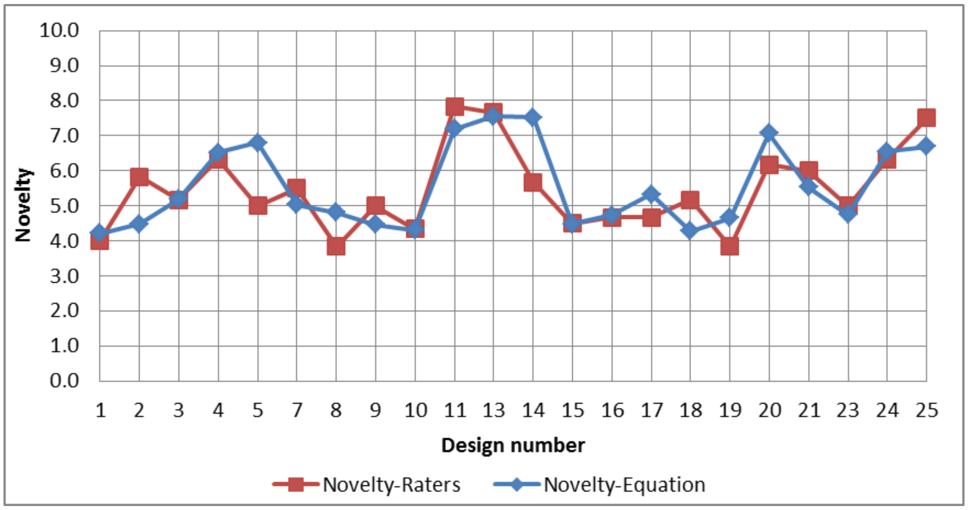


Figure 5.2: Novelty assessment based on raters vs. novelty assessment based on analytical formula

ratings of the designs by human raters and the novelty estimations predicted by the above formula. Six raters graded the designs from Group 1. All raters were graduate students in Electrical Engineering or Computer Engineering. They were initially trained on how to assess novelty, quality, and usefulness. In order to ensure impartial grading, the raters were not allowed to communicate with each other or with the subjects. Using the equation for novelty, it was experimentally observed that the value w = 4 provides the best fit such that the difference between the raters' grades and the values given by the equation is minimized. The plot for novelty based on raters vs. novelty based on analytical formula is shown in Figure 2. For each design in studies 1 and 2, the novelty values computed based on the analytical formula were also verified for consistency by two more raters (different from the six raters).

Appendix A presents the novelty ratings for all four groups calculated using the analytical method. Table 3 summarizes the novelty assessment for each of the four groups, including the average novelty of the designs in the group, the two designs with highest novelty, and a description of their features.

The following terms were used through the remaining of the presentation: H0 - Null Hypothesis, FTR - Fail to Reject H0, ANH - Accept Null

Table 5.3: Novelty assessment of the designs in Study 1

Group 1		
Average novelty: 5.5	Designs 13 and 14 have highest novelty ratings (both are 7.5)	Design 13 (intelligent fridge): performs usual fridge functions; keeps track of food items; alerts the user by sending messages via wireless link; detects current location of user and provides information about the nearest store to buy missing products. Design 14 (kitchen-aid robot): assists cook in performing different tasks in the kitchen.
Group 2		
Average novelty: 6.1	Designs 14 and 17 have highest novelty ratings (7.5 and 7.4, respectively)	Design 14 (auto shut-off pressure cooker): monitors the temperature and pressure inside the cooker; uses the robotic arm to turn the stove on and off. Design 17 (iFetch robot): gets commands wirelessly; locates the object that the user wants; holds object using robotic arm and then travels to the location of the user.
Group 3		
Average novelty: 4.8	Designs 20 and 17 have highest novelty ratings (8.2 and 6.7, respectively).	Design 20 (smart stove and oven): regular operation; gives warning signals; downloads and displays recipes from the Internet; orders food online. Design 17 (automatic grocery shopping system): user uses the camera to take picture of an item; the device sends this picture along with address of the user (identified using GPS) to the nearest grocery store; requests a delivery of the item.
Group 4		
Average novelty: 4.9	Designs 6 and 10 have high novelty ratings (6.8 and 6.7, respectively).	Design 6: alarm system to warn against fire or gas leakage. Design 10: performs certain functions according the specific movements of the user.

Table 5.4: Novelty ratings for the four groups

ANOVA Results			
	F	p-value	Fcrit
Group 1 vs. Group 2 vs. Group 3 vs. Group 4 H0	4.504	0.006	2.751 RNH
Group 1 vs. Group 2 H0	2.345	0.134	4.105 ANH
Group 1 vs. Group 3 H0	3.847	0.058	4.113 ANH
Group 1 vs. Group 4 H0	2.224	0.146	4.149 ANH
Group 2 vs. Group 3 H0	11.399	0.002	4.160 RNH
Group 2 vs. Group 4 H0	9.023	0.006	4.210 RNH
Group 3 vs. Group 4 H0	0.133	0.718	4.225 ANH

Table 5.5: Quality ratings for the four groups

ANOVA Results			
	F	p-value	Fcrit
Group 1 vs. Group 2 vs. Group 3 vs. Group 4 H0	16.004	0.000 FTA	2.742 RNH
Group 1 vs. Group 2 H0	8.787	0.005 FTA	4.091 RNH
Group 1 vs. Group 3 H0	56.186	0.000 FTA	4.085 RNH
Group 1 vs. Group 4 H0	1.042	0.315 FTR	4.130 ANH
Group 2 vs. Group 3 H0	3.483	0.071 FTR	4.139 ANH
Group 2 vs. Group 4 H0	8.271	0.008 FTA	4.210 RNH
Group 3 vs. Group 4 H0	55.936	0.000 FTA	4.196 RNH

Hypothesis, FTA - Fail to Accept H0, RNH - Reject Null Hypothesis.

Table 4 presents the one-way ANOVA tests that compare the novelty ratings of the four groups. The Null Hypothesis was rejected for the comparison between all groups together. The comparison between pairs of groups indicates no significant difference in the means between Group 1 vs. Group 2, and Group 1 vs. Group 4. The Null Hypothesis is accepted even for Group 1 vs. Group 3 but only by a small margin. For Group 2, the Null Hypothesis is accepted only for comparison with Group 1. For the comparison between Group 3 vs. Group 4, the Null Hypothesis is strongly accepted. The analysis suggests that Hypothesis 3 is accepted. Hence, less related exemplars increase the average novelty of the combinations while more related exemplars lower the average novelty. Using common exemplars or having more exemplars does not help novelty.

Quality assessment: Only 6 out of 68 designs were found to be impractical. The average quality ratings are 7.7 for Group 1, 7.1 for Group 2, 6.6 for Group 3, and 7.9 for Group 4. We compared the quality ratings by performing one-way ANOVA tests. Table 5 presents the results. There is a significant difference in the means for all groups together. Therefore, we inspected two groups at a time. The Null Hypothesis was rejected for most cases

Table 5.6: Correlation between quality rating and implementation characteristics

	Closest to quality rating				Farthest to quality rating
Group 1	Precision	Size	EoI	Power	Cost
Group 2	Precision	Size	EoI	Power	Cost
Group 3	Size	EoI	Precision	Power	Cost
Group 4	Precision	Size	EoI	Power	Cost

except in the case Group1 vs. Group 4, and Group 2 vs. Group3. Therefore, there is no significant difference in means for these pairs of groups. Thus, the quality results are statistically equivalent for the case when the exemplars are commonly used (Group 1) and when more exemplars are given (Group 4). The average quality is high for these cases. Also, the results are equivalent for exemplars which are rarely used together (Group 2) and often used together (Group 3). The average quality is low for these two cases and lowest for Group 3. Hence, Hypothesis 4 is invalid as utilizing related exemplars does not improve solution quality as compared to using less related exemplars.

The quality metric was calculated using five factors that characterize the design solution: size, cost, processing precision, power consumption, and ease of interface (EoI). One-way ANOVAs and post-hoc analysis using standard error technique were performed to identify which factor affects the overall quality rating the most. Table 6 presents the results. The factors which affect the overall quality the most are precision, size followed by EoI. Cost is the factor which is farthest from the overall quality.

Usefulness assessment: The usefulness of a design was evaluated based on five factors: (1) Does the design satisfy a need? (2) Does it follow the design constraint? (3) Is there something similar available? (4) Is the proposed design better than the similar options? (5) Can it be used to build other things?

The average usefulness is 5.7 for Group 1, 7.5 for Group 2, 7.1 for Group 3 and 6.2 for Group 4. The usefulness ratings were compared by performing one-way ANOVAs. The results are shown in Table 7. For the ANOVA with all groups together, the Null Hypothesis was rejected. ANOVAs for pairs of groups

Table 5.7: Usefulness ratings for the four groups

ANOVA Results			
	F	p-value	Fcrit
Group 1 vs. Group 2 vs. Group 3 vs. Group 4 H0	8.138	0.000 FTA	2.742 RNH
Group 1 vs. Group 2 H0	17.468	0.000 FTA	4.091 RNH
Group 1 vs. Group 3 H0	10.792	0.002 FTA	4.085 RNH
Group 1 vs. Group 4 H0	1.698	0.201 FTR	4.130 ANH
Group 2 vs. Group 3 H0	1.061	0.310 FTR	4.139 ANH
Group 2 vs. Group 4 H0	8.304	0.008 FTA	4.210 RNH
Group 3 vs. Group 4 H0	3.949	0.057 FTR	4.196 ANH

Table 5.8: Correlation between usefulness rating and implementation characteristics

	Closest to usefulness ratings				Farthest to usefulness ratings
Group 1	Q2	Q1	Q3	Q5	Q4
Group 2	Q2	Q1	Q5	Q3	Q4
Group 3	Q2	Q1	Q3	Q4	Q5
Group 4	Q1	Q2	Q5	Q3	Q4

show that there is no significant difference in the means for Group 2 and Group 3. The average rating is very high for these two groups. The Null Hypothesis was accepted for Group 1 and Group 4 as these groups have comparatively lower ratings, especially Group 1 (common but well distinguished exemplars). The difference of means for Group 3 and Group 4 is such that the p-value is very close to 0.05, so that this pair is just close enough for the H0 to be accepted. Hence, Hypothesis 5 is invalid as the average usefulness of solutions created using less related exemplars is not lower than that if more related exemplars are used.

One-way ANOVA and post-hoc analysis using standard error technique was used to identify which of the five questions affects the overall usefulness the most. The results are as shown in the Table 8. Question 2 and Question 1

Table 5.9: Novelty assessment of the designs in Study 2

Group 1		
Average novelty: 5.3	Designs 5 and 6 have highest novelty ratings (6.9 and 6.8, respectively)	Design 5: uses robotic arms to stabilize the house in the event of an earthquake; it also uses gas sensor to detect gas leakage. Design 6: intelligent vending machine which provides helpful suggestions and performs actions to help the user.
Group 2		
Average novelty: 5.2 after Iteration 1 and 6.0 after Iteration 2	Designs 8 and 10 have highest novelty ratings (7.2 and 6.8, respectively)	Design 8 (Driverless car): navigates using GPS, robotic arm and other sensors. In the second iteration, the design is converted into a 'Thinking car', where the car remembers its previous decisions and therefore has the ability of self-learning. Design 10: its novelty increased by a huge margin from iteration 1 to 2. In first iteration, the design is for a security system to warn against gas leakage. Novelty rating is 3.9 (low). In second iteration, the design is modified to an earthquake monitoring system which is a novel concept. New novelty rating is 6.8.
Group 3		
Average novelty: 6.5 after Iteration 1 and 6.3 after Iteration 2	Designs 8 and 6 have highest novelty ratings (6.8 and 6.7, respectively)	Design 8 (bathroom mirror defogger): its novelty increased from iteration 1 to 2. In first iteration, the design automatically clears the mirror by detecting the humidity and temperature in the bathroom. In the second iteration, there is an added functionality to record reminders which increased the factor 8 rating. Design 6: had highest rating in first iteration and the rating decreased by a significant amount in the second iteration. In first iteration, it implements a security system in the kitchen to warn against fire and gas leakage. In second iteration, anti-burglary functionality is added, but its purpose changes from only kitchen to any room, resulting in decrease in the overall novelty.

are the factors which affect usefulness the most. Question 4 has the least effect. This implies that the subjects focused more on making sure that the design satisfied a need and followed constraints. However, they ignored whether the design is better than similar options.

5.3.2 Study 2

Experiments were conducted separately by three groups of students: each group had 12 students, for a total number of 36 students. 7 responses were discarded as they did not include any relevant information (the corresponding responses are marked as N/A in Appendix B). Participants worked individually to produce design solutions. The rating of the designs was similar to that in

Table 5.10: Novelty ratings for the four groups

ANOVA Results			
	F	p-value	Fcrit
Group 2 Step1 vs. Step2 H0	3.335	0.093 FTR	4.747 ANH
Group 3 Step1 vs. Step2 H0	0.630	0.436 FTR	4.351 ANH
Group 1 vs. Group 2 Step1 H0	0.008	0.930 FTR	4.600 ANH
Group 1 vs. Group 3 Step1 H0	11.799	0.003 FTA	4.414 RNH
Group 2 Step1 vs. Group 3 Step1 H0	13.179	0.002 FTA	4.494 RNH
Group 2 Step2 vs. Group 3 Step2 H0	1.260	0.278 FTR	4.494 ANH

Study 1.

Novelty assessment: Appendix B presents the novelty of the designs created by three groups. Table 9 summarizes the novelty assessment for each of the three groups in Study 2, including the average novelty of the designs in the group, the two designs with highest novelty, and a description of their features.

Table 10 summarizes the ANOVA tests for the novelty ratings of the three groups and two design steps. For Group 2, the Null Hypothesis was accepted by a small margin for comparison between steps 1 and 2 as there is only a 0.8 difference between the average novelty for Step 1 and Step 2. In Group 3, the novelty for most of the designs remained the same from Step 1 to Step 2 and therefore, the Null Hypothesis was accepted. For Group 1, the Null Hypothesis was accepted for comparison with Step 1 of Group 2 but is rejected for comparison with Step 1 of Group 3. For comparison between Step 1 of groups 2 and 3, the Null Hypothesis was rejected but was accepted for comparison of Step 2. This suggests that the mean for the groups came closer in the second step. Thus, Hypothesis 6 is true as titles help improving the novelty of the subsequent design. Hypothesis 7 is false as briefs do not increase the novelty between the two steps.

Quality assessment: Appendix C presents the quality ratings for the

Table 5.11: Quality ratings for the three groups

ANOVA Results			
	F	p-value	Fcrit
Group 2 Step 1 vs. Step 2 H0	0.390	0.540 FTR	4.414 ANH
Group 3 Step 1 vs. Step 2 H0	0.558	0.464 FTR	4.351 ANH
Group 1 Step 1 vs. Group 2 Step 1 H0	0.450	0.511 FTR	4.381 ANH
Group 1 Step 1 vs. Group 3 Step 1 H0	1.015	0.326 FTR	4.351 ANH
Group 2 Step 1 vs. Group 3 Step 1 H0	0.410	0.530 FTR	4.381 ANH
Group 2 Step 2 vs. Group 3 Step 2 H0	0.031	0.862 FTR	4.381 ANH

three groups. The average quality ratings for both steps of Group 2 and Group 3 are very close to each other. As shown in Table 11, the Null Hypothesis is accepted for comparison between steps 1 and 2 for both groups. Also, there are no statistically significant difference in means between Step 1 of the groups 1, 2, and 3. While analyzing the designs, it was noticed that there were very few changes made when the design was modified from Step 1 to Step 2. Since most of the devices used in Step 1 were retained in Step 2, there is no major difference in the quality ratings.

Usefulness ratings: Table 12 shows that there is a small change in the usefulness ratings between steps 1 and 2 of both Group 2 and Group 3. However, the one-way ANOVA results suggest that this change is not statistically significant. Also, the Null Hypothesis was accepted for comparison between Step 1 of groups 1, 2, and 3. Hence, the overall usefulness remained the same for all the groups. However, it can be noticed that a few designs especially in Group 1 have very high ratings (e.g., designs 1 and 6) while some have very low ratings (i.e. Design 7). Therefore, there seems to be a lot of fluctuations in the ratings for different designs but the average rating matches that of the other groups.

Table 5.12: Usefulness ratings for the three groups

ANOVA Results			
	F	p-value	Fcrit
Group 2 Step 1 vs. Step 2 H0	1.336	0.267 FTR	4.600 ANH
Group 3 Step 1 vs. Step 2 H0	0.108	0.746 FTR	4.351 ANH
Group 1 Step 1 vs. Group 2 Step 1 H0	0.128	0.725 FTR	4.451 ANH
Group 1 Step 1 vs. Group 3 Step 1 H0	0.091	0.766 FTR	4.351 ANH
Group 2 Step 1 vs. Group 3 Step 1 H0	0.004	0.952 FTR	4.451 ANH
Group 2 Step 2 vs. Group 3 Step 2 H0	0.035	0.855 FTR	4.451 ANH

5.4 Discussion

Figure 3 summarizes the average novelty, quality, and usefulness of the designs in Study 1. The results show that, for modular, electronic embedded system design, it is more convenient to relate exemplars based on their functions using relations like input - output or event - response. It is difficult to combine exemplars such that properties of one exemplar are transferred to the other exemplar. This justifies the fact that the number of RBCs is significantly larger than the number of PBCs. The highest number of RBCs per design is generated if well known exemplars are considered as this generates more ideas on how to relate the exemplars. Surprisingly, having more or less related exemplars reduces the average number of RBCs per design. While less related exemplars might be more difficult to combine in various ways, more related exemplars might encourage fixation to a particular kind of design.

Related studies show that examples have a mixed role in creative design. It is known that exposure to examples can reduce innovation through fixation [96, 118]. The path-of-least-resistance model indicates that features of known examples are accessed first [171, 172]. Fixation to initial solutions can persist during the design process [168]. However, other work shows that examples can improve creativity too [151, 159]. Agogue et al. [130] distinguish examples with fixating role from those with a role in expanding the solution features

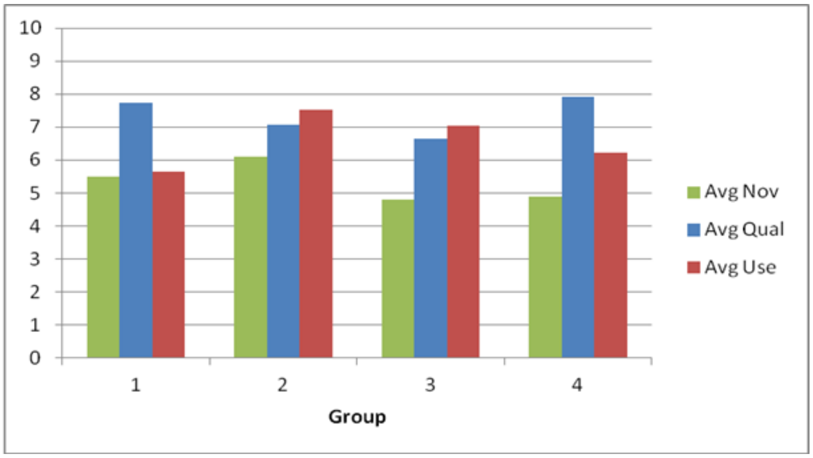


Figure 5.3: Average novelty, quality, and usefulness of the designs in Study 1

beyond the features of the initial design. Far-field, less common examples improve novelty and quality in mechanical engineering design [137]. Early exposure to examples possibly followed by prototyping and re-exposure to examples also helps creativity [156].

We observed that fixation induced by some of the exemplars was an important factor in deciding the nature of solutions, including their purpose and place of being utilized. To a certain degree, exemplars have a constraining effect similar to examples [118]. For example, having gas and temperature sensors among the exemplars given to Group 1 resulted in more applications on home safety related to toxic gases and dangerous temperatures, including fire. Group 2 had voice recorder, robotic arm, and cooking stove among its exemplars. These exemplars encouraged more solutions like voice controlled embedded systems and smart cooking devices placed in the kitchen and with robotic arm. The camera and the alarm signal in the set provided to Group 3 favored more home security applications against intruders. Group 4 had designs similar to Groups 1 and 3. Some exemplars, like capacitive sensors, were rarely used even though they are very popular in modern embedded systems, like smart phones and music players. This observation suggests that the exemplars (through their fixating constraints) have different impact on deciding the application and design solution.

Design solutions can be grouped into three categories: (1) solutions highly similar to already existing designs, e.g., home security systems, (2) solutions that apply an available solution for a new problem or in a new place, i.e. using a robotic arm for pumping gas in a gas station, and (3) solutions that include new relations between the given exemplars, like using a hair drier to disperse smoke in a kitchen. The novelty of the first category is low. The novelty of the second category is high, if the identified purpose or place are less common. This category resembles to using analogies in solving engineering problems [125]. There was one instance in which the solution represented a metaphor-based design. The application entitled "Voice follows you" proposed an automated voice amplification system that adjusts automatically the strength of the received voice signal based on the position of the participants. This solution has the potential of enabling many more applications that involve high-quality communication between people in distributed spaces. The third category included solutions with unique relations but the novelty of the overall solution or its usefulness were low.

Figure 4 plots and displays (in the bottom tables) the novelty, quality, and usefulness ratings of each design in Study 1. The gaps in the plots represent subjects that did not respond to the problem. These cases are marked as N/A in Appendix A. Novelty was assessed using the formula in Section 2.2 computed based on the features of the designs. Note that novelty was an explicit design requirement in our studies: In the problem description, participants were instructed to create novel designs which are dissimilar to designs seen in textbooks, technical magazines and journals, Internet, etc. The designs with novelty ratings above average were highlighted with a gray box if their usefulness was above average and with a black box if their usefulness was below average.

In about 60% of the cases, the novelty and usefulness of solutions are inversely correlated, meaning that higher novelty rate comes at the expense of

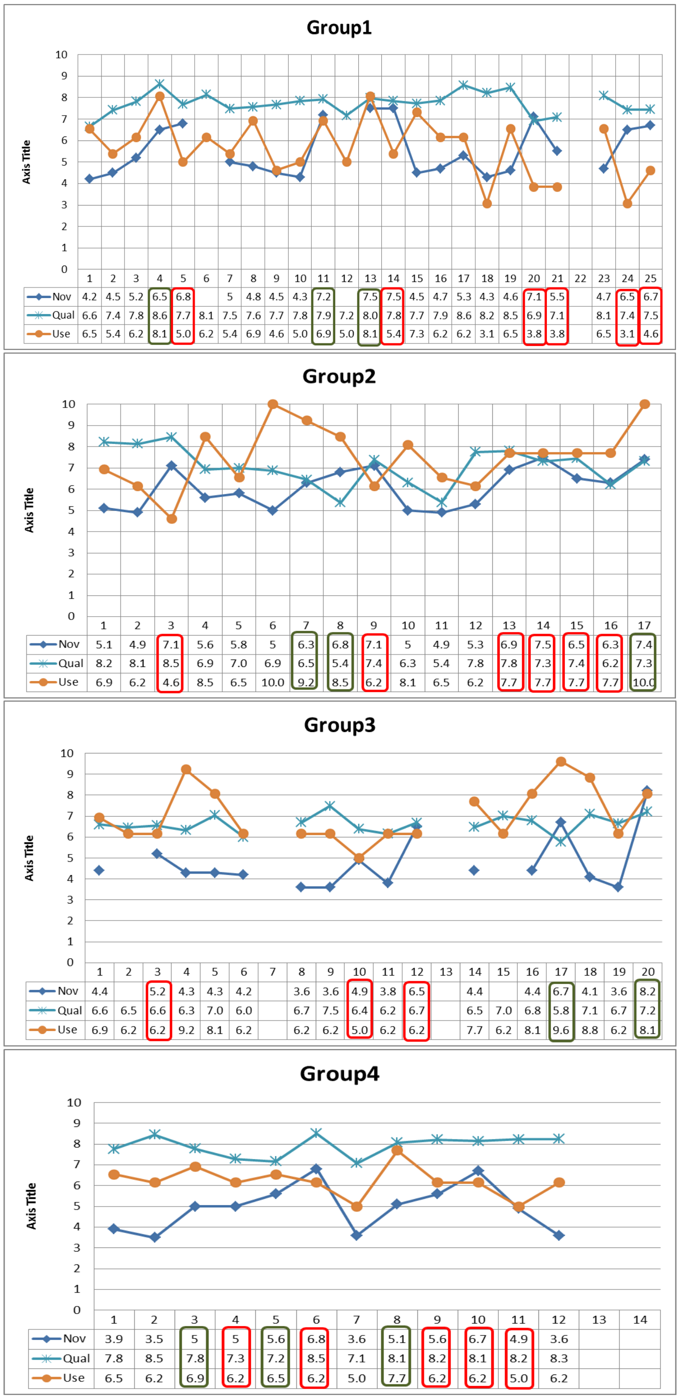


Figure 5.4: Novelty, quality, and usefulness of individual designs in Study 1

lower usefulness. Some new solutions had often low usefulness as the invented relations were considered unrealistic or marginal as more traditional designs already solved well the same problem, e.g., using the noise of car engine to gain access to parking lots. Three designs, "AI-fridge" (Design 13 in Group 1), "iFetch robot" (Design 17 in Group 2), and "smart stove and oven" (Design 20 in Group 3) have high novelty and usefulness. Such designs are truly creative but difficult to produce. More traditional solutions scored high for usefulness. Solution quality and novelty are less related as shown in the figure.

Producing design solutions based mainly on RBCs (and virtually no PBCs) can be justified by the fact that the given exemplars have orthogonal functionalities with few overlapping properties. This is similar with experiments that suggest that overlapping features favor property-based combinations [174]. Another explanation is that traditional design methodologies focus only on relating building blocks to achieve a certain purpose and not on how properties can be transferred among exemplars. Developing novel design methodologies that target PBCs can create opportunities to improve novelty, but this requires reasoning with design attributes (e.g., design performance) in addition to the more traditional way of reasoning based on functional features. A possibility would be to incorporate visual descriptions to reflect properties and self-explanations [131, 132, 142]. Novel design methods must address less considered issues regarding quality and usefulness, like relating the developed design to alternative solutions and the capability of a design to be used as an enabler for future solutions. Also, designers must be encouraged to tackle in more detail aspects like solution cost and easiness of interfacing (EoI).

The experimental results indicate that relation-based combinations in electronic embedded system designs can be distinguished based on their purpose into four types:

1. Core relations exist if the functionality of the resulting block is explicitly derivable from the functionality of the composed blocks. For example, the functionality of a weighted adder circuit includes two separate basic blocks: input multiplication by constant weights and summation. The features of the overall block can be derived through linear composition

from the features of the basic blocks.

2. Enabling relations are produced if the functionality of one block enables the functioning of another block. Enabling relations implement input-output connections between blocks, such as if an amplification block strengthens the inputs signals of another block. The features of the second block remain unchanged but the amplification block enhances the range of the inputs to the block.

3. Performance-enhancing relations are if one block enhances the performance of another block but without conditioning the basic functionality of the design solution. For example, a memory module increases the amount of time over which a computed value is available.

4. Hidden relations are created when the connected structure has different capabilities than those of the composing blocks, e.g., filtering structures. Possible ways of constructing hidden relations is by producing circular connections (feedback loops) and cross-correlations in a circuit design. The new property is a resultant of repeatedly executing the loop. The new features represent either fixed points (invariants) of the cyclic structures, amplification of secondary features (while reducing the importance of main attributes), or patterns defined over variable dynamics, e.g., repetitive behavior like for sinusoidal oscillation. In the first case, there are both enforcing and reducing relations that reach equilibrium (the fixed point). The second situation indicates a reinforcement relationship that acts as positive feedback to amplify the secondary feature, similar to increase in salience in concept combination [176]. The third case represents relations that create a cyclic trace that proceeds, through positive amplification, to reach an extreme point followed then by changing the positive amplification to reach another extreme point, and so on. Characterizing novel features of hidden relations requires simulation (prediction) similar to situated simulation theory [134].

Two of the relation kinds are partially similar to relational categories [158, 160]. Enabling relations are similar to CAUSES, USED BY, USES, BY, and FOR modifiers. Performance-enhancing relations can act as specialization modifiers, like DERIVED FROM, LOCATED, and DURING. All three kinds of relations operate as HAS, MADE OF, and USES modifiers. Note that FOR, IS, and ABOUT modifiers relate to goals, which are not explicitly expressed through the relations of modular design, especially in situations in which the features of the solution are not summations of the features of its composing blocks. The relational categories in [158, 160] do not include anything similar to hidden relations as there are no implicit features. Enabling and performance-enhancing relations are also related to schemata [139]. However, schemata do not include cyclic structures which are important to produce for hidden relations. Enabling relations are a more natural way of relating blocks to each other as the final characteristics of solutions are easier to infer in terms of basic blocks. This is due to several reasons. Many building blocks used in design have complementary, non-overlapping features, and designer training relies on discussing examples that are relevant representatives for a given class of circuits. In addition, economical constraints enforce that the building blocks in a library have little overlapping features as they can lead to redundancies, and hence unwanted increase in cost. Also, circuit design practice relies often on formalisms that stress input-output dependencies between blocks, thus favor enabling relations. In contrast, hidden relations are more difficult to predict as their nature becomes evident only after additional, usually hard-to-acquire insight, e.g., through simulation and analysis. Hidden relations are usually learned and reused in similar situations. Hidden relations correspond to implicitly expressed relations or relations which are open-ended (under-constrained), thus allowing an infinity of solutions. This suggests that hidden relations have the potential of introducing features other than what the original blocks offer, thus extending the domain knowledge beyond additive composition of the individual block meanings.

Figure 5 summarizes the average novelty, quality, and usefulness of the designs produced by the three groups in Study 2. Using titles increases the novelty of the solutions in Group 2, Step 1 compared to Group 2, Step 2. Titles helped refine the designs' purposes and places of use, two of the dominant

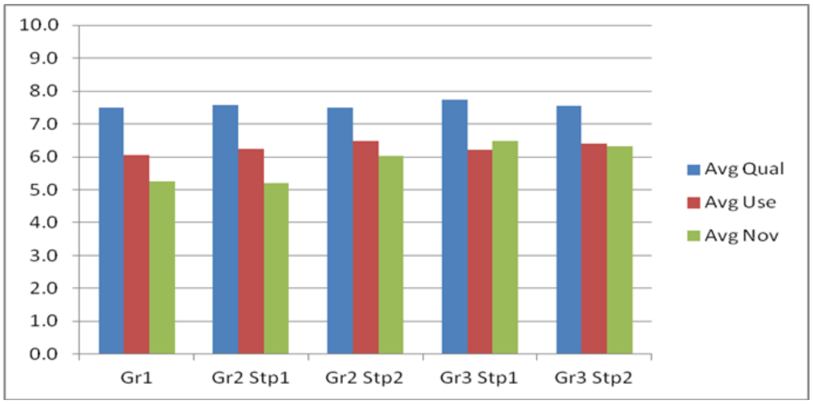


Figure 5.5: Average novelty, average quality, and average usefulness for all groups in Study 2

factors in deciding novelty. Deciding in Step 2 more precise purposes and places of use in contrast to the general, unspecified goals in Step 1 improved novelty. For example, titles focused a general safety system in which the robotic was used for protection (developed in Step 1) to a security system for banks (Step 2). Or a similar safety system (Step 1) is refined as a security system for industrial systems (Step 2). An interesting situation occurred for a design of a driverless car (Step 1) which was modified to a thinking car (Step 2) capable of active decision making and strategy finding. The title defined a novel PBC, the capability of the car to think. The observation that creativity is improved by more specific solutions (rather than broad, less detailed purposes) was also observed in the first study. Titles had little impact on quality and usefulness.

Short descriptions (briefs) did not improve the designs' novelty of Group 3, Step 1 compared to Group 3, Step 2. The result of briefs was to encourage the adding in Step 2 of more functionality to the initial design without changing its main purpose. For example, a system for seagull detection was extended also for searching extraterrestrial zones. Or a system used to pre-heat a car's engine was extended to distinguish between weekends and week days, which creates the premises of achieving a better quality of the product (e.g., customer satisfaction) but does not improve novelty. A typical situation was represented by a house security based on gas detection (Step 1) was extended to also incorporate movement detection (Step 2). The average novelty after Step 2 was slightly lower than Step 1. This is in contrast to the results reported for the novelty of product advertisements [154].This difference can be explained as follows. Briefs seem to help subjects identify incremental extensions of system without modifying the basic concept. As solutions are mainly based on RBCs, new RBCs are added in Step 2 but with minor impact on novelty. Briefs had little impact on quality and usefulness. The findings of the first study are likely to be valid for any domain characterized by modular solutions created by connecting building blocks with orthogonal functionalities. Such domains include computer architecture design, VLSI circuit design, and software design. For example, connecting processing units and wireless radio circuits, two types of modules rarely used previously together, has produced revolutionizing architectures for mobile computing systems. The findings of the second study probably hold for a broader range of domains, not only those dominated by relation-based combinations. Aids, like titles, can be used as tools to describe constraints that distinguish new solutions from their related precedents. A broad range of work explains the utility of constraints in improving creativity [162, 169].

5.5 Conclusions

This chapter presented two experimental studies on the characteristics of concept combinations in modular, electronic embedded system design as well as the relation between concept characteristics and novelty, quality, and usefulness of designs. The work explored the frequency of relation-based and property-based combinations in design solutions, and how the specifics of the used building blocks, i.e. salience, relatedness, and number, influenced the novelty, quality, and usefulness of combinations. The importance of two design aids, titles and short descriptions (briefs), in improving the novelty, quality, and usefulness of final designs was also analyzed. The two experimental studies are based on the work by Mobley, Doares and Mumford (1992).

Experimental results show that new electronic embedded systems are developed mainly through relation-based combinations. This conclusion does not depend on the kind or number of building blocks given to the subjects. The average novelty is higher for building blocks that have rarely been used together before. Also, the novelty of solutions is mainly related to factors that express the purpose (goal) and place of use (context) of the solutions, and less related to the number and specific kind of building blocks and the types of their interconnections. Some exemplars have a stronger role in deciding the purpose of the application by fixating the subjects to certain applications. The solution quality and usefulness are not correlated to the nature and number of building blocks. Subjects did pay little attention to issues like implementation cost, similar solutions being already available, and the possibility to incrementally develop the present design into future solutions. Novelty is aided by using titles that summarize the purpose and distinguishing features of solutions but short descriptions (briefs) do not help.

The two studies suggest that a strategy to improve design novelty is to attempt using existing design ideas (e.g., design patterns) in a new but specific place or for a new but precise purpose (goal). Using continuously titles to summarize and then focus on novel yet specific features improves innovation. Solutions that target general purposes should be avoided as their lack of details lowers the perceived usefulness and novelty. Second, it seems that many subjects had already selected the targeted purpose independently of the experiment, similar to pre-inventive structures [138]. The actual problem that they solved was partially related to a problem which they had identified before and separately from the experiment. Future work will study how exemplars with expansive role [130] can help broadening the space of the considered goals. Also, using less related exemplars helps increasing novelty as shown by work on analogical reasoning [138, 141, 172]. But the connection between unrelated exemplars and selecting new goals and opportunities is unclear. It seems hard to infer new needs in a bottom-up fashion, starting from exemplars and moving towards new purposes. However, combinations of distant concepts originated revolutionary technologies, like Internet and World Wide Web. Finally, we think that the insight gained through experimental studies can inspire the devising of new CAD tools targeted towards innovation in engineering design. We have devised computer algorithms for automatically comparing electronic circuits based on ideas inspired by structural alignment [146], theoretical models for domain knowledge representation in circuit design based on concept categorization [147], and circuit design flows inspired by problem solving heuristics [145]. Our ongoing work will continue this effort towards a complete CAD environment for innovation in electronic circuit design.

5.6 Appendix

APPENDIX A: Table 7.13
APPENDIX B: Table 7.14
APPENDIX C: Table 7.15

Table 5.13: Design novelty for the four groups

Design	Group 1	Group 2	Group 3	Group 4
1	4.2	5.1	4.4	3.9
2	4.5	4.9	N/A	3.5
3	5.2	7.1	5.2	5.0
4	6.5	5.6	4.3	5.0
5	6.8	5.8	4.3	5.6
6	N/A	5.0	4.2	6.8
7	5.0	6.3	N/A	3.6
8	4.8	6.8	3.6	5.1
9	4.5	7.1	3.6	5.6
10	4.3	5.0	4.9	6.7
11	7.2	4.9	3.8	4.9
12	N/A	5.3	6.5	3.6
13	7.5	6.9	N/A	N/A
14	7.5	7.5	4.4	N/A
15	4.5	6.5	N/A	
16	4.7	6.3	4.4	
17	5.3	7.4	6.7	
18	4.3		4.1	
19	4.6		3.6	
20	7.1		8.2	
21	5.5			
22	N/A			
23	4.7			
24	6.5			
25	6.7			
Average	5.5	6.1	4.8	4.9

Table 5.14: Novelty ratings for the three groups

Design	Group 1	Group 2		Group 3	
		Step 1	Step 2	Step 1	Step 2
1	5.5	N/A	N/A	6.4	6.4
2	3.9	N/A	N/A	N/A	N/A
3	4.6	N/A	N/A	6.0	6.0
4	N/A	5.5	5.9	6.4	6.2
5	6.9	5.3	5.8	6.6	6.5
6	6.8	4.8	5.4	7.3	6.7
7	N/A	N/A	N/A	6.5	6.2
8	N/A	6.9	7.2	6.7	6.8
9	4.5	N/A	N/A	6.9	6.5
10	4.4	3.9	6.8	6.5	6.4
11	4.9	5.8	5.8	5.3	5.2
12	5.8	4.4	5.4	6.7	6.4
Average	5.3	5.2	6.0	6.5	6.3

Table 5.15: Quality ratings for the three groups

Design	Group 1	Group 2		Group 3	
		Step1	Step2	Step1	Step2
1	7.2	N/A	N/A	7.5	7.5
2	7.4	7.8	7.5	N/A	N/A
3	8.1	7.4	7.5	7.6	7.6
4	N/A	7.5	7.3	7.7	7.5
5	7.4	7.3	6.7	7.8	7.8
6	7.2	7.4	7.5	6.7	6.8
7	7.9	N/A	N/A	7.2	6.8
8	7.9	7.7	7.7	9.3	8.4
9	6.8	7.7	7.7	7.3	7.0
10	7.7	7.7	7.4	7.6	7.6
11	7.1	7.4	7.7	8.4	8.1
12	7.6	8.0	8.0	7.9	7.7
Average	7.5	7.6	7.5	7.7	7.5

Chapter 6

Conclusions and Future Work

6.1 Conclusions

In this thesis report, we presented a procedure to construct robust data models using samples acquired through a grid network of embedded sensing devices with limited resources, like bandwidth and buffer memory. The procedure constructs local data models by lumping state variables, and then collects centrally thelocal models to produce global models. The modeling procedure uses a linear programming formulation to compute the lumping level at each node, and the parameters of the networked sensing platform, like data communication paths and bandwidths. Two algorithms are described to predict the trajectories of mobile energy sources/sinks as predictions can further reduce data loss and delays during communication. Experiments discuss the method's efficiency for thermal modeling of ULTRASPARC Niagara T1 architecture.

Experiments show that variable lumping reduces the overall error by up to 76.91% and delay by up to 57.62%, as compared to no lumping being used. The error is smallest if latency reduction has high priority. The attempt to minimize local error performs less lumping, however, results is larger data loss, and hence in more overall error. The attempt to reduce the overall error by minimizing the correlation error results in increased latency. As the network size increases from 25 nodes to 64 and 100 nodes, the larger communication traffic leads to further losses and delays. Therefore, accuracy-centered optimization becomes critical for performing reliable data extraction. Trajectory prediction using adaptive method (A2) reduces modeling error by up to about 10%.

We also presented a new approach for optimizing the accuracy of distributed data acquisition for tracking dynamic phenomena through a network of embedded sensors. The method considers three orthogonal facets: minimizing the sampling error of the individual embedded nodes through frontend reconfiguration, and selecting dynamically data paths to avoid data loss and delays due to embedded nodes running out of resources. For the sound-based tracking application, the accuracy of data sensing is improved by about 28.5%, data loss is zero in most situations, and delay reductions are more than 20% in most cases.

We proposed methods to detect and track emergent kinetic data representing clouds of physical entities in [3]. The main attributes of the considered kinetic data structures (KDS) are topography (position, boundary, and area), composition (signature), and concentration of the gas clouds. The report presents fully decentralized methods for optimized implementation of kinetic data on a grid network of wireless embedded sensing nodes. Adaptation policies are devised to switch between different parameters namely buffer memory size, communication bandwidth, radio power, and sampling precision. The efficiency of the algorithms is compared in [3] with similar methods proposed in the literature. The decentralized method improves latency up to 90% and data loss by 75% compared to centralized approaches, by 64% and 61% compared to cluster-based algorithms and latency improvement of about 25% compared to hierarchical techniques.

A methodology to model the dynamics of traffic scenes, including the participating vehicles, vehicle clusters, attributes and relations of all scene elements, and related events, like cluster merging and splitting was presented in Chapter 1. Experiments using this methodology are described in [4]. The main steps of the methodology find the elements of a scene, identify the relations among the elements, and construct analytical prediction models for the traffic scene dynamics. Compared to other methods, this methodology constructs the models online using sound input captured by the nodes equipped with two microphones each.

A study was presented about the role of precedents in illuminating creative ideas during iterative design procedures for open-ended problems in electronic embedded systems. Precedents were defined as the solutions and solution features developed during the iterative solving process by the participants in a group. Through an experiment grounded in cognitive psychology, this work explored the influence of precedents on novelty, variety, quality, and utility of design solutions devised during four consecutive iterations. Another tested hypothesis was that incrementally changing problem requirements improves design novelty. Experiments showed that precedents did not increase design novelty as compared to the group that did not utilize precedents. Precedents in iterative design flows reduced the variety of design solutions as the group setting seemed to filter out ideas that were perceived as less promising. In our experiments, precedents did not improve creativity as compared to the participants that did not use them.

We presented two experimental studies on the characteristics of concept combinations in modular, electronic embedded system design as well as the relation between concept characteristics and novelty, quality, and usefulness of designs. The work explored the frequency of relation-based and property-based combinations in design solutions, and how the specifics of the used building blocks, i.e. salience, relatedness, and number, influenced the novelty, quality, and usefulness of combinations. The importance of two design aids, titles and short descriptions (briefs), in improving the novelty, quality, and usefulness of final designs was also analyzed. Experimental results show that new electronic embedded systems are developed mainly through relation-based combinations. This conclusion does not depend on the kind or number of building blocks given to the subjects. Using continuously titles to summarize and then focus on novel yet specific features improves innovation.

Two application goals were explored for causal knowledge search: Creativity in electronic design and Thermal modeling of microprocessor. For each goal, causal graphs were extracted from distributed data using TETRAD. We analyzed the causal relationships between different variables in the graphs, under varying circumstances.

6.2 Future Work: Goal-oriented Causal Knowledge Search in Distributed Data

Due to the distributed nature of Cyber-physical systems, data models can be extracted at each individual node (node-level), between groups of nodes (local-level) or at the central node (global-level). At each level, the models express relationships between the involved variables or parameters. As mentioned in chapter 1, these relationships can be of two types: is-part relations and cause-effect relations. Is-part relations can be extracted using data mining, through classification and clustering. For example, using SVM classification, we identify a 'car' and classify that it is a part of the class of vehicles. However, cause-effect relations are tricky to isolate. Correlation of values between two variables does not necessarily imply causation. We need specialized algorithms to identify such causal relations. A graph of the causal relationships between variables is called a 'Causal Graph'. By observing the nature of relations between the same set of variables, at different nodes in the sensor network, we can try to gain an insight into the behavior of the physical entity being monitored.

A study of Cyber-physical systems includes observing behavior at each node in the network and their interactions. Similarly, in a group of human beings, we can analyze individual responses as well as their interaction with others in the group. Therefore, we will explore two application goals in this chapter: Creativity in electronic design (explained in chapters 6 and 7) and Thermal modeling of microprocessor (from chapter 1). For each goal, the aim is to extract causal graphs from distributed data and analyze the causal relationships between different variables under varying circumstances.

6.2.1 Causal Search

There has been significant research performed on the topic of causal analysis and search in the last two decades [177, 178]. In this chapter, the TETRAD tool [179] developed by Scheines et al. [178] was used to perform causal knowledge search. Raw numerical data was input to the tool. This data is used by the tool to extract the covariance statistics between the different variables. Based on the chosen algorithm, a causal graph is output at the end of the search operation. This graph can be parameterized as a 'Structural Equation Model' or a 'Bayes Model' and the parameters can be estimated using regressors. The search also provides a 'Model-fit' coefficient that gives an indication of the reliability of the graph compared to the data. So, we can use this number to get an idea of how closely the relationships in the graph are tied to the actual numerical data.

For example, as shown in Figure 6.1, the data block accepts numerical information. Background knowledge is included to reduce the sample space and help the algorithm achieve faster results. The Search block uses algorithms for model-based search and score-based search. If the search does not involve latent variables, the algorithm searches for 'Patterns'. There are special algorithms to search graphs that might contain latent variables, called Partial Ancestral Graphs (PAGs). The search outputs a causal graph that has directed as well as undirected edges. Graph manipulator is used to can be used to choose a desired direction for the undirected edges to create a fully-directed acyclic graph (DAG). This graph is converted to a parametric model such that each link is assigned a coefficient apart from the overall graph 'Model-fit' coefficient. These coefficients are estimated using the estimator block.

An example of causal search result for the thermal modeling case study is shown in Figure 6.2. As described in chapter 1, thermal data was generated for the ULTRASPARC T1 microprocessor. Variable 'dT/dt' represents change in temperature at a particular node in time 'dt', 'DTSum' is the sum of temperature difference of current node with neighbors and 'E/dt' denotes the energy injection due to power dissipation. As we know from equation 2.1, the change in temperature at a particular node is caused by thermal flow from neigh-

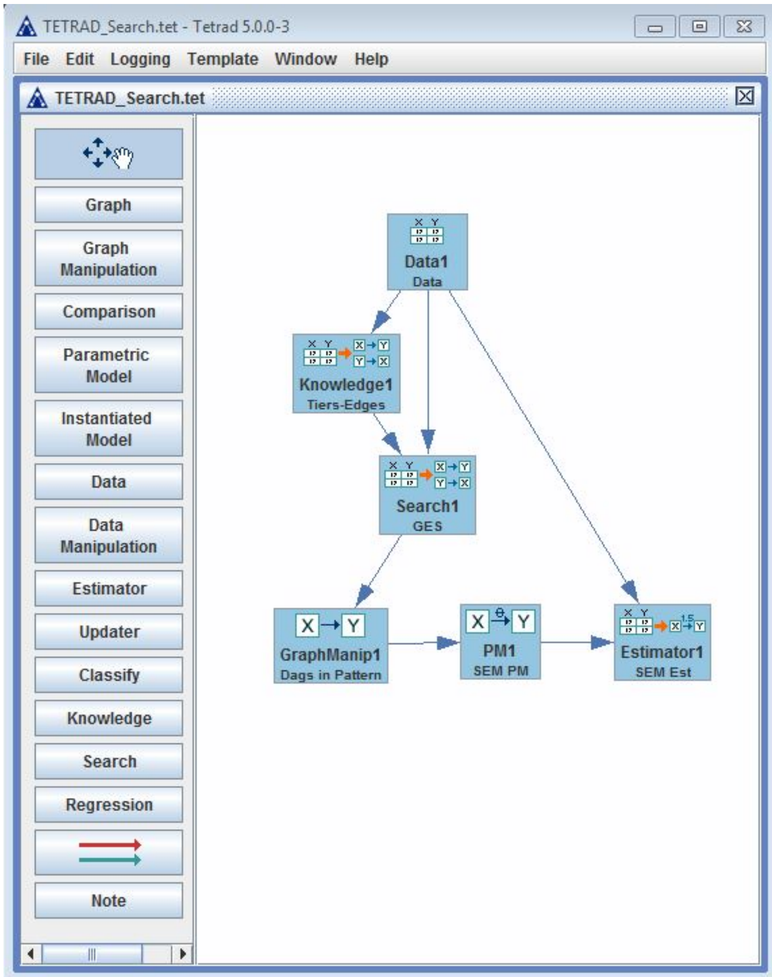


Figure 6.1: TETRAD procedure

bors and power dissipation. So, in the causal graph, we would expect to see directed links representing these cause-effect relationships. This is confirmed by Figure 6.2(top-left) using the GES (Greedy Equivalency Search) scoring-based search algorithm. The estimator output is shown in Figure 6.2(top-right) which shows the values of the edge coefficients. The output of FCI (Fast Causal Inference) algorithm, which searches for PAGs, is shown in Figure 6.2(bottom). We see an additional link between variables 'DTSum' and 'E/dt', which suggest that they might be some hidden dependency between the two. However, the 'bubbles' at the end of the link signify that the direction

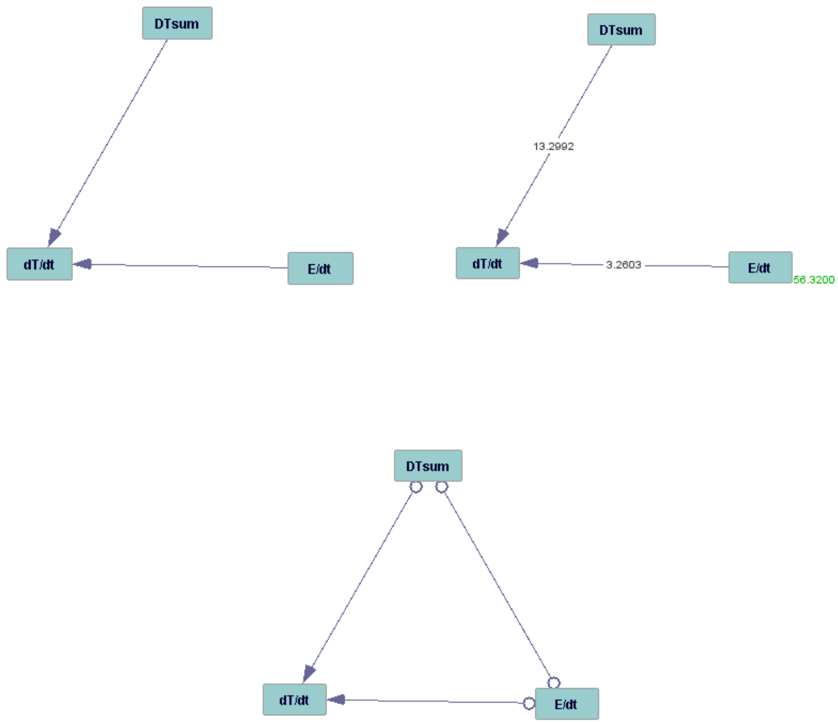


Figure 6.2: TETRAD example: Thermal modeling, (top-left) search using GES, (top-right) estimator output, (bottom) search using FCI

of link cannot be established due to insufficient information.

6.2.2 Preliminary Experiments and Observations

Two application goals will be explored in this section: Creativity in electronic design and Thermal modeling of microprocessor. For each goal, the aim is to extract causal graphs from distributed data and analyze the causal relationships between different variables under varying circumstances.

Creativity in electronic design

Problem statement: In chapter 5, two studies were performed to analyze creative concept combinations in design of modular embedded systems. For the first study (Study 1), subjects were asked to build a novel electronic embedded system to be used in the household. There were four groups of subjects and each group was given a different set of blocks to be used in their designs. The list of building blocks is shown in Table 5.1. Following were the important characteristics of the blocks for each group:

- Group 1: Used exemplars (building blocks) that are common in electronic embedded system problems but their functionality is orthogonal, thus well distinguished from each other.
- Group 2: Unrelated exemplars, such as the exemplars that are rarely used together in solutions.
- Group 3: Exemplars that are often used together.
- Group 4: A larger set of common exemplars.

The novelty ratings were computed using equation 5.1. Nine factors were involved in this analysis:

- Factor 1: number of exemplars used in the design
- Factor 2: types of relations connecting the exemplars
- Factor 3: topology of the combinations
- Factor 4: capabilities added by the combinations
- Factor 5: properties added to the system
- Factor 6: system topology
- Factor 7: type of the system
- Factor 8: purpose of the design

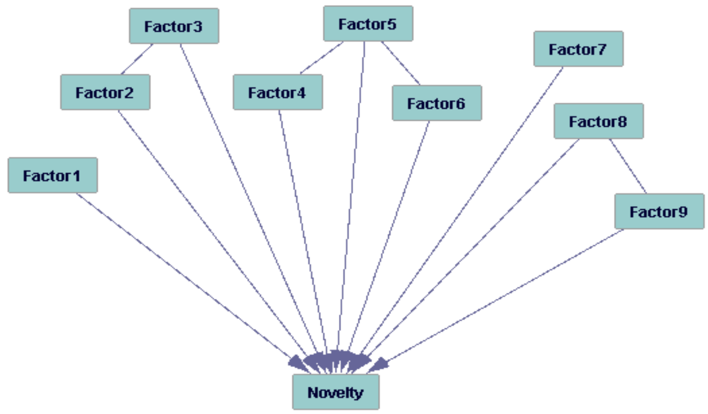
- Factor 9: place of use.

For each group, the individual scores for the nine factors as well as the overall novelty score were input to the TETRAD tool in order to perform causal search. The main aim was to identify any hidden correlations between the 9 factors that directly or indirect affect the overall novelty ratings.

Search Results: The search results for groups 1, 2, 3 and 4 are shown in Figures 6.3 and 6.4 respectively. As an additional analysis, the designs from all groups were combined into a single group and a separate causal search test was performed on this combined data. The result is shown in Figure 6.5.

For Group1, the model fit is 0.94 which indicates that this graph is highly significant compared to the data. All 9 factors affect the overall novelty rating. This is expected, due to the nature of the novelty expression. However, we can observe additional dependencies between different factors. For example, factor 2 (type of relations) and factor 3 (topology of combinations) are connected by an undirected edge. The undirected edge indicates that there is insufficient information to confirm the direction of the edge. It can also indicate that some members of the equivalence class contain edge in one direction while other members have the same edge pointing in the opposite direction. Also, factors 4 (capabilities added by combinations) and 5 (properties added to system), factors 5 and 6 (system topology), factor 8 (purpose)and 9 (place of use) are connected to each other respectively.

As seen in the graphs for Groups 2, 3 and 4, there are some similarities as well as differences in the causal relations between the factors, as compared to Group 1. The main reason for these dependencies might be the nature of the building blocks that were provided to the subjects, as explained earlier. For example, Group2 uses exemplars that are rarely used together. So, the purpose of the designs (factor 8), as chosen by the subjects, strongly affects the topology of the combinations of blocks (factor 3) as well as the system topology (factor 6). This behvior is not observed for any other group. On the contrary, Group used exemplars that are often used together. So, we see that the topology of combinations of blocks (factor 3) is directly affected by three factors, namely factors 2 (types of relations), 6 (system topology) and 7



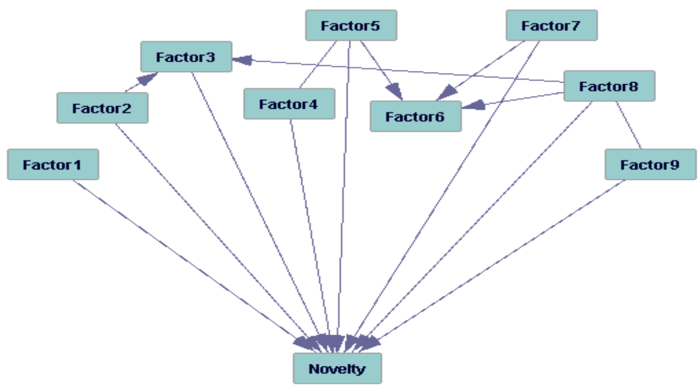


Figure 6.3: Causal graphs for Group1 (top) and Group2 (bottom), with model fit of 0.94 and 0.32 respectively

(type of system). The graph for Group 4 has the least number of links. This might be because of the larger set of building blocks or because group 4 was the smallest group with only 12 subjects.

A small test was performed to observe the effect of having different number of subjects in each group. All the groups were combined to from a big dataset called 'AllGroups' that contained 68 datapoints. This also gives an idea of the most common and prominent dependencies in the combined group of designs, irrespective of the building blocks that were provided. The results are shown in Figure 6.5. Most of the dependencies are similar to those

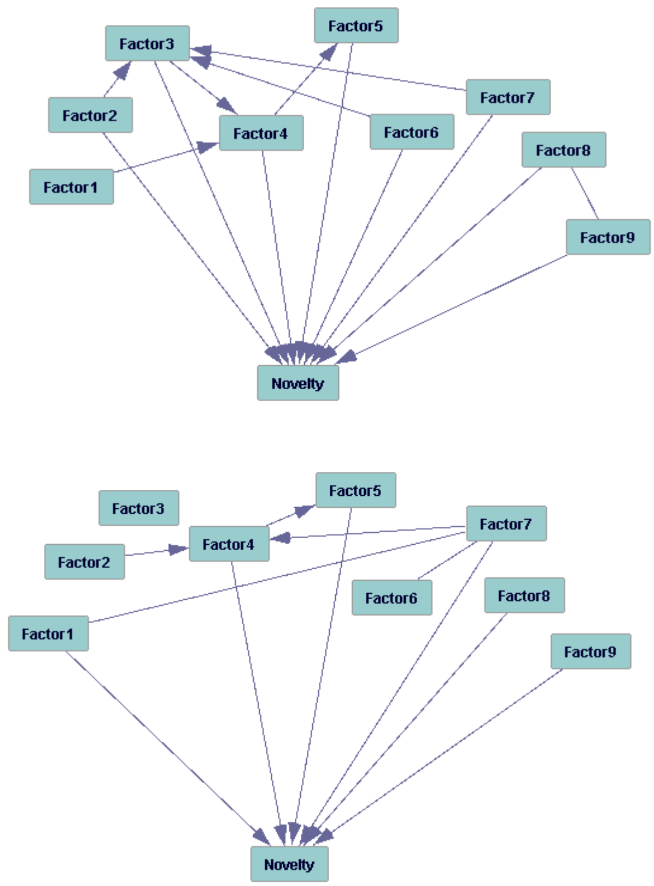


Figure 6.4: Causal graphs for Group3 (top) and Group4 (bottom), with model fit of 0.13 and 0.48 respectively

in Group 1. There are two additional links: one from factor 2 (types of relations) to factor 1 (number of exemplars) and another from factor 9 (place of use) to factor 2. The estimation of coefficients for this graph shows the edge coefficients. The model fit is 0.12 which indicates that this graphical model is statistically significant compared to the data. It is interesting to note that the edge coefficients are very close to the regression coefficients that can be estimated from the novelty expression in equation 5.1.

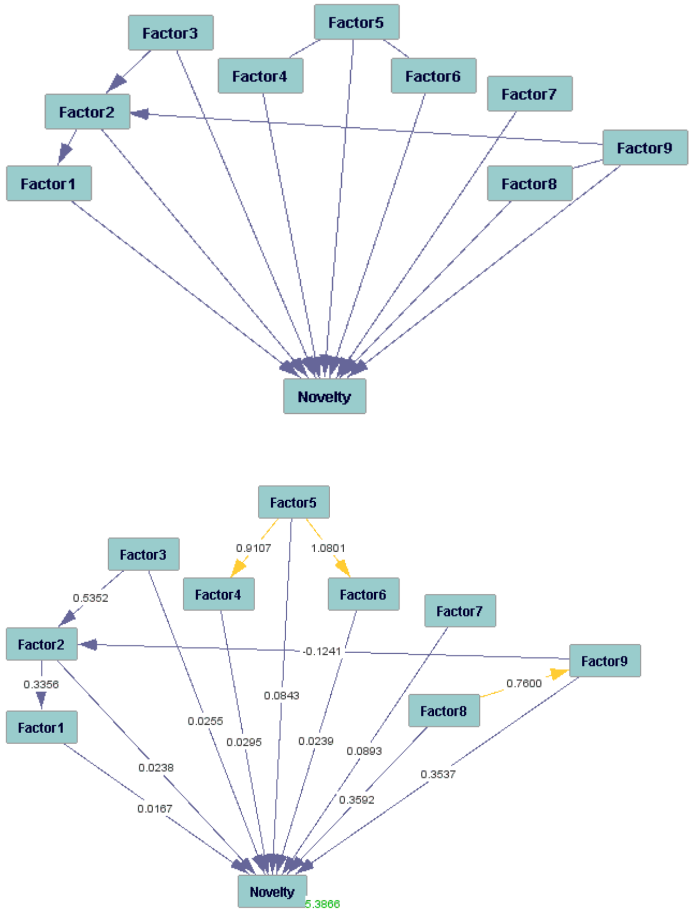


Figure 6.5: (top) Causal graph for all groups combined and (bottom) estimator output, Model fit of 0.12

Thermal modeling of microprocessor

Problem statement: As described in chapter 2, a thermal simulator was used to generate thermal data for an ULTRASPARC T1 architecture floorplan. A scenario that was simulated involved a single hotspot moving across either side of the floorplan. This dataset was labeled 'Dataset 2' (DS2) and is depicted in Figure 6.6. The X and Y axis denote the floorplan dimensions (um) while the Z axis denotes time (ms). The temperature scale (K) is shown on the colorbar.

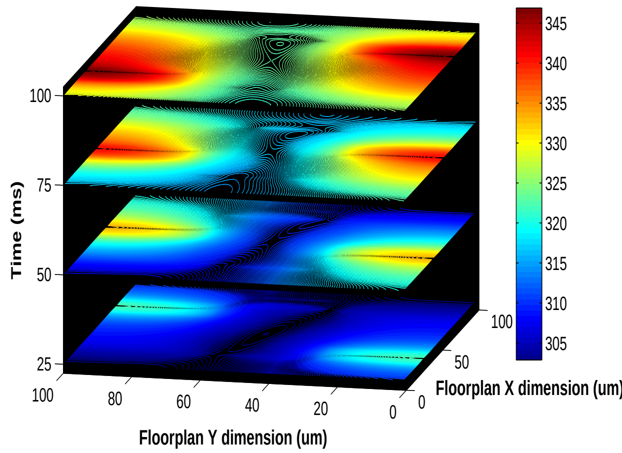


Figure 6.6: Dataset 2 thermal hotspots

For this dataset DS2, the following causal search tests were performed:

- Causal Pattern-search on DS2, using GES algorithm
- Causal Pattern-search on DS2 with microchannel cooling, using GES algorithm
- Causal PAG-search on DS2, using FCI algorithm
- Causal PAG-search on DS2 with microchannel cooling, using FCI algorithm

The aim is to observe the effect of cooling on the extracted causal graph. This should manifest in the form of changes in the caus-effect relationships in the graph as well as the value of model fit coefficient. Also, we can study the effectiveness of Pattern search using GES (scoring-based) algorithm versus PAG search using FCI algorithm.

Similar to the previous example on thermal modeling, variable 'dT/dt' represents change in temperature at a particular node in time 'dt' and 'E' denotes the energy injection due to power dissipation. 'DTup', 'DTdown', 'DTleft' and 'DTright' are the temperature difference of current node with the four neighbors in the grid network of sensor nodes. From equation 2.1, we know that the change in temperature at a particular node is caused by thermal flow from neighbors and power dissipation. So, in the causal graph, we would expect to see directed links representing these cause-effect relationships.

Search Results:

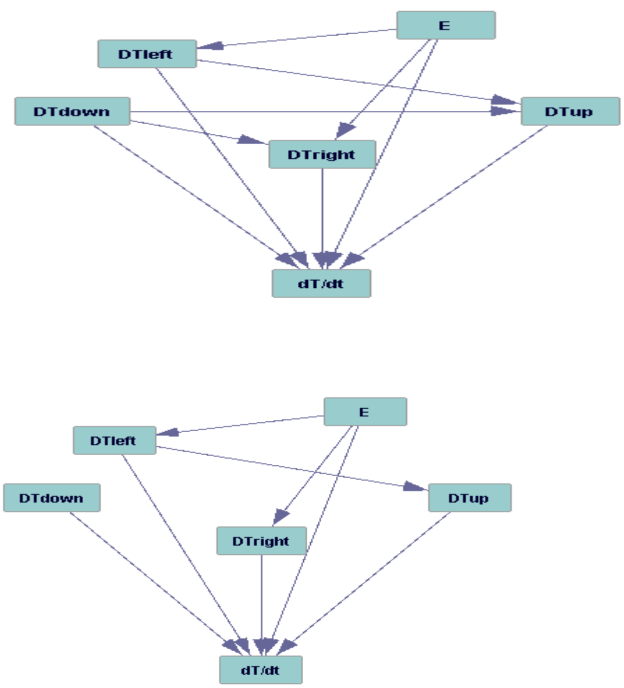
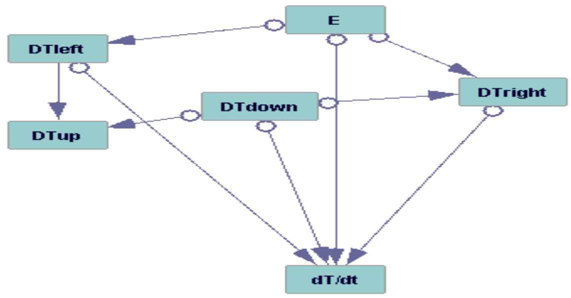


Figure 6.7: DS2 Thermal modeling, using GES with (bottom) and without (top) cooling, Model fit: 0.86,0.77

In Figure 6.7, on the left, we have the graph for DS2 using GES algorithm. The right-side shows the output when microchannel cooling is used. Without cooling, it can be seen that temperature at current node is affected by all neighbors as well as the energy injection. Also, 'E' affects the temperatures at the left and right neighbors, while the 'down' neighbor affects 'up' and 'right'. When cooling is involved, these links from the 'down' neighbor disappear. Also, the model fit drops from 0.86 to 0.77. These results might



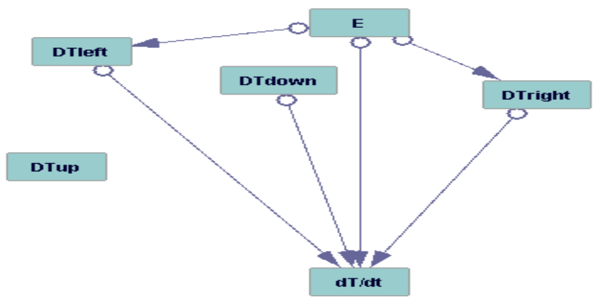


Figure 6.8: DS2 Thermal modeling, using FCI with (bottom) and without (top) cooling, Model fit: 0.54,0.43

get even worse for a different dataset and higher coolant flow-rate. Similar results can be observed in Figure 6.8 for the FCI algorithm. Without cooling, the links are similar to the GES results. When cooling was introduced, all links to 'DTup' disappeared as well as the links from the neighbor 'down' to 'up' and 'right'. The model fit reduced from 0.54 to 0.43.

6.2.3 Conclusions

Two application goals were explored in this section: Creativity in electronic design and Thermal modeling of microprocessor. For each goal, causal graphs were extracted from distributed data using TETRAD. We analyzed the causal relationships between different variables in the graphs, under varying circumstances.

Bibliography

[1] A. Umbarkar, V. Subramanian, A. Doboli, "Linear Programming-based Optimization for Robust Data Modeling in a Distributed Sensing Platform", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, (accepted for publication), 2014.

[2] V. Subramanian, "Building Distributed Data Models in a Performance-Optimized, Goal-Oriented Optimization Framework for Cyber-Physical Systems", PhD Thesis, Stony Brook Univeristy, 2012.

[3] V. Subramanian, A. Umbarkar, A. Doboli, "Decentralized detection and tracking of emergent kinetic data for wireless grids of embedded sensors", *NASA/ESA Conference on Adaptive Hardware and Systems*, 2012.

[4] A. Umbarkar, S. Kodasara, A. Doboli, "Online Construction of Analytical Prediction Models for Physical Environments: Application to Traffic Scene Modeling", *AVICPS Workshop, IEEE Real-Time Systems Symposium*, 2012.

[5] S. K. Rajagopal, "Traffic Scene Understanding using Sound-based Localization, SVM Classification and Clustering", M.S. Thesis, Stony Brook Univeristy, 2010.

[6] A. Doboli, A. Umbarkar,"The Role of Precedents in Increasing Creativity during Iterative Design of Electronic Embedded Systems", *Design Studies, Elsevier*, Volume 35, Issue 3, May 2014.

[7] A. Umbarkar, V. Subramanian, A. Doboli, S. Doboli,"Two Experimental Studies on Creative Concept Combinations in Modular Design of Electronic

Embedded Systems", *Design Studies, Elsevier*, Volume 35, Issue 1, Jan 2014.

[8] A. Umbarkar, V. Subramanian, A. Doboli, "Low-Cost Sound-based Localization using Programmable Mixed-Signal Systems-on-Chip", *Microelectronics Journal*, Vol. 42, No. 2, February 2011, pp. 382-395.

[9] N. Guarino, C. Welty, "Unity and Individuality: Towards a Formal Toolkit for Ontological Analysis", *Proc. European Conference on AI*, 2000.

[10] N. Guarino, C. Welty, "Towards a Methodology for Ontology based Model Engineering", *Proc. ECOOP Workshop on Model Eng.*, 2000.

[11] D. Halupka, J. Mathai, P. Aarabi, Sheikholeslami, *A. Robust sound localization in 0.18 um cmos*, IEEE Trans. Signal Processing, 53(6), 2005.

[12] R. Neapolitan, "Learning Bayesian Networks", Prentice Hall, 2004.

[13] J. Peral, "Causality. Models, Reasoning and Inference", *Cambridge University Press*, 2000.

[14] P. Spirtes, C. Glymour, R. Scheines, "Causation, Prediction, and Search", *The MIT Press*, 2000.

[15] V. Subramanian, M. Gilberti, A. Doboli, D. Pescaru, A Goal-Oriented Programming Framework for Performance Optimized Execution on Grid Sensor Networks with Reconfigurable Embedded Nodes, *ACM Transactions on Embedded Computing Systems*, 2011.

[16] V. Subramanian, A. Umbarkar, A. Doboli, Maximizing the Accuracy of Sound Based Tracking via a Low-Cost Network of Reconfigurable Embedded Nodes, *NASA/ESA Conference on Adaptive Hardware and Systems*, 2011.

[17] L. Guibas, Kinetic Data Structures, *Handbook of Data Structures and Applications*, 2004.

[18] E. Lee, Cyber physical systems: Design challenges, In *University of California, Berkeley Technical Report No. UCB/EECS-2008-8*, 2008.

[19] F. Zhao, J. Liu, L. Guibas and J. Reich, Collaborative signal and information processing: An information-directed approach, *Proceedings of IEEE*, 91(8):1199–1209, August 2003.

[20] Q. Fang, F. Zhao, and L. Guibas, Counting Targets: Building and Managing Aggregates in Wireless Sensor Networks, *Technical Report P2002-10298*, Palo Alto Research Center, June 2002.

[21] Cypress semiconductor corporation, psoc mixed signal array. *Document No. PSoC TRM 1.21*, 2005.

[22] T. Banerjee, K. Chowdhury, and D. Agrawal. Distributed data aggregation in sensor networks by regression based compression. In *Proceedings of MASS*, 2005.

[23] D. Basmadjian. *The Art of Modeling in Science and Engineering.* Chapman & Hall, 1999.

[24] A. Doboli and E. Currie. *Introduction to Mixed-Signal, Embeddd Design.* Springer, 2010.

[25] Coskun et al. *Temperature Aware Task Scheduling in MPSoCs.* DATE, 2007.

[26] Skadron et al. Temperature-aware microarchitecture: Modeling and implementation. *ACM Transactions on Architecture and Code Optimization*, 2004.

[27] Zanini et al. Hierarchical thermal management policy for high-performance 3d systems with liquid cooling. *IEEE Transactions on emerging and selected topics in circuits and systems*, 2011.

[28] C. Ferent, V. Subramanian, and A. Doboli. Linear programming approach for performance-driven data aggregation in networks of embedded sensors. In *Proc. Design, Automation and Test in Europe*, 2010.

[29] A. Fourmigue, G. Beltrame, G. Nicolescu, and E.M. Aboulhamid. *A Linear-Time Approach for the Transient Thermal Simulation of Liquid-Cooled 3D ICs*. CODES+ISSS, 2011.

[30] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden. *Distributed regression: an efficient framework for modeling sensor network data*. IPSN, 2004.

[31] T. He, B. Blum, J. Stankovic, and T. Abdelzaher. Aida: Addaptive application-independent data aggregation in wireless sensor networks. *ACM Transactions on Embedded Computing Systems*, 3(2):426–457, May 2004.

[32] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of MobiCOM*, pages 174–185, 1999.

[33] M. Ketel, Dogan N.S., and A. Ugur. *Distributed wireless sensor networks with power efficient data modeling*. SSST, 2005.

[34] P. Kongetira, K. Aingaran, and K. Olukotun. Niagara: A 32-way multithreaded sparc processor. *IEEE Micro*, 25(2):21–29, 2005.

[35] P. Li, L. Pileggi, M. Asheghi, and R. Chandra. *Efficient Full-Chip Thermal Modeling and Analysis*. Proc. ICCAD, 2004.

[36] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, and T. He. Rap: A real-time communication architecture for large-scale wireless sensor networks. In *Proceedings of IEEE RTAS*, 2002.

[37] H. Mizunuma and et al. *Thermal modeling for 3D-ICs with integrated microchannel cooling*. ICCAD, 2009.

[38] P. Nie and B. Li. *A cluster-based data aggregation architecture in WSN for Structural Health Monitoring*. IWCMC, 2011.

[39] L. Parolini, N. Tolia, B. Sinopoli, and B. Krogh. *A Cyber-Physical Systems Approach to Energy Management in Data Centers*. Proc. ICCPS, 2010.

[40] J. Ranieri, A. Vincenzi, A. Chebira, D. Atienza, and M. Vetterli. *EigenMaps: Algorithms for Optimal Thermal maps Extraction and Sensor Placement on Multicore Processors*. DAC, 2012.

[41] M. Rewienski and J. White. Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations. *Linear Algebra and Its Applications*, 415(2-3):426–454, 2006.

[42] R. Rutenbar, G. Gielen, and J. Roychowdhury. Hierarchical modeling, optimization, and synthesis for system-level analog and rf design. *Proceedings of the IEEE*, 95(3):640–669, March 2007.

[43] M. Schetzen. *The Volterra and Wiener Theories of Nonlinear Systems*. John Wiley & Sons, 1980.

[44] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[45] M. Shoaib and W. Song. *Data aggregation for Vehicular Ad-hoc Network using particle swarm optimization*. APNOMS, 2012.

[46] A. Sridhar and et al. *3D-ICe, http://esl.epfl.ch/3D-ICE*.

[47] A. Sridhar and et al. *3D-ICE: Fast Compact Transient Thermal Modeling for 3D ICs with Inter-tier Liquid Cooling*. ICCAD, 2010.

[48] V. Subramanian, A. Umbarkar, and A. Doboli. Maximizing the accuracy of sound based tracking via a low-cost network of reconfigurable embedded nodes. In *Adaptive and Hardware Systems (AHS)*, 2011.

[49] E. Vladislavleva, G. Smits, and D. Den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *IEEE Transactions on Evolutionary Computation*, 13(4):333–349, 2009.

[50] S. C. Wong and A. Barhorst. Polynomial interpolated taylor series method for parameter identification of nonlinear dynamic system. *Journal of Computational Nonlinear Dynamics*, 3(1):248–256, March 2006.

[51] F. Xue, R. Subbu, and P. Bonissone. Locally weighted fusion of multiple predictive models. In *Proc. of International Joint Conference on Neural Networks*, pages 2137–2143, 2006.

[52] K. Yuen, B. Liang, and B. Li. A distributed framework for correlated data gathering in sensor networks. *IEEE Transactions on Vehicular Technology*, 57(1):578–593, Jan 2008.

[53] F. Zanini, M. Sabry, D. Atienza, and G. De Micheli. Hierarchical thermal management policy for highperformance 3d systems with liquid cooling. *IEEE Transactions on Emerging and Selected Topics in Circuits and Systems*, 1(2):88–101, 2011.

[54] M. Zhu, S. Ding, R. Brooks, Q. Wu, N. Rao, and S. Iyengar. Fusion of threshold rules for target detection in sensor networks. *ACM Transactions on Sensor Networks*, 6(2), 2010.

[55] R. Brooks et al., *Self-organized Distributed Sensor Network Entity Tracking*, Int. Journal High-Perf. Computer Archit., Vol 16. No. 3, 2002.

[56] D. Halupka et al., *Robust Sound Localization in 0.18 um CMOS*, IEEE Trans. Signal Processing, 53(6), 2005.

[57] J. Han, M. Kamber, *Data Mining*, Morgan Kaufman, 2006.

[58] T. Hnat et al., *A Modular and Extensible Macroprogramming Compiler*, Proc. ACM SESENA, 2010.

[59] C. Intanagonwiwat et al., *Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks*, ACM MobiCom, 2000.

[60] S. Kokegar et al., *Constraint-Guided Dynamic Reconfiguration in Sensor Networks*, Proc. ACM IPSN, 2004.

[61] K. Lorincz et al., *Resource Aware Programming in the Pixie OS*, Proc. Sensys, 2008.

[62] H. Lu et al., *Energy-Efficient Spatially-Adaptive Clustering and Routing in Wireless Sensor Networks*, Proc. DATE, 2010.

[63] G. Mainland et al., *Decentralized, Adaptive Resource Allocation for Sensor Networks*, Proc. Symposium on Networked System Design & Implementation, 2005.

[64] A. Munir et al., *An MDP-based Application Oriented Optimal Policy for Wireless Sensor Networks*, Proc. CODES + ISSS, 2009.

[65] Y. Shoham et al., *Multiagent Systems. Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge Press, 2009.

[66] A. Umbarkar and et al., *Improved Sound-based Localization through a Network of Reconfigurable Mixed-Signal Nodes*, Proc. ROSE, 2010.

[67] V. Subramanian, A. Doboli, *Online Adaptation Policy Design for Grid Sensor Networks with Reconfigurable Nodes*, Proc. DATE Conference, 2009.

[68] G. Werner-Allen et al., *Lance: Optimizing High-resolution Signal Collection in Wireless Sensor Networks*, Proc. Sensys, 2008.

[69] Akin, O, "Case-based instruction strategies in architecture", *Design Studies 23*, 407-431, 2002.

[70] Altshuller, G. S., "Creativity as an exact science", *Gordon and Breach*, 1988.

[71] Amabile, T. M., "The social psychology of creativity", *Springer*, 1983.

[72] Atman, C., Chimka, J., Bursic, K., & Nachtmann, H., "A comparison of freshman and senior engineering design process", *Design Studies 20*, 131-152, 1999.

[73] Bink, M. L., and Marsh, R. L, " Cognitive regularities in creative activity", *Review of General Psychology 4*, 59-78, 2000.

[74] Bledow, R., Frese, M., Anderson, N., Erez, M. & Farr, J., "A dialectic perspective on innovation: Conflicting demands, multiple Pathways, and ambidexterity", *Industrial and Organizational Psychology 2*, 305-337, 2009.

[75] Chua, R. Y.-J., & Iyengar, S., "Creativity as a matter of choice: Prior experience and task instruction as boundary conditions for the positive effect of choice in creativity", *Journal of Creative Behavior 42*, 164-180, 2008.

[76] Conradi, P., "Reuse in electronic design: from information modelling to intellectual properties", *John Wiley & Sons*, 1999.

[77] Coskun, H., Paulus, P., Brown, V., & Sherwood, J., "Cognitive stimulation and problem representation in idea generating groups", *Group Dynamics: Theory, Research and Practice 4*, 307-329, 2000.

[78] Coyne, R., "Wicked problems revisited", *Design Studies 26*, 5-17, 2005.

[79] Dahmus, J., Gonzalez-Zugasti, J., & Otto, K., "Modular product architecture", *Design Studies 22*, 409-424, 2001.

[80] Darlington, M. J., & Culley, S. J., "A model of factors influencing the design requirement", *Design Studies 25*, 329-350, 2004.

[81] de Vries, B., & Achten, H. H., "DDDoolz: designing with modular masses", *Design Studies 23*, 515-531, 2002.

[82] Dorst, K., Cross, N., "Creativity in the design process: co-evolution of problem-solution", *Design Studies 22*, 425-437, 2001.

[83] Downing, F., "Transcending memory: remembrance and the design of place", *Design Studies 2003*, 213-235, 2003.

[84] Dubois, S., Eltzer, T., & De Guio, R., "A dialectical based model coherent with inventive and optimization problems", *Computers in Industry 60*, 575-583, 2009.

[85] Dugosh, K., & Paulus, P., "Cognitive and social comparison processes in brainstorming", *Journal of Experimental Social Psychology 41*, 313-320, 2005.

[86] Dzbor, M., & Zdrahal, Z., "Design as interactions of problem framing and problem solving", *In van Harmelen, F. (Eds.), Proceedings 15th European Conference on Artificial Intelligence (ECAI)*, 210-214, 2002.

[87] Eilouti, B. H., "Design knowledge recycling using precedents-based analysis and synthesis models", *Design Studies 30*, 340-368, 2009.

[88] Fan, Z., Wang, J., & Goodman, E., "Exploring open-ended design space of mechatronic Systems", *International Journal of Advanced Robotic Systems 1*, 295 - 302, 2004.

[89] Goldenberg, J., Mazursky D., & Solomon, S., "Toward identifying the inventive templates of new products: a channeled ideation approach", *Journal of Marketing Research 36*, 200-210, 1999.

[90] Goldenberg, J., Lehmann, D., & Mazursky, D., "The idea itself and the circumstances of its emergence as predictors of new product success", *Management Science 47*, 69-84, 2001.

[91] Goldschmidt, G., "Capturing indeterminism: representation in the design problem space", *Design Studies 18*, 441-455, 1997.

[92] Guilford, J. P., "Creativity", *The American Psychologist 5*, 444-454, 1950.

[93] Guilford, J. P., "The Structure of Intellect", *Psychological Bulletin 53*, 267-293, 1956.

[94] Heylighen, A., Deisz, P., & Verstijnen, I., "Less is More Original?", *Design Studies 28*, 499-512, 2007.

[95] Howard, T. J., Culley, S. J., & Dekoninck, E., "Describing the creative design process by the integration of engineering design and cognitive psychology literature", *Design Studies 29*, 160–180, 2008.

[96] Jansson, D. G., & Smith, S. M., "Design fixation", *Design Studies 12*, 3-11, 1991.

[97] Kohn, N., Paulus, P., & Choi, Y., "Building on the ideas of others: An examination of the idea combination process", *Journal of Experimental Social Psychology 47*, 554-561, 2011.

[98] Kryssanov, V., Tamaki, H., & Kitamura, S., "Understanding design fundamentals: how synthesis and analysis drive creativity, resulting in emergence", *Artificial Intelligence in Engineering 15*, 329-342, 2001.

[99] Lawson, B., "Schemata, gambits and precedents: some factors in design expertise", *Design Studies 25*, 443-457, 2004.

[100] Maimon, O., & Horowitz, R., "Sufficient conditions for inventive solutions", *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews 29*, 349 - 361, 1999.

[101] Mansfield, R., Busse, T., & Krepelka, E., "The effectiveness of creative thinking", *Review of Educational Research 48*, 517-536, 1978.

[102] Mobley, M. I., Doares, L. M., & Mumford, M. D., "Process analytic models of creative capacities: evidence for the combination and reorganization process", *Creative Research Journal 5*, 125-155, 1992.

[103] Nijstad, B., Stroebe, W., & Lodewijkx, H., "Cognitive stimulation and interference in groups: exposure effects in an idea generation task", *Journal of Experimental Social Psychology 38*, 535-544, 2002.

[104] Perttula, M., & Liikkanen, L., "Structural tendencies and exposure effects in design idea generation", *Proceedings of ASME 2006 Design Engineering Technical Conference and Computers and Information in Engineering Conference*, 2006.

[105] Poole, S., & Simon, M., "Technological trends, product design and the environment", *Design Studies 18*, 237-248, 1997.

[106] Poon, J., & Maher, M. L., "Co-evolution and emergence in design", *Artificial Intelligence in Engineering 11*, 319-327, 1997.

[107] Robertson B. F., & Radcliffe, D. F., "Impact of CAD tools on creative problem solving in engineering design", *Computer-Aided Design 41*, 136-146, 2009.

[108] Pugh, S., "Creating innovative products using total design", *in D. Clausing and R. Andrade (eds.)*, Addison-Wesley, 1996.

[109] Schon, D., "The reflective practitioner. How professionals think in action", *Basic Books*, 1983.

[110] Schunn, C. D., Lovell, M. R., Wang, Y., & Yang, A., "Measuring innovative apples and oranges: Towards more robust and efficient measures of product innovation", *In J. Gero (Ed.) Studying Design Creativity*, Springer, 2010.

[111] Senbel, M., Girling, C., White, J., Kellett, R., & Chan, P. F., "Precedents reconsidered: Urban design learning catalysed through data rich 3-D digital models", *Design Studies 34*, 74-92, 2013.

[112] Shah, J., Vargas-Hernandez, N., & Smith, S., "Metrics for measuring ideation effectiveness", *Design Studies 24*, 111-134, 2003.

[113] Shai, O. (2003), "Transforming engineering problems through graph representations", *Advanced Engineering Informatics 17*, 77-93, 2003.

[114] Shai, O., Reich, Y., & Rubin, D., "Creative conceptual design: extending the scope by infused design", *Computer-Aided Design 41*, 117-135, 2009.

[115] Shang, Y., Huang, K. & Zhang, Q. P., "Genetic model for conceptual design of mechanical products based on functional Surface", *International Journal of Advanced Manufacturing Technology 42*, 211-221, 2009.

[116] Shelly, M. W. II, & Bryan, G. L., "Human judgments and optimality", *John Wiley & Sons*, 1964.

[117] Simonton, D. K., "Creative thought as blind-variation and selective-retention: Combinatorial models of exceptional creativity", *Physics of Life Reviews 7*, 156-179, 2010.

[118] Smith, S. M., Ward, T. B., & Schumacher, J. S., "Constraining effects of examples in a creative generation task", *Memory & Cognition 21*, 837-845, 1993.

[119] Sternberg, R., "Implicit theories of intelligence, creativity, and wisdom", *Journal of Personality and Social Psychology 49*, 607-627, 1985.

[120] Sternberg, R., "Handbook of Creativity", *Cambridge University Press*, 1999.

[121] Stone, R., Wood, K., & Crawford, R., "A heuristic method for identifying modules for product architectures", *Design Studies 21*, 5-31, 2000.

[122] Thagard, P., and Stewart, T. C., "The AHA! Experience: creativity through emergent binding in neural networks", *Cognitive Science 35*, 1-33, 2011.

[123] Tseng, I., Moss. J., Cagan, J., & Kotovsky, K., "The role of timing and analogical similarity in the simulation of idea generation in design", *Design Studies 29*, 203-221, 2008.

[124] Van Knippenberg, D., De Dreu, C. K. W., and Hofman, A. C., "Work diversity and group performance: an integrative model and research agenda", *Journal of Applied Psychology 89*, 1008-1022, 2004.

[125] Vosniadou, S., & Ortony, A., "Similarity and Analogical Reasoning", *Cambridge University Press*, 1989.

[126] Ward, T. B., Patterson, M. J., & Sifonis, C. M., "The role of specificity and abstraction in creative idea generation", *Creativity Research Journal 16*, 1-9, 2004.

[127] Yang, G.-Z., "Body sensor networks", *Springer*, 2006.

[128] Zheng, L., Proctor, R., & Salvendy, G., "Creativity in ergonomic design: A supplemental value-adding source for product and service development", *Human Factors: The Journal of the Human Factors and Ergonomic Society*, 2010, published online http://hfs.sagepub.com/content/52/4/503.

[129] Zheng, L., Proctor, R., & Salvendy, G., "Can traditional divergent thinking tests be trusted in measuring and predicting real-world creativity?", *Creativity Research Journal 3*, 24-37, 2011.

[130] Agogue, M., Kazakci, A., Weil, B., & Cassotti, M., "The Impact of Examples on Creative Design: Explaining Fixation and Stimulation Effects", *Proceedings of the International Conference on Engineering Design, Vol. 2, Eds. Culley, S.J., Hicks, B.J., McAloone, T.C.; Howard, T.J. & Reich, Y.*, 266-274, 2011.

[131] Ainsworth, S., & Loizou, A. Th., "The effects of self-explaining when learning with text or diagrams", *Cognitive Science 27*, 669-681, 2003.

[132] Ainsworth, S., Prain, V., & Tytler, R., "Drawing to Learn in Science", *Science 333*, 1096-1097, 2011.

[133] Amabile, T. M., "Social psychology of creativity: A consensual assessment technique", *Journal of Personality and Social Psychology 43*, 997-1013, 1982.

[134] Barsalou, L., "Situated Simulation in the Human Conceptual System", *Language and Cognitive Processes 18*, 513-562, 2003.

[135] Baughman, W. A., & Mumford, M. D., "Process-Analytic Models of creative Capacities: Operating Influencing the Combination-and-Reorganization Process", *Creativity Research Journal 8*, 37-62, 1995.

[136] Bilda, Z., Gero, J., & Purcell, T., "To sketch or not to sketch? That is the question", *Design Studies 27*, 587-613, 2006.

[137] Chan, J., Fu, K., Schunn, C., Cagan, J., Wood, K., & Kotovsky, K., "On the Benefits and Pitfalls of Analogies for Innovative Design: Ideation

Performance Based on Analogical Distance, Commonness, and Modality of Examples", *Journal of Mechanical Design 133*, 081004, 2011.

[138] Christensen, B. T., & Schunn, C. D., "The relationship of analogical distance to analogical function and preinventive structure: The case of engineering design", *Memory & Cognition 35*, 29-38, 2007.

[139] Cohen, B., & Murphy, G., "Models of concepts", *Cognitive Science 8*, 27-58, 1984.

[140] Collado-Ruiz, D., "Influence of environmental information on creativity", *Design Studies 31*, 479-498, 2010.

[141] Dahl, D. W., & Moreau, P., "The influence and value of analogical thinking during new product ideation", *Journal of Marketing Research 39*, 47-60, 2002.

[142] Davenport, J. L., Yaron, D., Klahr, D. & Koedinger, K., "When do diagrams enhancing learning? A framework for designing relevant representations", *Proceedings of the 8th International conference for the learning sciences*, 191-198, 2008.

[143] Downing, P., "On the Creation and Use of English Compound Nouns", *Language 53*, 810-842, 1977.

[144] Estes, Z., & Ward, T. B., "The Emergence of Novel Attributes in Concept Modification", *Creativity Research Journal 14*, 149-156, 2002.

[145] Ferent, C., Montano, S., & Doboli, A., "A Prototype Framework for Conceptual Design of Novel Analog Circuits", *Proceedings of International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design*, 2012.

[146] Ferent, C. Doboli, A., & Doboli, S., "Symbolic Matching and Constraint Generation for Systematic Comparison of Analog Circuits", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 32*, 2013.

[147] Ferent, C., Doboli, A. & Doboli, S., "An Axiomatic Model for Concept Structure Description and Its Application to Circuit Design", *Knowledge-based Systems*, 2013, http://dx.doi.org/10.1016/j.knosys.2013.02.013.

[148] Hampton, J. A., "Inheritance of attributes in natural concept conjunctions", *Memory & Cognition 15*, 55-71, 1987.

[149] Hampton, J. A., "Emergent Attributes in Combined Concepts", *In T. B. Ward, S. M. Smith, J. Viad (Eds.), Conceptual Structures and Processes: Emergence Discovery and Change*, Washington DC: American Psychological Association, 1996.

[150] Hastie, R., Schroeder, C., & Weber, R., "Creating complex social conjunction categories from simple categories", *Bulleting of the Psychonomic Society 28*, 242-247, 1990.

[151] Herring, S. R., Chang, C.-C., Krantzler, J., & and Bailey, B. P., "Getting Inspired! Understanding How and Why Examples are Used in Creative Design Practice", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 87-96, 2009.

[152] Hwang, W.-Y., Chen, N.-S., Dung, J.-J., & Yang, Y.-L., "Multiple Representation Skills and Creativity Effects on Mathematical Problem Solving using a Multimedia Whiteboard System", *Educational Technology & Society 10*, 191-212, 2007.

[153] Kaeslin, H., "Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication", *Cambridge University Press*, 2008.

[154] Kilgour, A. M., "Improving the creative process: Analysis of the effects of divergent thinking techniques and domain specific knowledge on creativity", *International Journal of Business and Society 7*, 79-107, 2006.

[155] Kilgour, M., & Koslow, S., "Why and how do creative thinking techniques work?: Trading off originality and appropriateness to make more creative advertising", *Journal of the Academy of Marketing Science 37*, 298-309, 2009.

[156] Kulkarni, C., Dow, S. P., & Klemmer, S. R., "Early and Repeated Exposure to Examples Improves Creative Work", *Proceedings CogSci*, 2012.

[157] Lagne, C. L., "Relation-Based Combinations Versus Property-Based Combinations: A Test of the CARIN Theory and the Dual-Process Theory of Conceptual Combinations", *Journal of Memory and Language 42*, 365-389, 2000.

[158] Lagne, C. L., "The competition-among-relations-in-nominals theory of conceptual combinations: implications for stimulus class formation and class expansion", *Journal of the Experimental Analysis of Behavior 78*, 551-565, 2002.

[159] Lee, B., Srivastava, S., Kumar, R., Brafman, R., & Klemmer, S. R., "Designing with Interactive Example Galleries", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2257-2266, 2010.

[160] Levi, J. N., "The syntax and semantics of complex nominals", *Academic Press*, 1978.

[161] Markman, A. B., & Wisniewski, E. J., "Similar and different: The differentiation of basic-level categories", *Journal of Experimental Psychology: Learning, Memory, and Cognition 23*, 54-70, 1997.

[162] Marguc, J., Forster, J., & Van Kleef, G. A., "Stepping back to see the big picture: when obstacles elicit global processing", *Journal of Personality and Social Psychology 101*, 883-901, 2011.

[163] McDonnell, J., Lloyd, P., & Valkenburg, R., "Developing design expertise through the construction of video stories", *Design Studies 25*, 509-525, 2004.

[164] Oxman, R., "Design by re-representation: a model of visual reasoning in design", *Design Studies 18*, 329-347, 1997.

[165] Rye, N., "The design brief as carrier of client information during the construction process", *Design Studies 25*, 231-249, 2004.

[166] Segers, N. M., de Vries, B., & Achten, H. H., "Do word graphs stimulate design", *Design Studies 26*, 625-647, 2005.

[167] Smith, E. E., Osherson, D. N., Rips, L. J., & Keane, M., "Combining Prototypes: A Selective Modification Model", *Cognitive Science 12*, 485-527, 1988.

[168] Smith, S.M., & Blankenship, S.E., "Incubation and the persistence of fixation in problem solving", *American Journal of Psychology 104*, 61-87, 1991.

[169] Stokes, P., "Creativity from constraints", *Springer Publishing Company*, 2006.

[170] Vosniadou, S., & Brewer, W. F., "Theories of Knowledge Restructuring in Development", *Review of Educational Research 57*, 51-67, 1987.

[171] Ward, T. B., "Structured Imagination: The role of category structure in exemplar generation", *Cognitive Psychology 27*, 1-40, 1994.

[172] Ward, T. B., "Analogical distance and purpose in creative thought: Mental leaps versus mental hops", *In K. J. Holyoak, D. Gentner, & B. N. Kokinov (Eds.), Advances in analogy research: Integration of theory and data from the cognitive, computational, and neural sciences*, pp. 221-230, 1998.

[173] Wilkenfield, M. J., & Ward, T. B., "Similarity and Emergence in Conceptual Combinations", *Journal of Memory and Language 45*, 21-38, 2001.

[174] Wisniewski, E. J., "Construal and Similarity in Conceptual Combination", *Journal of Memory and Language 35*, 434-453, 1996.

[175] Wisniewski, E. J., "When concepts combine", *Psychonomic Bulletin & Review 4*, 167-183, 1997.

[176] Wisniewski, E. J., & Love, B. C., "Relations versus Properties in Concept Combination", *Journal of Memory and Language 38*, 177-202, 1998.

[177] J. Pearl, "Causality: Models, Reasoning and Inference", *Cambridge University Press*, Second Edition, 2009.

[178] R. Scheines, P. Spirtes, C. Glymour, C. Meek and T. Richardson, "The TETRAD Project: Constraint Based Aids to Causal Model Specification", *Multivariate Behavioral Research*, Vol. 33, Pages 65-117, 1998.

[179] R. Scheines and et al., *TETRAD, http://www.phil.cmu.edu/tetrad*