Chapter 5

**Knowledge Representation
and Reasoning for the Design
of Resilient Sensor Networks**

David Kelle, Touria El-Mezyani, Sanjeev Srivastava,
and David Cartes

Contents

5.1 Introduction 94

5.2 Background..... 95

 5.2.1 Simplicial Complexes..... 95

 5.2.2 Predicate Relations on Simplicial Complexes 96

 5.2.3 Path Algebra 97

5.3 Tools for Representation 98

5.4 Tools for Reasoning 99

 5.4.1 Facet Chain Algebra 99

 5.4.2 Finding the Best Facet Chain 101

 5.4.3 Reasoning Algorithm 103

5.5 Sensing System Design for Power Systems..... 104

 5.5.1 Development of Knowledge 106

 5.5.1.1 Redundancy and Agreement among Sensors 106

 5.5.1.2 Representation of Knowledge..... 107

 5.5.1.3 Strategy Chains for Redundancy Checks 108

 5.5.2 Reasoning Algorithm for Sensor Failure Detection and Data Restoration 110

 5.5.2.1 Simulation Results 110

 5.5.2.2 Future Application: Using Analogies in the Reasoning Algorithm for
 Sensor Team Selection 112

5.6 Conclusion 112

References 113

In this chapter, we propose a new approach for the design of resilient sensor networks using a knowledge representation (KR) formalism that will facilitate reasoning and collaboration among sensors. This approach utilizes abstract simplicial complexes (ASCs), which serve as the building blocks of this KR, permitting the encoding of more complex relationships than models based on graph theory due to their additional structure. These mathematical constructions exist in the framework of combinatorial algebraic topology (CAT), which uses discrete methods to formalize how algebraic equations relate to the structure of a space. Its algebraic nature allows formulation of algorithms and automation of decision making. Electrical power systems are a natural choice for an application of this KR because of their large size, distributed information, and complex dynamics. The developed approach can be used to design resilient sensor networks that can assist the power management, control, and monitoring systems in inferring and predicting the system state, as well as determining the health of all components, including sensors. This approach is illustrated on an IEEE 14 bus power system.

5.1 Introduction

Sensing and measurement of large-scale system variables such as weather conditions (wind speed, temperature, etc.), power flow, and market prices are essential for improving the monitoring and control of engineered systems. This information can be gathered by means of a distributed heterogeneous network of sensors. The heterogeneous nature of knowledge (e.g., electrical, weather, etc.) results in significant challenges involving control and monitoring systems. Usually, engineers design the sensing and data processing for control and monitoring by considering one particular type of knowledge for a given physical domain. Advanced sensing techniques will collect valuable information about the system conditions throughout the network for multiple physical domains. By considering these multiple forms of knowledge in concert, one can improve problem solving.

Using new communications such as the Internet and reliable media such as wireless, broadband power lines, or fiber optics, a network of distributed sensors will have the ability to communicate quickly in order to infer valuable information. For example, in large-scale systems, sensors will have the ability to interact and coordinate with control and monitoring systems for optimal system control and decision making.

While the study of sensor network design for the state estimation problem is fairly widespread [1,13], their use in distributed, large-scale systems for managing complexity and uncertainty, as well as real-time decentralized control and monitoring is seen as an increasingly important avenue of research. As such, our objective in this research is to develop tools that assist sensors to infer the system state, detect and diagnose failures, as well as to learn and adapt to their changing environment. In order to achieve these objectives, we introduce an approach based on a sensor management system (SMS). This SMS will be able to quickly collect important data and process it as well as monitor the health of sensors and restore erroneous or missing data. This system will inform (1) the controller about the state of the system, and (2) the monitoring system about the health of sensors and assist decisions. By optimizing the communication pathway between sensors and the control or monitoring system, as well as minimizing the number of sensors required for performing a task, we can improve system resilience and avoid problems, such as high maintenance cost, as well as latency and congestion.

KR is the key to the design of an intelligent SMS because it provides the basic cognitive structure of reasoning and machine learning. This can enable an SMS to predict possible outcomes and required actions from its perception of a situation. KR and reasoning is the study of thought

as a computational process. It is concerned with how a system uses its knowledge to infer solutions and make decisions. While logical reasoning produces answers to problems based on fixed rules, analogical reasoning produces conclusions by extending a known comparison, strategy, or concept in one domain called the source, to another domain, called the target. Analogies can help gain understanding of new concepts by comparing them with past experiences. There are many references such as [3,7–10] that discuss the theory of analogical reasoning in the context of the human mind. Hadamard in [10] suggests that human creative insight comes from the application of an analogy from a different domain of knowledge, and Collins in [3] explores its role in individuals with incomplete knowledge of the source domain. However, most work done to represent analogical reasoning focuses on graph-theoretic representations which are limited in the dimension of allowed relations.

The goal of this chapter is to develop tools for analogical reasoning and problem solving using CAT. We present a method to form solution strategies as well as make analogies, applying successful strategies to new situations. These methods use ASCs as a formalism for KR. An ASC is an extension of a graph, where the idea of an edge is generalized to arbitrary dimensions. This allows much richer structures such as the mutual adjacency of any number of vertices. Models based on simplicial complexes can encode more complex relationships than the traditional graph-theoretic models due to their additional structure, making ASCs a good choice for the building blocks of an analogy-capable KR. In [5], the authors developed a computer model for analogy solving using simplicial complexes. The source and the target analogs are represented as simplices and the analogy is modeled as topological deformations of these simplices along a chain sequence according to a set of rules. However, this proposed approach was adapted to the specific problem of solving IQ test questions, and was not applied to general intelligent systems. In order to apply this approach to engineering systems we need to develop a basic mathematical language of reasoning. This will be used to (1) model the knowledge collected about the system (e.g., mathematical constraints, etc.), (2) use the KR technique, and (3) develop reasoning process using analogies. The reasoning process will then analyze system conditions, build strategies for communication and interaction between sensors, and initiate necessary actions in the case of missing data or loss of sensors. The benefit of the proposed sensing systems offers the ability of fast data collection, processing, and reliability and resilience in the presence of failures. This new intelligent sensing system promise to increase the system resilience and improve performance of system monitoring and control.

5.2 Background

In this section, we present an overview and background of the existing mathematical tools applied in this chapter. Simplicial complexes form the backbone of the proposed knowledge structure. We first define some important concepts that involve ASCs and relations [4], and then discuss how ASCs can be used to represent knowledge [5].

5.2.1 Simplicial Complexes

Simplicial complexes are abstract structures developed in algebraic topology [14]. Geometrically, simplices are an extension of triangles and tetrahedra to arbitrary dimensions.

Definition 5.1 (Simplex, face, facet) An m -simplex $S = \{x_1, \dots, x_{m+1}\}$ is the smallest convex set in $\mathbb{R}^m$ containing the given set of $m + 1$ points (Figure 5.1). A simplex formed from a subset of the points of S is called a face of S , and a maximal simplex under inclusion is called a facet.

Figure 5.1 m -simplexes for $0 \leq m \leq 3$.

Simplices may be glued together along their faces. Such a geometric structure is called a *simplicial complex* Δ if every face of every simplex is also considered to be a simplex in Δ . A simplicial complex that is contained in another is called a *sub-complex*. However, we are only interested in the combinatorial properties of a simplicial complex. With this in mind, we consider an *ASC*.

Definition 5.2 (Abstract simplicial complexes) An *ASC* is a collection Δ of subsets of a set of vertices $V = \{x_1, \dots, x_n\}$, such that if S is an element of Δ then so is every nonempty subset of S [11,14]. The dimension of an ASC is the dimension of its largest facet. An ASC is called *pure* if all of its facets have the same dimension [6].

5.2.2 Predicate Relations on Simplicial Complexes

Dowker provides a way to represent binary relations as a simplicial complex [4]. Let $R \subset A \times B$ be a binary relation between sets A and B . As a subset of $A \times B$, $(a, b) \in R$ means that a is related to b by the relation R , that is, aRb . For each fixed $b \in B$, $\Delta_A(B)$ describes what elements of A are related to b and vice versa. In this way, two dual simplicial complexes,* $\Delta_A(B)$ and $\Delta_B(A)$, are associated to R . This can be taken further by replacing the arbitrary set B with a set of predicates $P = \{p_1, \dots, p_q\}$. This allows us to define R so that $(a, p_i) \in R$ if and only if a satisfies the predicate p_i , that is, $p_i(a)$ is true. $\Delta_A(P)$ tells what elements of A satisfy each predicate $p_i \in P$, whereas $\Delta_P(A)$ tells all the predicates satisfied by each $a \in A$. The use of simplicial complexes to represent data is part of a larger theory called Q-analysis [2].

Example 5.1: 1-ary predicate

The belief “*Per, Hans, and Leif are male*” is represented as follows:

$$(Per, male), (Hans, male), (Leif, male) \in R$$

where *male* is the predicate “*is male*.” The simplex $\{Per, Hans, Leif\}$ is associated to the predicate *male*,[†] as shown in Figure 5.2.

* In the rest of the chapter, we drop the reference to the predicate and object sets and just refer to the complex as Δ .

† From here the simplex will just be referred to as the predicate itself.

Figure 5.2 1-ary predicate.

5.2.3 Path Algebra

An algebra is a collection of mathematical objects along with specific rules for their interactions. Algebras can provide languages in which problems of different types can be discussed. A path algebra in graph theory is a set of paths equipped with two binary operations that satisfy certain requirements. Path algebras are used to solve problems like finding optimal paths between two vertices [12], finding all paths emanating from a source or converging to a target [17], determining shortest and longest paths, etc. In [5], the authors applied simplicial complexes to the problem of solving IQ test analogy questions by selecting deformations that had fewer steps and preserved the most properties, but did not develop a rigorous algebra of these deformations or use them for reasoning. We define this algebra by extending path algebras from graph theory to ASC so that it can be used to solve problems with analogies in a simplicial KR.

First, we present some background notions from [12] about path algebras. A path algebra is defined to be a set P , with two binary operations $\vee$ and $\circ$ that satisfy the following properties:

- $\vee$ is idempotent, commutative, and associative. That is, for all $a, b, c \in P$,

$$\begin{aligned} a \vee a &= a, \\ a \vee b &= b \vee a, \\ (a \vee b) \vee c &= a \vee (b \vee c). \end{aligned}$$

- $\circ$ is associative as well as left and right distributive. For all $a, b, c \in P$,

$$\begin{aligned} (a \circ b) \circ c &= a \circ (b \circ c), \\ a \circ (b \vee c) &= (a \circ b) \vee (a \circ c), \\ (a \vee b) \circ c &= (a \circ c) \vee (b \circ c). \end{aligned}$$

- There exist elements $\epsilon, \lambda \in P$ such that for any $a \in P$,

$$\begin{aligned} \epsilon \circ a &= a = a \circ \epsilon, \\ \lambda \vee a &= a = a \vee \lambda, \\ \lambda \circ a &= \lambda. \end{aligned}$$

The operation $\vee$ is called the join and $\circ$ is called the product. The elements ϵ and λ are the units of the product and join operations, respectively. By defining the join operator to select the shortest path and the product as path concatenation, Manger shows that the closure (Definition 5.4) of the adjacency matrix of a graph yields the shortest path from node i to node j in its i, j th entry [12]. The join and product of two matrices is defined in terms of the join and product of paths.

Definition 5.3 (Join and product) Let A, B be $n \times n$ matrices over P . Define the operations $\vee$ and $\circ$ on the matrices A, B :

$$A \vee B = [a_{ij} \vee b_{ij}]$$

$$A \circ B = \left[\bigvee_{k=1}^n a_{ik} \circ b_{kj} \right]$$

The powers of the matrix A are defined inductively with the product operation.

Definition 5.4 (Stability index and closure) A square matrix A is said to be *stable* if for some integer q ,

$$\bigvee_{k=1}^q A^k = \bigvee_{k=1}^{q+1} A^k$$

The smallest such q is called the stability index of A , and the join of the matrices A^k as k ranges from 1 to q is called the (weak) closure of A , written $\hat{A}$:

$$\hat{A} = \bigvee_{k=1}^q A^k$$

5.3 Tools for Representation

Our goal is to extend the binary relation used in [4,5] in order to describe relationships that are more complex than the sharing of common properties.

Example 5.2: Necessity of extension

The belief “*Per, Hans, and Leif are brothers*” cannot be encoded as in *Example 5.1* by just replacing the predicate m ; the statements “*Per is a brother, Leif is a brother, and Hans is a brother*” do not have the intended meaning.

Instead, we extend the set A to a product of sets $A_1 \times A_2 \times \cdots \times A_m$, and define the predicate as follows.

Definition 5.5 (m-Ary predicate) Let m be the minimal number of variables required to express a relationship. We define the m -ary predicate p on m variables to be the map from the product of sets $A_1 \times A_2 \times \cdots \times A_m$ to the set $\{True, False\}$:

$$p : A_1 \times A_2 \times \cdots \times A_m \rightarrow \{True, False\}.$$

Predicates with $m = 1$ are called properties.

Figure 5.3 Binary predicate.

Example 5.3: Binary predicates

Per, *Hans*, and *Leif* are brothers. The following beliefs imply the desired belief:

1. *Per* and *Hans* are brothers.
2. *Per* and *Leif* are brothers.
3. *Leif* and *Hans* are brothers.

Figure 5.3 shows the topological representation of the brotherhood of *Per*, *Hans*, and *Leif*.*

5.4 Tools for Reasoning

The use of analogy in problem solving allows an intelligent system to connect different knowledge domains via method, strategy, or context to an unfamiliar problem.

In this section, we develop an algebra for reasoning by analogy by defining a concept similar to a path called a facet chain. We discuss how this algebra can be used to find a “best” facet chain, and then introduce an algorithm for reasoning.

5.4.1 Facet Chain Algebra

We begin by creating some basic terminology about facet chains and define the operations necessary for creating an algebra.

Definition 5.6 (Links and size) Let $\mathcal{F}$ be the collection of facets of a finite ASC. We define a link to be a set containing two facets $L_{i,j} = \{F_i, F_j\}$. A connected link is a link whose facets are adjacent. The size $|L_{i,j}|$ of the link $L_{i,j}$ is the number of vertices in the intersection $F_i \cap F_j$. The link $\{F_i, F_j\} = \{F_j\} = \epsilon_j$ is called a trivial link. For convenience, we sometimes write $L_{i,j}^{|L_{i,j}|}$ to denote the link along with its size.

Consider an alphabet $\mathcal{L}$ as the set of links of a finite ASC. Then a word over $\mathcal{L}$ is a sequence of links; the set of words over $\mathcal{L}$ is denoted $\mathcal{L}^*$.

* Together, the statements exhibit logical dependencies; e.g., (1), (2) $\Rightarrow$ (3).

Definition 5.7 (Facet chains and length) We refer to an element $\Gamma \in \mathcal{L}^*$ as a facet chain.* A connected facet chain is a facet chain whose links are all connected. The length of a facet chain is written as $|\Gamma|$ and is equal to the number of its nontrivial links. We write a facet chain from F_i to F_j as $\Gamma(i, j)$ and the set of facet chains from F_i to F_j as $\mathcal{C}_{ij}$.

Definition 5.8 (Critical size) Let $\Gamma \in \mathcal{L}^*$. We refer to the link in Γ with the smallest size as the critical link of Γ and its size as the critical size $|\Gamma|_c$:

$$|\Gamma|_c = \min\{|L_{i,i+1}| : i = 1, \dots, |\Gamma| - 1\}$$

Definition 5.9 (Aspect ratio) Let $\Gamma \in \mathcal{L}^*$. Define the aspect ratio AR of Γ to be the ratio of its length to its critical size:

$$AR(\Gamma) = \frac{|\Gamma|}{|\Gamma|_c}$$

If the denominator is zero, define the aspect ratio to be ∞ .

Depending on the ASC, there can be many facet chains connecting two facets. We develop the chain operations join and product along with a norm on chains to select the desired facet chain.

Definition 5.10 (Chain norm) Let α be a positive rational number, $\alpha \in \mathbb{Q}^+$. Define the function ρ_α to assign to every chain Γ in $\mathcal{F}^*$ a value in $\mathbb{Q}^+ \cup \{\infty\}$:

$$\rho_\alpha(\Gamma) = \alpha \cdot AR(\Gamma)$$

The chain norm allows the join to select chains based on two requirements: short length and large critical size. For low aspect ratios, it prioritizes the length requirement while prioritizing the size for high aspect ratios. The constant α determines the aspect ratio at which this distinction occurs.

Remark 5.1 (Connected facet chains) The norm $\rho_\alpha(\Gamma)$ is finite if and only if Γ is connected and has finite length.

Remark 5.2 ($\mathcal{F}$ is a metric space) The set of facets $\mathcal{F}$ is a metric space with distance between two facets defined as $d(F_i, F_j) = \min\{\rho(\Gamma(i, j)) | \Gamma(i, j) \in \mathcal{C}_{ij}\}$.

Now we can define the operations join and product for facet chains.

* Our definition of a facet chain differs from a chain in algebraic topology, which is defined as a linear combination of simplices.

Definition 5.11 (Join and product of facet chains) Let $\Gamma(i, j), \Gamma(k, l)$ be two facet chains in $\mathcal{F}^*$. Define the join of $\Gamma(i, j)$ and $\Gamma(k, l)$ to be the one with the smallest norm:

$$\Gamma_{ij} \vee \Gamma_{kl} = \begin{cases} \Gamma(i, j) : & \rho_\alpha(\Gamma(i, j)) < \rho_\alpha(\Gamma(k, l)) \\ \Gamma(k, l) : & \rho_\alpha(\Gamma(k, l)) < \rho_\alpha(\Gamma(i, j)) \\ \Gamma(i, j) : & \rho_\alpha(\Gamma(k, l)) = \rho_\alpha(\Gamma(i, j)), \\ & i < k \end{cases}$$

Now, we define the product of $\Gamma(i, j)$ and $\Gamma(k, l)$ to be concatenation of the two facet chains when the last link of $\Gamma(i, j)$ equals the first link of $\Gamma(k, l)$:

$$\Gamma(i, j) \circ \Gamma(k, l) = \begin{cases} \Gamma(i, j) * \Gamma(k, l) : & j = k \\ \lambda : & \text{otherwise} \end{cases}$$

The set of chains $\mathcal{F}^*$ along with the operations $\circ$ and $\vee$ form the facet chain algebra.

5.4.2 Finding the Best Facet Chain

Recall that Definition 5.3 extends the operations of join and product to square matrices. Applied to the aforementioned definitions, the join operation $\bigvee_{k=1}^n \Gamma(i, k) \circ \Gamma(k, j)$ operates on the set of chains in $\mathcal{C}_{ij}$. From Definition 5.4, the closure of the adjacency matrix can yield useful information about paths from one vertex to another. We now define two matrices that allow us to quickly compute the best chain between two facets; the first plays a similar role as the adjacency matrix in graph theory, and the second encodes the instructions for following the chain.

Definition 5.12 (Link matrix) Let $\mathcal{I}$ be the square matrix of connected links:

$$\mathcal{I} = [L_{ij}^{|L_{ij}|}]$$

We can easily think of a chain $\Gamma(i, j)$ as transforming the facet F_i into F_j . However, this transformation requires knowledge of all the facets involved. In a situation where a system must learn from an analogy between two domains, we might not have this complete knowledge in the target domain. By considering a sequence of vertex permutations that correspond to the links in the chain $\Gamma(i, j)$, we are able to obtain an explicit sequence of instructions that is independent of any facet knowledge. If the size of a link is one less than the dimension of the facets, then there is only one possible permutation as each facet has only one vertex that is not in their intersection. If it is smaller than this, then there are several permutations that change the facets as desired. However, since we are only dealing with combinatorial information, each facet is viewed as a set and sets are invariant under ordering. We consider two permutations to be equivalent if they both correspond to the same link. Each equivalence class is represented by one of its elements.

Definition 5.13 (Permutation matrix) To a link L_{ij} between facets F_i and F_j , associate the permutation $\sigma_{ij} = (i_1 j_1)(i_2 j_2) \dots (i_n j_n)$, where $n = |F_j \setminus (F_i \cap F_j)|$ and the vertices $i_1, \dots, i_n \in F_i$ and $j_1, \dots, j_n \in F_j$.

Define the matrix $\mathcal{P} = [\sigma_{ij}]$ to be the matrix of these permutations.

Figure 5.4 Example ASC with seven facets.

Proposition 5.1 The i, j th entry of the matrix $\hat{\mathcal{I}}$ is the chain from F_i to F_j with the smallest chain norm, and the i, j th entry of $\hat{\mathcal{P}}$ is a sequence of permutations on vertices that when applied to the facet F_i will sequentially transform it into F_j . The i th column of $\hat{\mathcal{I}}$ represents all possible best facet chains originating at F_i and the j th row of $\hat{\mathcal{I}}$ represents all best facet chains terminating at F_j .

Therefore, the matrix $\hat{\mathcal{I}}$ tells us, for every two facets in the ASC, which facet chain takes the shortest number of steps and preserves the most properties during each step; the matrix $\hat{\mathcal{P}}$ tells us how to actually perform the corresponding transformation.

Example 5.4:

Let Δ be the simplicial complex shown in Figure 5.4. The matrix $\mathcal{I}$ is given as follows:

$$\begin{bmatrix} \epsilon_1 & L_{1,2}^2 & L_{1,3}^1 & 0 & L_{1,5}^2 & L_{1,6}^1 & 0 \\ L_{2,1}^2 & \epsilon_2 & L_{2,3}^2 & L_{2,4}^1 & L_{2,5}^2 & 0 & 0 \\ L_{3,1}^1 & L_{3,2}^2 & \epsilon_3 & L_{3,4}^2 & L_{3,5}^1 & 0 & L_{3,7}^1 \\ 0 & L_{4,2}^1 & L_{4,3}^2 & \epsilon_4 & 0 & 0 & L_{4,7}^1 \\ L_{5,1}^2 & L_{5,2}^1 & L_{5,3}^1 & 0 & \epsilon_5 & L_{5,6}^2 & L_{5,7}^1 \\ L_{6,1}^1 & 0 & 0 & 0 & L_{6,5}^2 & \epsilon_6 & L_{6,7}^2 \\ 0 & 0 & L_{7,3}^1 & L_{7,4}^1 & L_{7,5}^1 & L_{7,6}^2 & \epsilon_7 \end{bmatrix} = \mathcal{I}$$

Similarly, we have the matrix $\mathcal{P}$ shown in Equation 5.1:

$$\mathcal{P} = \begin{bmatrix} e & (1\ 4) & (1\ 5)(2\ 4) & \dots & (2\ 8) & (2\ 7)(3\ 8) & \dots \\ (4\ 1) & e & (2\ 5) & (2\ 6)(3\ 5) & (2\ 8)(4\ 1) & \dots & \dots \\ (1\ 5)(2\ 4) & (2\ 5) & e & (3\ 6) & (4\ 8)(5\ 1) & \dots & (4\ 7)(3\ 8) \\ \dots & (2\ 6)(3\ 5) & (6\ 3) & e & \dots & \dots & (4\ 7)(6\ 8) \\ (2\ 8) & (4\ 8)(5\ 7) & (4\ 8)(5\ 7) & \dots & e & (3\ 7) & (3\ 5)(1\ 7) \\ (2\ 7)(3\ 8) & \dots & \dots & \dots & (3\ 7) & e & (1\ 5) \\ \dots & (4\ 7)(3\ 8) & (4\ 7)(3\ 8) & (4\ 7)(6\ 8) & (3\ 5)(1\ 7) & (1\ 5) & e \end{bmatrix} \quad (5.1)$$

The elements of $\mathcal{P}$ written as three dots are permutations corresponding to nonconnected links called *teleportations*.* These are the links of zero size, which are suppressed for clarity.

* Rodriguez refers to disconnected paths in graph theory as teleportations, and we extend the terminology here [17].

Teleportation chains have potential utility for reasoning with hypothetical knowledge of uncertain or future events.

Remark 5.3 (Norm of teleportations) The norm of a teleportation is always ∞ . This can be easily seen as the critical size of a teleportation is zero. Since teleportations have infinite norms, they are never selected by the join when a connected link is available.

There can be many facet chains connecting a pair of facets. For example, $\Gamma_a(1, 4)$, $\Gamma_b(1, 4)$ are two facet chains* connecting F_1 to F_4 :

$$\begin{aligned}\Gamma_a &= L_{1,2}^2, L_{2,3}^2, L_{3,4}^2 \\ \Gamma_b &= L_{1,2}^2, L_{2,4}^1\end{aligned}$$

Since the aspect ratios of the two facet chains are greater than one, the chain norm for $\alpha = 1$ gives priority to the size of the critical link. The corresponding chain norms are shown as follows:

$$\begin{aligned}\rho_1(\Gamma_a) &= \frac{3}{2} \\ \rho_1(\Gamma_b) &= \frac{2}{1}\end{aligned}$$

The chain Γ_a has the smallest chain norm of the three, with length three and norm two, whereas Γ_b has length two and norm one. This yields $\rho_1(\Gamma_a) < \rho_1(\Gamma_b)$, so $\Gamma_a \vee \Gamma_b = \Gamma_a$.

5.4.3 Reasoning Algorithm

The reasoning process starts with the ASC representing the target domain called the *target complex*. The target complex is where the problem statement can be represented. The system may not have complete knowledge of the target domain and may only know the initial target facet and a few vertices of the final target facet. The next step is to find a suitable source complex that is rich enough to contain strategies relevant to the problem. The answers to questions posed in the target complex should then be inferred from reasoning in the source complex (see Figure 5.5). The problem of analogy-finding can be formulated as follows: the source complex, initial target facet, and the correspondence ϕ between the two complexes are known, as well as a partial vertex set of the final target facet; we want to determine the best facet chain that connects the initial target facet with a final target facet containing the partial vertex set. In this work, we assume ϕ is a bijection from the target complex to a subset of the source complex.

In practice, α can be chosen based on the specifics of the ASC. For example, choosing α to be smaller than the aspect ratios of all facet chains in the system would result in reasoning that prefers keeping as many connections as possible at the expense of using more steps. This reasoning is more cautious in a sense. Selecting α to be large would result in more hasty reasoning that prefers quick solutions with few steps, at the expense of maintaining fewer connections.

* From here, we drop the reference to the facets F_1 and F_4 .

Figure 5.5 The inverse map ϕ^{-1} sends the solution Γ from the source to the target.

The analogy-finding algorithm consists of four steps*:

1. Map initial target facet and the partial final target vertex set, respectively F_i^T and V_f^T , to the initial source facet and partial final source vertex set, respectively:

$$\begin{aligned}\phi(F_i^T) &= F_i^S \\ \phi(V_f^T) &= V_f^S\end{aligned}$$

2. Find the set of facet chains $\mathcal{C}_{if}^S$ that start at F_i^S and end at a facet containing V_f^S . This is done by taking entries of the i th column of $\hat{\mathcal{I}}$ whose last facets contain V_f^S . The join of the facet chains in $\hat{\mathcal{I}}$ is the desired facet chain $\Gamma(F_i^S, F_f^S) \in S$.
3. Map the sequence of permutations corresponding to the chain from Step 2 to the target complex:

$$\phi^{-1}(\sigma_{if}^S) = \sigma_{if}^T$$

4. Apply the permutations in order to the initial target facet until the resulting facet contains V_f^T . This sequence of facets is the desired facet chain in T .

5.5 Sensing System Design for Power Systems

The integration of distributed energy resources in power grids and the development of smart grids have raised several challenges such as power grid monitoring and control. Sensing systems play a vital role in solving these challenges and in improving the power grid reliability. When sensors experience hardware or software failures, the resulting contingencies and missing data can cause problems with the data processing algorithms utilized in power systems [16]. We refer to the sensors with missing or erroneous data as lost sensors.

* If the final target facet is known, use F_f^T in place of V_f^T .

Sensors can be the primary sources of failure. Missing or erroneous data sent to the relays or controls can lead to false tripping of circuit breakers or other switching devices, which may lead to blackouts. Sequences of fault events like lost sensors or line sags can be recorded before the blackout happens [15]. If these events are identified quickly and sensor failures are corrected efficiently, major blackouts could be avoided.

To handle this rapid identification of events, we propose to integrate an intelligent SMS with the control and monitoring systems, allowing tasks or services to be accomplished as well as the selection of optimal sensor teams needed. We consider a service to be the monitoring and detection of fault events, and the correction and reconstruction of erroneous or missing data. The SMS should feature communication among sensors with the goal of identifying fault events and correcting errors in data that lead to contingencies.

The integration of the SMS with the sensing system will increase the system resilience and reliability. The SMS can be developed as a software layer in the sensing system, which selects the teams of sensors needed for tasks, checks the health of sensors, corrects and reconstructs data, and infers decisions to controllers.

In the following section, we illustrate the reasoning process with the IEEE 14 bus system (Figure 5.6). We have chosen this case as a simple example for illustration purposes. Although there are simpler techniques that can accomplish the computations shown, our contribution is that this structure of reasoning can be extended to much more difficult problems where those methods fail and flexible reasoning is needed.

Figure 5.6 IEEE 14 bus system.

5.5.1 Development of Knowledge

In this application, we develop sensing services performed by a given team T_i of sensors that are able to detect fault events and reconstruct data from lost sensors. The proposed KR contains information about the services needed for the identification of sensor failures. These services use teams of sensors in a given power line and its adjacent lines. Each service records data to be processed in order to detect and locate the failure.

5.5.1.1 Redundancy and Agreement among Sensors

In order to monitor the sensor network, sensors are checked using their redundancy. The verification of sensor values is done using Kirchhoff's Current Law (KCL) or Voltage Law (KVL). For the examples in this chapter, we use KCL. Consider the sensors at buses 1, $\dots$, 5 in the IEEE-14 bus system as shown in Figure 5.7. At each transmission line $L_{i,j}$, two sensors are placed in the head and the tail respectively. A sensor $s_{i,j}$ is indexed such that i corresponds to the nearest bus and j corresponds to the remaining adjacent bus. The current measured by sensor $s_{i,j}$ is written $I_{i,j}$ and the current measured by sensor $s_{j,i}$ is $I_{j,i}$. The reference direction of current is defined to be toward the higher numbered bus, toward the load, or away from the generator. Since sensors $s_{1,2}$ and $s_{2,1}$ are on the same line, a redundancy check can be done without any computation and simply amounts to checking if $I_{i,j} = I_{j,i}$.

Definition 5.14 (Directly and indirectly redundant teams) Assuming non-fault conditions, we refer to a team of sensors as a directly redundant team if the sensors are all duplicates or backups of each other. For current measurements, this is true only for sensors that are all on the same line. Applying KCL or KVL to a network yields a system of linear equations. If a team of sensors satisfies exactly one of these equations, it is called an indirectly redundant team.

Figure 5.7 Example of team coverage and redundancy.

Example 5.5: Directly and indirectly redundant teams

In Figure 5.7, the team of current sensors $T_1 = \{s_{1,2}, s_{2,1}\}$ is directly redundant, while the team $T_2 = \{s_{2,1}, s_{2,3}, s_{2,4}, s_{2,5}, s_{2,L}, s_{2,G}\}$ is indirectly redundant. The second case can be seen from applying KCL to bus 2 in Figure 5.7. We have

$$I_{2,1} + I_{2,G} = I_{2,3} + I_{2,4} + I_{2,5} + I_{2,L}. \quad (5.2)$$

The variable $I_{2,1}$ is the current measured by sensor $s_{2,1}$ and is defined to flow from bus 1 to bus 2.

Definition 5.15 (Composite redundancy) We say a team satisfies a composite redundancy if it is a union of directly and indirectly redundant teams.

Example 5.6:

In Figure 5.7, the team $T_3 = \{s_{1,2}, s_{2,1}, s_{2,3}, s_{2,4}, s_{2,5}, s_{2,L}, s_{2,G}\}$ is composite as it is a union of T_1 and T_2 . However, $T_4 = \{s_{1,5}, s_{1,2}, s_{2,1}, s_{2,5}, s_{2,4}\}$ is not composite. Every sensor in a composite team can be validated by and contribute to a redundancy check. In order to make T_4 composite, we would either have to add $s_{2,3}$ or remove $s_{2,4}$ and $s_{2,5}$.

5.5.1.2 Representation of Knowledge

We begin by specifying the predicates of the KR and creating facets that satisfy these predicates. Since we are interested in the satisfaction of redundancies, we choose the two predicates*: “The team of sensors is directly redundant” and “The team of sensors is indirectly redundant.” We refer to their respective simplexes as direct and indirect.

Example 5.7:

Take the two sensor teams $\{s_{1,2}, s_{2,1}\}$ and $\{s_{1,G}, s_{1,5}, s_{1,2}\}$ from the network in Figure 5.7. The two simplexes associated to these teams are shown in Figure 5.8.

Let $s_{i,j}$ and $s_{j,i}$ denote the sensors placed on $L_{i,j}$ respectively at the head and tail of the line. In order to design the ASC of sensors in power networks, we first assume that one or more sensors are placed in every transmission line. This allows the power network to have full sensor coverage. The ASC can then be developed considering three steps: (1) At the first step, we consider only the

Figure 5.8 (a) Simplex for a directly redundant team and (b) simplex for an indirectly redundant team.

* The resulting equations such as Equation 5.2 can be seen as a modification of the facet ideal in [6].

Figure 5.9 Constructing the ASC for full coverage. (a) Construction of indirect simplices and line identification. (b) Direct simplices are added. (c) ASC is completed by adding remaining indirect simplices.

indirectly redundant teams and at each line $L_{i,j}$ we consider a possible measurement* (Figure 5.9a); (2) at the second step, we add sensors in directly redundant teams as shown in Figure 5.9b; and finally (3) we add simplices by connecting the new vertex to the rest of the vertices (Figure 5.9c) so that each of the added simplices is itself an indirectly redundant team.

The result of step (1) is shown in Figure 5.10. Step (3) produces the ASC in Figure 5.11 whose simplices represent algebraic substitution of the direct redundancy into the indirect redundancy.

5.5.1.3 Strategy Chains for Redundancy Checks

Here we use facet chains to represent the strategy of checking sensors for contingency from the knowledge of their redundancy. Each of the predicates in the KR represented in Figure 5.11 can be used for redundancy checking, providing strategies for the monitoring service to detect sensor failures. Consider s_i as the service of the agreement check that is executed by the team $T_2 = s_{2,1}, s_{2,5}, s_{2,4}, s_{2,3}, s_{2,L}, s_{2,G}$ of indirectly redundant sensors.

Given a lost sensor to recover data from, form a chain with the following steps:

1. Delete vertices from the SC that do not correspond to existing sensors in the network.
2. If a simplex has more than one vertex removed, delete the simplex.
3. Compute the transitive closure of the link matrix of the ASC.
4. Pick the row of the transitive closure matrix corresponding to a simplex that contains the lost sensor vertex. From this row, select the entry with the smallest chain norm. Call this entry the optimal chain.

The resulting chain identifies the sensors whose data are needed for the reconstruction.

* In the case of KCL, each simplex will correspond to a bus of the network.

Figure 5.10 Simplicial complex of full sensor coverage in IEEE 14-bus system.

5.5.2 Reasoning Algorithm for Sensor Failure Detection and Data Restoration

In order to detect sensor failures, the SMS will use a reasoning algorithm, which will utilize the sensors redundancy to identify the faulty sensors and restore their data. The agreement of sensors is tested by checking for satisfaction of the constraint equation. We say that a team of sensors agrees if their measured quantities satisfy the given constraint equation, in this case one of the Kirchhoff's laws.

Example 5.8: Using Chains from Extended Complex

Figure 5.12 shows hypothetical sensor failures in the complex from Figure 5.9. When one sensor fails, the chain composed of the single link $\{F_1, F_2\}$ still provides redundancy. When another sensor fails, both s_{ij} and $s_{i,b}$ can be computed by substitution. Denote the measured quantity at sensor s_{ij} by s_{ij}^* :

$$F_1 : s_{i,b} = s_{i,a}^* + s_{i,j} \quad (5.3)$$

$$F_2 : s_{i,j} = s_{j,c}^* + s_{j,d}^* \quad (5.4)$$

$$\{F_1, F_2\} : s_{i,b} = s_{i,a}^* + s_{j,c}^* + s_{j,d}^* \quad (5.5)$$

If they all agree, the algorithm declares that all sensors in the team are healthy, while if there is a team who's sensors do not agree with each other, the algorithm will identify the sensors that do not agree and they will be declared faulty. After a sensor is declared faulty, a process of service restoration will begin. If a failure occurs in one sensor, the SMS will accommodate the fault by isolating the faulty sensor and reconstructing their data from the remaining healthy ones.

5.5.2.1 Simulation Results

To validate our approach for the contingency test, the IEEE-14 bus system is simulated in a MATLAB®/Simulink® environment with three different scenarios for sensor faults: for failures in the sensors $s_{1,2}$ at the time interval $[0.1, 0.3s]$, $s_{2,1}$ at the time interval $[0.4, 0.6s]$, and $s_{2,5}$ at the time interval $[0.7, 0.9s]$.

Figure 5.12 (a) First sensor fails, but the chain still provides redundancy; (b) second sensor fails, the chain no longer provides redundancy, but the values of both sensors can still be reconstructed.

Consider the following teams of sensors :

1. Composite: $T_1 = \{s_{1,2}, s_{2,1}, s_{2,3}, s_{2,4}, s_{2,5}, s_{2,L}, s_{2,G}\}$
2. Indirect: $T_2 = \{s_{2,1}, s_{2,3}, s_{2,4}, s_{2,5}, s_{2,L}, s_{2,G}\}$
3. Direct: $T_3 = \{s_{1,2}, s_{2,1}\}$

To test the argument between sensors in a team T_i , we consider the following analytic relations:

$$\mathcal{A}(T_1) : s_{1,2}^* - s_{2,1}^* - s_{2,3}^* - s_{2,4}^* - s_{2,5}^* - s_{2,L}^* - s_{2,G}^* = 0$$

$$\mathcal{A}(T_2) : s_{2,1}^* - s_{2,3}^* - s_{2,4}^* - s_{2,5}^* - s_{2,L}^* - s_{2,G}^* = 0$$

$$\mathcal{A}(T_3) : s_{1,5}^* - s_{1,G}^* - s_{1,2}^* = 0$$

where $\mathcal{A}(T_i)$ represent the agreement relation. The team T_i is in agreement when the relation $\mathcal{A}(T_i)$ is satisfied, otherwise T_i is in disagreement. If a team of sensors disagrees, then one or more sensors in the team is faulty. Figure 5.13 shows the residues of the agreement test conducted in teams T_1 , T_2 , and T_3 . For the three fault scenarios, the agreement relations $\mathcal{A}(T_1)$, $\mathcal{A}(T_2)$, and $\mathcal{A}(T_3)$ are not satisfied and have non-zero residues when failures occur (Figure 5.13). However, only team T_1 is able to detect the maximum number of failures. In order to identify which sensor

Figure 5.13 Agreement test.

has failed, we compare the agreement of the composite team against the agreement of the direct and the indirect redundant teams:

- At the time interval $[0.1, 0.3s]$, team T_2 is in agreement, while teams T_1 and T_3 are not. This means that the faulty sensor is an element of T_1 and T_3 and not an element of T_2 .
- At the time interval $[0.4, 0.6s]$, none of the teams T_1 , T_2 , and T_3 are in agreement. So the faulty sensor must be an element of T_1 , T_2 , and T_3 .
- At the time interval $[0.7, 0.9s]$, team T_3 agrees, while teams T_1 and T_2 are not in agreement. This means that the faulty sensor is an element of T_1 and T_2 and not an element of T_3 .

This comparison allows us to identify the sensor failures when it occurs in power grids.

5.5.2.2 *Future Application: Using Analogies in the Reasoning Algorithm for Sensor Team Selection*

When not quickly and accurately detected, faults in a power network may lead to a cascade of other failures, for example, a transmission line tripping can cause a transient, overloading lines in other areas. Cascading failures in power systems are the major cause of large blackouts. Even simple events such as line sag close to nearby trees can be the primary source of a cascading failure.

As power systems are generally large-scale distributed systems, redundancy checking can be computationally expensive.* To reduce this burden, the SMS should be able to select only specific teams to test the redundancy when a fault event occurs, instead of checking every redundancy. To achieve this objective, analogies can be a good candidate solution for the team selection. This solution will be based on the strategies of chains that have been used when some specific fault events occurred.

In a power system, due to similar components or topologies, certain fault events can be fairly common and repeat themselves in different areas of the grid. By using analogies to identify these similar events, it may be possible to get a better understanding of the spatial and temporal distribution of similar faults. The knowledge of this distribution can be used by the SMS to select only the teams that are useful to detect the given fault type.

5.6 Conclusion

The integration of a new SMS with the capability of learning, updating knowledge, and making decisions without human intervention can answer the control and monitoring needs of complex engineered systems. The objective of this research is to use a CAT-based KR to develop reasoning algorithms for a distributed intelligent SMS, which will be integrated as a software layer in the control and monitoring system.

To achieve this objective, we developed a set of topological tools and a theory for KR and reasoning. The motivation behind developing an analogical reasoning process using this model is to allow an autonomous intelligent system to act according to structural comparisons among its beliefs about the world as opposed to just what is explicitly representable with logical rules.

We use ASCs as a tool to model knowledge that will serve as a basis for reasoning operations. ASCs have been chosen in this research because of their rich structure, which allows the encoding

* The presented method relies on matrix multiplication and so has complexity $O(n^3)$.

of a large range of knowledge. These developed tools will have an impact beyond the field of sensing and measurement systems, as the generality of our approach will yield a theory that can be used in numerous branches of control, such as system health management, robotics, and unmanned space exploration. The development of technologies (e.g., actuators, sensors, etc.) presents an opportunity to develop a new intelligent control paradigm based on reasoning by analogy with the goal of improving the capability of systems. Furthermore, some of the biggest obstacles in large-scale complex systems are (1) uncertainty, (2) controllability, and (3) unpredictability because of the system's dynamic nature. The proposed tools for using analogies for problem solving in real-time will go a long way in addressing these concerns.

References

1. A. Abur and A. Gómez-Expósito. *Power System State Estimation Theory and Implementations*. Marcel Dekker, New York, NY, 2004.
2. R.H. Atkin. From cohomology in physics to q-connectivity in social science. *International Journal of Man-Machine Studies*, 4(2):139–167, 1972.
3. A. Collins, E.H. Warnock, N. Aiello, and M.L. Miller. Reasoning from incomplete knowledge. Technical Report, Bolt Beranek and Newman Inc., Cambridge, MA, March 1975.
4. C.H. Dowker. Homology groups of relations. *The Annals of Mathematics, 2nd Series*, 56(1):84–95, 1952.
5. J.-L. Giavitto and E. Valencia. Algebraic topology for knowledge representation in analogy solving. In *ECAI98: European Conference on Artificial Intelligence*, Brighton, England, 1998.
6. S. Faridi. The facet ideal of a simplicial complex. *Manuscripta Mathematica*, 109:159–174, 2002. 10.1007/s00229-002-0293-9.
7. D. Gentner and B. Bowdle. *Metaphor as Structure Mapping*, New York: Cambridge University Press, 2008, pp. 109–128.
8. D. Gentner and K. Forbus. Computational models of analogy. *WIREs Cognitive Science*, 2:266–276, 2011.
9. D. Gentner, K.J. Holyoak, and B.N. Kokinov. *The Analogical Mind: Perspectives from Cognitive Science*. The MIT Press, Cambridge, MA, March 2001.
10. J. Hadamard. *The Psychology of Invention in the Mathematical Field*. Mineola, NY: Dover Publications, 1954.
11. A. Hatcher. *Algebraic Topology*, Vol. 227. Cambridge University Press, Cambridge, UK, 2002.
12. R. Manger. A new path algebra for finding paths in graphs. In *26th International Conference on Information Technology Interfaces*, Cavtat, Croatia, 2004.
13. A. Monticelli. Electric power system state estimation. *Proceedings of the IEEE*, 88(2):262–282, 2000.
14. J. Munkres. *Elements of Algebraic Topology*. New York: Perseus Books, 1993.
15. D.P. Nedic, I. Dobson, D.S. Kirschen, B.A. Carreras, and V.E. Lynch. Criticality in a cascading failure blackout model. *International Journal of Electrical Power & Energy Systems*, 28(9):627–633, 2006.
16. W. Qiao, G.K. Venayagamoorthy, and R.G. Harley. Missing-sensor-fault-tolerant control for SSSC facts device with real-time implementation. *IEEE Transactions on Power Delivery*, 24(2):740–750, 2009.
17. M.A. Rodriguez and P. Neubauer. A path algebra for multi-relational graphs. In *IEEE 27th International Conference—Data Engineering Workshops*, Hannover, Germany, 2011.