Topological Semantics for Lumped Parameter Systems Modeling

Randi Wang Vadim Shapiro
Spatial Automation Laboratory
University of Wisconsin - Madison

October 18, 2018

Abstract

Behaviors of many engineering systems are described by lumped parameter models that encapsulate the spatially distributed nature of the system into networks of lumped elements; the dynamics of such a network is governed by a system of ordinary differential and algebraic equations. Languages and simulation tools for modeling such systems differ in syntax, informal semantics, and in the methods by which such systems of equations are generated and simulated, leading to numerous interoperability challenges. Logical extensions of SysML aim specifically at unifying a subset of the underlying concepts in such languages.

We propose to unify semantics of all such systems using standard notions from algebraic topology. In particular, Tonti diagrams classify all physical theories in terms of physical laws (topological and constitutive) defined over a pair of dual cochain complexes and may be used to describe different types of lumped parameter systems. We show that all possible methods for generating the corresponding state equations within each physical domain correspond to paths over Tonti diagrams. We further propose a generalization of Tonti diagram that captures the behavior and supports canonical generation of state equations for multi-domain lumped parameter systems.

The unified semantics provides a basis for greater interoperability in systems modeling, supporting automated translation, integration, reuse, and numerical simulation of models created in different authoring systems and applications. Notably, the proposed algebraic topological semantics is also compatible with spatially and temporally distributed models that are at the core of modern CAD and CAE systems.

Keywords: Interoperability, Algebraic topology, System modeling languages; Tonti diagrams

1 Introduction

1.1 Motivation

Lumped parameter models are commonly used to describe behaviors of many engineering systems [1]. In such systems, spatially and temporally distributed physical phenomena are approximated by a finite network of abstract components that store, dissipate, or transform energy; the phenomena-specific constitutive properties of the components (e.g. generalized impedances) are estimated by domain integrals from the actual system by a process of "reticulation"[2]. Bond graphs [2], linear graphs [3], Modelica [4] and Simulink/Simscape [5] are commonly used physical modeling languages for creating, editing, and simulating lumped parameter models. These languages may differ widely in their syntax, but have similar (though not identical) semantics that specifies interconnectivity

and constitutive relations of individual components in a system; the resulting model of a system created in such languages is then compiled into a system of state equations that may be numerically solved to simulate the system's dynamic response. Formally, the dynamic behavior of a lumped parameter model is described by the state equations, a set of ordinary differential equations (ODEs) or differential algebraic equations (DAEs), whose solutions depend on the initial conditions.

Generally speaking, all modeling languages can handle the same broad class of problems but with non-trivial differences in system types, representations, state equation derivation and simulation mechanisms [6, 3, 7, 8]. For example, in Simulink, components exchange numeric information uni-directionally and are not subject to conservation laws; by contrast, the energy flow between components is bi-directional in other languages, satisfying conservation laws. While every linear graph model may be represented by a bond graph, the converse statement is not true [9]. Furthermore, parallel/serial junctions in a bond graph do not specify ordering of branches in the junctions, which implies that every bond graph in fact corresponds to a family of (dynamically) equivalent graph-based models in other languages. Many languages generate state equations using efforts and flows as variables, but their integral forms may also be used, for example in bond graphs[10]. The above and other differences in syntax and semantics of system modeling languages lead to challenges in exchange, translation, and composition of models created in these languages. Such interoperability difficulties are only likely to increase due to ubiquitous and growing adoption of physical modeling languages by industry and standards organizations.

Conceptually, there are two possible approaches to dealing with semantic interoperability issues: "point-to-point" correspondence[1] between models created in different languages, or standardization on a single neutral format that can be translated to/from models in any such languages. The first approach is more practical but is problematic because it requires $O(n^2)$ such translators, which is not only expensive, but discourages development of new languages and simulation solutions. The second approach is similar in spirit to STEP for product models, which requires the neutral format to be formally defined and include the superset of models present in any such language. These difficulties can be observed in a recent effort to extend System Modeling Language (SysML) with packages for direct communication with multiple simulation tools [11]. Recently proposed Functional Mockup Interface (FMI) [12] attempts to sidestep the semantic interoperability issues by supporting model exchange and integration via standardized XML and C-code interfaces; this approach simply shifts the responsibility for semantic correctness of these tasks to the authoring systems and users.

Irrespectively of the selected approach, semantic interoperability requires establishing formal correspondence between concepts and constructs in distinct modeling languages. This is the main goal of this paper. Our approach relies on tools from algebraic topology and well known classification of physical theories developed over the years by Tonti [13, 14], Roth [15], Branin [16], Kron [17], and others. Importantly, this classification generalizes to higher-dimensional physical models, suggesting that the proposed framework may be extended to include spatially-distributed models represented by three-dimensional solid models, partial differential equations and finite element methods.

[1] Such correspondence may take a form of direct translation or using APIs.

1.2 Outline

Section 2 briefly surveys and compares the main system modeling languages and surveys related work. In Section 3, we summarize the well known algebraic topological model of physical systems; this model serves as the basis for Tonti's diagrams that classify physical variables, laws, and theories [14]. Tonti's classification is a starting point for the unified semantics proposed in this paper. The main results of the paper are contained in Sections 4 and 5.

Single domain lumped parameters systems are examined in detail in Section 4. Because all such systems are isomorphic in terms of their physical variables, laws, and Tonti's diagrams, we only discuss the models of electrical systems – with understanding that the discussion also applies to lumped parameters systems in other physical domains. In particular, we show that paths over the corresponding Tonti diagram correspond to all possible ways to generate state equations for an electrical network, including well known node-potential and mesh-current methods that date to Maxwell [19]. While the discussion and the semantics proposed in Section 4 apply to *any one* single-domain lumped-parameter system model, in Section 5 we consider the more general multi-domain systems (that may combine electrical, mechanical, hydraulic, thermal, and other single-domain subsystems). We show that all such systems may be represented by an algebraic topological model in terms of generalized state variables associated with extended and generalized Tonti diagrams. We conclude in Section 6 with discussion of practical applications and consequences of the proposed semantic unification, open issues, and possible extensions.

2 Background

2.1 Brief history of lumped-parameter modeling

Informally, the significance of lumped-parameter models and languages can be attributed to two main factor factors. Lumped parameter models are the simplest models of system behavior (usually, but not always dynamic) that encapsulate and abstract away all detailed geometric information. This allows modeling behavior of systems without dealing with detailed embodiment of such systems – either because these details are too complex or not known (for example in early design stages). Secondly, the lumped parameter models in different physical domains (electrical, mechanical motion, fluid flow, etc.) have identical mathematical structure that unifies all such models. Below we briefly summarizes the conceptual evolution of these developments.

The efforts to classify and unify different physical models and theories date back at least to James Clerk Maxwell, who expressed the desire to establish a formal analogy between various physical quantities based on their mathematical form [19], and continued through 20^{th} century (for example see [20]). Maxwell also introduced the so called *mesh* and *node* methods for solving electrical networks [18, 19] that we will also use to derive state equations for lumped parameter systems in Section 4.

Advances in analysis and simulation of (analog) electrical systems motivated many to use such analogies in order to leverage the advances in electrical engineering in other physical domains [21, 22]. In particular, Nickle proposed to predict the response of a mechanical dynamical system by translating it into a dynamically equivalent analog electrical circuit [21]. This electrical-network approach to modeling and simulation of mechanical systems became widely used and culminated in the work of Gabriel Kron, an influential and controversial electrical engineer, who showed that most distributed mechanical systems may also be modeled and efficiently simulated by analog electrical

networks [23, 24]. Kron's work influenced many others, including F. Branin, P. Roth, E. Tonti, and H. Paynter.

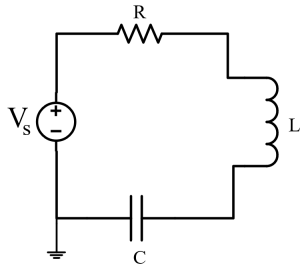
More specifically, Branin recognized that the versatility of electrical networks is an indication of the combinatorial and topological nature of the classical vector calculus [25]. Roth identified specific algebraic topological model for stationary electrical networks in terms of chain and cochain sequences [26], whereas Branin generalized Roth diagram to two dual cochain sequences and revealed that this model underlies general three-dimensional network models. Tonti [14] built on these insights to develop a complete classification of space-time physical theories in terms of their underlying topological structures and codified this classification in terms of concise (Tonti) diagrams. We will review the Tonti diagrams in Section 3 and we will adopt them as a convenient way to describe and visualize models of lumped parameter systems. More details of these developments can be found in Appendix D of [14].

Increasingly, these algebraic topological models are finding applications in computational modeling of engineering systems. For example, several researchers proposed that all geometric and physical computations should be based on a common (co)chain complex model [27, 28, 29], and that a vast majority of physical laws may be enforced as topological invariants of common transformations used in automated design systems [30]. Algebraic topological structure has been used explicitly in variety of modeling systems spanning diverse areas such interactive physics [31], computer graphics [32], and manufacturing systems [33].

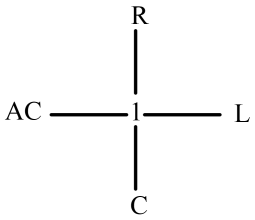
2.2 Languages for lumped parameter modeling

Paynter's work [34] originally focused on methodology for use of lumped parameters models for design and analysis of engineering systems and culminated in the language of bond graphs. He described a systematic method of converting a general system to a simplified lumped parameter model by a process of "reticulation" (creating a network model through discretization of a spatially-distributed system) and developed a symbolic language of *bond graphs* for describing such models. A major conceptual and practical significance of bond graphs is that they model all lumped parameter systems in terms of small number of abstract generalized variables: efforts, flows, their derivatives, integrals, and constitutive relationships. This allowed bond graphs to represent lumped parameter models of complex physically heterogeneous systems that are coupled using abstract transformer and gyrator elements and can be easily translated into a system of ordinary differential equations.

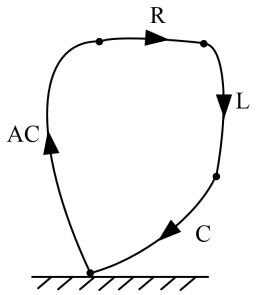
Arguably, all lumped parameter models and languages are based on concepts similar to those used in bond graphs and are now a standard textbook material [10, 3]. Lumped parameter models abstract a spatially and temporaly distributed system by a network of components. The purpose of the network is to model the flow of energy through the system's components as a function of time; detailed geometry of the system is accounted for by integral properties of the components. Formally, the lumped parameter model is described in terms of ordinary differential and algebraic equations with variables that are specific to physical phenomena modeled by the network. However, all such models are isomorphic based on the well known 'analogies' between different physical theories, as summarized in Table 1. In each physical domain, the power is a product of two generalized variables that are classified as across or through (in case of Modelica and linear graph models), or effort and flow (in case of bond graphs). The network includes abstract sources of these variables, and transforms the energy by storing or dissipating in three types elements that are commonly described by analogy with electrical networks as generalized resistors, inductances, and capacitors. In addition, the energy may be transformed between different physical domains using transformer



(a) A simple RLC electrical circuit



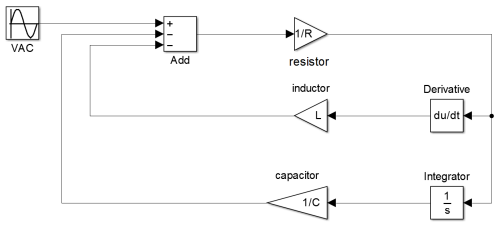
(b) bond graph model



(c) linear graph model

```
model SimpleRLC
  Modelica.Electrical.Analog.Sources.SineVoltage AC(V = 5, freqHz = 10);
  Modelica.Electrical.Analog.Basic.Resistor R(R = 1);
  Modelica.Electrical.Analog.Basic.Capacitor C(C = 0.01);
  Modelica.Electrical.Analog.Basic.Inductor L(L = 0.1);
  Modelica.Electrical.Analog.Basic.Ground Ground;
equation
  connect(Ground.p, AC.n);
  connect(C.p, AC.n);
  connect(AC.p, R.p);
  connect(R.n, L.p);
  connect(L.n, C.n);
end SimpleRLC;
```

(d) Modelica model



(e) Simulink model

Figure 1: Comparison of different physical modeling languages for a simple electrical circuit problem

and gyrator components in the network. More general components with complex behavior may be incorporated into the network, but all such components may be modeled as a composition of these basic components listed above [2]. The formal reason for these analogies is the common underlying algebraic topological structure that we will discuss in Section 3.

The differences between various modeling languages stem from their approach to describing the basic components, their connectivity in the network, and the method by which the governing system of equations is generated. Thus, bond graph uses 0 and 1 junctions to represent parallel or serial connections of components; linear graph uses directed edges to connect components (nodes), while Modelica/Simulink uses connectors to link library components. Behavior of components is described in terms of state variables that are related by constitutive relations: bond graph associates effort, flow and constitutive relationships to 1-ports; linear graph associates across and through variables to directed edges, while Modelica and Simulink associate these variables with blocks.

The governing state equations is a system of ODEs (or more generally DAEs) that are generated from such descriptions by methods that differ in the choice of state variables and generation algorithms. Of course, all such equations describe identical dynamical behavior of the modeled system

Table 1: Analogy of physical variables

Variables type	Effort	Flow	Momentum	Displacement
Mechanical (translation)	Force	Velocity	Momentum	Displacement
Mechanical (rotational)	Torque	Angular velocity	Angular momentum	Angle
Electrical	Voltage	Current	Flux linkage variable	Charge
Hydraulic	Pressure	Volume flow rate	Pressure momentum	Volume

that is determined by initial conditions. To be concrete, let us briefly compare how such languages would be used to model a simple example of a serial RLC electrical circuit shown in Figure 1a. Figure 1 shows how the same circuit would be described in four different languages: bond graphs, linear graphs, Modelica, and Simulink. As we already noted, electrical circuits themselves have also been used as a general purpose modeling language for modeling lumped parameter system, for example in [17], but they have been largely superceded by more modern languages in use today. The following is a brief summary of steps to generate state equations for the circuit in each of the languages.

Bond graph : (1) assign causality to the bond graph; (2) select input and energy state variables; (3) generate an initial set of system equations; (4) simplify initial equations to state-space form [10], giving

$$\frac{d}{dt}\left\{\begin{array}{c} q_C \\ p_L \end{array}\right\} = \left[\begin{array}{cc} 0 & \frac{1}{L} \\ -\frac{1}{C} & -\frac{R}{L} \end{array}\right]\left\{\begin{array}{c} q_C \\ p_L \end{array}\right\} + \left\{\begin{array}{c} 0 \\ V_{AC} \end{array}\right\} \tag{1}$$

where, q_C is the time integral of the current of capacitor, p_L is the time integral of the voltage of inductor. V_{AC} represents the voltage of voltage source.

Linear graph: (1) generate a normal tree from the linear graph; (2) select the primary and secondary variables, system order and state variables; (3) formulate constitutive, continuity and compatibility equations then simplify these equations to state equations [35], giving

$$\frac{d}{dt}\left\{\begin{array}{c} i_L \\ v_C \end{array}\right\} = \left[\begin{array}{cc} -\frac{1}{L} & -\frac{R}{L} \\ 0 & \frac{1}{C} \end{array}\right]\left\{\begin{array}{c} i_L \\ v_C \end{array}\right\} + \left\{\begin{array}{c} \frac{1}{L} \\ 0 \end{array}\right\} V_{AC} \tag{2}$$

where, i_L represents the current of inductor, v_C represents the voltage of capacitor. V_{AC} represents the voltage of voltage source.

Modelica/Simulink: (1) based on the established model in Modelica/Simulink, extract constitutive equations of each component from library; (2) select dynamic variables; (3) express the problem as the smallest possible ODE system [4, 5]. In case of Modelica, we get

$$\left\{\begin{array}{c} der(L.i) \\ der(C.v) \end{array}\right\} = \left[\begin{array}{cc} -\frac{1}{L} & -\frac{R}{L} \\ 0 & \frac{1}{C} \end{array}\right]\left\{\begin{array}{c} L.i \\ C.v \end{array}\right\} + \left\{\begin{array}{c} \frac{1}{L} \\ 0 \end{array}\right\} V_{AC} \tag{3}$$

where, $der(*)$ represents the derivative of $*$ in terms of time. $L.i$ is the current of inductor, $C.v$ is the voltage of capacitor. Similarly, for Simulink, we obtain

$$L\frac{di}{dt} = V_{AC} - Ri - \frac{1}{C}\int_{-\infty}^{t} i(t)dt \tag{4}$$

It is easy to verify that these equations are equivalent in terms of their solutions, but are distinct in appearance due to the different choices of state variables and procedures by which they are generated in each modeling system. For instance, linear graph uses across (effort) and through (flow) variables, but bond graph uses their integral forms as state variables.

2.3 Interoperability of lumped parameter languages and systems

Broadly speaking, interoperability subsumes the problems of exchange and integration of simulation models created in different systems. The latter often manifests itself as the need for co-simulation [36] and/or for executable semantics in system modeling tools. For example, system engineering languages, such as SysML [37] need to integrate with simulation models in order to predict the modelled system behavior; however, each specific tool needs a different interface, as proposed in [38, 39] for Matlab/Simulink [38, 39], in [40, 41] for Modelica, and for bond graph in [42]. An effort to overcome these challenges was recently described in [11], where a SysML extension is proposed specifically for the purpose of generating such interfaces automatically. One of our goals in this effort is to provide formal semantics for this and other interoperability efforts. Without such formal semantics, integration of distinct simulation tools requires a non-trivial software development effort that must resolve individual assumptions and differences of distinct models. The purpose of the recently proposed FMI standard [44] is to streamline and simplify such efforts through generation of uniform C code. Unifying semantics of distinct models would support automatic generation of such FMI interfaces.

Model to model conversion is an effective method for achieving interoperability between different modeling languages. The conversion between the bond graph and linear graph models has been studied in the 70's. Ort and Martens proposed a topological procedure for converting the bond graph to the linear graph by identifying the correspondence of bonds in a bond graph and edges of a linear graph [45]. While theoretically not every bond graph has a corresponding linear graph [9], Birkett gave a deterministic cut-and-paste method for converting any physical bond graph to the corresponding linear graph model [46]. Bond graphs can also be converted to equivalent models in Simulink or Modelica. Specifically, [47] used bond graphs through Simulink to analyze dynamic systems by transforming bond graphs to equivalent block diagrams; researchers in [48, 49] observed that translating non-causal bond graph models to Modelica is in principle a straightforward process, even if causal assignment cannot be specified in Modelica. The reverse conversion from Modelica to bond graphs has been studied in [50]. Strictly speaking, not every Modelica model can be translated into a bond graph, because power continuity (the energy balance) is strictly enforced by the junction structure of bond graph but does not have to be enforced by interconnection of components in Modelica [51]. Similarly, not all Simulink models may be translated into Modelica models, but translation of selective Simulink models to equivalent Modelica models was considered in [52]. Our goal is to identify a subset of physical models that is supported by all of these languages and provide unified semantics that would remove any ambiguities in such conversions and other interoperability tasks.

A fundamental limitation underlying all of the above approaches is that none of them offer a systematic path towards interoperability with full 3D (spatially distributed) models governed by partial differential equations and discretized by finite element, volume, or difference methods. This area of research is in its infancy and exemplified by rather limited efforts. For example, in [53] finite element model of 3D deformable objects are converted to the corresponding lumped mass-spring models by minimizing the difference of stiffness matrices of these two models. Others

proposed various methods for associating geometric information with components in the lumped parameter systems. Thus, integral properties from solid models in CAD systems may be mapped to lumped parameters, and mating constraints in assembly models are mapped to transformers in the corresponding Modelica models [54, 55, 56]. Such methods also allow translating Adams rigid body simulation models to Modelica code [57]. Additional constraints and annotations may be introduced in order to explicitly maintain bilateral correspondence between parameters of lumped parameter and three-dimensional models [58, 59, 60]. By contrast, the semantics of lumped parameter models proposed in this papers naturally extends to full 3D models of spatially distributed phenomena.

3 Algebraic topological models of lumped parameter systems

Algebraic topological interpretations of network models and various types of electrical circuits are well known in literature [16, 14] and are now a standard textbook material [61]. We now apply such an interpretation to lumped parameter models and show that it provides a neutral standard semantics for all such systems. Superficially, all algebraic topological formulations are identical, but important semantic difference emerge in details. As a starting point, we adopt Tonti's classification [14] of physical theories in terms of their algebraic topological models; however, specific requirements of the lumped parameter system require significant extensions and modifications that we discuss below. Strictly speaking, a proper setting for physical modeling is 4D spacetime. But, most engineering models are set in space×time, where space and time models are treated separately.

3.1 Lumped parameter models as cochains

In algebraic topological view of physics, physical properties are distributed in spacetime over finite chunks of space called p-dimensional cells, or p-cells, ($p = 0, 1, 2...$) that fit together to form a cell complex that decomposes the undelying physical space. Many choices of cells are possible; they can be open or closed, p-simplicies, p-balls, or p-manifolds; specific choices are dictated by convenience and applications and define the type of cell complex[62]. All cells are endowed with orientation, or sense of direction, which becomes important in order to properly assign signs to physical properties associated with cells.

As we already saw, all lumped parameter models are formulated using 2-dimensional cells complexes: 0-cells (nodes), 1-cells (edges), and 2-cells (cycles or "meshes"). These complexes are abstract in the sense that geometric coordinates or shapes of the cells are immaterial; only their connectivity carries important physical information.[2] The distribution of physical properties is described by assigning their types and quantities to the individual cells in this complex. The formal mechanism for doing so requires discretizing the property g over p-cells e^p_α as a *p-cochain* C^p, a formal sum

$$C^p = \sum_{\alpha=1}^{n_p} g_\alpha e^p_\alpha \tag{5}$$

The relation between physical properties is governed by two types of fundamental laws: metric laws and topological laws. Metric laws usually involve measurement while topological laws relate

[2]This is in stark contrast to spatially distributed physical phenomena modeled governed by partial differential equation where geometry of cells becomes critical.

physical properties associated with space and its boundary. Topological laws can be formulated using formal linear coboundary δ operations on cochains. Specifically, coboundary δ_p operating on a p-cochain produces a $(p+1)$-cochain by transferring and adding the coefficient of the p-cochain to its cofaces (Eq.6). Formally,

$$\delta_p(C^p) = \delta_p \left(\sum_{\alpha=1}^{n_p} g_\alpha e_\alpha^p \right) = \sum_{\beta=1}^{n_{p+1}} \left(\sum_{\alpha=1}^{n_p} h_{\alpha\beta} \cdot g_\alpha \right) \cdot e_\beta^{p+1}, \tag{6}$$

where n_p represents the number of cells in the p-cochain. The incidence coefficient $h_{\alpha\beta} = [e_\alpha^p, e_\beta^{p+1}] \in \{0, \pm 1\}$ is determined by relative orientation of p-cell e_α^p and it cofaces $(p+1)$-cell e_β^{p+1} [14]. If e_β^{p+1} is not a coface of e_α^p, then $h_{\alpha\beta} = 0$; otherwise, if the orientations of e_α^p and e_β^{p+1} are consistent, then $h_{\alpha\beta} = +1$, otherwise, $h_{\alpha\beta} = -1$. If we denote the usual p-incidence matrix as $\mathbf{A} = [h_{\beta\alpha}]$, then the coboundary operator δ_p is commonly represented by its transpose $\mathbf{A}^T$.

Informally, the coboundary operations capture the essence of balance, equilibrium, conservation, compatibility, and other topological laws. For cochains on finite cell complexes, coboundary operators $\delta_p, p = 1, 2, 3$ correspond to the usual vector calculus operators of gradient, curl, and divergence respectively. The vector calculus identities $\nabla \times (\nabla \phi) = 0$, and $\nabla \cdot (\nabla \times \mathbf{F}) = 0$ are simply instances of the Poincare lemma stating that $\delta_p(\delta_{p-1}()) = \mathbf{0}$, where $\mathbf{0}$ denotes a null cochain. A collection of cochains and coboundary operators on a cell complex form a cochain complex[62].

3.2 Physical theories as Tonti diagrams

Every physical theory is conceptualized in terms of relationships between two types of dual physical quantities that are referred to by various authors as configuration/source[14], through/across[16], or effort/flow[6]. In what follows we will adopt Tonti's convention and distinguish between configuration type variables, that are modeled as cochains on primary cell complex decomposition of space, and source variables that are modeled as cochains on the dual cell complex decomposition of the same space. This notion is illustrated in Figure 7, where the primal cell complex is shown in blue the dual cell complex is shown in black. (The actual geometry of cells is irrelevant for this discussion.) Each p-dimensional cell in the primal cell complex is dual to a unique $(n-p)$-dimensional cell in the dual cell complex. By duality, it also follows that the coboundary operator $\widetilde{\delta}_p$ on the dual cell complex can be represented by the transpose of $\delta_p = \mathbf{A}^T$, or simply $\mathbf{A}$.

This conceptualization of physical quantities in terms cochains on dual cell complexes is not arbitrary: it arises from first principles based on how the postulated quantities are measured. In each case, the measurement process implies the intrinsic dimension of the associated quantity (e.g., displacements are measured at a point, currents are measured across the surface, voltage drop is measured along a path, and so on). The decision whether a particular quantity belongs to the primal or dual complex is determined by the oddness principle that requires change of sign under change of orientation of the relevant cell. The primal cells are are endowed with inner orientation, while the dual cells are oriented relative to the containing (outer) space. The reader is referred to [14] for detailed discussion of these concepts.

Figure 2 shows the correspondence between the primal and dual cochains of variables and naming conventions used by different authors and lumped-parameter modeling systems. We note that the adopted classification in terms of primal and dual cochains is consistent with the conventions in linear graphs [3], Modelica [4], Simscape [5] and NIST models [11], but differs slightly from that in bond graphs [1].

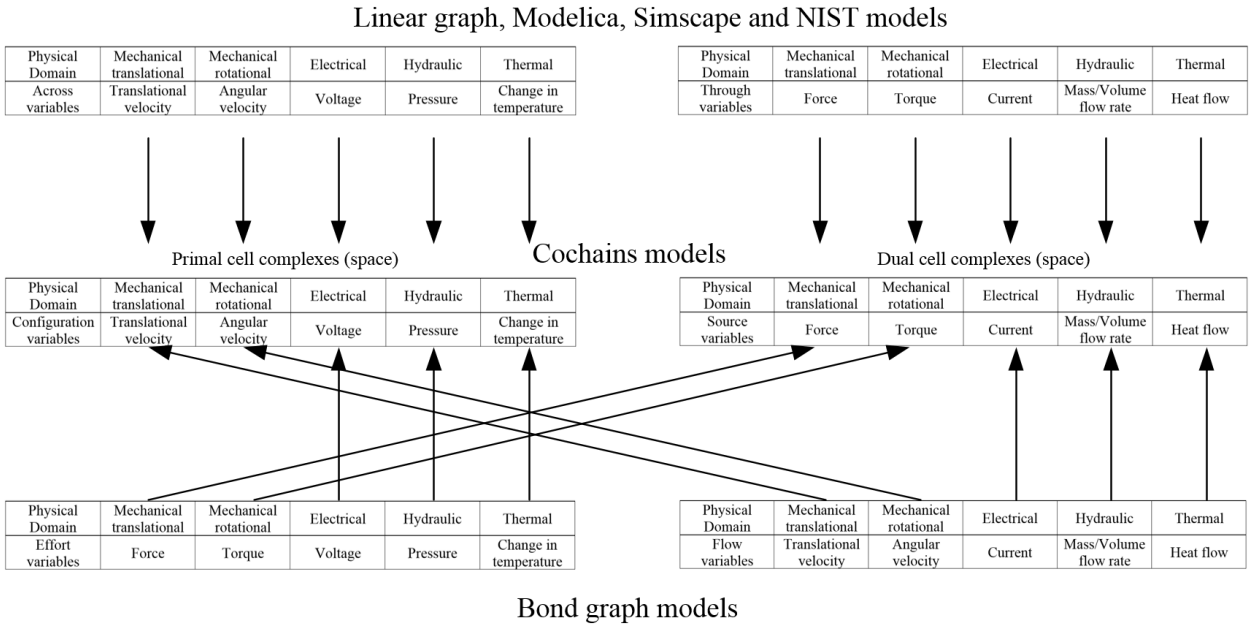


Figure 2: Categorization of physical variables category in different approaches to physical modeling

Physical laws (topological and metric) relate different types of variables within each physical theory. Tonti proposed a systematic method for representing these laws using a diagram that can be considered an evolved combination of the so-called Roth diagrams [63] in terms of cochain sequences and "Maxwell house" diagram to represent all topological and metric relationships in electromagnetism [64]. A simple example of such a diagram is the Tonti diagram of electrical (static) network theory is shown in (Figure 3). It describes the network systems that satisfy Kirchhoff Current Law (KCL) and Kirchhoff Voltage Law (KVL) using a pair of cochains complexes dual to each other [14].

The diagram consists of two vertical sequences corresponding to the primal (left) and dual (right) cochain complexes, ordered by dimension. The vertical arrows correspond to the coboundary operations, going down for primal cochains and going up for dual cochains. Formally, the two sequences are exact[3] and form two dual cochain complexes:

$$\text{primal}: \quad \mathbf{e}^0 \xrightarrow{\delta_0} \mathbf{v}^1 \xrightarrow{\delta_1} \mathbf{0}^2 \tag{7}$$

$$\text{dual}: \quad \mathbf{0}^2 \xleftarrow{\widetilde{\delta}_1} \mathbf{j}^1 \xleftarrow{\widetilde{\delta}_0} \mathbf{i}^0 \tag{8}$$

The measured relationships between dual quantities are represented by the horizontal links in the diagram. In the case of network theory, as shown in Figure 3, the primal variable are node potentials $\mathbf{e}$ associated with 0-cells, voltage drops $\mathbf{v}$ and sources associated with 1-cells, and voltage drops associated with 2-cells (meshes or cycles) that are identically $\mathbf{0}$ as the consequence of KVL. The cochains of adjacent dimensions satisfy topological laws expressed by the corresponding coboundary operations depicted as down-facing vertical arrows. Thus, 1-cochain of voltage drops $\mathbf{v}^1 = \mathbf{A}^T\mathbf{e}^0$ is implied by the coboundary operation δ_0 on 0-cochain of node potentials $\mathbf{e}^0$; and KVL

[3]The cochain sequence is exact if it satisfies $\delta_p \circ \delta_{p-1} = 0$ $(p \geq 1)$

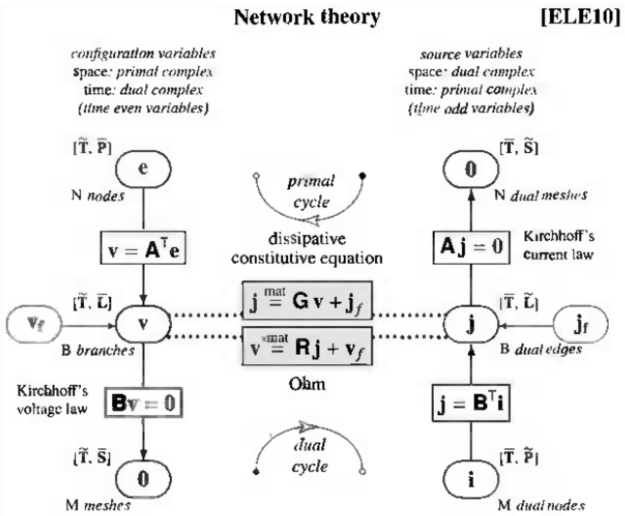


Figure 3: Tonti Diagram of network theory - constitutive equations are modified to account for voltage and current sources

is just a restatement of the Poicare lemma. Similarly, the dual source (current) variables: 0-cochain **i**, 1-cochains **j**, and 2-cochain **0** are indicated in the right branch of the diagram, related by the sequence of two coboundary operations, indicated as arrows going up and expressing KCL.

The constitutive relation between 1-cochain of voltage drops **v** and 1-cochain of currents **j** satisfies Ohm's Law (or its inverse). The diagram also reveals two (dual) methods of generating the governing state equations for network models, depending on the choice of state variables. The two methods are indicated by primal and dual 'cycles' which refer to two different ways of composing topological and metric laws. For instance, in the primal cycle, 1-cochain voltage drop **v** is obtained by coboundary operation on 0-cochain node potential **e**. Using the Ohm's law, **v** is converted to 1-cochain branch currents **j** in the dual cell complex, where coboundary operation **A** on **j** equals zero. The physical meaning of the latter coboundary operation **A** is that the algebraic sum of branch currents of a dual loop equals zero, as required by KCL. Similarly, the dual cycle relies on KVL to generate the dual state equations for the same system.

Even though the diagram in Figure 3 describes a static phenomena, the reader will notice that the configuration and source variables are associated with time instances: primal instances for source variables and dual instances for configuration variables. This distinction becomes critical in dynamic physical models where, once again, the primary time elements (0-dimensional instances and 1-dimensional intervals) are distinguished from the dual time elements based on the oddness principle that requires sign change under reversal of motion [14]. Strictly speaking, a proper setting for all physical models is a 4D spacetime, which we chose to represent as direct product of space and time. In other words, for each type of spatial variable, we can also consider its behavior in time which is represented by a pair of dual 1-dimensional time complexes, as shown in Figure 4. Here 0-cells $\overline{I}_*$ and 1-cells $\overline{T}_*$ represent primal time instances and intervals, while 0-cells $\widetilde{I}_*$ and 1-cells $\widetilde{T}_*$ represent the dual time instances and intervals, respectively.

The introduction of time cell complexes in time has two consequences. First, it identifies the usual time derivative with the corresponding coboundary operator δ^t. Thus, if **q** is a 0-cochain of primal (or dual) time instances, then $\overline{I}$ (respectively, $\widetilde{I}$) is the 1-cochain of time intervals of **q** defined

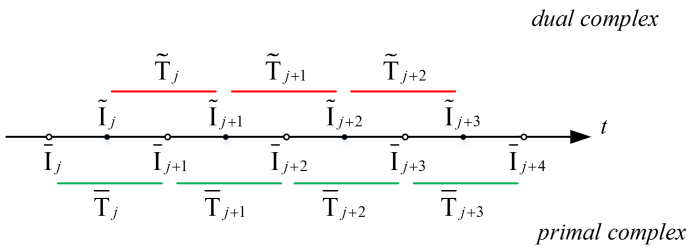


Figure 4: Cell complexes model of time

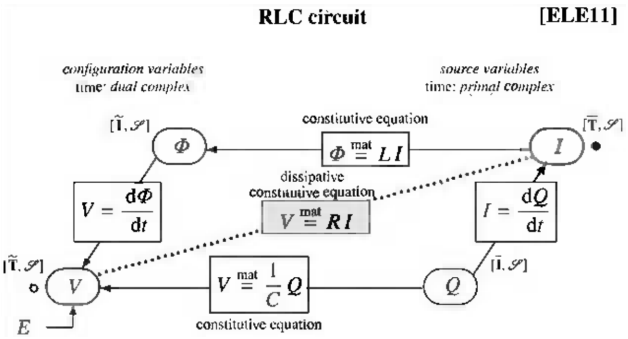


Figure 5: Tonti diagram of RLC circuit - only voltage sources are included

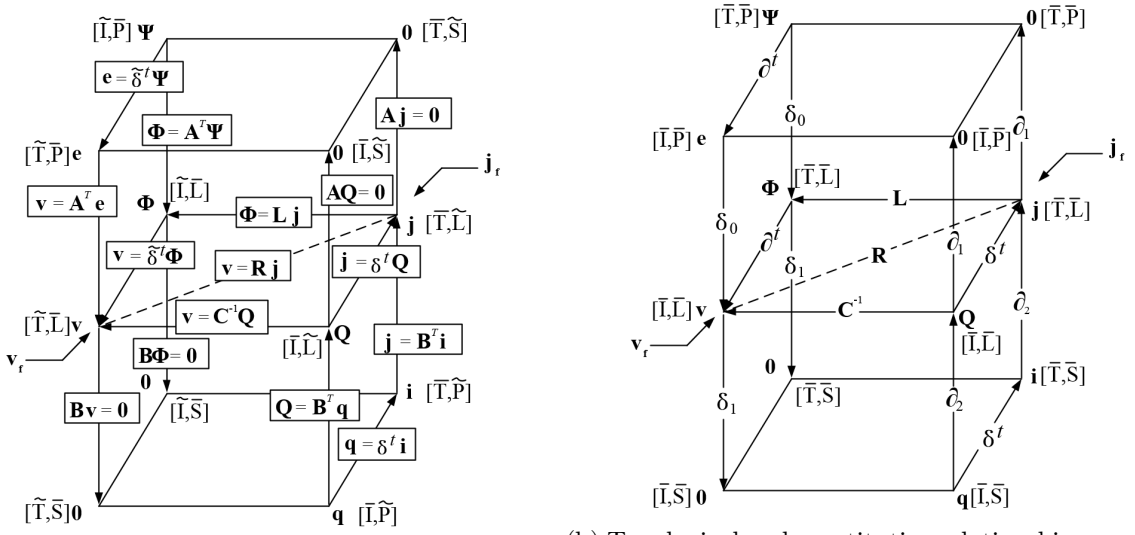
by $\delta_0^t \mathbf{q}$ (respectively $\tilde{\delta}_0^t \mathbf{q}$). On a finite cell complex, δ_0^t become the time difference operation. Secondly, all space Tonti diagrams now acquire an additional time dimension, giving rise to horizontal sections of the diagram. One such section is shown in Figure 5, which corresponds to the Tonti diagram of RLC circuit systems [14]. Here, the usual differential relations $V = d\Phi/dt$ and $I = dQ/dt$ are consequence of topological relations $\mathbf{V} = \tilde{\delta}_0^t \mathbf{\Phi}$ and $\mathbf{I} = \delta_0^t \mathbf{Q}$, respectively, where Φ is magnetic flux and Q is electric charge. Two new constitutive equations describe the capacitance relation between electric charge Q and voltage drop v and the inductance relation between magnetic flow Φ and currents j. The second time derivative is the result of composition of two first time derivatives. For instance, the second order equation differential equation $V = d\Phi/dt = Ld\,(I)/dt = Ld^2Q/dt^2$ in Figure 5 can be expressed as $\mathbf{V} = \tilde{\delta}_0^t \mathbf{\Phi} = \mathbf{L}\tilde{\delta}_0^t \mathbf{I} = \tilde{\delta}_0^t \mathbf{L} \delta_0^t \mathbf{Q}$ [14, 65]. Finally, we note that the vertical space diagram in Figure 3 and the horizontal time diagram in Figure 5 can be combined into a single three-dimensional diagram, as described in Section 4 and shown in Figure 6a.

In addition to physical quantities in Figure 3 and 5, the diagram in Figure 6a includes magnetic flux potential Ψ and mesh electric charge q that are related by coboundary operators to magnetic flux Φ and electric charge Q respectively. In other words, the extended Tonti diagram includes two additional cochain complexes defined by the two sequences:

$$\text{primal}: \quad \mathbf{\Psi}^0 \xrightarrow{\delta_0} \mathbf{\Phi}^1 \xrightarrow{\delta_1} \mathbf{0}^2 \tag{9}$$

$$\text{dual}: \quad \mathbf{0}^2 \xleftarrow{\tilde{\delta}_1} \mathbf{Q}^1 \xleftarrow{\tilde{\delta}_0} \mathbf{q}^0 \tag{10}$$

Additional horizontal arrows indicate the corresponding time coboundary (derivative) operations. With the added constitutive, topological equations and physical variables, the extended Tonti diagram includes all possible quantities, as well as constitutive and topological laws of RLC



(a) Matrix operators on dual cell complexes

(b) Topological and constitutive relationships on a single cell complex

Figure 6: Extended Tonti diagram for RLC network system - with voltage and current sources

electrical circuit systems. In the following section, we explicitly distinguish between the constitutive and topological laws as they are commonly used in practice in a single cell complex network model that is shown in Figure 6b.

3.3 Dual cochain complexes on a single cell complex

The ultimate goal of physical modeling is numerical simulation which is usually performed on a discretization of spacetime. If one were to accept that every physical theory is formulated in terms of dual cochain complexes, it would be reasonable to expect that most modeling and simulation tools are also formulated in terms of dual discretizations (one for configuration variables and another one for the source variable) as illustrated in Figure 7 (on the left). In fact, such dual discretizations are often advocated in literature as more natural and numerically stable alternatives, for example, in mimetic discretization schemes [66], cell methods [67, 65], discrete exterior calculus [68] and other modeling approaches. However, the vast majority of numerical schemes appear to be based on a single discretization of space, which supports evaluation of both primal and dual cochains. For example, in most finite element, finite difference, and finite volume methods, all configuration and source variables are associated with cells (often nodes) in the underlying mesh, and their duality is hidden within the numerical scheme itself [69].

In lumped parameter models, dual discretizations are particularly counter-intuitive, since all spatially distributed properties have already been integrated (lumped) and only connectivity of the underlying cell complex remains visible. That connectivity often directly corresponds to the physical embedding; for example, a single electrical network carries both voltage and current information. Similarly, general network model is a single cell complex where primal and dual cochains are represented. The mapping of dual cochains on the primal cell complex is straightforward and is accomplished by mapping the dual $(n-p)$-cells to their corresponding primal p-cells, as shown in Figure 7. Consider how this mapping would work for the cochains in Figure 3. With this mapping,

node potentials $\mathbf{e}$, voltage drops $\mathbf{v}$ and sources, and voltage drops of cycles $\mathbf{0}$ are still associated with primal 0-cells (e.g. A), 1-cells (e.g. L), and 2-cells (e.g. M) respectively. However, the dual source (current) variables: mesh currents $\mathbf{i}$, branch currents $\mathbf{j}$, and currents merging at nodes $\mathbf{0}$ which were originally associated with dual 0-cells (e.g. a), 1-cells (e.g. l), and 2-cells (e.g. m) are now associated with primal 2-cells (e.g. M), 1-cells (e.g. L), and 0-cells (e.g. A) respectively.

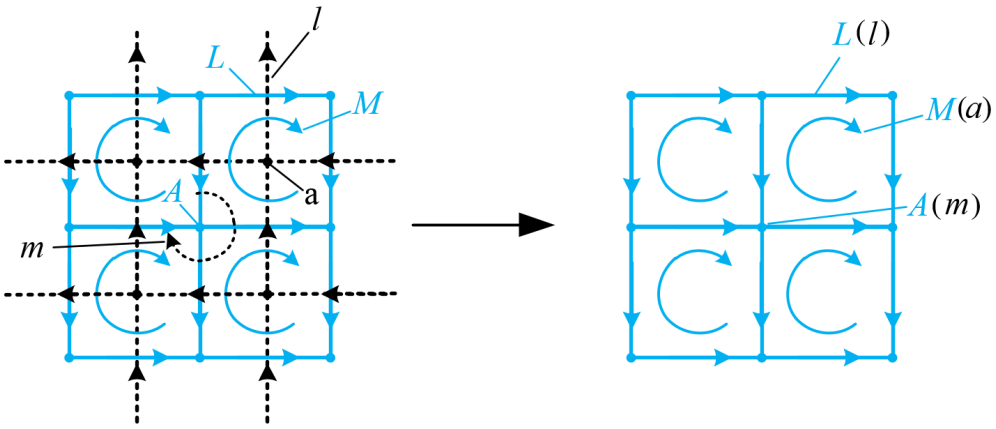


Figure 7: Dual cochain complexes on a single cell complex

With this mapping, all $(n-p)$-coboundary operations in dual cell complexes would become the p-boundary operations ∂ in the primal cochain complex, which operate on a p-cochains and produce a $(p-1)$-cochains:

$$\partial_p(C^p) = \partial_p(\sum_{\alpha=1}^{n_p} g_\alpha e_\alpha^p) = \sum_{\beta=1}^{n_{p-1}} \left(\sum_{\alpha=1}^{n_p} h_{\alpha\beta} \cdot g_\alpha \right) \cdot e_\beta^{p-1}, \tag{11}$$

which is similar to Eq.6, except that the coefficients are transferred from p-cells to their $(p-1)$-faces. This implies that the coboundary operation $\widetilde{\delta}_{n-p}$ on the dual cells complex is mapped into the boundary operation ∂_p on the primal cell complex,[4] i.e.,

$$\partial_p = \widetilde{\delta}_{n-p} \tag{12}$$

This explains, for example, why the KCL on the primal complex is described by condition $\partial_1 \mathbf{j}^1 = \mathbf{0}$, stating that the branch currents must add up to zero at every node.

The single complex network model is summarized by a new type of Tonti diagram shown in Figure 6b. Here the dual (source) cochains have been mapped to the corresponding cochains on the primal cell complex to form the dual cochain complex with boundary ∂_p operators replacing the original dual $\widetilde{\delta}_{n-p}$ coboundary operators. In other words, the dual cochain sequences (8) and (10) become respectively

$$\mathbf{i}^2 \xrightarrow{\partial_2} \mathbf{j}^1 \xrightarrow{\partial_1} \mathbf{0}^0 \tag{13}$$

$$\mathbf{q}^2 \xrightarrow{\partial_2} \mathbf{Q}^1 \xrightarrow{\partial_1} \mathbf{0}^0 \tag{14}$$

[4]This observation also justifies use of ∂_p operator, usually reserved for p-chains, on p-cochains

In the rest of the report, we will assume that an algebraic topological model of a lumped parameter system is described by such single (primal) cell complex and a corresponding Tonti diagram with four cochain complexes (correspong to the four vertical 'legs' of the diagram) and all relationships between them.[5] The two primal cochain complex include: (1) 0-cochain node potentials $\mathbf{e}^0$, 1-cochain voltage drops $\mathbf{v}^1$, 2-cochain of mesh voltages (which is $\mathbf{0}$ by KVL); and (2) 0-cochain magnetic flux potentials $\mathbf{\Psi}^0$, 1-cochain magnetic fluxes $\mathbf{\Phi}^1$, and 2-cochain of mesh magnetic fluxes and 2-cochain of mesh voltages (which is $\mathbf{0}$ by KVL). The cochains in each complex are related by the spatial coboundary operators, while the corresponding p-cochains in the two complexes are related by the time boundary operator. Similarly, the two dual cochain complexes are: (1) 0-cochain of node currents (which is $\mathbf{0}$ by KCL), 1-cochain currents $\mathbf{j}^1$, 2-cochain mesh currents $\mathbf{i}^2$; and (2) 0-cochain of node electric charges, 1-cochain electric charges $\mathbf{Q}^1$, 2-cochain mesh electric charges $\mathbf{q}^2$. The cochains in each of the dual complexes related by the spatial boundary operators, while the corresponding p-chains in the two dual complexes are related by the time coboundary operators.

4 Single-domain lumped parameter systems

In this section, we will show how to use Tonti diagrams to describe lumped parameter systems and introduce an automated method of generating system state equations. We will focus on lumped parameter models of a single physical domain, exemplified by classical RLC electrical circuit systems. Application to other physical domains is immediate, since all such models are isomorphic.

4.1 Static systems

Classical single-domain RLC electrical circuits consist of five types of physical elements: resistors, capacitors, inductors, voltage sources and current sources. We will first consider a special case of static resistive circuits that contain only constant resistors, as well as voltage and current sources; later we will extend the approach to general dynamic electrical circuits. The algebraic topological structure of static electrical circuits relies only on the two dual cochain complexes modeled over a single cell complex. The primal cochain complex includes 0-cochain of node potentials $\mathbf{e}^0$, 1-cochain of voltage drops $\mathbf{v}^1$, and 2-cochain of mesh voltages (which is $\mathbf{0}$ by KVL), related by the coboboundry operators; the dual cochain complex consists of 0-cochain of node currents (which is $\mathbf{0}$ by KCL), 1-cochain of branch currents $\mathbf{j}^1$, and 2-cochain mesh currents $\mathbf{i}^2$ related by the boundary operators. The topological and constitutive relations between these cochains are given by the diagram in Figure 3, and two methods of generating the equations are indicated by primal and dual 'cycles' respectively in the diagram. In the context of the more general model, these cycles correspond to red and blue paths in the extended Tonti diagram as shown in (Figure 8). Each path is defined by a sequence of the arrows in the diagram indicating composition of the corresponding physical laws.

For example, if we use the red path in Figure 8, then the 0-cochain $\mathbf{e}^0$ is selected as the state variable. The system state equation can be generated by composition of three steps starting with a 0-cochain $\mathbf{e}^0$ of node potentials. First, the downward red arrow indicates that the node potentials give rise to voltage drops associated with incident branches using the coboundary operator $\mathbf{v}^1 =$

[5] Strictly speaking, use ∂_p operators give rise to chain complexes, but it should be clear that these chain complexes on the primal cell complex are isomorphic to the cochain complexes over the dual cell complex.

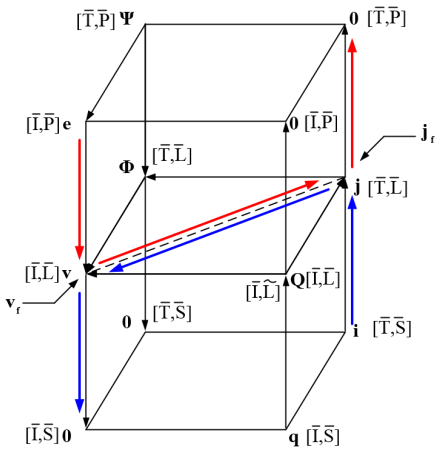


Figure 8: Paths corresponding to primal and dual cycle in Figure 3

$\delta_0\mathbf{e}^0$. The second lateral red arrow correspond to the constitutive relation $\mathbf{j}^1 = \mathbf{G}\left(\mathbf{v}^1 + \mathbf{v}_\mathrm{f}^1\right) + \mathbf{j}_\mathrm{f}^1$, which accounts for the contribution of 1-cochains of voltage sources $\mathbf{v}_\mathrm{f}^1$ and source currents $\mathbf{j}_\mathrm{f}^1$ in each branch of the electrical network. Finally, the upward vertical red arrow indicates the application of KCL $\partial_1\mathbf{j}^1 = \mathbf{0}$ requiring that branch currents add up to zero. The composition of the three laws yields:

$$\partial_1\left(\mathbf{G}\left(\delta_0\mathbf{e}^0 + \mathbf{v}_\mathrm{f}^1\right) + \mathbf{j}_\mathrm{f}^1\right) = \mathbf{0} \tag{15}$$

Eq.15 is usually written in a more traditional form as

$$\partial_1\left(\mathbf{G}\left(-\delta_0\mathbf{e}^0 - \mathbf{v}_\mathrm{f}^1\right) + \mathbf{j}_\mathrm{f}^1\right) = \mathbf{0} \tag{16}$$

As explained in [14], the minus sign in front of $\delta_0\mathbf{e}^0$ is due to the largely historical assumption that the vertices (0-cells) of the primal cell complex are oriented as sinks. The second minus sign in front of $\mathbf{v}_\mathrm{f}^1$ signifies the fact that a (positive) voltage source should be subtracted from the (positive) voltage drop in every branch.

An alternative method for generating the system state equations follows the dual blue path in Figure 8. The process starts with the dual 2-cochain of mesh currents $\mathbf{i}^2$ selected as the state variable and amounts to composition of the analogous three steps: boundary operator ∂_2 applied to mesh currents in order to generate branch currents, constitutive Ohm's law $\mathbf{R}$ that relates the branch currents to voltage drops, and coboundary operator δ_1 applied to the voltage drops in accordance with KVL. Taking into account the voltage and current sources, the process results in:

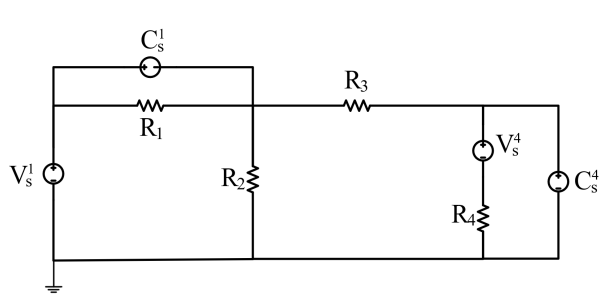
$$\delta_1\left(\mathbf{R}(\partial_2\mathbf{i}^2 - \mathbf{j}_\mathrm{f}^1) + \mathbf{v}_\mathrm{f}^1\right) = \mathbf{0} \tag{17}$$

Collecting the terms with known voltage and current sources and moving them to the right hand side, the system state equations Eq.16 and Eq.17 transform to Eq.18 and Eq.19, respectively.

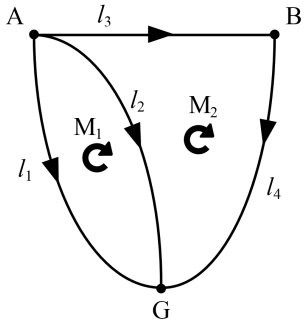
$$\partial_1\mathbf{G}\delta_0\mathbf{e}^0 = \partial_1\left(\mathbf{j}_\mathrm{f}^1 - \mathbf{G}\mathbf{v}_\mathrm{f}^1\right) \tag{18}$$

$$\delta_1\mathbf{R}\partial_2\mathbf{i}^2 = \delta_1\left(-\mathbf{v}_\mathrm{f}^1 + \mathbf{R}\mathbf{j}_\mathrm{f}^1\right) \tag{19}$$

When boundary and cobounadry operators are replaced by the corresponding incidence matrices describing a cell complex underlying a specific lumped parameter system, the state equations become systems of linear equations that can be solved for the unknown state variables (node potentials $\mathbf{e^0}$ and mesh currents $\mathbf{i^2}$ respectively).



(a) An electrical circuit of pure resistance



(b) Topological structure

Figure 9: An electrical circuit of pure resistance and its topological structure

Example 4.1 *We will use an example of static electrical circuit in Figure 9a to illustrate the derivation of Eq.18 and Eq.19 in concrete setting. The shown electrical network contains four constant resistors* $R_1 = 6\Omega$, $R_2 = 3\Omega$, $R_3 = 2\Omega$, $R_4 = 4\Omega$, *two constant current sources* $C_s^1 = 3A$, $C_s^4 = 1A$ *and two constant voltage sources* $V_s^1 = 8V$, $V_s^4 = 4V$. *Topologically, the network is a 2-dimensional complex shown in Figure 9b and consisting of three 0-cells (A,B,G), four 1-cells*(l_1,l_2,l_3,l_4) *and two 2-cells*(M_1,M_2).

The algebraic topological model of the electrical circuit contains: primal 0-cochain node potentials ($\mathbf{e}^0 = e_A \cdot A + e_B \cdot B + e_G \cdot G$), *primal 1-cochain voltage drops* ($\mathbf{v}^1 = V_1 \cdot l_1 + V_2 \cdot l_2 + V_3 \cdot l_3 + V_4 \cdot l_4$), *dual 1-cochain of branch currents* ($\mathbf{j}^1 = j_1 \cdot l_1 + j_2 \cdot l_2 + j_3 \cdot l_3 + j_4 \cdot l_4$), *dual 2-cochain of mesh currents* ($\mathbf{i}^2 = i_1 \cdot M_1 + i_2 \cdot M_2$) *and two cochains that are always* $\mathbf{0}$: *the dual 0-cochain of node currents and the primal 2-cochain of loop voltage drops. In order to get unique solution of the state equations, we consider 0-cell G as the reference node, implying the boundary condition of* $e_G = 0$. *Following the red path on the Tonti diagram generates equation Eq.18 with individual terms as follows:*

$$\partial_1 = \begin{bmatrix} -1 & -1 & -1 & 0 \\ 0 & 0 & +1 & -1 \\ +1 & +1 & 0 & +1 \end{bmatrix} \tag{20}$$

$$\delta_0 = \partial_1^T \tag{21}$$

$$\mathbf{G} = \begin{bmatrix} R_1^{-1} & 0 & 0 & 0 \\ 0 & R_2^{-1} & 0 & 0 \\ 0 & 0 & R_3^{-1} & 0 \\ 0 & 0 & 0 & R_4^{-1} \end{bmatrix} = \begin{bmatrix} 1/6 & 0 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/4 \end{bmatrix} \tag{22}$$

$$\mathbf{j}_{\mathrm{f}}^1 = \begin{bmatrix} C_s^1 & 0 & 0 & -C_s^4 \end{bmatrix}^T = \begin{bmatrix} 3 & 0 & 0 & -1 \end{bmatrix}^T \tag{23}$$

$$\mathbf{v}_{\mathrm{f}}^1 = \begin{bmatrix} V_s^1 & 0 & 0 & V_s^4 \end{bmatrix}^T = \begin{bmatrix} 8 & 0 & 0 & 4 \end{bmatrix}^T \tag{24}$$

Substituting Eq.20 ∼ Eq.24 into Eq.18 we obtain a linear system of equations that has the solution of $\mathbf{e}^0 = [-1/2,\ 7/3,\ 0]^T$. *Once the 0-cochain node potentials* $\mathbf{e}^0$ *is known, it is easy to obtain 1-cochain voltage drops* $\mathbf{v}^1$ *and 1-cochain branch currents* $\mathbf{j}^1$ *by using the topological equation* $\mathbf{v}^1 = -\left(\delta_0 \mathbf{e}^0\right)$ *and the constitutive equation* $\mathbf{j}^1 - \mathbf{j}_{\mathrm{f}}^1 = \mathbf{G}\left(\mathbf{v}^1 - \mathbf{v}_{\mathrm{f}}^1\right)$.

Similarly following the blue path in the Tonti diagram, the generated system state equation Eq.19 would involve:

$$\delta_1 = \begin{bmatrix} -1 & +1 & 0 & 0 \\ 0 & -1 & +1 & +1 \end{bmatrix} \tag{25}$$

$$\partial_2 = {\delta_1}^T \tag{26}$$

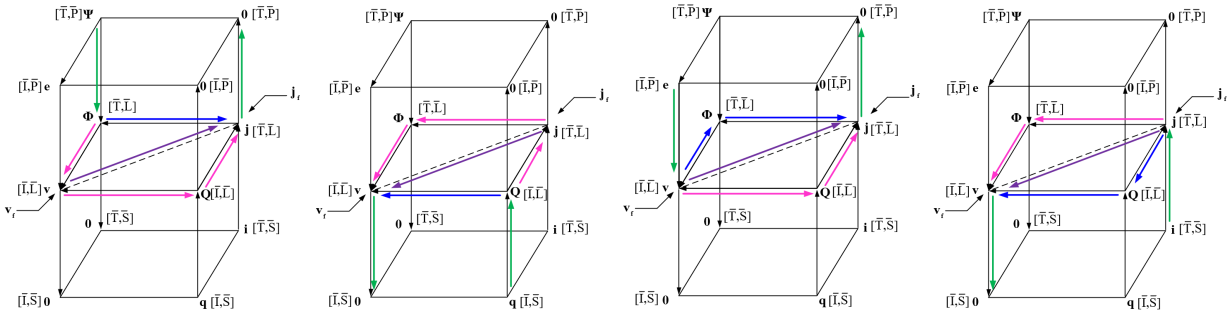
$$\mathbf{R} = \begin{bmatrix} R_1 & 0 & 0 & 0 \\ 0 & R_2 & 0 & 0 \\ 0 & 0 & R_3 & 0 \\ 0 & 0 & 0 & R_4 \end{bmatrix} = \begin{bmatrix} 6 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix} \tag{27}$$

Substituting Eq.23 ∼ Eq.27 into Eq.19, we obtain another system of linear equations that yields the solutions for the mesh currents $\mathbf{i}^2 = \begin{bmatrix} -19/12 & -17/12 \end{bmatrix}^T$. *The 1-cochain branch currents* $\mathbf{j}^1$ *and 1-cochain voltage drops* $\mathbf{v}^1$ *are immediately obtained by applying the topological relation* $\mathbf{j}^1 = \partial_2 \mathbf{i}^2$ *and the constitutive equation* $\mathbf{v}^1 - \mathbf{v}_{\mathrm{f}}^1 = \mathbf{R}\left(\mathbf{j}^1 - \mathbf{j}_{\mathrm{f}}^1\right)$.

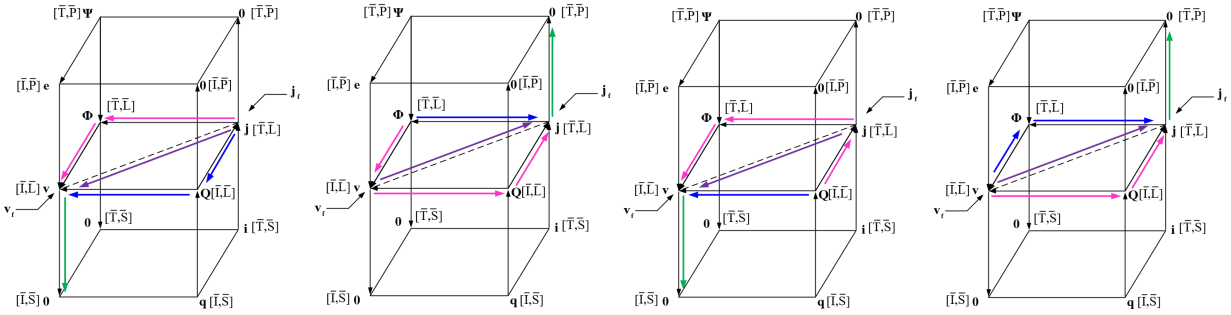
4.2 Dynamic systems

In this section, we will extend the above approach to general dynamic electrical circuits. The algebraic topological structure of dynamic electrical circuits relies on cochains from all four cochain complexes modeled over a single cell complex. The topological and constitutive relations between these cochains are given by the diagram in Figure 6b and eight different methods of generating the state equations are indicated by paths in the diagrams shown in Figure 10. Just as with static systems, each path is a sequence of the arrows indicating composition of the corresponding physical laws. In contrast to static systems, the middle horizontal section of the diagram allows three alternative (pink, purple and blue) paths relating the primal 1-cochain of voltage drops $\mathbf{v}^1$ and the dual 1-cochain of branch currents $\mathbf{j}^1$ corresponding to capacitance, resistance, and inductance constitutive relationship respectively. The presence of alternative paths indicate superposition of the corresponding equations generated by each path.

For example, if we use the paths in Figure 10a to generate the state equations, then the 0-cochain $\mathbf{\Psi}^0$ is selected as the state variable. The system state equation can be generated by composition of five physical laws (two topological and three constitutive) starting with a 0-cochain $\mathbf{\Psi}^0$. First, coboundary operator in space δ_0 applied to potential magnetic fluxes $\mathbf{\Psi}^0$ in order to generate magnetic fluxes $\mathbf{\Phi}^1$. Now the path splits in two: the blue arrow corresponds to constitutive law $\mathbf{L^{-1}}$ (of the inductor) that relates the magnetic fluxes to branch currents of inductors; the pink arrow takes the magnetic fluxes to generate voltage drops $\mathbf{v}$ via the boundary operator in time ∂_1^t. From here the path splits in two again: the purple arrow corresponds to the constitutive Ohm's law $\mathbf{G}$ that relates the voltage drops to branch currents of resistors; continuing along the pink path,



(a) State variable: $\mathbf{\Psi}^0$ (b) State variable: $\mathbf{q}^2$ (c) State variable: $\mathbf{e}^0$ (d) State variable: $\mathbf{i}^2$



(e) State variable: $\mathbf{j}^1$ (f) State variable: $\mathbf{\Phi}^1$ (g) State variable: $\mathbf{Q}^1$ (h) State variable: $\mathbf{v}^1$

Figure 10: State equation generation paths on the extended Tonti diagram

the constitutive capacitance law $\mathbf{C}$ relates the voltage drops to electric charges $\mathbf{Q}^1$ of capacitors is followed by the time coboundary operator δ_0^t applied to electric charges of capacitors to generate branch currents of capacitors. Note that the three paths corresponding to the three constitutive laws merge into a single 1-cochain of branch currents $\mathbf{j}^1$, which is then transformed one more time by the upward green arrow corresponding to KCL $\partial_1 \mathbf{j}^1 = \mathbf{0}$. Taking into account the voltage and current sources, above processes results in following state equations:

$$\partial_1 \left(\delta_0^t \mathbf{C} \left(-\partial_1^t \left(\delta_0 \mathbf{\Psi}^0 \right) - \mathbf{v}_\mathrm{f}^1 \right) + \mathbf{R}^{-1} \left(-\partial_1^t \left(\delta_0 \mathbf{\Psi}^0 \right) - \mathbf{v}_\mathrm{f}^1 \right) + \mathbf{L}^{-1} \left(- \left(\delta_0 \mathbf{\Psi}^0 \right) - \delta_0^t \mathbf{v}_\mathrm{f}^1 \right) + \mathbf{j}_\mathrm{f}^1 \right) = \mathbf{0} \quad (28)$$

Collecting the terms with known voltage and current sources and moving them to the right hand side, the system state equations Eq.28 can be written in a more intuitive form as

$$\partial_1 \left(\underbrace{\delta_0^t \mathbf{C} \partial_1^t \delta_0 \mathbf{\Psi}^0}_{\text{currents of C}} + \underbrace{\mathbf{R}^{-1} \partial_1^t \delta_0 \mathbf{\Psi}^0}_{\text{currents of R}} + \underbrace{\mathbf{L}^{-1} \delta_0 \mathbf{\Psi}^0}_{\text{currents of L}} \right) = \partial_1 \left[\underbrace{\mathbf{j}_\mathbf{f}^1}_{\text{current sources}} \quad - \underbrace{\left(\delta_0^t \mathbf{C} + \mathbf{R}^{-1} + \mathbf{L}^{-1} \delta_0^t \right) \mathbf{v}_\mathbf{f}^1}_{\substack{\text{equivalent current sources} \\ \text{generated from} \\ \text{voltage sources}}} \right] \quad (29)$$

Other methods for generating the system state equation follow the different paths in Figure 10b - Figure 10h. For example, in Figure 10b, the process starts with the dual 2-cochain of mesh

charge $\mathbf{q}^2$ selected as the state variable and amounts to another composition of the five physical laws indicated by the corresponding paths. The blue, purple, and pink path corresponds to the three constitutive laws (capacitance, resistance, and inductance respectively), relating the branch electric charges to the branch voltage drops. The two green arrows correspond to the ∂_2 operator transforming mesh charges $\mathbf{q}^2$ to branch charges $\mathbf{Q}^1$ and application of KVL ($\delta_1 \mathbf{v}^1 = \mathbf{0}$). Putting it all together and taking into account the voltage and current sources, the composition procedures results in:

$$\delta_1 \left(\partial_1^t \mathbf{L} \left(\delta_0^t \partial_2 \mathbf{q}^2 - \mathbf{j}_\mathrm{f}^1 \right) + \mathbf{R} \left(\delta_0^t \partial_2 \mathbf{q}^2 - \mathbf{j}_\mathrm{f}^1 \right) + \mathbf{C}^{-1} \left(\partial_2 \mathbf{q}^2 - \partial_1^t \mathbf{j}_\mathrm{f}^1 \right) + \mathbf{v}_\mathrm{f}^1 \right) = \mathbf{0} \tag{30}$$

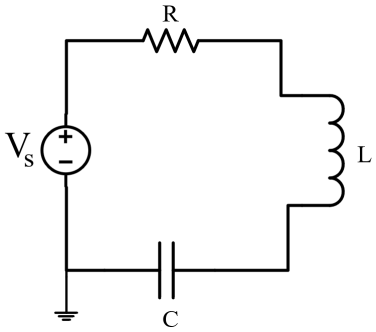
or in a more intuitive form similar to Eq.29:

$$\delta_1 \left(\underbrace{\partial_1^t \mathbf{L} \delta_0^t \partial_2 \mathbf{q}^2}_{\text{voltage drops of L}} + \underbrace{\mathbf{R} \delta_0^t \partial_2 \mathbf{q}^2}_{\text{voltage drops of R}} + \underbrace{\mathbf{C}^{-1} \partial_2 \mathbf{q}^2}_{\text{voltage drops of C}} \right) = \delta_1 \left[\underbrace{-\mathbf{v}_\mathbf{f}^1}_{\text{voltage sources}} + \underbrace{\left(\partial_1^t \mathbf{L} + \mathbf{R} + \mathbf{C}^{-1} \partial_1^t \right) \mathbf{j}_\mathbf{f}^1}_{\substack{\text{equivalent voltage drops} \\ \text{generated from} \\ \text{current sources}}} \right] \tag{31}$$

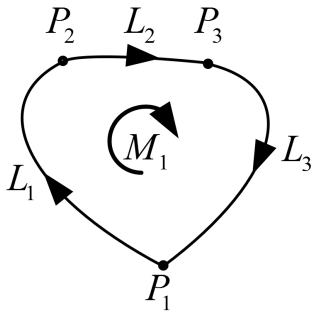
It can be observed that Eq.29 and Eq.31 respectively represent the current and voltage equilibrium of the system. The generated system state equations may be viewed as algebraic with coboundary operators interpreted as finite difference operators on a finite cell complex. However, as we already observed, in lumped parameter systems space and time are treated separately, and discretization of time is often delayed until a particular numerical integration scheme is chosen. In this case, viewing boundary ∂_1^t and coboundary δ_0^t operations as differentiation in time syntactically transforms Eq.29 and Eq.31 to a more familiar form:

$$\partial_1 \left(\underbrace{\mathbf{C} \delta_0 \ddot{\mathbf{\Psi}}^0}_{\text{currents of C}} + \underbrace{\mathbf{R}^{-1} \delta_0 \dot{\mathbf{\Psi}}^0}_{\text{currents of R}} + \underbrace{\mathbf{L}^{-1} \delta_0 \mathbf{\Psi}^0}_{\text{currents of L}} \right) = \partial_1 \left[\underbrace{\mathbf{j}_\mathbf{f}^1}_{\text{current sources}} - \underbrace{\left(\mathbf{C} \dot{\mathbf{v}}_\mathbf{f}^1 + \mathbf{R}^{-1} \mathbf{v}_\mathbf{f}^1 + \mathbf{L}^{-1} \int \mathbf{v}_\mathbf{f}^1 dt \right)}_{\substack{\text{equivalent current sources} \\ \text{generated from} \\ \text{voltage sources}}} \right] \tag{32}$$

$$\delta_1 \left(\underbrace{\mathbf{L} \partial_2 \ddot{\mathbf{q}}^2}_{\text{voltage drops of C}} + \underbrace{\mathbf{R} \partial_2 \dot{\mathbf{q}}^2}_{\text{voltage drops of R}} + \underbrace{\mathbf{C}^{-1} \partial_2 \mathbf{q}^2}_{\text{voltage drops of L}} \right) = \delta_1 \left(\underbrace{-\mathbf{v}_\mathbf{f}^1}_{\text{voltage sources}} + \underbrace{\mathbf{L} \dot{\mathbf{j}}_\mathbf{f}^1 + \mathbf{R} \mathbf{j}_\mathbf{f}^1 + \mathbf{C}^{-1} \int \mathbf{j}_\mathbf{f}^1 dt}_{\substack{\text{equivalent voltage drops} \\ \text{generated from} \\ \text{current sources}}} \right) \tag{33}$$



(a) A simple RLC electrical circuit



(b) Topological structure

Figure 11: A simple RLC electrical circuit and its topological structure

Example 4.2 *We will use a very simple example of RLC electrical circuit (Figure 1a) to illustrate the derivation of Eq.33. A copy of Figure 1a is shown in Figure 11a. The shown electrical network contains one resistor R, one capacitor C, one inductor L and one voltage source V_s. Topologically, the network is a 2-dimensional cell complex shown in Figure 11b and consists of three 0-cells (P_1,P_2,P_3), three 1-cells (L_1,L_2,L_3) and one 2-cell (M_1).*

The algebraic topological model of the electrical circuit contains: primal 0-cochain node potentials ($\mathbf{e}^0 = e_1 \cdot P_1 + e_2 \cdot P_2 + e_3 \cdot P_3$), primal 0-cochain magnetic flux potentials ($\mathbf{\Psi}^0 = \psi_1 \cdot P_1 + \psi_2 \cdot P_2 + \psi_3 \cdot P_3$), primal 1-cochain voltage drops ($\mathbf{v}^1 = v_1 \cdot L_1 + v_2 \cdot L_2 + v_3 \cdot L_3$), primal 1-cochain magnetic fluxes ($\mathbf{\Phi}^1 = \phi_1 \cdot L_1 + \phi_2 \cdot L_2 + \phi_3 \cdot L_3$), dual 1-cochain currents ($\mathbf{j}^1 = j_1 \cdot L_1 + j_2 \cdot L_2 + j_3 \cdot L_3$), dual 1-cochain electric charges ($\mathbf{Q}^1 = Q_1 \cdot L_1 + Q_2 \cdot L_2 + Q_3 \cdot L_3$), dual 2-cochain mesh currents ($\mathbf{i}^2 = i \cdot M_1$), dual 2-cochain mesh electric charges ($\mathbf{q}^2 = q \cdot M_1$), and four cochains that are always $\mathbf{0}$*: 2-cochain loop voltage drops, 2-cochain loop magnetic fluxes, 0-cochain node currents and 0-cochain of node electric charges. Following the paths in Figure 10b generates Eq.33, with individual terms as follows:*

$$\delta_1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \tag{34}$$

$$\partial_2 = {\delta_1}^T \tag{35}$$

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{36}$$

$$\mathbf{R} = \begin{bmatrix} R & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{37}$$

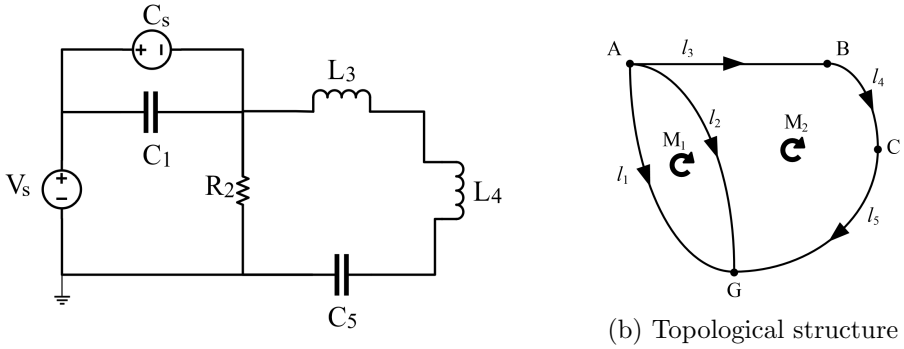
$$\mathbf{C}^{-1} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & C^{-1} \end{bmatrix} \tag{38}$$

$$\mathbf{v}^1_{\mathbf{f}} = \begin{bmatrix} -V_s & 0 & 0 \end{bmatrix}^T \tag{39}$$

$$\mathbf{j}_\mathbf{f}^1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \tag{40}$$

Substituting Eq.34 ∼ Eq.40 into Eq.33, we obtain the system state equations as shown in Eq.41 . Since there is only one 2-cell, the mesh current equals the branch current, so Eq.41 is equivalent to the system state equation generated by the Simulink model shown in Eq.4.

$$L\ddot{q} + R\dot{q} + C^{-1}q = V_s \tag{41}$$



(a) An RLC electrical circuit

(b) Topological structure

Figure 12: An RLC electrical circuit and its topological structure

Example 4.3 *We will use another example of electrical circuit in Figure 12a to illustrate the derivation of Eq.32 . The shown electrical circuit contains one resistor R_2, two capacitors C_1, C_5, two inductors L_3, L_4, one current source C_s and one voltage source V_s. Topologically, the network is a 2-dimensional cell complex shown in Figure 12b, and consisting of four 0-cells (A,B,C,G), five 1-cells(l_1,l_2,l_3,l_4,l_5) and two 2-cells(M_1,M_2).*

The algebraic topological model of the electrical circuit contains: primal 0-cochain node potentials ($\mathbf{e}^0 = e_A \cdot A + e_B \cdot B + e_C \cdot C + e_G \cdot G$), primal 0-cochain magnetic flux potentials ($\mathbf{\Psi}^0 = \psi_A \cdot A + \psi_B \cdot B + \psi_C \cdot C + \psi_G \cdot G$), primal 1-cochain voltage drops ($\mathbf{v}^1 = v_1 \cdot l_1 + v_2 \cdot l_2 + v_3 \cdot l_3 + v_4 \cdot l_4 + v_5 \cdot l_5$), primal 1-cochain magnetic fluxes ($\mathbf{\Phi}^1 = \phi_1 \cdot l_1 + \phi_2 \cdot l_2 + \phi_3 \cdot l_3$), dual 1-cochain currents ($\mathbf{j}^1 = j_1 \cdot l_1 + j_2 \cdot l_2 + j_3 \cdot l_3 + j_4 \cdot l_4 + j_5 \cdot l_5$), dual 1-cochain electric charges ($\mathbf{Q}^1 = Q_1 \cdot l_1 + Q_2 \cdot l_2 + Q_3 \cdot l_3 + Q_4 \cdot l_4 + Q_5 \cdot l_5$), dual 2-cochain mesh currents ($\mathbf{i}^2 = i_1 \cdot M_1 + i_2 \cdot M_2$), dual 2-cochain mesh electric charges ($\mathbf{q}^2 = q_1 \cdot M_1 + q_2 \cdot M_2$), and four cochains that are always $\mathbf{0}$: 2-cochain loop voltage drops, 2-cochain loop magnetic fluxes, 0-cochain node currents and 0-cochain of node electric charges. In order to get unique solution of the state equations, we consider 0-cell G as the reference node, implying the boundary condition of $e_G = 0$. Following the paths in Figure 10a, generates Eq.32, with individual terms as follows:

$$\partial_1 = \begin{bmatrix} +1 & +1 & 0 & 0 & +1 \\ -1 & -1 & -1 & 0 & 0 \\ 0 & 0 & +1 & -1 & 0 \\ 0 & 0 & 0 & +1 & -1 \end{bmatrix} \tag{42}$$

$$\delta_0 = \partial_1^T \tag{43}$$

$$\mathbf{C} = \begin{bmatrix} C_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_5 \end{bmatrix} \tag{44}$$

$$\mathbf{R}^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1/R_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{45}$$

$$\mathbf{L}^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1/L_3 & 0 & 0 \\ 0 & 0 & 0 & 1/L_4 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{46}$$

$$\mathbf{v}_\mathbf{f}^1 = \begin{bmatrix} V_s & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{47}$$

$$\mathbf{j}_\mathbf{f}^1 = \begin{bmatrix} C_s & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{48}$$

Substituting Eq.42 $\sim$ Eq.48 into Eq.32, we obtain the system state equations as follows:

$$\begin{bmatrix} C_1+C_5 & -C_1 & 0 & -C_5 \\ -C_1 & C_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -C_5 & 0 & 0 & C_5 \end{bmatrix} \ddot{\mathbf{\Psi}}^0 + \begin{bmatrix} 1/R_2 & -1/R_2 & 0 & 0 \\ -1/R_2 & 1/R_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \dot{\mathbf{\Psi}}^0 + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1/L_3 & -1/L_3 & 0 \\ 0 & -1/L_3 & 1/L_3+1/L_4 & -1/L_4 \\ 0 & 0 & -1/L_4 & 1/L_4 \end{bmatrix} \mathbf{\Psi}^0 = \begin{bmatrix} C_s - \dot{V}_s C_1 \\ -C_s + \dot{V}_s C_1 \\ 0 \\ 0 \end{bmatrix} \tag{49}$$

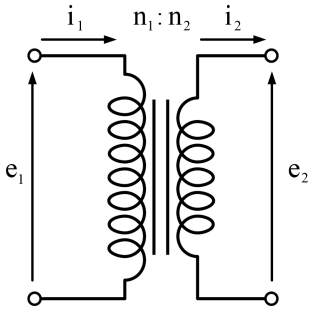
5 Multi-domain lumped parameter systems

5.1 Interactions of single-domain models

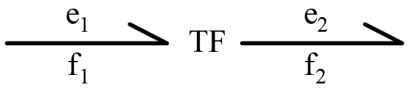
Engineering systems are usually constructed as compositions of single-domain subsystems in order to perform complex engineering tasks. Representative examples include electric motors (electro-mechanical systems), ovens (electro-thermal systems), hydraulic pumps (hydraulic-mechanical systems), etc. We will refer to such systems as multi-domain systems, where lumped-parameter behavior of each single-domain is governed by an extended Tonti diagram as described in the previous section. It should be understood that multi-domain systems subsume the special case of homogeneous systems where all single-domain systems are of the same type, e.g. all electrical or all mechanical. Composition of two single-domain systems is associated with the process of energy conversion between the two systems, also called *transduction* [10]. Specific means of transduction

vary from system to system, but common examples of transducers (devices the perform transduction) include gears and levers, electrical transformers, motors, piezoelectric devices, hydraulic pumps, and so on.

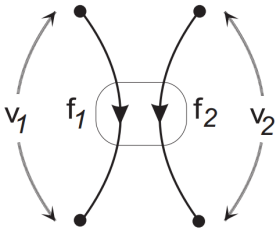
Devices used to couple the same type variables of different physical domains are usually called transformers (e.g. electrical transformers), while devices coupling the dual type variables are called gyrators (e.g. electric motors) [10]. Physical transformers and gyrators always have energy loss during the energy transduction, but if the leakage is small enough then the energy loss is usually neglected during modeling. Transformers and gyrators with no energy loss are called ideal transformers and gyrators [10]. System modeling languages generally use ideal transformers and gyrators as abstract constructs in order to avoid modeling complex structures of physical transformers and gyrators devices. For instance, bond graphs use abstract 2-port transformers and gyrators which can be used to connect any 1-ports [10]. Figure 13 and Figure 14 show several transformer and gyrator models from different system modeling languages.



(a) A schematic ideal electrical transformer



(b) Bond graph model of transformers



(c) Linear graph model of transformers

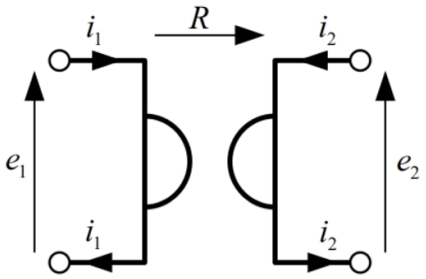
```
model IdealTransformer "Ideal electrical transformer"
  extends Modelica.Electrical.Analog.Interfaces.TwoPort;
  parameter Real n=1 "Turns ratio";
equation
  v1 = n*v2;
  i2 = -n*i1;
end IdealTransformer;
```

(d) Modelica model of the ideal electrical transformer

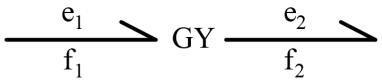
Figure 13: Comparison of transformers of different modeling languages

Such abstract transformers and gyrators in each language can represent hundreds of complex transformer and gyrator devices. For example, Figures 15 and 16 show two examples of multi-domain models represented by different modeling languages. Figure 15 is an example of a homogeneous multi-domain system, where all sub-domains are electrical domains. Figure 16 is an example of a heterogeneous multi-domain system, where one electrical domain and one mechanical domain are connected by a DC motor.

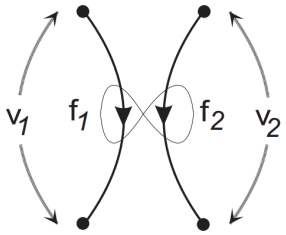
In our proposed combinatorial model of lumped parameter systems, physical transducer devices can be abstracted by additional relations between primal and dual variables in each of the subsystems; such relations may be governed by additional constitutive, interaction, or conservation constrains imposed on the multi-domain system [71]. For example, consider the three-domain



(a) a schematic ideal electrical gyrator



(b) Bond graph model of gyrators



(c) Linear graph model of gyrators

```
model IdealGyrator "Ideal gyrator"
  extends Modelica.Electrical.Analog.Interfaces.TwoPort;
  parameter SIunits.Conductance G=1 "Gyration conductance";
equation
  i1 = G*v2;
  i2 = -G*v1;
end IdealGyrator;
```

(d) Modelica model of the ideal gyrator

Figure 14: Comparison of gyrators of different modeling languages

system consisting of a hydraulic pump that is controlled by an electrical motor. Behaviors of the subsystems (electrical, rotational motion, and hydraulic) are described by the corresponding extended Tonti diagrams) that are further constrained as shown in Figure 17.

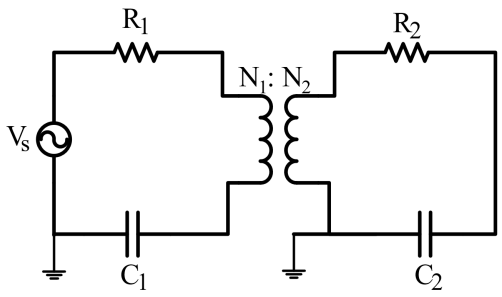
In principle, such a representation is sufficient for capturing the behavior of a multi-domain system. Each Tonti diagram corresponds to a system of ordinary differential equations that are coupled by the transducer constraints. When the constraints are algebraic, this representation corresponds to the usual system of differential-algebraic equations. More complex transducer relationships may involve multiple physical variables as well as non-linear and differential constraints [72], resulting in more complex models of behaviors.

However, this representation neither recognizes nor takes advantage of the fact that all single-domain behaviors are isomorphic, which allows to treat the whole multi-domain system as a collection of four constrained cochain complexes on a *single* cell complex model. Below we will define such a model, which takes a form of a generalized Tonti diagram. We then show that, in the presence of two most common transducers: ideal transformers and gyrators, the governing equations for such a model may be generated by following the paths on the generalized Tonti diagram. This results extends the result of Section 3 to multi-domain lumped parameter systems.

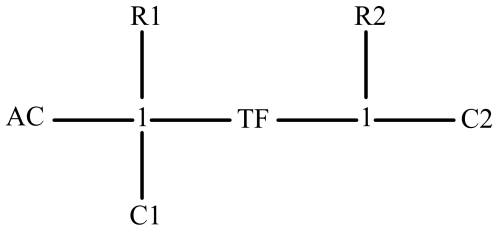
5.2 Generalized Tonti diagram for multi-domain systems

Since lumped parameter models in different physical domain are isomorphic, so are their corresponding Tonti diagrams. In this sense, a single Tonti diagram describes behavior of all lumped parameter systems, provided that the variable of the same space-time type are identified and generalized.

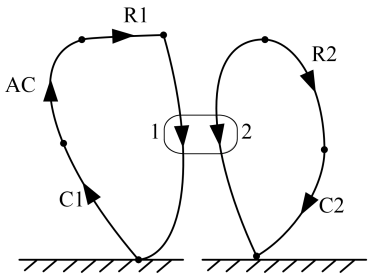
Two most common generalizations are mechanical (generalized displacement-force model) and electrical (generalized voltage-current model). For the sake of consistency with the discussion



(a) A multi-domain RLC electrical circuit



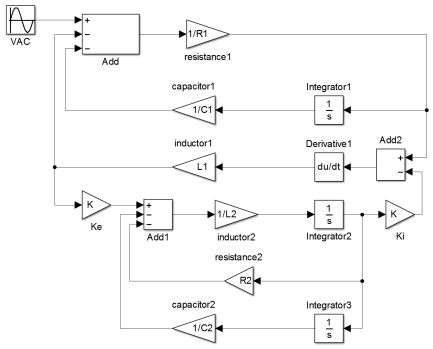
(b) Bond graph model



(c) Linear graph model

```
model homogeneous
  Modelica.Electrical.Analog.Basic.Resistor R1(R = 5);
  Modelica.Electrical.Analog.Sources.SineVoltage AC(V = 5, freqHz = 10);
  Modelica.Electrical.Analog.Basic.Capacitor C1(C = 0.01);
  Modelica.Electrical.Analog.Basic.Transformer T(L1 = 0.2, L2 = 0.1, M = 0);
  Modelica.Electrical.Analog.Basic.Resistor R2(R = 5);
  Modelica.Electrical.Analog.Basic.Capacitor C2(C = 0.001);
  Modelica.Electrical.Analog.Basic.Ground ground1;
  Modelica.Electrical.Analog.Basic.Ground ground2;
equation
  connect(ground1.p, AC.n);
  connect(ground2.p, C2.n);
  connect(C2.p, T.n2);
  connect(R2.n, C2.n);
  connect(R2.p, T.p2);
  connect(R1.n, T.p1);
  connect(AC.p, R1.p);
  connect(C1.p, AC.n);
  connect(T.n1, C1.n);
end homogeneous;
```

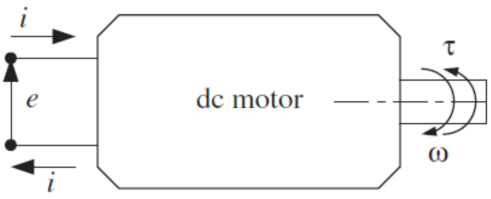
(d) Modelica model



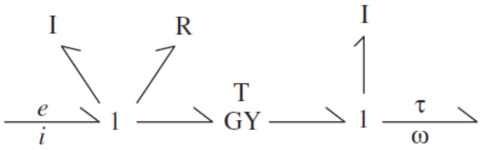
(e) Simulink model

Figure 15: Different models of a multi-domain RLC electrical circuit

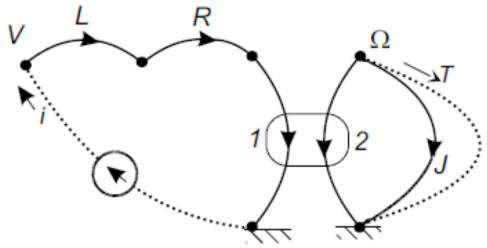
in section 4, we will adopt the generalized electrical model. For example, the electrical voltage, mechanical translational velocity and hydraulic pressure difference are all considered to be of the same type called the generalized voltage; the electrical resistors, mechanical dampers and hydraulic resistors are all identified as generalized resistors, and so on. In order to emphasize the generalized nature of all physical quantities and to distinguish them from the actual physical electrical network model, we will choose a different set of symbols. Specifically, the generalized Tonti diagram is defined by four exact cochain sequences on a single cell complex:



(a) A schematic DC motor



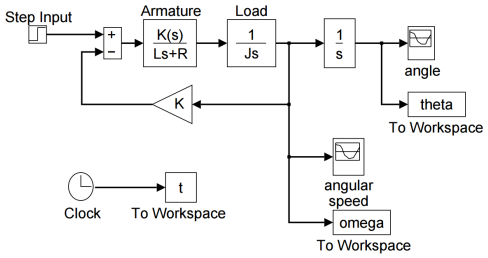
(b) Bond graph model [10]



(c) Linear graph model [3]

```
model DC_motor
  Modelica.Electrical.Analog.Sources.ConstantVoltage constantVoltage1;
  Modelica.Electrical.Analog.Basic.Inductor inductor1;
  Modelica.Electrical.Analog.Basic.Resistor resistor1;
  Modelica.Electrical.Analog.Basic.EMF emf;
  Modelica.Mechanics.Rotational.Components.Inertia inertia1;
  Modelica.Electrical.Analog.Basic.Ground ground1;
equation
  connect(emf.flange, inertia1.flange_a);
  connect(emf.n, constantVoltage1.n);
  connect(resistor1.n, emf.p);
  connect(inductor1.n, resistor1.p);
  connect(constantVoltage1.p, inductor1.p);
  connect(ground1.p, constantVoltage1.n);
end DC_motor;
```

(d) Modelica model



(e) Simulink model [70]

Figure 16: Different models of a DC motor

$$\text{primal}: \quad \mathbf{p}^0 \xrightarrow{\delta_0} \mathbf{a}^1 \xrightarrow{\delta_1} \mathbf{0}^2 \tag{50}$$

$$\mathbf{d}^0 \xrightarrow{\delta_0} \mathbf{u}^1 \xrightarrow{\delta_1} \mathbf{0}^2 \tag{51}$$

$$\text{dual}: \quad \mathbf{s}^2 \xrightarrow{\partial_2} \mathbf{t}^1 \xrightarrow{\partial_1} \mathbf{0}^0 \tag{52}$$

$$\mathbf{n}^2 \xrightarrow{\partial_2} \mathbf{m}^1 \xrightarrow{\partial_1} \mathbf{0}^0, \tag{53}$$

where $\mathbf{p}^0$ is a 0-cochain generalized potentials, $\mathbf{a}^1$ is a 1-cochain generalized voltages, $\mathbf{d}^0$ is a 0-cochain generalized potential magnetic fluxes, $\mathbf{u}^1$ is a 1-cochain generalized magnetic fluxes, $\mathbf{s}^2$ is a 2-cochain generalized mesh currents, $\mathbf{t}^1$ is a 1-cochain generalized currents, $\mathbf{n}^2$ is 2-cochain generalized mesh charges, and $\mathbf{m}^1$ is a 1-cochain generalized electric charges. There are also four cochains that are always $\mathbf{0}$: 2-cochain of generalized mesh magnetic fluxes, 2-cochain of generalized mesh voltages, 0-cochain of generalized node currents and 0-cochain of generalized node electric charges.

With such a generalization, all the physical variables of the same space-time type are replaced by their generalized counterparts, effectively transforming model of the heterogeneous multi-domain

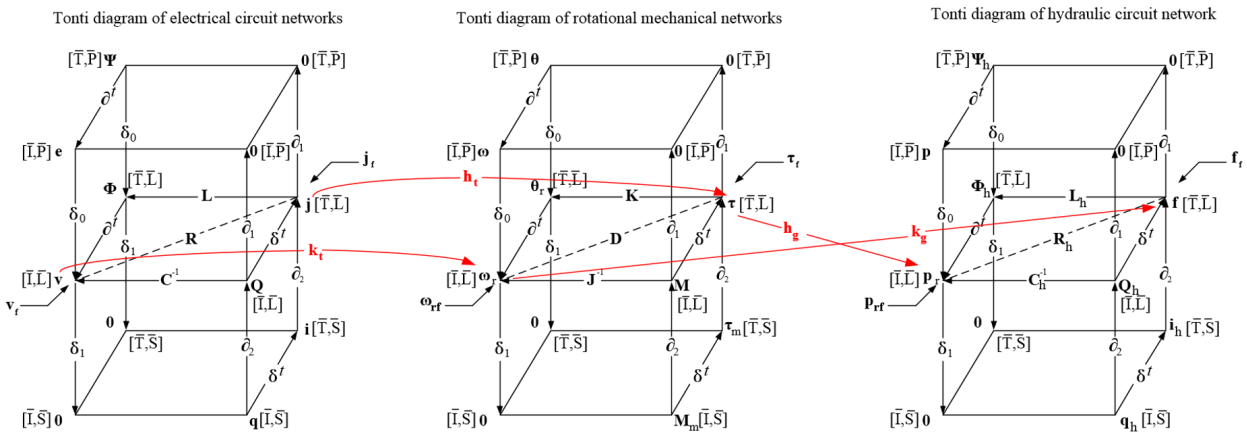


Figure 17: Behavior of three-domain system, a hydraulic pump driven by electrical motor, is abstracted by relations between three interacting extended Tonti diagrams.

Commonly-used physical variables in the Tonti diagram of rotational mechanical networks: $\boldsymbol{\theta}$ - angle, $\boldsymbol{\theta}_r$ - relative angle, $\boldsymbol{\omega}$ - rotational velocity, $\boldsymbol{\omega}_r$ - relative rotational velocity, $\boldsymbol{\tau}$ - torque, $\mathbf{M}$ - angular momentum, $\mathbf{K}$ - rotational stiffness, $\mathbf{D}$ - rotational damping coefficient and $\mathbf{J}$ - moment of inertia. Commonly-used physical variables in the Tonti diagram of hydraulic networks: $\boldsymbol{\Psi}_h$ - potential of hydraulic flux, $\boldsymbol{\Phi}_h$ - hydraulic flux, $\mathbf{p}$ - pressure, $\mathbf{p}_r$ - pressure drop, $\mathbf{Q}_h$ - flow of volume, $\mathbf{f}$ - flow rate, $\mathbf{L}_h$ - hydraulic inductance, $\mathbf{R}_h$ - hydraulic resistance, $\mathbf{C}_h$ - hydraulic capacitance. $\mathbf{h}_t$ and $\mathbf{k}_t$ are transformer modulus. $\mathbf{k}_t$ is the ratio of relative rotational velocity of motor to voltage drop and $\mathbf{h}_t$ is the ratio of torque to current. $\mathbf{h}_g$ and $\mathbf{k}_g$ are gyrator modulus. $\mathbf{k}_g$ is the ratio of flow rate to relative rotational velocity of motor and $\mathbf{h}_g$ is the ratio of pressure drop to torque.

system in an abstract (generalized) homogeneous system. The behavior of this system is governed by the generalized Tonti diagram shown in Figure 18. As before, the generalized primal and dual cochains are related by (generalized) constitutive relations: resistance $\mathbf{R}_g$, capacitance $\mathbf{C}_g$, and inductance $\mathbf{L}_g$.

Furthermore, since all physical quantities are generalized, the actions of ideal transformers and gyrators can be modeled simply as additional constraints on the cochains in the generalized Tonti diagram. Traditionally, a transformer is abstracted as a linear transformation

$$\begin{bmatrix} a_1 \\ t_1 \end{bmatrix} = \begin{bmatrix} k_t & 0 \\ 0 & 1/k_t \end{bmatrix} \begin{bmatrix} a_2 \\ t_2 \end{bmatrix} \tag{54}$$

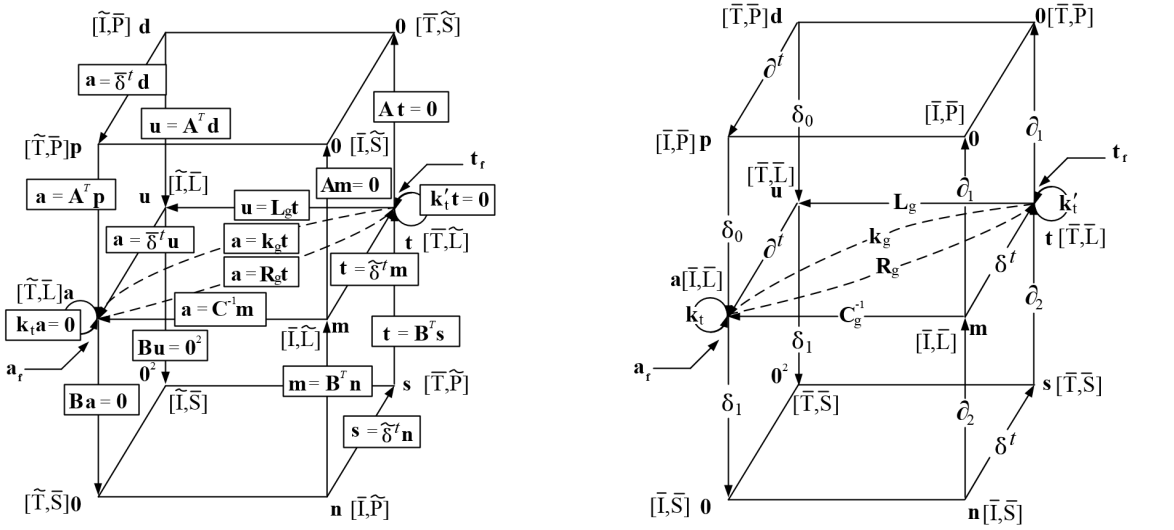
where k_t is the transformer's modulus measuring the ratio between two (generalized) voltages a_1 and a_2, as well as the reciprocal ratio between the generalized currents t_1 and t_2 in order to enforce energy balance $a_1 t_1 = a_2 t_2$. It is easy to see that the same relationships may be enforced by a pair of linear constraints

$$\begin{cases} \begin{bmatrix} 1 & -k_t \end{bmatrix} \cdot \begin{bmatrix} a_1 & a_2 \end{bmatrix}^T = 0 \\ \begin{bmatrix} 1 & -k_t^{-1} \end{bmatrix} \cdot \begin{bmatrix} t_1 & t_2 \end{bmatrix}^T = 0 \end{cases} \tag{55}$$

Generalizing, every ideal transformer can be represented by a pair of linear constraints

$$\mathbf{k}_t \mathbf{a} = 0; \qquad \mathbf{k}_t' \mathbf{t} = 0$$

on cochains of generalized voltages $\mathbf{a}$ and currents $\mathbf{t}$. These constraints are indicated on the generalized Tonti diagram in Figure 18 by two cycles.



(a) Matrix operators on dual cell complexes

(b) Topological operators on a single cell complex

Figure 18: Generalized extended Tonti diagram for generalized RLC network system

Similarly, the effect of an abstract gyrator is usually described by a linear transformation

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 & k_g \\ k_g & 0 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} \tag{56}$$

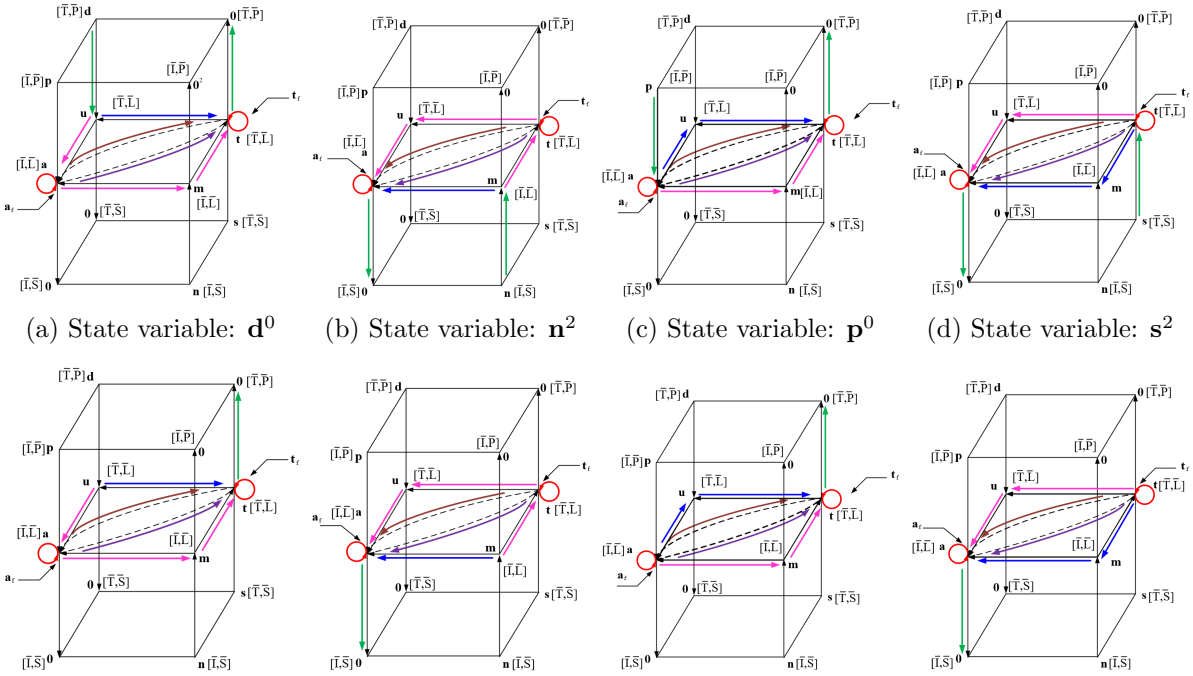
where the modulus k_g relates the dual quantities in two interacting domains: generalized voltage a_1 of the first domain is proportional to the generalized current t_2 of the second domain, and vice versa, again satisfying the ideal energy balance law. Equivalently, a generalized gyrator may be represented by a linear transformation $\mathbf{k}_g$ that relates the cochains of generalized voltages and currents, as indicated by a dotted arrow in the Tonti diagram in Figure 18.

5.3 System state equations of multi-domain systems

With all physical variables generalized, the heterogeneous multi-domain system now becomes a homogeneous multi-domain system in terms of generalized physical variables. Instead of multiple 2-cochain complexes associated with different types of physical variables, the algebraic topological model of the multi-domain system is now a set of 2-cochain complexes associated with the same (generalized) type of physical variables that are defined over a single cell complex and are constrained by abstract transformers and gyrators.

Eight different methods of generating the state equations are indicated by paths in the generalized Tonti diagram shown in Figure 19. Just as with the single domain systems, each path is a sequence of the arrows indicating composition of the corresponding physical laws. In contrast to the single domain diagram, the middle horizontal section of the generalized Tonti diagram allows an additional alternative path relating the primal 1-cochain of generalized voltages $\mathbf{a}^1$ and the dual 1-cochain of generalized currents $\mathbf{t}^1$ by gyrators as well as two alternative cyclic paths (shown in red), which respectively constrain the generalized voltages and currents of transformers.

For example, if we use the paths in Figure 19a to generate the state equations, then the 0-cochain $\mathbf{d}^0$ is selected as the state variable. The system state equation can be generated by composition



(a) State variable: $\mathbf{d}^0$ (b) State variable: $\mathbf{n}^2$ (c) State variable: $\mathbf{p}^0$ (d) State variable: $\mathbf{s}^2$

(e) State variable: $\mathbf{u}^1$ (f) State variable: $\mathbf{m}^1$ (g) State variable: $\mathbf{a}^1$ (h) State variable: $\mathbf{t}^1$

Figure 19: State equation generation paths on the generalized extended Tonti diagram

of seven physical laws (two topological and five constitutive) starting with a 0-cochain $\mathbf{d}^0$. First, coboundary operator in space δ_0 applied to generalized potential magnetic fluxes $\mathbf{d}^0$ in order to generate generalized magnetic fluxes $\mathbf{u}^1$. Now the path splits into two: the blue arrow corresponds to constitutive law $\mathbf{L}_g^{-1}$ that relates generalized magnetic fluxes to generalized currents of generalized inductors; the pink arrow takes the generalized magnetic fluxes to generate generalized voltages $\mathbf{a}^1$ via the boundary operator in time ∂_1^t. From here the path splits into four: (1) the purple arrow corresponds to the constitutive law $\mathbf{R}_g^{-1}$ that relates generalized voltages to generalized currents of generalized resistors; (2) the brown arrow corresponds to the constitutive law $\mathbf{k}_g^{-1}$ that relates generalized voltages to generalized currents of gyrators; (3) the left red cyclic arrow corresponds to constitutive law $\mathbf{k}_t$ that constrains the generalized voltages of generalized transformers; (4) the pink arrow takes the generalized voltages to generate generalized electric charge of generalized capacitors by using constitutive law $\mathbf{C}_g$, followed by taking the generalized electric charge to generate generalized currents of generalized capacitors via the coboundary operator in time δ_0^t. Note that these four paths and the right red cyclic path[6] corresponding to five constitutive laws merge into a single 1-cochain of generalized through variables $\mathbf{t}^1$, which is then transformed one more time by the upward green arrow corresponding to KCL $\partial_1 \mathbf{t}^1 = \mathbf{0}$. Taking into account the generalized sources, collecting the terms with known generalized sources and moving them to the right hand side, above procedure results in the state equation Eq.57, with Eq.58 and Eq.59 being the constraint equations generated from two cyclic red paths.

[6]Note that the generalized currents of transformers $\mathbf{t}_T^1$ cannot be directly obtained from the generalized voltages of transformers $\mathbf{a}_T^1$, so we treat them as additional unknown variables in the system state equation.

$$\partial_1\left(\underbrace{\delta_0^t\mathbf{C}_g\partial_1^t\delta_0\mathbf{d}^0}_{\substack{\text{generalized currents}\\ \text{of generalized C}}}+\underbrace{\mathbf{R}_g^{-1}\partial_1^t\delta_0\mathbf{d}^0}_{\substack{\text{generalized currents}\\ \text{of generalized R}}}+\underbrace{\mathbf{L}_g^{-1}\delta_0\mathbf{d}^0}_{\substack{\text{generalized currents}\\ \text{of generalized L}}}+\underbrace{\mathbf{k}_g^{-1}\partial_1^t\delta_0\mathbf{d}^0}_{\substack{\text{generalized currents}\\ \text{of gyrators}}}+\underbrace{\mathbf{t}_T^1}_{\substack{\text{generalized currents}\\ \text{of transformers}}}\right)$$
$$=\partial_1\left[\underbrace{\mathbf{t}_\mathbf{f}^1}_{\substack{\text{generalized currents}\\ \text{sources}}}\quad\underbrace{-\left(\delta_0^t\mathbf{C}_g+\mathbf{R}_g^{-1}+\mathbf{k}_g^{-1}+\mathbf{L}_g^{-1}\delta_0^t\right)\mathbf{a}_\mathbf{f}^1}_{\substack{\text{equivalent generalized current}\\ \text{sources generated from}\\ \text{generalized voltage sources}}}\right] \tag{57}$$

$$\mathbf{k}_t\left(\partial_1^t\delta_0\mathbf{d}^0-\mathbf{a}_\mathrm{f}^1\right)=\mathbf{0} \tag{58}$$

$$\mathbf{k}_t'\left(\mathbf{t}^1-\mathbf{t}_\mathrm{f}^1\right)=\mathbf{0} \tag{59}$$

Assuming that the number of state variables is N and the number of transformers is M, then the system of equations Eq.57 has *N+2M* unknowns. The two transformer's constraints generate *2M* constraint equations, while the other paths generate N state equations. As expected, the number of unknowns equals to the total number of state and constraint equations.

Other methods for generating the system state equation follow the different paths in Figure 19b $\sim$ Figure 19h. For example, in Figure 19b, the process starts with the dual 2-cochain of generalized mesh electric charges $\mathbf{n}^2$ selected as the state variable and amounts to another composition of the seven physical laws indicated by the corresponding paths. The blue, purple, pink and brown path corresponds to the four generalized constitutive laws (generalized capacitance, resistance, inductance and gyrator), relating the generalized currents to generalized voltages. The two green arrows correspond to the ∂_2 operator transforming $\mathbf{n}^2$ to generalized electric charges $\mathbf{m}^1$ and application of KVL ($\delta_1\mathbf{a}^1=\mathbf{0}$). The two red cyclic arrows correspond to the constitutive equations of transformers. Putting it all together and taking into account the generalized sources, the composition procedures results in Eq.60 , with Eq.61 and Eq.62 being the constraint equations generated from two cyclic red paths.

$$\delta_1\left(\underbrace{\partial_1^t\mathbf{L}_g\delta_0^t\partial_2\mathbf{n}^2}_{\substack{\text{generalized voltages}\\ \text{of generalized L}}}+\underbrace{\mathbf{R}_g\delta_0^t\partial_2\mathbf{n}^2}_{\substack{\text{generalized voltages}\\ \text{of generalized R}}}+\underbrace{\mathbf{C}_g^{-1}\partial_2\mathbf{n}^2}_{\substack{\text{generalized voltages}\\ \text{of generalized C}}}+\underbrace{\mathbf{k}_g\delta_0^t\partial_2\mathbf{n}^2}_{\substack{\text{generalized voltages}\\ \text{of gyrators}}}+\underbrace{\mathbf{a}_T^1}_{\substack{\text{generalized voltages}\\ \text{of transformers}}}\right)$$
$$=\delta_1\left[\underbrace{-\mathbf{a}_\mathbf{f}^1}_{\substack{\text{generalized voltages}\\ \text{sources}}}\quad\underbrace{+\left(\partial_1^t\mathbf{L}_g+\mathbf{R}_g+\mathbf{k}_g+\mathbf{C}_g^{-1}\partial_1^t\right)\mathbf{t}_\mathbf{f}^1}_{\substack{\text{equivalent generalized voltage}\\ \text{sources generated from}\\ \text{generalized current sources}}}\right] \tag{60}$$

$$\mathbf{k}_t'\left(\delta_0^t\partial_2\mathbf{n}^2-\mathbf{t}_\mathrm{f}^1\right)=\mathbf{0} \tag{61}$$

$$\mathbf{k}_\mathrm{t}\left(\mathbf{a}^1-\mathbf{a}_\mathrm{f}^1\right)=\mathbf{0} \tag{62}$$

As with any dynamic system, interpreting boundary ∂_1^t and coboundary δ_0^t operations as differentiation in time syntactically transforms Eq. 57 $\sim$ Eq.59 and Eq.60 $\sim$ Eq.62 to a more familiar form:

$$\begin{aligned}
&\partial_1 \left(\underbrace{\mathbf{C}_g \delta_0 \ddot{\mathbf{d}}^0}_{\substack{\text{generalized currents}\\ \text{of generalized C}}} + \underbrace{\mathbf{R}_g^{-1} \delta_0 \dot{\mathbf{d}}^0}_{\substack{\text{generalized currents}\\ \text{of generalized R}}} + \underbrace{\mathbf{L}_g^{-1} \delta_0 \mathbf{d}^0}_{\substack{\text{generalized currents}\\ \text{of generalized L}}} + \underbrace{\mathbf{k}_g^{-1} \delta_0 \dot{\mathbf{d}}^0}_{\substack{\text{generalized currents}\\ \text{of gyrators}}} + \underbrace{\mathbf{t}_T^1}_{\substack{\text{generalized currents}\\ \text{of transformers}}} \right) \\
&= \partial_1 \left[\underbrace{\mathbf{t}_\mathbf{f}^1}_{\substack{\text{generalized current}\\ \text{sources}}} \quad \underbrace{- \left(\mathbf{C}_g \dot{\mathbf{a}}_\mathbf{f}^1 + \mathbf{R}_g^{-1} \mathbf{a}_\mathbf{f}^1 + \mathbf{k}_g^{-1} \mathbf{a}_\mathbf{f}^1 + \mathbf{L}_g^{-1} \int \mathbf{a}_\mathbf{f}^1 dt \right)}_{\substack{\text{equivalent generalized current}\\ \text{sources generated from}\\ \text{generalized voltage sources}}} \right]
\end{aligned} \tag{63}$$

$$\mathbf{k}_t \left(\delta_0 \dot{\mathbf{d}}^0 - \mathbf{a}_\mathrm{f}^1 \right) = \mathbf{0} \tag{64}$$

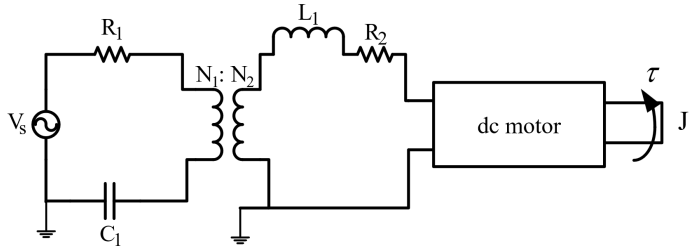
$$\mathbf{k}_t' \left(\mathbf{t}^1 - \mathbf{t}_\mathrm{f}^1 \right) = \mathbf{0} \tag{65}$$

$$\begin{aligned}
&\delta_1 \left(\underbrace{\mathbf{L}_g \partial_2 \ddot{\mathbf{n}}^2}_{\substack{\text{generalized voltages}\\ \text{of generalized L}}} + \underbrace{\mathbf{R}_g \partial_2 \dot{\mathbf{n}}^2}_{\substack{\text{generalized voltages}\\ \text{of generalized R}}} + \underbrace{\mathbf{C}_g^{-1} \partial_2 \mathbf{n}^2}_{\substack{\text{generalized voltages}\\ \text{of generalized C}}} + \underbrace{\mathbf{k}_g \partial_2 \dot{\mathbf{n}}^2}_{\substack{\text{generalized voltages}\\ \text{of gyrators}}} + \underbrace{\mathbf{a}_T^1}_{\substack{\text{generalized voltages}\\ \text{of transformers}}} \right) \\
&= \delta_1 \left(\underbrace{-\mathbf{a}_\mathbf{f}^1}_{\substack{\text{generalized voltages}\\ \text{sources}}} \quad \underbrace{+ \mathbf{L}_g \dot{\mathbf{t}}_\mathbf{f}^1 + \mathbf{R}_g \mathbf{t}_\mathbf{f}^1 + \mathbf{k}_g \mathbf{t}_\mathbf{f}^1 + \mathbf{C}_g^{-1} \int \mathbf{t}_\mathbf{f}^1 dt}_{\substack{\text{equivalent generalized voltage}\\ \text{sources generated from}\\ \text{generalized current sources}}} \right)
\end{aligned} \tag{66}$$

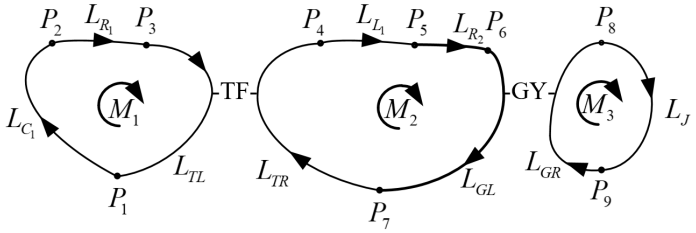
$$\mathbf{k}_t' \left(\partial_2 \dot{\mathbf{n}}^2 - \mathbf{t}_\mathrm{f}^1 \right) = \mathbf{0} \tag{67}$$

$$\mathbf{k}_\mathrm{t} \left(\mathbf{a}^1 - \mathbf{a}_\mathrm{f}^1 \right) = \mathbf{0} \tag{68}$$

Example 5.1 *We will use an example of multi-domain electrical-mechanical system in Figure 20a to illustrate the derivation of Eq.63 and Eq.66 . The shown electrical-mechanical system contains two resistors R_1, R_2, one capacitor C_1, one inductor L_1, one voltage source V_s, one moment of inertia J, one external torque τ, one electrical transformer and one DC motor. The transformer between two electrical domains is an ideal electrical transformer, where the ratio of voltage drops (currents) equals the ratio (inverse ratio) of the winding numbers N_1/N_2; the gyrator between the electrical and the mechanical domain is an ideal DC motor, where the ratio (inverse ratio) of the voltage drop (current) and the rotational velocity (torque) is a constant number k_g. Topologically, the system is a 2-cell complex shown in Figure 20b, and consisted of nine 0-cells (P_1,P_2,P_3,P_4,P_5,P_6,P_7,P_8,P_9), nine 1-cells (L_{R_1},L_{C_1},L_{TL},L_{TR}, L_{R_2},L_{L_1},L_{GL},L_{GR},L_J) and three 2-cells (M_1,M_2,M_3). We use*



(a) A multi-domain electro-mechanical system



(b) Topological structure

Figure 20: An electro-mechanical system and its topological structure

symbol -TF- to represent the abstract transformer and a symbol -GY- to represent the abstract gyrator. These two symbols identify the cells where the energy transaction may occur.

The algebraic topological model of the system contians: primal 0-cochain generalized potential magnetic fluxes ($\mathbf{d}^0 = d_1 \cdot P_1 + d_2 \cdot P_2 + d_3 \cdot P_3 + d_4 \cdot P_4 + d_5 \cdot P_5 + d_6 \cdot P_6 + d_7 \cdot P_7 + d_8 \cdot P_8 + d_9 \cdot P_9$*), primal 0-cochain generalized potentials (*$\mathbf{p}^0 = p_1 \cdot P_1 + p_2 \cdot P_2 + p_3 \cdot P_3 + p_4 \cdot P_4 + p_5 \cdot P_5 + p_6 \cdot P_6 + p_7 \cdot P_7 + p_8 \cdot P_8 + p_9 \cdot P_9$*), primal 1-cochain generalized magnetic fluxes (*$\mathbf{u}^1 = u_1 \cdot L_{R_1} + u_2 \cdot L_{C_1} + u_3 \cdot L_{TL} + u_4 \cdot L_{TR} + u_5 \cdot L_{R_2} + u_6 \cdot L_{L_1} + u_7 \cdot L_{GL} + u_8 \cdot L_{GR} + u_9 \cdot L_J$*), primal 1-cochain generalized voltages (*$\mathbf{a}^1 = a_1 \cdot L_{R_1} + a_2 \cdot L_{C_1} + a_{TL} \cdot L_{TL} + a_{TR} \cdot L_{TR} + a_5 \cdot L_{R_2} + a_6 \cdot L_{L_1} + a_{GL} \cdot L_{GL} + a_{GR} \cdot L_{GR} + a_9 \cdot L_J$*), dual 1-cochain generalized currents (*$\mathbf{t}^1 = t_1 \cdot L_{R_1} + t_2 \cdot L_{C_1} + t_{TL} \cdot L_{TL} + t_{TR} \cdot L_{TR} + t_5 \cdot L_{R_2} + t_6 \cdot L_{L_1} + t_{GL} \cdot L_{GL} + t_{GR} \cdot L_{GR} + t_9 \cdot L_J$*), dual 1-cochain generalized electric charges (*$\mathbf{m}^1 = m_1 \cdot L_{R_1} + m_2 \cdot L_{C_1} + m_{TL} \cdot L_{TL} + m_{TR} \cdot L_{TR} + m_5 \cdot L_{R_2} + m_6 \cdot L_{L_1} + m_{GL} \cdot L_{GL} + m_{GR} \cdot L_{GR} + m_9 \cdot L_J$*), dual 2-cochain generalized mesh electric charges $\mathbf{n}^2 = n_1 \cdot M_1 + n_2 \cdot M_2 + n_3 \cdot M_3$, dual 2-cochain generalized mesh currents $\mathbf{s}^2 = s_1 \cdot M_1 + s_2 \cdot M_2 + s_3 \cdot M_3$ and four cochains that are always $\mathbf{0}$: 2-cochain of generalized mesh magnetic fluxes, 2-cochain of generalized mesh voltages, 0-cochain of generalized node currents and 0-cochain of generalized node electric charges. In order to obtain unique solution of the state equations, we consider 0-cells P_1, P_7 and P_9 as the reference node. Following the paths shown in Figure 19a, generates Eq.63 $\sim$ Eq.65, with individual terms as follows:*

$$\partial_1 = \begin{bmatrix} -1 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & +1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & +1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & +1 \end{bmatrix} \tag{69}$$

$$\delta_0 = \partial_1^T \tag{70}$$

$$\mathbf{C}_g = \begin{bmatrix} C_{g_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{71}$$

$$\mathbf{R}_g^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & R_{g_1}^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{g_2}^{-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{72}$$

$$\mathbf{L}_g^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & L_{g_1}^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & L_{g_2}^{-1} \end{bmatrix} \tag{73}$$

$$\mathbf{k}_g^{-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_g^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_g^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{74}$$

$$\mathbf{k}_t = \begin{bmatrix} 0 & 0 & 1 & -N_1/N_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{75}$$

$$\mathbf{k'}_t = \begin{bmatrix} 0 & 0 & 1 & -N_2/N_1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{76}$$

$$\mathbf{t}_T^1 = \begin{bmatrix} 0 & 0 & t_{TL} & t_{TR} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T \tag{77}$$

$$\mathbf{a}_{\mathrm{f}}^1 = \begin{bmatrix} 0 & -a_{\mathrm{f}_1} & 0 & 0 & 0 & 0 & 0 & 0 & -a_{\mathrm{f}_2} \end{bmatrix}^T \tag{78}$$

$$\mathbf{t}_{\mathrm{f}}^1 = [\mathbf{0}]_{9\times 1} \tag{79}$$

Substitute Eq.69 $\sim$ Eq.79 to Eq.63 $\sim$ Eq.65, the generated system state equation is as follows:

$$\begin{cases} \begin{bmatrix} C_{g_1} & -C_{g_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -C_{g_1} & C_{g_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \ddot{\mathbf{d}}^0 + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & R_{g_1}^{-1} & -R_{g_1}^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -R_{g_1}^{-1} & R_{g_1}^{-1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{g_2}^{-1} & -R_{g_2}^{-1} & -k_g^{-1} & k_g^{-1} \\ 0 & 0 & 0 & 0 & 0 & -R_{g_2}^{-1} & R_{g_2}^{-1} & k_g^{-1} & -k_g^{-1} \\ 0 & 0 & 0 & 0 & 0 & -k_g^{-1} & k_g^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_g^{-1} & -k_g^{-1} & 0 & 0 \end{bmatrix} \dot{\mathbf{d}}^0 \\ + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_{g_1}^{-1} & -L_{g_1}^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -L_{g_1}^{-1} & L_{g_1}^{-1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & L_{g_2}^{-1} & -L_{g_2}^{-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -L_{g_2}^{-1} & L_{g_2}^{-1} \end{bmatrix} \mathbf{d}^0 + \begin{bmatrix} t_{TL} \\ 0 \\ -t_{TL} \\ t_{TR} \\ 0 \\ 0 \\ -t_{TR} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -R_{g_1}^{-1} a_{\mathrm{f}_1} \\ R_{g_1}^{-1} a_{\mathrm{f}_1} \\ 0 \\ 0 \\ 0 \\ 0 \\ -L_{g_2}^{-1} \int a_{\mathrm{f}_2} dt \\ L_{g_2}^{-1} \int a_{\mathrm{f}_2} dt \end{bmatrix} \\ \left(\dot{\mathbf{d}}_{(3)}^0 - \dot{\mathbf{d}}_{(1)}^0\right) \Big/ \left(\dot{\mathbf{d}}_{(7)}^0 - \dot{\mathbf{d}}_{(4)}^0\right) = N_1/N_2 \\ t_{TL}/t_{TR} = N_2/N_1 \end{cases} \tag{80}$$

Similarly following the paths in Figure 19b, generates Eq.66 ∼ Eq.68, with individual terms as follows:

$$\delta_1 = \begin{bmatrix} +1 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & +1 & +1 & +1 & +1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & +1 & +1 \end{bmatrix} \tag{81}$$

$$\partial_2 = {\delta_1}^T \tag{82}$$

$$\delta_1 \mathbf{L}_g \partial_2 = \delta_1 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & L_{g_1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & L_{g_2} \end{bmatrix} \partial_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & L_{g_1} & 0 \\ 0 & 0 & L_{g_2} \end{bmatrix} \tag{83}$$

$$\delta_1 \mathbf{R}_g \partial_2 = \delta_1 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & R_{g_1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & R_{g_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \partial_2 = \begin{bmatrix} R_{g_1} & 0 & 0 \\ 0 & R_{g_2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{84}$$

$$\delta_1 \mathbf{C}_g^{-1} \partial_2 = \delta_1 \begin{bmatrix} C_{g_1}^{-1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \partial_2 = \begin{bmatrix} C_{g_1}^{-1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{85}$$

$$\delta_1 \mathbf{k}_g \partial_2 = \delta_1 \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_g & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & k_g & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \partial_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & k_g \\ 0 & k_g & 0 \end{bmatrix} \tag{86}$$

$$\delta_1 \mathbf{a}_T^1 = \delta_1 \begin{bmatrix} 0 & 0 & \mathrm{a}_{TL} & \mathrm{a}_{TR} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T = \begin{bmatrix} \mathrm{a}_{TL} \\ \mathrm{a}_{TR} \\ 0 \end{bmatrix} \tag{87}$$

Substitute Eq.75 ∼ Eq.79, Eq.81 ∼ Eq.87 to Eq.66 ∼ Eq.68, the generated system state equation is as follows:

$$\begin{cases} \begin{bmatrix} 0 & 0 & 0 \\ 0 & L_{g_1} & 0 \\ 0 & 0 & L_{g_2} \end{bmatrix} \ddot{\mathbf{n}}^2 + \begin{bmatrix} R_{g_1} & 0 & 0 \\ 0 & R_{g_2} & k_g \\ 0 & k_g & 0 \end{bmatrix} \dot{\mathbf{n}}^2 + \begin{bmatrix} C_{g_1}^{-1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{n}^2 + \begin{bmatrix} \mathrm{a}_{TL} \\ \mathrm{a}_{TR} \\ 0 \end{bmatrix} = \begin{bmatrix} a_{\mathrm{f}_1} \\ 0 \\ a_{\mathrm{f}_2} \end{bmatrix} \\ \mathrm{a}_{TL}/\mathrm{a}_{TL} = N_1/N_2 \\ \dot{\mathbf{n}}_{(1)}^2 \Big/ \dot{\mathbf{n}}_{(2)}^2 = N_2/N_1 \end{cases} \tag{88}$$

6 Conclusion

6.1 Summary and significance

Formal semantics of engineering models is a principal ingredient of rigorous and algorithmic foundations for model-based systems engineering. Standardizing on such a semantics is also a key to solving widespread interchangeability and interoperability challenges that arise with rapid proliferation of model-based engineering language and tools. In this paper, we proposed a formal semantics for a large and important class of lumped-parameter systems that are widely used for systems engineering, physical modeling, and design activities. The proposed semantics relies only on standard tools from algebraic topology and known results in classification of physical theories and systems. The semantics is effectively 'representation free' in that it is independent of specific implementation assumptions, coordinates, linguistic constructs, or numerical simulation schemes.

We showed that (extended and generalized) Tonti diagrams provide a canonical method for representing behaviors of lumped parameter systems computationally. Based on known classification of physical theories, behavior of any lumped parameter system may be described either as a collection of interacting single-domain Tonti diagrams or as a single generalized Tonti diagram with energy transduction represented by additional constraints. In other words, a Tonti diagram can be viewed as a representation scheme and a data structure for representing most known physical behaviors. We have also seen that this representation supports algorithmic generation of all possible forms of the governing state equations as paths in the (collection of) Tonti diagrams. The topological and constitutive operators appear as labels on the edges of the diagram and may be interpreted either symbolically, giving differential equations, or numerically (e.g. as finite difference operators), corresponding to executable approximations of such models.

6.2 Interchangeability and interoperability

Achieving greater interoperability of lumped parameter systems tools and languages was the main motivation for developing the proposed canonical semantic model. The existence of common semantics implies that all compliant models become fully interchangeable irrespective of specific syntax or modeling concepts adapted in a particular simulation tool. The representation of this semantics

using the corresponding Tonti diagrams can serve as a formal neutral format for exchange of all such models. Furthermore, this representation supports exchange not only of the system models but also of the adapted simulation approaches in individual systems, now represented as paths in the corresponding Tonti diagrams. In practical terms, the existence of the common semantic model also eliminates the need for customized point-to-point translators or new super-languages attempting to subsume and unify currently existing tools. New compatible languages and modeling tools, for example in SysML, could be developed rigorously and rapidly, as long as they adhere to the adapted semantic model.

While model translation and exchange is a widely adapted and practiced in industry, it is a rather limited form of interoperability. When distinct tools and languages are based on a common reference semantics, they should be able to interoperate directly in a truly object-oriented fashion, where distinct portions of the system models are represented and simulated in different tools. This would allow better protection of proprietary models and fine-tuning selection of tools that are most appropriate for specific modeling tasks.

6.3 Open issues and promising directions

The semantic model proposed in this paper provides a foundation and a starting point for developing a more general semantic framework for modeling behavior of physical systems. In particular, the strongly-typed classification of physical theories and their representation by Tonti diagrams tacitly assumes that the constructed models are subject to energy balance laws. The majority of lumped-parameter system models satisfy this assumption. However, many modeling systems support numerous additional operations and constructs, such as signal flows and arbitrary mathematical transformations, that are not necessarily based on first principles. Whether and how such constructs should be included in the common semantics model is not entirely clear. In principle, they could be treated as special cases of energetic processes (as, for example, is advocated in bond graph literature [6]), or they may require introduction of additional types of constraints to the underlying topological model.

In this paper we focused on lumped parameter systems that are intrinsically two-dimensional and devoid of geometry. However, both classification of physical theories and Tonti diagrams span the entire spectrum of full-dimensional physical models and behaviors. This implies that the proposed formal semantic model can be extended to all spatially distributed models whose behavior is commonly described by integral and partial differential equations. In fact, developing a semantics model for an single-domain physical model whose behavior is described by a known Tonti diagram is a straightforward task that would follow the development in Section 4 of this paper. Different paths on such a diagram would correspond to different methods to generate governing equations; replacing symbolic operators by numerical approximations yields rich variety of standard numerical simulation methods [69, 73, 65]. However, in contrast to lumped parameter systems that are all isomorphic, spatially distributed models are heterogeneous in their dimension, type of variables, and topological structures. Interaction, composition, and transformation of such Tonti diagrams and the systems they represent remain active open research issues.

Treating the common semantic model and its representation by Tonti diagrams as the first class objects, that explicitly represent physical behaviors, opens up a number of promising and exciting opportunities in computational design and model-based engineering. Algorithmic construction, editing, composition, and transformation of such models would support a broad range of design engineering activities, from concept generation to detailed system modeling. Because many such

engineering activities are performed using SysML, it may be reasonable to expect that Tonti diagrams should appear as one of the standard diagrams in SysML in the near future. Such a diagram would provide immediate support for including physical behaviors and their simulation into a broad range of model-based systems engineering activities.

Acknowledgements

The authors would like to thank Conrad Bock for numerous discussions and encouragement. This research was supported in part by the National Institute of Standards under cooperative agreement 70NANB14H248, by NSF grants CMMI-1344205, CMMI-1361862, and CMMI-1547189, and by Defense Advanced Research Projects Agency's FUNdamental DESign Program. The responsibility for errors and omissions lies solely with the authors.

References

[1] Craig A Kluever. *Dynamic systems: modeling, simulation, and control*. John Wiley & Sons, 2015.

[2] Henry M Paynter. *Analysis and design of engineering systems*. MIT press, 1961.

[3] Derek Rowell and David N Wormley. *System dynamics: an introduction*. Prentice Hall, 1997.

[4] Hilding Elmqvist and Sven Erik Mattsson. An introduction to the physical modeling language modelica. In *Proceedings of the 9th European Simulation Symposium, ESS*, volume 97, pages 19–23. Citeseer, 1997.

[5] Devendra K Chaturvedi. *Modeling and simulation of systems using MATLAB and Simulink*. CRC Press, 2009.

[6] Dean C Karnopp, Donald L Margolis, and Ronald C Rosenberg. System dynamics: a unified approach. 1990.

[7] Peter Fritzson and Vadim Engelson. Modelica - a unified object-oriented language for system modeling and simulation. In *ECOOP98 Object-Oriented Programming*, pages 67–90. Springer, 1998.

[8] Dingyü Xue and YangQuan Chen. *System simulation techniques with Matlab and Simulink*. John Wiley & Sons, 2013.

[9] Alan S Perelson and George F Oster. Bond graphs and linear graphs. *Journal of the Franklin Institute*, 302(2):159–185, 1976.

[10] Dean C Karnopp, Donald L Margolis, and Ronald C Rosenberg. *System dynamics: modeling, simulation, and control of mechatronic systems*. John Wiley & Sons, 2012.

[11] Conrad Bock, Raphael Barbau, Ion Matei, and Mehdi Dadfarnia. An extension of the systems modeling language for physical interaction and signal flow simulation. *Systems Engineering*, 20(5):395–431, 2017.

[12] Martin Otter. Functional mockup interface (fmi) for model exchange. *Modelica Newsletter*, 1, 2010.

[13] Enzo Tonti. *On the formal structure of physical theories*. Istituto di matematica del Politecnico di Milano, 1975.

[14] Enzo Tonti. *The mathematical structure of classical and relativistic physics*. Springer, 2013.

[15] J Paul Roth. An application of algebraic topology to numerical analysis: On the existence of a solution to the network problem. *Proceedings of the National Academy of Sciences*, 41(7):518–521, 1955.

[16] Franklin H Branin. The algebraic-topological basis for network analogies and the vector calculus. In *Symposium on generalized networks*, pages 453–491, 1966.

[17] Gabriel Kron. *Diakoptics: the piecewise solution of large-scale systems*, volume 2. MacDonald, 1963.

[18] Paul Bamberg and Schlomo Sternberg. *A Course in Mathematics for Students of Physics*. Cambridge University Press, Cambridge, England, 1988,1990.

[19] James Clerk Maxwell. On the mathematical classification of physical quantities. In W.D. Niven, editor, *The Scientific Papers of James Clerk Maxwell*, pages 257 – 266. Cambridge University Press, 1890.

[20] D. van Dantzig. On the geometrical representation of elementary physical objects and the relations between geometry and physics. *Nieuw Archief voor Wiskunde*, 3(II):73–89, 1954.

[21] C. A. Nickle. Oscillographic solution of electro-mechanical systems. *Transactions of the American Institute of Electrical Engineers*, 44:844–856, 1925.

[22] H. F. Olson. *Dynamical Analogies*. Van Nostrand, Princeton, 1943.

[23] Gabriel Kron. Numerical solution of ordinary and partial differential equations by means of equivalent circuits. *Journal of Applied Physics*, 126:172–186, March 1945.

[24] Gabriel Kron. *Diakoptics – The Piecewise Solution of Large-Scale Systems*. The Electrical Journal, London, 1957-1959. A series of 20 articles beginning June 7, 1957.

[25] Franklin H. Branin. The algebraic-topological basis for network analogies and the vector calculus. In *Proceedings of the Symposium on Generalized Networks*, volume 16, pages 453 – 491, Brooklyn, New York, 1966. Polytechnic Institute of Brooklyn.

[26] J. P. Roth. An application of algebraic topology to numerical analysis: On the existence of a solution to the network problem. *Proc. Nat. Acad. Sci.*, 41:518–521, 1955.

[27] Richard S Palmer and Vadim Shapiro. Chain models of physical behavior for engineering analysis and design. *Research in Engineering Design*, 5(3-4):161–184, 1993.

[28] Antonio DiCarlo, Franco Milicchio, Alberto Paoluzzi, and Vadim Shapiro. Chain-based representations for solid and physical modeling. *IEEE Transactions on Automation Science and Engineering*, 6(3):454–467, 2009.

[29] Antonio DiCarlo, Alberto Paoluzzi, and Vadim Shapiro. Linear algebraic representation for topological structures. *Computer-Aided Design*, 46:269–274, 2014.

[30] Vasu Ramaswamy and Vadim Shapiro. Combinatorial laws for physically meaningful design, March 2004.

[31] Jeffrey A Chard and Vadim Shapiro. A multivector data structure for differential forms and equations. *Mathematics and Computers in Simulation*, 54(1):33–64, 2000.

[32] Richard Egli and Neil F Stewart. Chain models in computer simulation. *Mathematics and Computers in Simulation*, 66(6):449–468, 2004.

[33] Oyvind Bjorke. *Manufacturing Systems Theory*. Akademika Pub, 1995.

[34] Henry M. Paynter. *Analysis and Design of Engineering Systems*. The M.I.T. Press, Cambridge, Massachusetts, 1961.

[35] Herman E Koenig, YeLMAZ TOKAD, and Hiremaglur K Kesavan. Analysis of discrete physical systems. 1967.

[36] Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. Co-simulation: State of the art. *arXiv preprint arXiv:1702.00686*, 2017.

[37] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann, 2014.

[38] Ahsan Qamar, Carl During, and Jan Wikander. Designing mechatronic systems, a model-based perspective, an attempt to achieve sysml-matlab/simulink model integration. In *Advanced Intelligent Mechatronics, 2009. AIM 2009. IEEE/ASME International Conference on*, pages 1306–1311. IEEE, 2009.

[39] Andrea Sindico, Marco Di Natale, and Gianpiero Panci. Integrating sysml with simulink using open-source model transformations. In *SIMULTECH*, pages 45–56, 2011.

[40] OMG SE DSIG SysML-Modelica Working Group et al. Sysml-modelica transformation specification, 2009.

[41] Thomas Johnson, Aleksandr Kerzhner, Christiaan JJ Paredis, and Roger Burkhart. Integrating models and simulations of continuous dynamics into sysml. *Journal of Computing and Information Science in Engineering*, 12(1):011002, 2012.

[42] Skander Turki and Thierry Soriano. A sysml extension for bond graphs support. In *Proc. of the International Conference on Technology and Automation (ICTA), Greece*. Citeseer, 2005.

[43] Ion Matei and Conrad Bock. *SysML Extension for Dynamical System Simulation Tools*. US Department of Commerce, National Institute of Standards and Technology, 2012.

[44] Torsten Blochwitz, Martin Otter, Martin Arnold, Constanze Bausch, H Elmqvist, A Junghanns, J Mauß, M Monteiro, T Neidhold, D Neumerkel, et al. The functional mockup interface for tool independent exchange of simulation models. In *Proceedings of the 8th International Modelica Conference; March 20th-22nd; Technical Univeristy; Dresden; Germany*, number 063, pages 105–114. Linköping University Electronic Press, 2011.

[45] JR Ort and HR Martens. A topological procedure for converting a bond graph to a linear graph. *Journal of Dynamic Systems, Measurement, and Control*, 96(3):307–314, 1974.

[46] SH Birkett and PH Roe. The mathematical foundations of bond graphsi. algebraic theory. *Journal of the Franklin Institute*, 326(3):329–350, 1989.

[47] José Antonio Calvo, Carolina Álvarez-Caldas, and José Luis San Román. *Analysis of Dynamic Systems Using Bond Graph Method Through SIMULINK.* INTECH Open Access Publisher, 2011.

[48] Jan F Broenink. Bond-graph modeling in modelica. In *Proceedings of 9th European simulation symposium*, pages 19–22. Passau Germany, Oct, 1997.

[49] Jan F Broenink. Object-oriented modeling with bond graphs and modelica. *SIMULATION SERIES*, 31:163–168, 1999.

[50] Mariana C Dabreu and Gabriel A Wainer. A bond-graph mapping mechanism for m/cd+. 2006.

[51] Wolfgang Borutzky. *Bond graph methodology: development and analysis of multidisciplinary dynamic system models*. Springer Science & Business Media, 2009.

[52] Mike Dempsey. Automatic translation of simulink models into modelica using simelica and the advancedblocks library. In *Paper presented at the 3rd International Modelica Conference*. Citeseer, 2003.

[53] Suriya Natsupakpong and M Cenk Çavuşoğlu. Determination of elasticity parameters in lumped element (mass-spring) models of deformable objects. *Graphical Models*, 72(6):61–73, 2010.

[54] Vadim Engelson, Hakan Larsson, and Peter Fritzson. A design, simulation and visualization environment for object-oriented mechanical and multi-domain models in modelica. In *Information Visualization, 1999. Proceedings. 1999 IEEE International Conference on*, pages 188–193. IEEE, 1999.

[55] Peter Bunus, Vadim Engelson, and Peter Fritzson. Mechanical models translation, simulation and visualization in modelica. In *Visualization in Modelica. Proc. of Modelica 2000 Workshop*. Citeseer, 2000.

[56] Vadim Engelson, Peter Bunus, Lucian Popescu, and Peter Fritzson. Mechanical cad with multibody dynamic analysis based on modelica simulation. In *Proceedings of the 44th Scandinavian Conference on Simulation and Modeling*, pages 18–19, 2003.

[57] Paul Bowles, Michael Tiller, Hilding Elmqvist, Dag Brück, Sven Erik Mattsson, Andreas Möller, Hans Olsson, and Martin Otter. Feasibility of detailed vehicle modeling. Technical report, SAE Technical Paper, 2001.

[58] Arquimedes Canedo, Zhi Zhang, Ahmad Albaqsami, Jiang Wan, and Mohammad Abdullah Al Faruque. Maintaining the design intent in the synthesis of 3-d and 1-d system models using constraints. *IEEE Systems Journal*, 2016.

[59] Vadim Engelson, Peter Fritzson, and Ulf Sellgren. An easy-to-use generic model configurator for models in multiple tool formats. In *5th EUROSIM Congress on Modeling and Simulation, Paris, France*, 2004.

[60] Thomas Hoeft and Christoph Nytsch-Geusen. Design and validation of an annotation-concept for the representation of 3d-geometries in modelica. In *Proceedings of the 6th International Modelica Conference*, 2008.

[61] Paul Bamberg and Shlomo Sternberg. *A course in mathematics for students of physics*, volume 2. Cambridge University Press, 1991.

[62] Allen Hatcher. Algebraic topology. 2001.

[63] J. Paul Roth. Existence and uniqueness of solution to electrical network problem via homology sequences. In *SIAM-AMS Proceedings, Vol. III*, pages 113–118. American Mathematical Society, 1971.

[64] Alain Bossavit. *Computational electromagnetism: variational formulations, complementarity, edge elements*. Academic Press, 1998.

[65] Elena Ferretti. The cell method: An overview on the main features. *Curved and Layered Structures*, 2(1), 2015.

[66] José E Castillo and Guillermo F Miranda. *Mimetic discretization methods*. CRC Press, 2013.

[67] Enzo Tonti. Why starting from differential equations for computational physics? *Journal of Computational Physics*, 257:1260–1290, 2014.

[68] Anil Nirmal Hirani. *Discrete exterior calculus*. PhD thesis, California Institute of Technology, 2003.

[69] Claudio Mattiussi. An analysis of finite volume, finite element, and finite difference methods using some concepts from algebraic topology. *Journal of Computational Physics*, 133:289–309, 1997.

[70] Robert Babuska and Stefano Stramigioli. Matlab and simulink for modeling and control. *Delft University of Technology*, 1999.

[71] Enzo Tonti. A classification diagram for physical variables, 2003.

[72] Henry E Singleton. *Theory of nonlinear transducers*. PhD thesis, Massachusetts Institute of Technology, 1950.

[73] Carlos A Felippa. Introduction to finite element methods. *Course Notes, Department of Aerospace Engineeing Sciences, University of Colorado at Boulder, available at http://www. colorado. edu/engineering/Aerospace/CAS/courses. d/IFEM. d*, 2004.