IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 10, 2020, accepted May 31, 2020, date of publication June 8, 2020, date of current version June 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3000129

Port-Hamiltonian Modeling of Multiphysics Systems and Object-Oriented Implementation With the Modelica Language

FRANCISCO M. MÁRQUEZ[1,2], PEDRO J. ZUFIRIA[2], AND LUIS J. YEBRA[3]
[1]Departamento de Automática, Universidad de Alcalá (UAH), 28801 Alcalá de Henares, Spain
[2]Information Processing and Telecommunications Center, Departamento de Matemática aplicada a las Tecnologías de la Información y las Comunicaciones, ETSI Telecomunicación, Universidad Politécnica de Madrid (UPM), 28040 Madrid, Spain
[3]Plataforma Solar de Almería, CIEMAT, 04200 Tabernas, Spain

Corresponding author: Francisco M. Márquez (francisco.marquez@uah.es)

The work of Luis J. Yebra was supported by the Universidad de Alcalá, through the Programa propio Giner de los Ríos Research Grant.

ABSTRACT In this article we present the implementation in Modelica language of a library with the fundamental components for modeling a wide variety of multiphysics systems. Modelica is an object-oriented modeling language, which allows to make a simple, systematic and elegant design of the library. The mechanisms of inheritance and composition of Modelica facilitate the modeling and reuse of components in different domains of Physics. To model the behavior of each component in a systematic framework we have used the theory of port-Hamiltonian systems, formulated mainly by means of differential geometry. The port-Hamiltonian approach allows a methodical definition of complex systems by connecting simple systems that exchange energy through connection ports. To graphically represent the components of a system and their connections, we have employed slightly modified bond graphs symbols for easier reading. The general and systematic applicability of the library is illustrated via two examples framed in different domains of Physics: the mechanical Sun-Earth-Moon system where we perform an analysis of errors that justifies the employed system of units, and the electrical nonlinear Chua circuit, modeled by composition of port-Hamiltonian subsystems. Both derived models have been built and simulated based on the more general models of mechanical and electrical systems, which are also part of the library developed with the port-Hamiltonian approach.

INDEX TERMS Bond graphs, Dirac structures, Modelica language, multiphysics systems, object oriented modeling, port-Hamiltonian systems.

I. INTRODUCTION

A. MOTIVATION

The methodical modeling and analysis of multidisciplinary complex systems is one of the major challenges in science and engineering [23]. In the case of physical or chemical systems, energy is a magnitude that plays an important role in building consistent models of complex systems composed of interconnected subsystems defined in different physical domains (e.g.: mechanics electricity, thermodynamics, etc.), also called *multiphysics systems*. As stated in [44, p.6], energy and power are ''*the* 'lingua franca' *between different physical domains*''. This is the case, for example, of solar thermal power plants, where the models must take into account processes of transformation, transport, and storage of electromagnetic (sunlight), mechanical (fluid movement), and thermal (heat transfer) energy.

The leading role of energy based modeling of physical systems was established two centuries ago in frameworks of modeling, such as the Euler-Lagrange and Hamiltonian formalisms, developed in the field of Physics and enriched by contributions from mathematical areas such as differential geometry. At the same time, some branches of engineering, such as electrical engineering, have achieved remarkable success in coping with complex systems using modeling techniques based on the interconnection of multiple simple systems, giving rise to a paradigm called ''network theory''. One example is the port-based modeling, pioneered by

The associate editor coordinating the review of this manuscript and approving it for publication was Jesus Felez.

H. Paynter in the 1960s, who developed a graphical notation, known as bond graphs, to represent the connection and power flow between different parts of multiphysics systems [37].

The port-Hamiltonian formalism brings together the aforementioned assets and provides a framework for modeling physical systems based on the storage and exchange of energy through power ports that interconnect different parts of the system with each other, as well as with the environment.

The port-Hamiltonian approach enables the rigorous and scalable construction of complex systems because, on the one hand, the models are formulated in terms of differential geometry giving rise to the well endowed port-Hamiltonian systems [42]. On the other hand, the models of complex systems are constructed as a network of simple connected port-Hamiltonian systems whose connection structure, called Dirac's structure [14], [22], is also defined in terms of differential geometry, giving rise to a composed port-Hamiltonian system [12].

The Modelica language [3] allows to describe the behavior of a complex dynamic system from models of its constituent parts. The mechanisms of inheritance, instantiation, and composition offered by the language have been used for this purpose in the developed library.

Inheritance allows to define abstract and very general models, which can be refined to derive the concrete models of interest. Instantiation allows to create multiple copies of the same parametrizable model. Finally, composition is a mechanism that allows to define the model of a complex system by connecting the models of its constituent parts. These are all common characteristics to object-oriented languages which can be used hierarchically, where at each level of the hierarchy only its specific behavior needs to be modeled.

Hierarchical composition is a suitable way to implement formal concepts about port-Hamiltonian systems because the connection of port-Hamiltonian systems provides a system having also port-Hamiltonian structure.

The elements of Modelica that allow to work with the above concepts are classes, models, and connectors. Starting from general classes, one can extend or instantiate specific classes which, when connected, define the structure of the system to be modeled. The behavior of the system is completed by declaring the equations of the behavior of its components.

Tools such as Dymola, OpenModelica, or JModelica process symbolically the above description to simplify the equations and to generate a computationally efficient instance of the model that allows to directly simulate the behavior of the system.

Due to the described characteristics, port-Hamiltonian approach is an ideal framework to model multiphysics systems, and Modelica is a suitable language to implement models developed with the port-Hamiltonian approach.

The main objective of this paper is to present a generic Modelica library (pHlib) strictly based on the mathematical port-Hamiltonian formalism, rigorously founded on differential geometry, which serves as a unifying framework for modeling multiphysics systems. The Modelica Standard Library (MSL) is the public reference library normally used for modeling multi-domain systems, and we have used it to compare its results with those of models developed with pHlib.

B. LITERATURE REVIEW

The pioneering work in port-based system modeling and its bond graph representation by Paynter [37] has been used for decades for modeling multidisciplinary engineering dynamical systems [7] and it has partly inspired the formalism of the port-Hamiltonian systems. The evolution and current state of Paynter's ideas, as well as an study of interesting examples of applications of the bond graph methodology are illustrated in [6] and [30].

The theory of port-Hamiltonian systems makes use of methods from differential geometry (see [33] for a review of smooth manifolds), and from the behavioral approach to dynamical systems ([45], [46]). In [14] and [22] the Dirac structure and Dirac manifold are introduced to extend the Hamiltonian formalism to the study of dissipative systems; and in [42] the above concepts are used to define port-Hamiltonian systems. Reference [43] provides a wide introduction to port-Hamiltonian systems, whereas [44] gather an updated review with a presentation of the theoretical foundations and some simple practical applications of port-Hamiltonian systems. For more complex examples refer to the collection of articles in [23]. The representation of Dirac structures and the connection of port-Hamiltonian systems are discussed in [5], [12], [17], [36]. The relationship between bond graphs and port-Hamiltonian systems can be found in [26], and the derivation of port-Hamiltonian models from bond graphs is discussed in [21] and [39].

The first published work developing a library for modeling with bond graphs is [10], where the Dymola language [24] is used. This language was a predecessor of Modelica and used by the tool with the same name. Currently, Dymola is the name of a commercial modeling and simulation tool based on Modelica. The specification of Modelica language can be found in [3], and a comprehensive explanation of the use of the language to model systems from multiple domains is provided in [25]. In [11], [48], and [19], one can find important examples of building Modelica libraries for modeling with bond graphs.

In [16] and [15] the authors present the Port-Hamiltonian Systems Modeling Language (PHSML) and the algorithms to translate from PHSML to Modelica. This approach combines the modeling rigor of the port-Hamiltonian formulation and the efficiency of Modelica simulation tools.

C. CONTRIBUTIONS

In this paper we propose an object-oriented library of the fundamental components for modeling and simulating multiphysics systems, rooted in the formal definitions of Dirac structures and port-Hamiltonian systems.

Our approach brings together three modeling traditions that have been coexisting but with little connection between them: i) port-based modeling with bond graphs, which originated in the electrical engineering field; ii) Hamiltonian modeling, rooted in analytical mechanics; and iii) object oriented modeling of physical systems, an approach initiated in the late 1980s [4].

For this purpose: i) we have implemented the components of our library with Modelica, a modern, expressive, and powerful object oriented language designed for system modeling; ii) we have founded such implementation on the concepts of Dirac structures and port-Hamiltonian systems, both of them formalized in terms of differential geometry; iii) to represent the fundamental components, we propose new symbols derived from the ones used in bond graphs (so that they remain recognizable by bond graph modelers), which are more expressive and make it easier to identify the constituent elements of the models.

Our approach has the additional advantage (unlike bond graph modeling) that no causality analysis is needed when building the models, since the Modelica language and port-Hamiltonian modeling allow the definition of a system behavior making use of a set of differential and algebraic equations (DAE), without performing any causality assignment.

D. PAPER ORGANIZATION

This paper is organized as follows. In section II, we summarize the main results on Dirac structures and finite dimensional port-Hamiltonian systems concerning their definition, interconnection, and representation. General definitions are presented in a coordinate-free style by using the language of differential geometry whose notation is summarized in the section VII.

In section III, we present some of the fundamental blocks for building, by composition, complex port-Hamiltonian systems from simple subsystems. These components are: the junction structures, the storage elements (which include the state variables of the system), the generators (which impose the boundary conditions), and the dissipative elements.

In section IV, we present the library pHlib, where the components defined in the previous section are implemented using the Modelica language.

In section V and section VI, we present the implementations of the packages Mechanics and ElectricalNetwork, which are derived from the pHlib library to extend it to the domains of mechanics and electrical networks respectively. The general and systematic applicability of these packages is illustrated via two examples framed in different physical domains: the model of the N-body problem (with the particular case of the Sun-Earth-Moon system, composed of three bodies); and the model of the Chua circuit (which shows a chaotic behavior depending on the initial conditions).

II. PORT-HAMILTONIAN SYSTEMS

In this paper, all vector spaces and smooth manifolds are real and finite dimensional. The section VII summarizes the notation used in this article.

A. POWER PORTS AND Dirac STRUCTURES

We begin by defining the constant and modulated power ports as some fundamental mathematical tools to model the energy transfer between the system and its environment, and the energy interchange between the elements of the system, respectively.

Definition 1: [42, p.109] If the vector space F (called space of *flows*) and its dual $\mathsf{E} = \mathsf{F}^*$ (called space of *efforts*) meet that the product $\langle e \,|\, f \rangle$ ($f \in \mathsf{F}, e \in \mathsf{E}$) has physical dimensions of power, we refer to $\mathsf{F} \oplus \mathsf{E}$ as a *constant power port*, and $P = \langle e \,|\, f \rangle$ represents the power traversing the port.

If $\mathcal{X}$ is a smooth manifold with tangent and cotangent bundles $T\mathcal{X}$ and $T^*\mathcal{X}$, a *modulated power port* is now defined as the *fiber product* $P = T\mathcal{X} \oplus T^*\mathcal{X}$ and, for each $x \in \mathcal{X}$, the tangent and cotangent spaces $T_x\mathcal{X}$ and $T_x^*\mathcal{X}$ are spaces of flows and efforts respectively.

Remark 2: The definition of power port indicates that the product $\langle e|f \rangle$ has physical dimension of power in order to focus on the modeling of physical systems. This definition is consistent with the definition of a port in the bond graph methodology [6, p.20-23]. Some examples of $\langle$*effort*|*flow*$\rangle$ pairs taken from different physical domains are [23, p.23]: $\langle$*force*|*velocity*$\rangle$ translational mechanics, $\langle$*torque*|*angular velocity*$\rangle$ rotational mechanics, $\langle$*voltage*|*current*$\rangle$ electricity, $\langle$*pressure*|*volume flow*$\rangle$ hydraulic, and $\langle$*temperature*|*entropy flow*$\rangle$ thermodynamics. In the thermal domain variants of the bond graph modeling have been proposed where the previous rule is relaxed. This is the case of the so-called *pseudo-bond graphs* [6, p.427], where the temperature and the heat flow, whose product is not a power, are taken as effort and flow. This is often justified in practical engineering because it is easier to work with heat flows than with entropy flows [30, p.548]. There also exist applications of the bond graph methodology to non-physical domains, for example, to economic systems [9], [47]. These applications require an abstraction of the concepts of energy and power to coherently assign the variables of effort and flow.

For a unified treatment of all energy transfers, let now define the vector bundle $\mathcal{Q} = T\mathcal{X} \oplus T^*\mathcal{X} \oplus \mathsf{F} \oplus \mathsf{E}$ and the non-degenerate symmetric bilinear form $\Omega : \mathcal{Q} \oplus \mathcal{Q} \to \mathbb{R}$ that applies $q = X \oplus \omega \oplus e \oplus f$ and $q' = X' \oplus \omega' \oplus e' \oplus f'$ to

$$\Omega(q, q') = \frac{\langle \omega \,|\, X' \rangle + \langle \omega' \,|\, X \rangle + \langle e \,|\, f' \rangle + \langle e' \,|\, f \rangle}{2}. \qquad (1)$$

It is straightforward to show that $\Omega(q, q) = \langle \omega \,|\, X \rangle + \langle e \,|\, f \rangle = \langle \omega \oplus e \,|\, X \oplus f \rangle$ and then $\Omega(q, q)$ is the sum of the power traversing the ports $T\mathcal{X} \oplus T^*\mathcal{X}$ and $\mathsf{F} \oplus \mathsf{E}$. The Ω-*orthogonal complement* of a subbundle $\mathcal{D} \subset \mathcal{Q}$ is the subbundle $\mathcal{D}^\perp = \left\{q \in \mathcal{Q} : \Omega(q, q') = 0 \text{ for all } q' \in \mathcal{D}\right\}$.

Definition 3: [14], [22] A *Dirac structure* on the vector bundle $\mathcal{Q} = T\mathcal{X} \oplus T^*\mathcal{X} \oplus \mathsf{F} \oplus \mathsf{E}$ is a subbundle $\mathcal{D} \subset \mathcal{Q}$ maximally isotropic under the bilinear form Ω defined in (1), i.e., $\mathcal{D} = \mathcal{D}^\perp$.

Remark 4: It follows from the definition that $\Omega(q, q) = 0$ for all $q \in \mathcal{D}$, i.e., the energy traversing the ports

$T\mathcal{X} \oplus T^*\mathcal{X}$ and $\mathsf{F} \oplus \mathsf{E}$ is conserved and, due to this property, Dirac structures are useful to model the energy-conserving part of physical systems.

Proposition 5: Let $\mathcal{X}$ be a smooth manifold and F a vector space. The constant rank vector subbundle $\mathcal{D} \subset \mathcal{Q} = T\mathcal{X} \oplus T^*\mathcal{X} \oplus \mathsf{F} \oplus \mathsf{E}$ is a Dirac structure if and only if:

1) $\langle \omega \,|\, X \rangle + \langle e \,|\, f \rangle = 0$ *for all* $X \oplus \omega \oplus f \oplus e \in \mathcal{D}$ *and,*
2) $\operatorname{rank} \mathcal{D} = \operatorname{rank} (T\mathcal{X} \oplus \mathsf{F}) = \dim \mathcal{X} + \dim \mathsf{F}$.

Proof: See [17]. ☐

Henceforward, only constant rank Dirac structures will be considered.

B. FINITE DIMENSIONAL PORT-HAMILTONIAN SYSTEMS

Definition 6: [42] A *port-Hamiltonian system* is a state space dynamical system defined by the 4-tuple $(\mathcal{X}, \mathsf{F}, \mathcal{D}, H)$ where the smooth manifold $\mathcal{X}$ is the state space, the flow space F and the effort space $\mathsf{E} = \mathsf{F}^*$ define a power port $\mathsf{F} \oplus \mathsf{E}$ to interchange energy with the environment, the subbundle $\mathcal{D} \subset T\mathcal{X} \oplus T^*\mathcal{X} \oplus \mathsf{F} \oplus \mathsf{E}$ is a Dirac structure, $H \in C^\infty(\mathcal{X})$ is a smooth function $H : \mathcal{X} \to \mathbb{R}$ that models the energy of the system, and the dynamics is specified by

$$-\dot{x} \oplus \mathrm{d}H \oplus f \oplus e \in \mathcal{D}. \tag{2}$$

Remark 7: The evolution of the state, $\dot{x} \in T\mathcal{X}$, is a vector field defined on the manifold $\mathcal{X}$ and the differential of the Hamiltonian function, $\mathrm{d}H \in T^*\mathcal{X}$, is a covector field or 1-form [33]. The product $\langle \mathrm{d}H \,|\, -\dot{x} \rangle$ is the power outgoing from the storage elements of the system and incoming in the Dirac structure (see Fig.1).

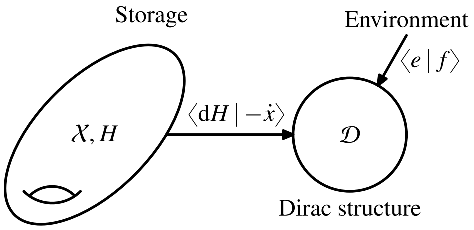


FIGURE 1. **Power distribution in a port-Hamiltonian system.**

Eq. 2 reads $\langle \mathrm{d}H \,|\, -\dot{x} \rangle + \langle e \,|\, f \rangle = 0$ or

$$\langle \mathrm{d}H \,|\, \dot{x} \rangle = \langle e \,|\, f \rangle\,, \tag{3}$$

and it says that "the power $\langle \mathrm{d}H \,|\, \dot{x} \rangle$ incoming in the storage element of the pHS is the power $\langle e \,|\, f \rangle$ incoming in the Dirac structure from the environment". For local coordinates $(x^i) = (x^1, \ldots, x^n)$ in a neighborhood of $x \in \mathcal{X}$ the power $\langle \mathrm{d}H \,|\, \dot{x} \rangle$ reads

$$\begin{aligned} \langle \mathrm{d}H \,|\, \dot{x} \rangle &= \left\langle \frac{\partial H}{\partial x^i} \mathrm{d}x^i \,\middle|\, \frac{\mathrm{d}x^j}{\mathrm{d}t} \frac{\partial}{\partial x^j} \right\rangle = \frac{\partial H}{\partial x^i} \frac{\mathrm{d}x^j}{\mathrm{d}t} \mathrm{d}x^i \left(\frac{\partial}{\partial x^j} \right) \\ &= \frac{\partial H}{\partial x^i} \frac{\mathrm{d}x^j}{\mathrm{d}t} \delta^i_j = \frac{\partial H}{\partial x^i} \frac{\mathrm{d}x^i}{\mathrm{d}t} = \frac{\mathrm{d}H}{\mathrm{d}t}, \end{aligned}$$

and tells us that "if the system does not interact with the environment (i.e., $\langle e \,|\, f \rangle = 0$), then the energy is a conserved quantity ($\mathrm{d}H/\mathrm{d}t = 0$)".

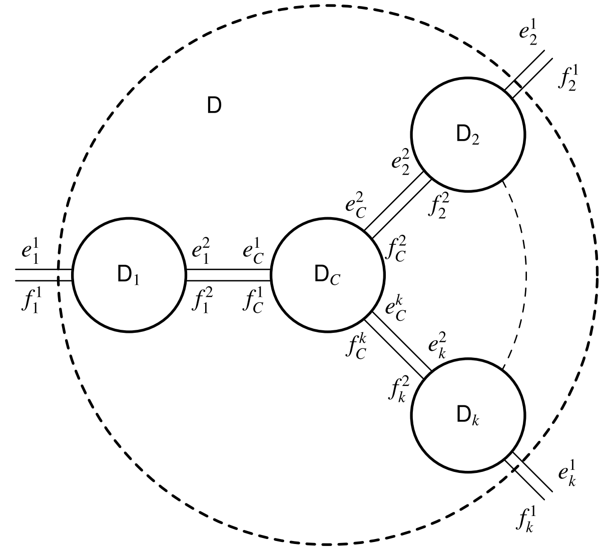


FIGURE 2. **Composition of Dirac structures.**

C. INTERCONNECTION OF PORT-HAMILTONIAN SYSTEMS

The following propositions formalize that the Dirac structures and the port-Hamiltonian systems meet a fundamental requirement for constructing complex system by interconnecting simpler ones: the composition of Dirac structures leads to another Dirac structure and, likewise, the interconnection of port-Hamiltonian systems provides again a port-Hamiltonian system.

Proposition 8: Let $\mathsf{D}_i \subset \mathsf{Q}_i = \mathsf{F}_i^1 \oplus \mathsf{E}_i^1 \oplus \mathsf{F}_i^2 \oplus \mathsf{E}_i^2$, $1 \leq i \leq k$, be k Dirac structures defined with respect to the bilinear forms $\Omega_i = \mathsf{Q}_i \oplus \mathsf{Q}_i \to \mathbb{R}$ (see (1)) and let $\mathsf{D}_C \subset \mathsf{Q}_C = \mathsf{F}_C^1 \oplus \mathsf{E}_C^1 \oplus \cdots \oplus \mathsf{F}_C^k \oplus \mathsf{E}_C^k$ be another Dirac structure defined with respect to the bilinear form

$$\begin{aligned} \Omega_C : \mathsf{Q}_C \oplus \mathsf{Q}_C &\to \mathbb{R} \\ (q_C, q_C') &\to \frac{1}{2} \sum_{i=1}^{k} \left(\left\langle e_C^i \,\middle|\, f_C^{i\,\prime} \right\rangle + \left\langle e_C^{i\,\prime} \,\middle|\, f_C^i \right\rangle \right), \end{aligned}$$

and connected to the first ones so that $\langle e_C^i \,|\, f_C^i \rangle = -\langle e_i^2 \,|\, f_i^2 \rangle$, i.e., $\mathsf{F}_C^i \oplus \mathsf{E}_C^i \cong \mathsf{F}_i^2 \oplus \mathsf{E}_i^2$ but $P_C^i = -P_i^2$ (see Fig.2). Then $\mathsf{D} \subset \mathsf{Q} = \mathsf{F}_1^1 \oplus \mathsf{E}_1^1 \oplus \cdots \oplus \mathsf{F}_k^1 \oplus \mathsf{E}_k^1$ is a Dirac structure with respect to the bilinear form

$$\begin{aligned} \Omega : \mathsf{Q} \oplus \mathsf{Q} &\to \mathbb{R} \\ (q, q') &\to \frac{1}{2} \sum_{i=1}^{k} \left(\left\langle e_1^1 \,\middle|\, f_i^{1\,\prime} \right\rangle + \left\langle e_i^{1\,\prime} \,\middle|\, f_i^1 \right\rangle \right). \end{aligned}$$

Proof: See [12] and [5]. ☐

Proposition 9: The interconnection of k port-Hamiltonian systems $S_i = (\mathcal{X}_i, \mathsf{F}_i, \mathcal{D}_i, H_i)$, $1 \leq i \leq k$ (see Fig.3) across the environment ports $\mathsf{F}_i \oplus \mathsf{E}_i$ with the Dirac structure $\mathsf{D}_C \subset \mathsf{Q}_C = \mathsf{F} \oplus \mathsf{E} \oplus \mathsf{F}_C^1 \oplus \mathsf{E}_C^1 \oplus \cdots \oplus \mathsf{F}_C^k \oplus \mathsf{E}_C^k$ such that $P_C^i = -P_i$, defines another port-Hamiltonian system $S = (\mathcal{X}, \mathsf{F}, \mathcal{D}, H)$ where:

1) $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_k$,
2) $H = H_1 + \cdots + H_k$ *and,*

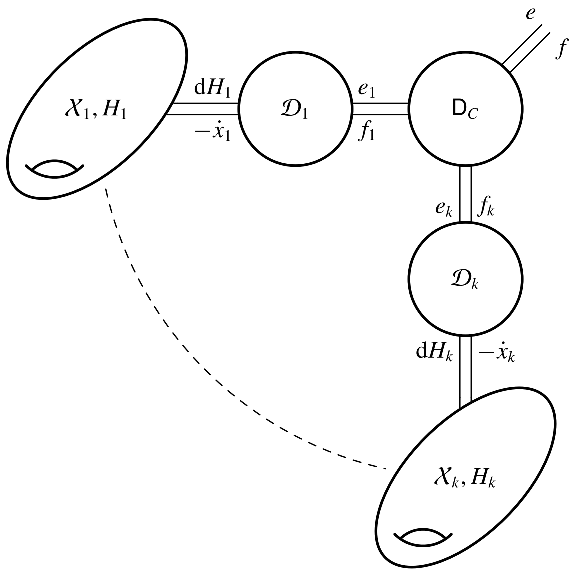


FIGURE 3. **Interconnection of port-Hamiltonian systems.**

3) *$\mathcal{D} \subset \mathcal{Q} = T\mathcal{X} \oplus T^*\mathcal{X} \oplus \mathsf{F} \oplus \mathsf{E}$ is the composition of the k Dirac structures $\mathcal{D}_i$ and D_C.*

Proof: See [12]. ☐

D. KERNEL AN IMAGE REPRESENTATION OF PORT-HAMILTONIAN SYSTEMS

The geometric definition of port-Hamiltonian systems (Definition 6) allows for a powerful coordinate-free reasoning; nevertheless, for engineering purposes where specific computations are required (such as simulation and design of control systems), it is necessary to represent port-Hamiltonian systems by means of differential-algebraic equations obtained from a coordinate based representation.

There are different ways of representing a Dirac structure such as the: kernel and image representation, constrained input-output representation, hybrid input-output representation [43], canonical coordinates representation [14], scattering representation [12], and spinor representation [36]. We will comment on the kernel representation, the most suitable for obtaining explicit port-Hamiltonian systems in the input-state-output form.

Proposition 10: Let $\mathcal{D} \subset \mathcal{Q} = T\mathcal{X} \oplus T^*\mathcal{X} \oplus \mathsf{F} \oplus \mathsf{E}$ be a Dirac structure of constant rank $n + m = \mathrm{rank}(T\mathcal{X} \oplus \mathsf{F})$, then there are two bundle morphisms $F : T\mathcal{X} \oplus \mathsf{F} \to \mathcal{V}$ and $E : T^*\mathcal{X} \oplus \mathsf{E} \to \mathcal{V}$ that satisfy the conditions

$$\begin{aligned} E \circ F^* + F \circ E^* &= 0 \\ \mathrm{rank}(F \oplus E) = \mathrm{rank}(T\mathcal{X} \oplus \mathsf{F}) &= n + m, \end{aligned} \tag{4}$$

and such that $\mathcal{D} = \ker(F \oplus E)$, i.e.,

$$\mathcal{D} = \{q \in \mathcal{Q} : F(X \oplus f) + E(\omega \oplus e) = 0\}\,. \tag{5}$$

The equating space $\mathcal{V}$ is a $(n + m)$-rank vector bundle with base space $\mathcal{X}$, and (5) is the kernel representation *of $\mathcal{D}$. For the same morphisms F and E, the Dirac structure is said to be represented in* image representation *if*

$$\begin{aligned} \mathcal{D} = \{q \in \mathcal{Q} : X \oplus f &= E^*(\lambda), \\ \omega \oplus e &= F^*(\lambda), \lambda \in \mathcal{V}^*\}. \end{aligned} \tag{6}$$

Proof: See [14] and [42]. ☐

Remark 11: Choosing local coordinates (x^i) in a neighborhood of $x \in \mathcal{X}$ and a base $(b_j) = \{b_1, \ldots, b_m\}$ for F with dual base $(b^j) = \left\{b^1, \ldots, b^m\right\}$, the morphisms F and E can be locally represented with matrices $F(x), E(x) \in \mathbb{R}^{(n+m)\times(n+m)}$ that satisfy

$$\begin{aligned} E(x)F^{\mathsf{T}}(x) + F(x)E^{\mathsf{T}}(x) &= 0 \\ \mathrm{rank}[F(x)|E(x)] &= n + m. \end{aligned} \tag{7}$$

For every fibre D_x, the matrix kernel representation is

$$\begin{aligned} \mathsf{D}_x = \{q_x = (v, w, f, e) \in \mathsf{Q}_x :& \\ F(x)\begin{bmatrix} v \\ f \end{bmatrix} + E(x)\begin{bmatrix} w \\ e \end{bmatrix} &= 0\}, \end{aligned} \tag{8}$$

and the local image representation is

$$\mathcal{D}_x = \left\{ q_x \in \mathcal{Q}_x : \begin{bmatrix} v \\ f \end{bmatrix} = E^{\mathsf{T}}(x)\lambda\,, \quad \begin{bmatrix} w \\ e \end{bmatrix} = F^{\mathsf{T}}(x)\lambda,\ \lambda \in \mathbb{R}^{1\times(n+m)} \right\}. \tag{9}$$

Remark 12: If we consider now the port-Hamiltonian system $(\mathcal{X}, \mathsf{F}, \mathcal{D}, H)$ and a kernel representation of $\mathcal{D}$, the Eq. 2 that defines the system dynamics can be written as

$$F(-\dot{x} \oplus f) + E(\mathrm{d}H \oplus e) = 0, \tag{10}$$

that yields the following local matrix equation

$$F(x)\begin{bmatrix} -\dot{x} \\ f \end{bmatrix} + E(x)\begin{bmatrix} \partial_x H \\ e \end{bmatrix} = 0, \tag{11}$$

where $\dot{x} = [\dot{x}^1, \ldots, \dot{x}^n]^{\mathsf{T}}$ and $\partial_x H = [\frac{\partial H}{\partial x^1}, \ldots, \frac{\partial H}{\partial x^n}]^{\mathsf{T}}$ are the matrix representation of $\dot{x}$ and $\mathrm{d}H$ respectively.

III. PORT-HAMILTONIAN SYSTEM BUILDING BLOCKS

In this section we define some elementary systems that are useful building blocks for modeling complex ones [37].

A. JUNCTION STRUCTURES

The junction structures are port-Hamiltonian system parts that instantaneously transfer energy without storage or conversion into heat.

Definition 13: A *0-junction* or *common effort junction* (see Fig.4.a) is a Dirac structure $\mathsf{D}_0 \subset \mathsf{F}_1 \oplus \mathsf{E}_1 \oplus \cdots \oplus \mathsf{F}_k \oplus \mathsf{E}_k$ defined by the following equations

$$\begin{aligned} f_1 + f_2 \cdots + f_k &= 0 \\ e_1 = e_2 = \cdots &= e_k. \end{aligned} \tag{12}$$

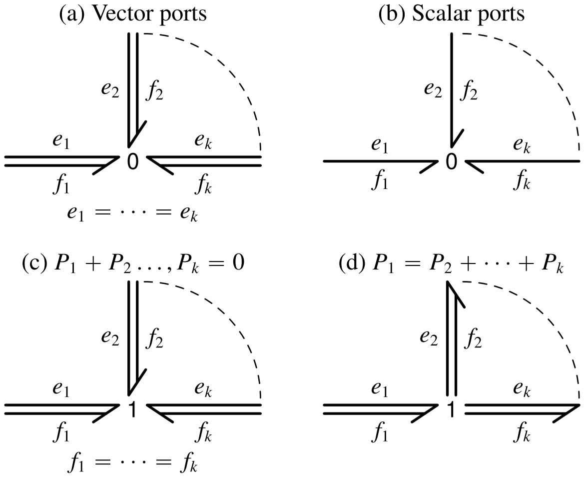


FIGURE 4. (a) 0-junction (or common effort junction) with vector bonds/ports, (b) 0-junction with scalar bonds, (c) 1-junction (or common flow junction), (d) 1-junction with a different power propagation direction assignment.

Remark 14: The following matrices correspond to the kernel representation of a $\mathsf{0}$-junction

$$F = \begin{bmatrix} I_m & I_m & \cdots & I_m \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \; m = \dim \mathsf{F}_1 = \cdots = \dim \mathsf{F}_k,$$

$$E = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ I_m & -I_m & 0 & \cdots & 0 \\ 0 & I_m & -I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -I_m \end{bmatrix},$$

and it is straightforward to prove that $EF^\intercal + FE^\intercal = 0$ and that $\operatorname{rank}[F|E] = k \cdot m$ (see (7)).

Definition 15: A *1-junction* or *common flow junction* (see Fig.4.c) is a Dirac structure $\mathsf{D}_1 \subset \mathsf{F}_1 \oplus \mathsf{E}_1 \oplus \cdots \oplus \mathsf{F}_k \oplus \mathsf{E}_k$ defined by the following equations

$$\begin{aligned} f_1 = f_2 \ldots &= f_k \\ e_1 + e_2 + \cdots + e_k &= 0. \end{aligned} \tag{13}$$

Remark 16: The following matrices correspond to the kernel representation of a $\mathsf{1}$-junction

$$F = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ I_m & -I_m & 0 & \cdots & 0 \\ 0 & I_m & -I_m & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -I_m \end{bmatrix},$$

$$E = \begin{bmatrix} I_m & I_m & \cdots & I_m \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}.$$

Remark 17: Proposition 5 can be used to verify that $\mathsf{0}$ and $\mathsf{1}$-junctions are Dirac structures [26]. For example, $\mathsf{0}$-junctions are energy-conserving; indeed, if we name $e = e_1 = \cdots = e_k$, then

$$\sum_{i=1}^{k} \langle e_i \,|\, f_i \rangle = \sum_{i=1}^{k} \langle e \,|\, f_i \rangle = \left\langle e \,\middle|\, \sum_{i=1}^{k} f_i \right\rangle = \langle e \,|\, 0 \rangle = 0.$$

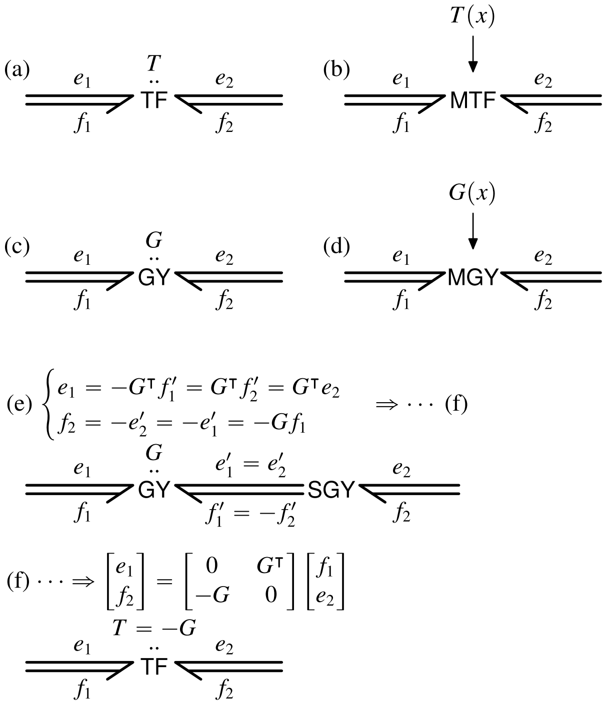
On the other hand, it is obvious that $\mathsf{0}$-junctions satisfy the rank condition. To show that, take k independent variables, say $e_1, f_2, \ldots, f_k$; the rest, $f_1, e_2, \ldots, e_k$, will be a linear combination of the previous ones. Then

$$\dim \mathsf{D}_0 = \dim (E_1 \oplus F_2 \oplus \cdots \oplus F_k) = \sum_{i=1}^{k} \dim F_i.$$

Definition 18: A 2-port *modulated transformer* MTF (see Fig.5.b) is a Dirac structure $\mathsf{D}_{\mathsf{MTF}} \subset \mathsf{F}_1 \oplus \mathsf{E}_1 \oplus \mathsf{F}_2 \oplus \mathsf{E}_2$ defined by the following constitutive relations

$$\begin{bmatrix} e_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & -T^\intercal(x) \\ T(x) & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ e_2 \end{bmatrix}, \tag{14}$$

where the elements of the matrix $T(x) \in \mathbb{R}^{\dim \mathsf{F}_2 \times \dim \mathsf{F}_1}$ are smooth functions on the manifold $\mathcal{X}$. If T does not depend on x then (14) defines a *constant* transformer TF (see Fig.5.a).



$$\text{(e)} \begin{cases} e_1 = -G^\intercal f_1' = G^\intercal f_2' = G^\intercal e_2 \\ f_2 = -e_2' = -e_1' = -G f_1 \end{cases} \Rightarrow \cdots \text{(f)}$$

$$\text{(f)} \cdots \Rightarrow \begin{bmatrix} e_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & G^\intercal \\ -G & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ e_2 \end{bmatrix}$$

FIGURE 5. (a) Constant transformer, (b) modulated transformer, (c) constant gyrator, (d) modulated gyrator and (e) the cascade composition of two gyrators is equivalent to a transformer (f).

Remark 19: The conditions of the kernel representation (see (7)) can be used to verify that the modulated transformer defines a Dirac structure. Indeed, (14) can be written

$$\underbrace{\begin{bmatrix} -T_{m_2\times m_1} & I_{m_2} \\ 0_{m_1} & 0_{m_1\times m_2} \end{bmatrix}}_{F} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0_{m_2\times m_1} & 0_{m_2} \\ I_{m_1} & T^{\mathsf{T}}_{m_1\times m_2} \end{bmatrix}}_{E} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = 0, \quad \begin{cases} m_1 = \dim \mathsf{F}_1 \\ m_2 = \dim \mathsf{F}_2, \end{cases}$$

$$\begin{aligned} &EF^{\mathsf{T}} + FE^{\mathsf{T}} \\ &= \begin{bmatrix} 0 & -TI_{m_1} + I_{m_2}T \\ -I_{m_1}T^{\mathsf{T}} + T^{\mathsf{T}}I_{m_2} & 0 \end{bmatrix} \\ &= 0, \\ &\operatorname{rank}\left[F|E\right] \\ &= \operatorname{rank}\begin{bmatrix} -T & I_{m_2} & 0 & 0 \\ 0 & 0 & I_{m_1} & T^{\mathsf{T}} \end{bmatrix} = m_1 + m_2. \end{aligned}$$

If $\dim \mathsf{F}_1 = \dim \mathsf{F}_2 = 1$ then $T(x)$ is a scalar and the equations that defines the transformer are $f_2 = Tf_1$ and $e_2 = -e_1/T$.

Definition 20: A 2-port *modulated gyrator* MGY (see Fig.5.d) is a Dirac structure $\mathsf{D}_{\mathsf{MGY}} \subset \mathsf{F}_1 \oplus \mathsf{E}_1 \oplus \mathsf{F}_2 \oplus \mathsf{E}_2$ defined by the following constitutive relations

$$\begin{bmatrix} e_1 \\ e_2 \end{bmatrix} = \begin{bmatrix} 0 & -G^{\mathsf{T}}(x) \\ G(x) & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \tag{15}$$

where the elements of the matrix $G(x) \in \mathbb{R}^{\dim \mathsf{F}_2 \times \dim \mathsf{F}_1}$ are smooth functions on the manifold $\mathcal{X}$. If G does not depend on x then (15) defines a *constant* gyrator GY (see Fig.5.c). If $\dim \mathsf{F}_1 = \dim \mathsf{F}_2 = 1$ then $G(x)$ is a scalar and the equations that defines the gyrator are $f_2 = -e_1/G$ and $e_2 = Gf_1$.

Remark 21: The following matrices correspond to the kernel representation of a gyrator

$$F = \begin{bmatrix} 0_{m_1} & G^{\mathsf{T}}_{m_1\times m_2} \\ -G_{m_2\times m_1} & 0_{m_2} \end{bmatrix},$$

$$E = \begin{bmatrix} I_{m_1} & 0_{m_1\times m_2} \\ 0_{m_2\times m_1} & I_{m_2} \end{bmatrix}, \quad \begin{cases} m_1 = \dim \mathsf{F}_1 \\ m_2 = \dim \mathsf{F}_2, \end{cases}$$

and is straightforward (see Remark 19) to prove: (Proposition 5.1) $EF^{\mathsf{T}} + FE^{\mathsf{T}} = 0$, and (Proposition 5.2) $\operatorname{rank}\left[F|E\right] = m_1 + m_2$.

The gyrator essentially interchanges the roles of effort and flow variables. The *symplectic gyrator* SGY is a special case of constant gyrator where $\dim \mathsf{F}_1 = \dim \mathsf{F}_2 = m$ and $G = I_m$. It can be used to define transformers (see Fig.5.e) and, in this sense, the gyrator is a more fundamental building block than the transformer.

B. BOUNDARY CONDITIONS, ENERGY STORAGE AND DISSIPATION

To complete the description of a port-Hamiltonian system, we to take account of the three fundamental phenomena related to the energy which happen in any non-isolated physical system:

1) The energy exchanged with the environment,
2) the energy storage in the system and,
3) the increase of the system entropy as its state evolves (second principle of thermodynamics [20], [34]).

Boundary conditions model the relation of a system with its environment and are defined with the *energy sources*.

Definition 22: A *flow source* $\mathsf{MSf} : f(t)$ modulated by $F(x)$ is a 1-port element than imposes the value $f = F(x)f(t)$ to the flow variable of a power port (see Fig.6). The symbol $\mathsf{Sf} : f(t)$ is used when the source does not depend on x. Likewise, an *effort source* $\mathsf{MSe} : e(t)$ modulated by $E(x)$ imposes the value $e = E(x)e(t)$ to the effort variable and the symbol $\mathsf{Se} : e(t)$ is used when the source does not depend on x.

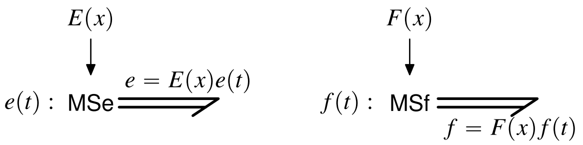


FIGURE 6. Modulated sources of effort (MSe) and flow (MSf).

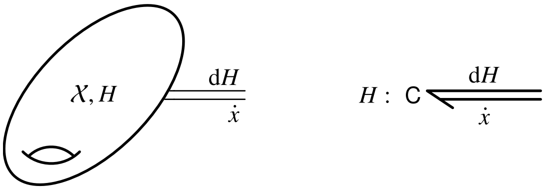


FIGURE 7. Storage element and its bond graph representation.

Definition 23: A *storage element* $\mathsf{C} : H$ is a 1-port element (see Fig.7) defined by the pair $(\mathcal{X}, H)$ where $\mathcal{X}$ is an n-dimensional smooth manifold and $H \in C^{\infty}(\mathcal{X})$ is the Hamiltonian that models the system energy. The flow and effort variables of the port are the vector field $\dot{x} \in T\mathcal{X}$ and the covector field $\mathrm{d}H \in T^{*}\mathcal{X}$ respectively.

Remark 24: Connecting the storage element $\mathsf{C} : H$ with the Dirac structure $\mathcal{D} \subset T\mathcal{X} \oplus T^{*}\mathcal{X} \oplus \mathsf{F} \oplus \mathsf{E}$ we obtain the port-Hamiltonian system $(\mathcal{X}, \mathsf{F}, \mathcal{D}, H)$.

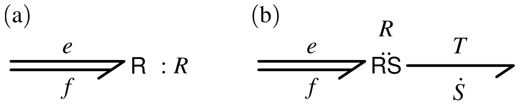


FIGURE 8. Isothermal (a) and non-isothermal (b) resistors.

Definition 25: A *resistor* $\mathsf{R} : R$ is a 1-port element (see Fig.8.a) that irreversibly transforms non-thermal energy into heat and is defined by the relations

$$\begin{aligned} \Phi_R(f, e) &= 0 \\ P_R &= \langle e|f \rangle > 0. \end{aligned} \tag{16}$$

Remark 26: RS resistors (see Fig.8.b) were introduced by [41] and, they are used for modeling non-isothermal energy dissipations, i.e., heat generation where the change of temperature concerns the system behavior. Temperature T and entropy flow $\dot{S}$ are the effort and flow variables employed in the thermal domain. Thus the power balance of an RS resistor is $\langle e|f\rangle = \langle T|\dot{S}\rangle$, and it is congruent with the second principle of thermodynamics ($\dot{S} > 0$, i.e., increase of the entropy of a system as its state evolves) because $\langle e|f\rangle > 0$ (see (16)) and $T > 0$.

IV. OBJECT ORIENTED LIBRARY (pHlib) OF PORT-HAMILTONIAN SYSTEM BUILDING BLOCKS

The pHLib library consists of two packages: Dirac and pHS. The Dirac package implements the concepts of power port, bond, Dirac structure, and junction structures (Dirac structures that have an equivalent bond graph). The following listings show the Modelica implementation of the previous elements:

```
connector Port "Power port (Definition 1)"
  Integer dir "Power direction: +1 (incoming power), −1 (outgoing power)
    ";
  parameter Integer dimPort "Port dimension"; // dim 𝒳
  replaceable Real e[dimPort] "Effort covector"; // e ∈ T*_x 𝒳
  replaceable Real f[dimPort] "Flow vector"; // f ∈ T_x 𝒳
end Port;

connector OneDimensionalPort
  extends Port(dimPort=1);
  extends Icons.OneDimensionalPort;
end OneDimensionalPort;

connector MultiDimensionalPort
  extends Port;
  extends Icons.MultiDimensionalPort;
end MultiDimensionalPort;
```

Remark 27: Ports (in port-Hamiltonian systems) and connectors (in Modelica) define the interfaces to form complex systems by composition of simple ones. The facility to define *parameterized* classes in Modelica allows a general implementation of the power port concept, which will later be instantiated according to the dimension of each port.

```
class Bond "Bond graph (Fig. 4)"
  Modelica.SIunits.Power P;
  replaceable connector BasePort=Port;
  BasePort from;
  BasePort to;
equation
  from.dir = −1;
  to.dir = +1;
  to.e = from.e;
  to.f = from.f;
  P = to.e * to.f; // P = ⟨e|f⟩
end Bond;

class OneDimensionalBond = Bond (
  redeclare replaceable connector BasePort = OneDimensionalPort);

class MultiDimensionalBond = Bond(
    redeclare replaceable connector BasePort = MultiDimensionalPort);
end MultiDimensionalBond;
```

Remark 28: The class Bond (a concept taken from the bond graph methodology) is used to connect the ports of the subsystems that make up a complex system. Each bond has an input port (from) and an output port (to) that set the power propagation direction. The facility to define *replaceable* classes in Modelica allows a general implementation of the bond concept, which will later be instantiated according to the class of port to be connected to. This will allow, for example, to grant a physical meaning to the variables of the model, replacing the type Real used in the general definitions, by types of the physical domain to which the model belongs. The Modelica instruction connect is used to associate the ports of a bond with the ports of the subsystems it connects (see Fig.9 and the associated listing pHSExample).

```
partial class DiracStructure "Dirac structure (Definition 3)"
  parameter Integer nPorts "Number of ports";
  parameter Integer dimPorts[nPorts] "Dimension of each port";
  replaceable Port port[nPorts](dimPort={dimPorts[i] for i in 1:nPorts});
  Modelica.SIunits P;
equation
  P = sum(port[i].e * port[i].f * port[i].dir for i in 1:nPorts);
  // Checking for Proposition 5.1
  assert(abs(P) < 1E−10,
    "Energy conservation violated in a Dirac structure.
     The system of equations may need to be rescaled to facilitate
     the convergence of the numerical solver.");
end DiracStructure;
```

Remark 29: Dirac structures are the fundamental components to define the power conserving connection structure between the different parts of a port-Hamiltonian system. DiracStructure is the base class (or superclass) in a class hierarchy that defines Dirac structures. The hierarchical composition property (Proposition 8) is achieved with the array of ports defined in DiracStructure, so that a complex Dirac structure can be constructed by connecting simple Dirac structures. The structure is also checked for energy balance (Proposition 5.1).

```
class ZeroJunction "0−Junction (Definition 13)"
  extends DiracStructure;
equation
  for i in 2:nPorts loop
    port[i].e = port[1].e;
  end for;
  sum(port[i].f * port[i].dir for i in 1:nPorts) = zeros(dimPorts[1]);
end ZeroJunction;
```

```
class OneJunction "1−Junction (Definition 15)"
  extends DiracStructure;
equation
  sum(port[i].e * port[i].dir for i in 1:nPorts) = zeros(dimPorts[1]);
  for i in 2:nPorts loop
    port[i].f = port[1].f;
  end for;
end OneJunction;
```

Remark 30: 0-junctions and 1-junctions are particular cases of Dirac structures. Using Modelica terminology we say that the classes ZeroJunction and OneJunction are *subclasses* of DiracStructure. Since derived classes inherit the characteristics of their parent class, including their equations, we have only added the equations (12) and (13) in the definition of ZeroJunction and OneJunction respectively. The same applies to the classes Transformer and Gyrator shown below.

```
class Transformer "Transformer (Definition 18)"
  extends DiracStructure(nPorts=2, label="TF");
  replaceable Real T[dimPorts[2], dimPorts[1]];
equation
  port[1].e * port[1].dir = −transpose(T) * port[2].e * port[2].dir;
  port[2].f = T * port[1].f;
end Transformer;
```

```
class Gyrator "Gyrator (Definition 20)"
  extends DiracStructure(nPorts=2, label="GY");
  replaceable Real G[dimPorts[2], dimPorts[1]];
equation
  port[1].e * port[1].dir = −transpose(G) * port[2].f * port[2].dir;
  port[2].e = G * port[1].f;
end Gyrator;
```

The pHS package implements the concepts of boundary conditions, energy storage and dissipation. The following three listings show the Modelica implementation of the previous elements.

```
class Se "Effort source (Definition 22)"
  replaceable Real e[port.dimPort] "Effort vector";
equation
  port.e = e ∗ (−port.dir);
end Se;

class Sf "Flow source"
  replaceable Real f[port.dimPort] "Flow vector";
equation
  port.f = f ∗ (−port.dir);
end Sf;
```

```
class C "Storage element (Definition 23)"
  replaceable parameter Real x0[port.dimPort] "Initial state"; // $x_0$
  replaceable Real x[port.dimPort](start=x0) "State"; // $x \in \mathcal{X}$
  replaceable Real dH[port.dimPort] "Differential of H";
  // $\mathrm{d}H : T\mathcal{X} \to \mathbb{R}$,
  // $\mathrm{d}H(x) : T_x\mathcal{X} \to T_{H(x)} \cong \mathbb{R}$, i.e., $\mathrm{d}H \in T^*\mathcal{X}$
  // $\mathrm{d}H$ has to be defined in the derived class.
equation
  {port.e, port.f} = {dH, der(x)};
end C;

class Linear "Linear storage element"
  replaceable Real ZMatrix[port.dimPort, port.dimPort];
  extends pHS.Storage.C;
equation
  ZMatrix ∗ dH = x; // $H(x) = \frac{1}{2}x^{\intercal}Z^{-1}x, \quad \partial_x H = Z^{-1}x$
end Linear;
```

Remark 31: The class C is the base class in the hierarchy for defining storage elements. The effort and flow variables of the power port of this component are $\mathrm{d}H$ and $\dot{x}$, respectively. The effort $\mathrm{d}H$ is calculated from the Hamiltonian H and must be defined for each class derived from C (this is the case, for example, of the class Linear, where $\mathrm{d}H = Z^{-1}x$). The flow $\dot{x}$ is obtained with the Modelica operator der(x), which represents the time derivative of the state variable x.

While the Modelica language allows to choose state variables when defining a model, the port-Hamiltonian approach facilitated this task, because state variables are imposed by the storage elements. In addition, this way of modeling allows to know the function that defines the total energy of the system. Indeed, as energy is an extensive magnitude, the total energy of a system is the sum of the energy of its components (see Proposition 9.2).

```
import SI = Modelica.SIunits;
class R "Resistor (Definition 25)"
replaceable Real PhiR[port.dimPort];
equation
PhiR = zeros(port.dimPort); // $\Phi_R(e, f) = 0$
end R;

class RS "Resistor with entropy generation (Remark 26)"
replaceable Real PhiR;
replaceable Real e = port[1].e[1];
replaceable Real f = port[2].f[1];
SI.Temperature T2 = port[2].e[1];
SI.EntropyFlowRate dot_S2 = port[2].f[1];
Port port[2];
equation
dot_S2 ∗ T2 = PhiR; // $\dot{Q} = \langle T_2 | \dot{S}_2 \rangle = \Phi_R(e, f)$
end RS;
```

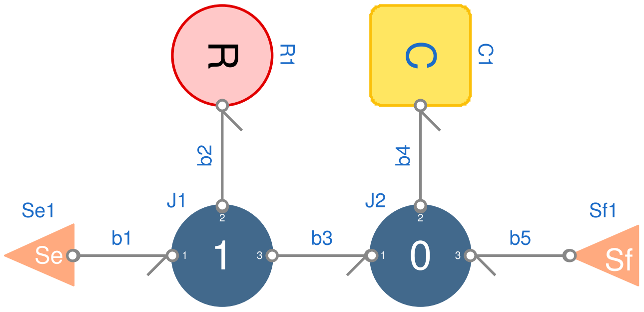


FIGURE 9. **Example of a port-Hamiltonian system with the proposed representation.**

In Fig.9 we can see the graphical representation of a port-Hamiltonian system composed of two sources (Se1, Sf1), a storage element (C1) and a dissipative element (Re1), all of them connected through a Dirac structure composed of two junctions (J1, J2). The proposed symbology for the pHlib library is slightly different from that used in traditional bond graphs but is easily recognizable. The following listing shows the Modelica code corresponding to this system, where the connection of the building blocks can be seen by means of OneDimensionalBond components (b1,…,b5) and the instruction connect:

```
model pHSExample
  Sources.Se Se1(redeclare OneDimensionalPort port);
  ThreePortOneJunction J1(
     redeclare OneDimensionalPort port1,
     redeclare OneDimensionalPort port2,
     redeclare OneDimensionalPort port3);
  ThreePortZeroJunction J2(
     redeclare OneDimensionalPort port1,
     redeclare OneDimensionalPort port2,
     redeclare OneDimensionalPort port3);
  Sources.Sf Sf1(redeclare OneDimensionalPort port);
  Dissipative.R R1(redeclare OneDimensionalPort port);
  Storage.C C1(redeclare OneDimensionalPort port, x0={0});
  OneDimensionalBond b1, b2, b3, b4, b5;
equation
  Se1.e = {5};
  R1.PhiR = R1.port.e − 3 ∗ R1.port.f;
  C1.dH = 2 ∗ C1.x;
  Sf1.f = {1};
  connect(Se1.port, b1.from); connect(b1.to, J1.port1);
  connect(J1.port2, b2.from); connect(b2.to, R1.port);
  connect(J1.port3, b3.from); connect(b3.to, J2.port1);
  connect(J2.port2, b4.from); connect(b4.to, C1.port);
  connect(Sf1.port, b5.from); connect(b5.to, J2.port3);
end pHSExample;
```

V. EXAMPLE I: SIMPLE MECHANICAL SYSTEMS

Definition 32: [1, p.187] In Hamiltonian mechanics, a simple system is the tuple $(\mathcal{X}, \omega, H)$ where the symplectic manifold $(\mathcal{X}, \omega)$ is the state space (also called phase space), $H \in C^{\infty}(\mathcal{X})$ is the Hamiltonian, and the equation $\dot{x} = (\mathrm{d}H)^{\sharp}$ defines the system dynamics.[1]

---

[1] $X_H = (\mathrm{d}H)^{\sharp}$ is the Hamiltonian vector field defined by H, thus it is the unique vector field that satisfies $\omega(X_H, Y) = \mathrm{d}H(Y)$ for every vector field $Y \in \mathfrak{X}(\mathcal{X})$ [33, p.574].

Let $\mathcal{Q}$ be an n-dimensional smooth manifold and $T^*\mathcal{Q}$ its cotangent bundle with the symplectic canonical form ω, then $(T^*\mathcal{Q}, \omega, H)$ is a Hamiltonian system and, if we write the states in canonical coordinates $x = (q, p) \in T^*\mathcal{Q}$; then $\omega = \sum_{i=1}^{n} \mathrm{d}q^i \wedge \mathrm{d}p^i$, $(\mathrm{d}H)^\sharp = \sum_{i=1}^{n} \left(\partial_{p^i}H \cdot \partial_{q^i} - \partial_{q^i}H \cdot \partial_{p^i}\right)$, and the equations of the system can be written

$$\begin{bmatrix}\dot{q}\\ \dot{p}\end{bmatrix} = \begin{bmatrix}\partial_p H\\ -\partial_q H\end{bmatrix} = \overbrace{\begin{bmatrix}0 & I_n\\ -I_n & 0\end{bmatrix}}^{J_{2n}} \begin{bmatrix}\partial_q H\\ \partial_p H\end{bmatrix} \Rightarrow |\dot{x}\rangle = J_{2n}\, |\mathrm{d}H\rangle\,.$$

This corresponds to a port-Hamiltonian system with a Dirac structure whose kernel representation is $F = -I_{2n}$ and $E = J_{2n}$ (see Fig.10).

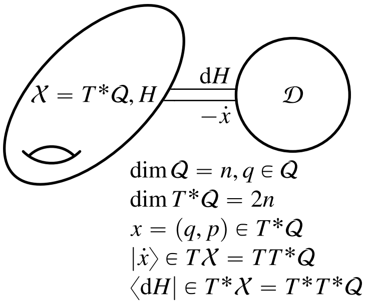


FIGURE 10. Port-Hamiltonian representation of a mechanical system. For local coordinates (q^i) in a neighborhood of $q \in \mathcal{Q}$, the coordinates of $|\dot{x}\rangle_{(q,p)}$ and $\langle \mathrm{d}H|_{(q,p)}$ are $[\dot{q}, \dot{p}]^\top$ and $[\partial_q H, \partial_p H]$ respectively.

In mechanics, $q \in \mathcal{Q}$ is called *generalized displacement* and $p \in T_q^*\mathcal{Q}$ *generalized momentum*. In translational mechanics, the flow $\dot{x}$ and effort $\mathrm{d}H$ variables have the physical dimensions $[\dot{q}, \dot{p}]$≡[*velocity*, *force*] and $[\partial_q H, \partial_p H]$≡[*force*, *velocity*] respectively. Hence, the dual product $\langle \mathrm{d}H|\dot{x}\rangle$ has the dimension of *force*×*velocity*+*velocity*×*force*=*power*. The Hamiltonian $H(q, p) = V(q) + K(p)$ of simple mechanical system has two components: the *potential energy* $V(q)$ that does not depend on the momentum, and the *kinetic energy* $K(p)$ that does not depend on the displacement [27, p.24]; and, because of this distinction, two different types of storage elements were defined in the bond graph terminology: the *capacitor* or *compliance* (C-element) that stores potential energy, and the *inertia* or *inertance* (I-element) that stores kinetic energy. In traditional bond graph modeling of translational mechanical system, the variables of flow f and effort e have physical dimension of velocity and force respectively, a C-store is defined with the relations $f = \dot{q}$, $e = \Phi_C(q) = \partial_q V(q)$, and an I-store with the relations $f = \Phi_I(p) = \partial_p K(p)$, $e = \dot{p}$. But in the general framework of port-Hamiltonian systems, velocities and forces can be variables of flow and effort indistinctly, and there is no reason to distinguish between C and I storage elements, except for procedural convenience, i.e., to facilitate the writing of the system equations. Moreover, every I-store can be defined from a C-store connected to a symplectic gyrator (see Fig.11).

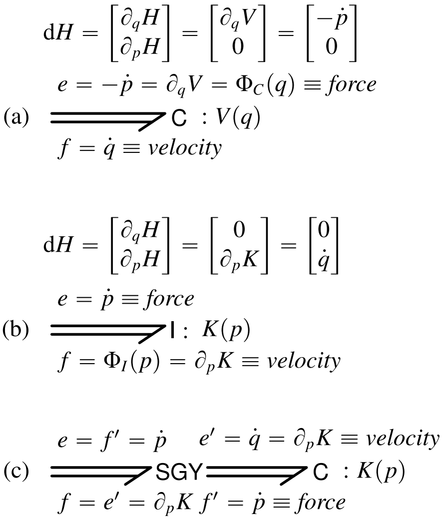


FIGURE 11. Simple mechanical systems: (a) C-element that stores potential energy, (b) I-element that stores kinetic energy and (c) its equivalent C-element.

A. MODELICA IMPLEMENTATION

The class SimpleSystem (see Fig.12 for a graphical representation) models the behavior of a simple mechanical system and is part of the package Mechanics.Systems:

```
class SimpleSystem // Fig. 10 and Fig. 12
  parameter Integer dimQ "Number of degrees of freedom of the system"
    ;
  parameter Real q0[dimQ] "Initial displacement";
  parameter Real p0[dimQ] "Initial momentum";
  Real q[dimQ]=storage.x[1:dimQ] "Generalized displacement";
  Real p[dimQ]=storage.x[dimQ + 1:2*dimQ] "Generalized momentum";
  Real dH[2*dimQ]=storage.dH "The dimension of the phase space is
    twice the number of degrees of freedom.";
  pHS.Storage.C storage(
    redeclare MultiDimensionalPort port(dimPort=2*dimQ),
    x0={(if i <= dimQ then q0[i] else p0[i - dimQ]) for i in 1 : 2*dimQ})
  pHS.Storage.C storage(
    redeclare replaceable MultiDimensionalPort port(dimPort=2*dimQ));
  TwoPortSlicer Slc(
    redeclare MultiDimensionalPort port1(dimPort=dimQ),
     redeclare MultiDimensionalPort port2(dimPort=dimQ),
     redeclare MultiDimensionalPort port3(dimPort=2*dimQ));
  Dirac.Components.SGY SGY(
    redeclare MultiDimensionalPort port1(dimPort=dimQ),
    redeclare MultiDimensionalPort port2(dimPort=dimQ));
  MultiDimensionalBond b1(
    redeclare connector BasePort = MultiDimensionalPort(dimPort=
      dimQ));
  MultiDimensionalBond b2(
    redeclare connector BasePort = MultiDimensionalPort(dimPort=
      dimQ));
  MultiDimensionalBond b3(
    redeclare connector BasePort = MultiDimensionalPort(dimPort=2*
      dimQ));
equation
  connect(storage.port, b3.from); connect(b3.to, Slc.port3);
  connect(Slc.port1, b1.from); connect(b1.to, SGY.port1);
  connect(Slc.port2, b2.from); connect(b2.to, SGY.port2);
end SimpleSystem;
```

The component Slc shown in Fig.12 is a *two port slicer* introduced to separate the $2n$-dimension port $\langle dH|\dot{x}\rangle = \langle(\partial_q H, \partial_p H)|(\dot{q}, \dot{p})\rangle$ into two n-dimension ports $\langle \partial_q H|\dot{q}\rangle$ and $\langle \partial_p H|\dot{p}\rangle$. The implementation of this component is as follows:

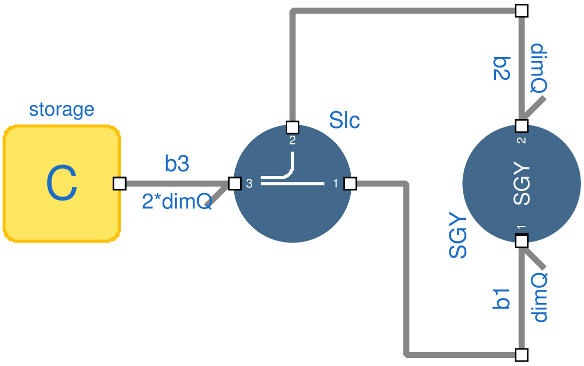


FIGURE 12. Simple mechanical systems.

```
class PortSlicer "Port slicer"
  extends DiracStructure;
equation
  for i in 1:nPorts−1 loop
    port[nPorts].e[sum(port[i].dimPort for j in 1:i−1)+1:sum(port[i].
      dimPort for j in 1:i)] = port[i].e;
    port[nPorts].f[sum(port[i].dimPort for j in 1:i−1)+1:sum(port[i].
      dimPort for j in 1:i)] = port[i].f;
  end for;
  //
  for i in 1:nPorts−1 loop
    assert(port[i].dir <> port[nPorts].dir,
      "The directionality of the largest port of a portSlicer has to be
        opposite to that of other ports.");
  end for;
end PortSlicer;
```

B. IMPLEMENTATION OF THE N-BODY PROBLEM

The N-body problem studies the movement in $\mathbb{R}^3$ of N matter particles subjected to gravitational attraction forces. Each body $i \in 1, \ldots, N$ has a mass m_i, a position $q_i = [q_i^1, q_i^2, q_i^3]^\mathsf{T}$ and a momentum $p_i = [p_i^1, p_i^2, p_i^3]^\mathsf{T} = m_i \dot{q}_i$. With the notation $M = \mathrm{diag}(m_1, \ldots, m_n)$, $q = [q_1, \ldots, q_N]^\mathsf{T}$ and $p = [p_1, \ldots, p_N]^\mathsf{T}$; the matrix representation of x, H, and $\mathrm{d}H$ is

$$\begin{aligned} x &= \begin{bmatrix} q \\ p \end{bmatrix} \in \mathbb{R}^{6N}, \\ H(q,p) &= \tfrac{1}{2} p^\mathsf{T} M^{-1} p - \sum_{1 \le i < j \le N} G \frac{m_i m_j}{\|q_i - q_j\|}, \\ \mathrm{d}H &= M^{-1} p + \sum_{1 \le i \ne j \le N} G m_i m_j \frac{q_i - q_j}{\|q_i - q_j\|^3}. \end{aligned} \tag{17}$$

The partial model nBodyProblem models the behavior of the N-body problem and is part of the package Mechanics.Examples. As you can see below, the template is parameterized with the Sunits package that defines the system of units used. By default, Sunits takes the value of the International System of Units (SI), available in the package Mechanics.Examples.SIunits.

```
partial model nBodyProblem
  replaceable package Sunits = Mechanics.Examples.SIunits;
  //
  parameter Integer nBodies "Number of bodies";
  parameter Sunits.Mass M[nBodies] "Mass of each body";
  parameter Sunits.Position[3] q0[nBodies] "Initial positions";
  parameter Sunits.Velocity[3] v0[nBodies] "Initial velocities";
  //
  Mechanics.Systems.SimpleSystem System(dimQ = 3∗nBodies,
    q0 = {if rem(i,3)==0
          then q0[div(i,3),3]
          else q0[div(i,3)+1,rem(i,3)] for i in 1:3∗nBodies},
    p0 = {if rem(i,3)==0
          then M[div(i,3)]∗v0[div(i,3),3]
          else M[div(i,3)+1]∗v0[div(i,3)+1,rem(i,3)] for i in 1:3∗nBodies});
  //
  Sunits.Position[3] q[nBodies];
  Sunits.Momentum[3] p[nBodies];
  //
  Sunits.Force[3] dH_dq[nBodies]; // ∂H/∂q
  Sunits.Velocity[3] dH_dp[nBodies]; // ∂H/∂p
equation
  for i in 1:nBodies loop
    dH_dq[i] = Sunits.G∗(
     sum(M[i]∗M[j]∗(q[i]−q[j])/sqrt((q[i]−q[j])∗(q[i]−q[j]))^3 for j in 1:i
       −1) +
     sum(M[i]∗M[j]∗(q[i]−q[j])/sqrt((q[i]−q[j])∗(q[i]−q[j]))^3 for j in i+1:
       nBodies));
    dH_dp[i] = p[i]/M[i];
  end for;
  for i in 0:nBodies−1 loop
    System.q[3∗i+1:3∗i+3] = q[i+1];
    System.p[3∗i+1:3∗i+3] = p[i+1];
    // Marshaling of the ∂_qH and ∂_pH components to form the covector
    // dH = [∂_qH, ∂_pH].
    // [..., ∂H/∂q_i^1, ∂H/∂q_i^2, ∂H/∂q_i^3, ...;    1 ≤ i ≤ N
    System.dH[3∗i+1:3∗i+3] = dH_dq[i+1];
    // ..., ∂H/∂p_i^1, ∂H/∂p_i^2, ∂H/∂p_i^3, ...];    1 ≤ i ≤ N
    System.dH[3∗nBodies+3∗i+1:3∗nBodies+3∗i+3] = dH_dp[i+1];
  end for;
```

TABLE 1. Masses and initial state for the simulation of the Sun-Earth-Moon system.

	Mass (kg)	Initial position (m)	Initial velocity (m/s)
Sun	1.98892×10^{30}	[0, 0, 0]	[0, 0, 0]
Earth	5.97420×10^{24}	$[1.521 \times 10^{11}, 0, 0]$	[0, 29290, 0]
Moon	7.34200×10^{22}	$[1.52484 \times 10^{11}, 0, 0]$	[0, 30310, 0]

C. SIMULATION OF THE SUN-EARTH-MOON SYSTEM

The Sun-Earth-Moon system is a particular case formed by three bodies that can be implemented very easily from the nBodyProblem partial model using the instantiation mechanism of Modelica as shown below in the model SunEarthMoon. The data used to carry out the simulation can be seen in Table 1. These data are transformed in the model to fit the Astronomical Unit System (AUS) [29], [35], defined in the ASunits package, where the units of length, mass and time are: the astronomical unit (1 AU=149597870700 m), the solar mass (1 $\mathrm{M}_\odot$ = 1.98892×10^{30} kg), and the day duration (1 day=86400 s). Fig.13 and Fig.14 show simulation results for time spans of 2 and 10 years.

```
model SunEarthMoonSystem
  extends nBodyProblem(
    redeclare package Sunits = Mechanics.Examples.ASunits,
    nBodies=3,
    M={Sunits.mSun, Sunits.mEarth, Sunits.mMoon},
    q0=[0, 0, 0;
        1.52100E+11, 0, 0;
        1.52484E+11, 0, 0] / Sunits.unitOfLength,
    v0=[0, 0, 0;
        0, 29290, 0;
        0, 29290 + 1020, 0] ∗ Sunits.unitOfTime / Sunits.unitOfLength);
  Sunits.Length distanceSunEarth = sqrt((q[1] − q[2]) ∗ (q[1] − q[2]));
  Sunits.Length distanceSunMoon = sqrt((q[1] − q[3]) ∗ (q[1] − q[3]));
  Sunits.Length distanceEarthMoon = sqrt((q[2] − q[3]) ∗ (q[2] − q[3]));
end SunEarthMoonSystem;
```

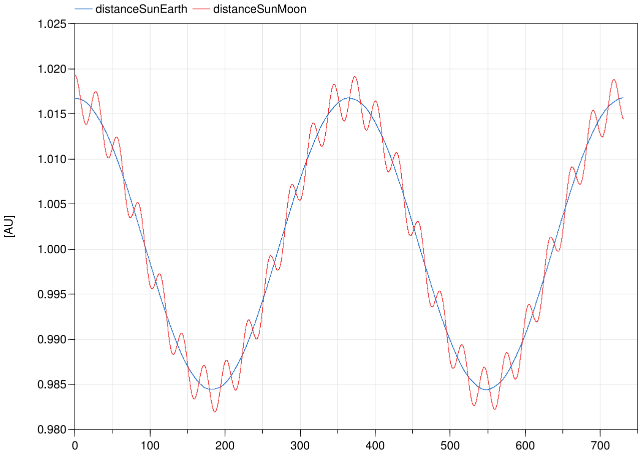


FIGURE 13. Simulation of the Sun-Earth-Moon system for a time span of 2 years.

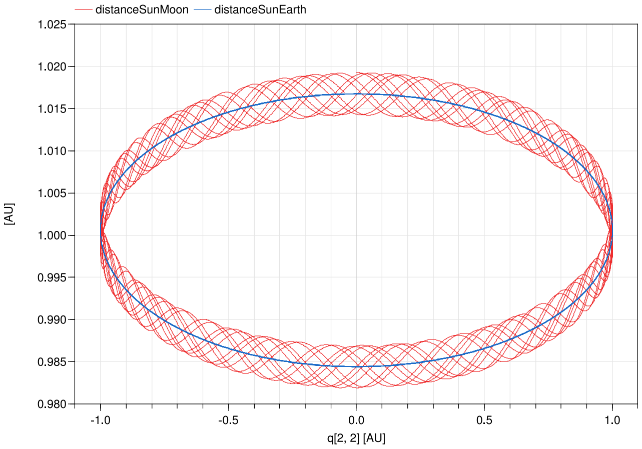


FIGURE 14. Simulation of the Sun-Earth-Moon system for a time span of 10 years. The abscissa axis is the *y* coordinate of the Earth.

```
package ASunits "Astronomical system of units"
  import IAU = Mechanics.Examples.IAU2009;
  //
  type AngularMomentum = Real (
    quantity="AngularMomentum", unit="MS.AU2/day");
  type Energy = Real (quantity="Energy", unit="MS.AU2/day2");
  type Force = Real (quantity="Force", unit="MS.AU/day2");
  type Length = Real (quantity="Length", unit="AU");
  type Mass = Real (quantity="Mass", unit="MS");
  type Momentum = Real (quantity="Momentum", unit="MS.AU/day");
  type Position = Length;
  type Power = Real (quantity="Power", unit="MS.AU2/day3");
  type Velocity = Real (quantity="Velocity", unit="AU/day");
  //
  constant Length unitOfLength = IAU.AU;
  constant Mass unitOfMass = IAU.mSun;
  constant Real unitOfTime = IAU.day;
  //
  constant Real G(unit="MS/(AU3.day2)") =
    IAU.G ∗ (unitOfMass ∗ unitOfTime^2) / unitOfLength^3 "Constant of
      gravitation";
  constant Mass mSun = IAU.mSun/unitOfMass "Solar mass";
  constant Mass mEarth = IAU.mEarth/unitOfMass "Mass of the Earth";
  constant Mass mMoon = IAU.mMoon/unitOfMass "Lunar mass";
end ASunits;
```

```
package IAU2009 "IAU (International Astronomical Union) 2009 system
  of astronomical constants  [35]"
  import SI = Modelica.SIunits;
  constant Real G(unit="m3/(kg.s2)") = 6.67428E−11 "Constant of
    gravitation";
  constant SI.Mass mSun = 1.98892E+30 "Solar mass";
  constant SI.Mass mEarth = 5.9742E+24 "Mass of the Earth";
  constant SI.Mass mMoon = 7.342E+22 "Lunar mass";
  constant SI.Length au = 149597870700 "Astronomical Unit  [29]";
  constant Real second = 1;
  constant Real minute = 60 ∗ second;
  constant Real hour = 60 ∗ minute;
  constant Real day = 24 ∗ hour;
  constant Real year = 365.25 ∗ day;
end IAU2009;
```

D. LOCAL ERROR ANALYSIS

The Sun-Earth-Moon model has been numerically integrated with the DASSL solver used by the Dymola simulation tool. DASSL ([8], [38]) is a variable-order, variable-step numerical integration method that has proven to be able to solve with robustness and stability an important set of non-linear ODEs and DAEs, like those presented in this document.

The global error made in the integration of a DAE system is the difference between the true solution of the initial value problem and the computed solution in the whole experiment. The numerical methods used by Dymola are coded to control the local error instead of the global error.

The maximum local error value e_ℓ in each integration step can be expressed as

$$|e_\ell| < tol_{rel} \cdot (|x^i| + nominal(x^i)) \tag{18}$$

where tol_{rel} is the relative tolerance that the tool passes as argument to the numerical integrator, x^i is the i-component of the state vector, and $nominal(x^i)$ is the Modelica attribute nominal declared for the variable x^i (Dymola normally uses a default value of 1). Details about this computation can be found in [18, p.649].

In the Sun-Earth-Moon model the distances are in the order of 10^{11} m and the momenta in the order of 10^{29} kg·m/s. If the solvers work with relative tolerances up to 10^{-12} then $e_\ell(q) < 0.1$ and $e_\ell(p) < 10^{17}$ (when expressed in SI units and the attribute nominal has the default value of 1):

$$\begin{aligned}|e_\ell(q)| &< 10^{-12} \cdot (|10^{11}| + 1) \approx 10^{-1}\\ |e_\ell(p)| &< 10^{-12} \cdot (|10^{29}| + 1) \approx 10^{17}.\end{aligned}$$

So, the state $x = (q, p)$ has to be expressed in a suitable system of units to limit the maximum local error at each integration step. This is the case with the AUS units, where the distances are in the order of 1 AU and the momenta in the order of 10^{-8} $M_\odot$·AU/day. In both cases $e_\ell < 10^{-12}$ with the value 1 as default value for nominal:

$$\begin{aligned}|e_\ell(q)| &< 10^{-12} \cdot (|1| + 1) \approx 10^{-12}\\ |e_\ell(p)| &< 10^{-12} \cdot (|10^{-8}| + 1) \approx 10^{-12}.\end{aligned}$$

VI. EXAMPLE II: ELECTRICAL NETWORKS

The building blocks defined in section section III have a one-to-one correspondence with the elements used in electrical lumped parameter networks (see Table 2).

TABLE 2. Port-Hamiltonian modeling of electrical networks.

Element/Law	pHS block
charge q magnetic flux linkage λ	state $x = (q, \lambda) \in \mathbb{R}^{n+m}$
$\left\{\begin{array}{ll}\text{voltage} & v \\ \text{current} & i\end{array}\right.$	port variables $\left\{\begin{array}{ll}\text{effort} & e \\ \text{flow} & f\end{array}\right.$
node (common voltage) Kirchhoff's current law (KCL): $\sum_{i_\alpha \in \text{node}_\beta} i_\alpha = 0$	0-junction (common effort) $\sum_{f_\alpha \in \text{0-junction}_\beta} f_\alpha = 0$
loop (common current) Kirchhoff's voltage law (KVL): $\sum_{v_\alpha \in \text{loop}_\beta} v_\alpha = 0$	1-junction (common flow) $\sum_{e_\alpha \in \text{1-junction}_\beta} e_\alpha = 0$
ideal transformer $n : 1$ ideal gyrator [40] $n : 1$	transformer TF : n gyrator GY : n
voltage generator $v(t)$ current generator $i(t)$ capacitor inductor resistor R	effort source Se : $v(t)$ flow source Sf : $i(t)$ storage C : $H_C(q)$ storage I : $H_L(\lambda)$ resistor R : R

Definition 33: A storage element with electrical and magnetic parts is the pair $(\mathbb{R}^{n+m}, H)$ where $x = (q, \lambda) \in \mathbb{R}^{n+m}$ is the state, $q \in \mathbb{R}^n$ is the electric charge, $\lambda \in \mathbb{R}^m$ is the magnetic flux linkage, and $H \in C^\infty(\mathbb{R}^{n+m})$ is the Hamiltonian.

In electrical networks, the flow $\dot{x}$ and effort $\mathrm{d}H$ variables have the physical dimensions $[\dot{q}, \dot{\lambda}] \equiv [\mathit{current}, \mathit{voltage}]$ and $[\partial_q H, \partial_\lambda H] \equiv [\mathit{voltage}, \mathit{current}]$ respectively. Hence, the dual product $\langle \mathrm{d}H | \dot{x} \rangle$ has the dimension of $\mathit{current} \times \mathit{voltage} + \mathit{voltage} \times \mathit{current} = \mathit{power}$.

The Hamiltonian $H(q, \lambda) = H_C(q) + H_L(\lambda)$ of simple electrical network has two components: the *electrical energy* $H_C(q)$, and the *magnetic energy* $H_L(\lambda)$. Because of this distinction, two different types of storage elements were defined in the bond graph terminology: the *capacitor* (C-element) that stores electrical energy and the *inductor* (I-element) that stores magnetic energy (compare with simple mechanical systems section V).

In traditional bond graph modeling of electrical networks, the variables of flow f an effort e have physical dimension of current (i) and voltage (v) respectively: a C-store is defined with the relations $i_C = \dot{q}$, $v_C = \Phi_C(q) = \partial_q H_C(q)$; and a I-store with the relations $v_L = \dot{\lambda}$, $i_L = \Phi_I(\lambda) = \partial_\lambda H_L(\lambda)$. But in the general framework of port-Hamiltonian systems, currents and voltages can be variables of flow and effort indistinctly and there is no reason to distinguish between C and I storage elements, except for procedural convenience, i.e., to facilitate the writing of the system equations. Moreover, every capacitor can be defined from an inductor connected to a symplectic gyrator (see Fig.15).

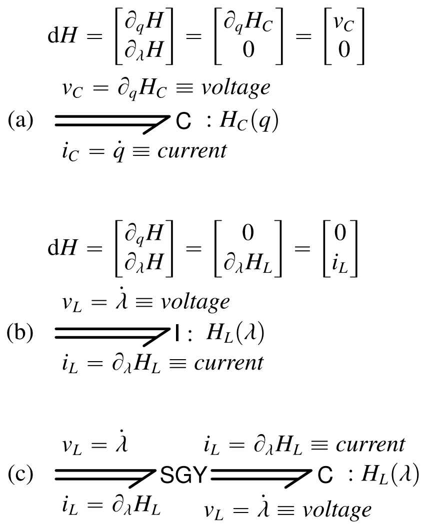


FIGURE 15. Storage elements: (a) C-element that stores electrical energy, (b) I-element that stores magnetic energy and (c) its equivalent C-element.

A. LEGENDRE TRANSFORMATION OF THE HAMILTONIAN

While currents and voltages can be easily measured in an electrical network, the state variables that we have introduced in our model, electric charges and magnetic flux linkages, have to be inferred from such currents and voltages. Note also that the design of control systems is facilitated by working with models whose state variables are observable or even directly measurable; this fact justifies the use of models of electrical networks where currents and voltages are the state variables. The appropriate tool to perform this formulation is the *Legendre transformation*.

Let $Y(x^1, \ldots, x^\ell, x^{\ell+1} \ldots, x^r) = Y(x_\ell, x_{r-\ell})$ be a function with partial derivatives $p = \partial Y / \partial x_\ell$. The Legendre transformation of Y with respect to x_ℓ is a function of the independent variables $(p, x_{r-\ell}) = (p^1, \ldots, p^\ell, x^{\ell+1}, \ldots, x^r)$, defined [2, p.64]

$$\begin{aligned} Y(p, x_{r-\ell}) &= p^{\mathsf{T}} x_\ell - Y(x) \\ &= \sum_{k=1}^{\ell} p^k x^k - Y(x^1, \ldots, x^\ell, \ldots, x^r). \end{aligned} \tag{19}$$

The latter equation yields the following Hamiltonian for an electrical network:

$$\begin{aligned} H(v_C, i_L) &= [v_C^{\mathsf{T}}, i_L^{\mathsf{T}}] \begin{bmatrix} q \\ \lambda \end{bmatrix} - H(q, \lambda) \\ &= v_C^{\mathsf{T}} q + i_L^{\mathsf{T}} \lambda - H(q, \lambda) \\ &= \sum_{k=1}^{n} v_C^k q^k + \sum_{k=1}^{m} i_L^k \lambda^k - H(q, \lambda). \end{aligned} \tag{20}$$

B. LINEAR STORAGE ELEMENTS

For linear electrical networks the Hamiltonian is

$$\begin{aligned} H(q, \lambda) &= H_C(q) + H_L(\lambda) \\ &= \tfrac{1}{2} q^{\mathsf{T}} C^{-1} q + \tfrac{1}{2} \lambda^{\mathsf{T}} L^{-1} \lambda, \end{aligned} \tag{21}$$

where $C \in \mathbb{R}^n$ (capacitance) and $L \in \mathbb{R}^m$ (inductance) are symmetric matrices. The voltage in the capacitor and the current in the inductor are

$$\begin{aligned} v_C &= \partial H / \partial q = \partial H_C / \partial q = C^{-1} q, \\ i_L &= \partial H / \partial \lambda = \partial H_L / \partial \lambda = L^{-1} \lambda. \end{aligned} \tag{22}$$

Hence, the Legendre transformation of $H(q, \lambda)$ with respect to (v_C, i_L) is

$$\begin{aligned} H(v_C, i_L) &= [v_C^\mathsf{T}, i_L^\mathsf{T}] \begin{bmatrix} Cv_C \\ Li_L \end{bmatrix} \\ &\quad -\tfrac{1}{2}(Cv_C)^\mathsf{T} C^{-1}(Cv_C) \\ &\quad -\tfrac{1}{2}(Li_L)^\mathsf{T} L^{-1}(Li_L) \\ &= \tfrac{1}{2}v_C^\mathsf{T} Cv_C + \tfrac{1}{2}i_L^\mathsf{T} Li_L. \end{aligned} \tag{23}$$

Since this is a linear case, we can obtain the above result by substituting the values of q and λ (see (22)) directly into (21). In addition, Fig.16 shows the equivalence between both models of a storage element (with canonical state variables q and λ, or with state variables v_C and i_L).

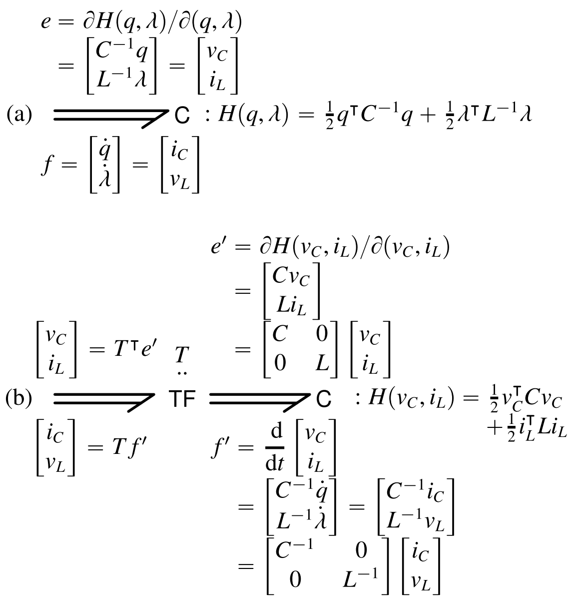


FIGURE 16. (a) Linear storage element with canonical state variables (q, λ). (b) Equivalent component modeled from a storage element with non-canonical state variables (v_C, i_L), connected to a transformer with transformation matrix $T = \mathrm{diag}(C^{-1}, L^{-1})$.

C. MODELICA IMPLEMENTATION

The following listings show the implementation of the port-Hamiltonian elements described above. All of them are part of the ElectricalNetworks package and have been derived from the classes defined in the pHS package, taking full advantage of the fact that Modelica is an object-oriented language.

```
import SI = Modelica.SIunits; // Units of the International System

connector ElectricPort = Port(redeclare SI.Voltage e, redeclare SI.Current
    f);

class ElectricBond
  SI.Voltage v[from.dimPort] "Effort";
  SI.Current i[from.dimPort] "Flow";
  extends Bond(redeclare replaceable connector BasePort = ElectricPort
    );
equation
  {v, i} = {from.e, from.f};
end ElectricBond;
```

```
class Voltage "Source of electric voltage"
  extends pHS.Sources.Se(
    redeclare replaceable ElectricPort port, label="v");
end Voltage;

class Current "Source of electric current"
  extends pHS.Sources.Sf(
    redeclare replaceable ElectricPort port, label="i");
end Current;
```

```
class LinearCapacitor
  replaceable SI.Capacitance CMatrix[port.dimPort, port.dimPort];
  SI.Charge q[port.dimPort] "Canonical state";
  parameter SI.Charge q0[port.dimPort] = zeros(port.dimPort) "Initial
    charge";
  extends pHS.Storage.Linear(
    redeclare replaceable ElectricPort port, x0=q0);
equation
  ZMatrix = CMatrix;
  x = q;
end LinearCapacitor;

class LinearInductor
// Modeled from a linear capacitor connected to a symplectic gyrator
  (Fig. 15.c)
  replaceable connector BasePort = ElectricPort;
  replaceable SI.Inductance LMatrix[port.dimPort, port.dimPort];
  SI.MagneticFlux lambda[port.dimPort] "Canonical state";
  parameter SI.MagneticFlux lambda0[port.dimPort] = zeros(port.
    dimPort) "Initial magnetic flux linkage";
  pHS.Storage.Linear storage(redeclare BasePort port, x0=lambda0);
  Dirac.Components.SGY SGY1(
    redeclare BasePort port1, redeclare BasePort port2) ;
  Interfaces.Bonds.ElectricBond b1(
    redeclare connector BasePort = BasePort);
equation
  storage.ZMatrix = LMatrix;
  storage.x = lambda;
  connect(SGY1.port1, b1.from); connect(b1.to, storage.port);
  connect(SGY1.port2, port);
end LinearInductor;
```

```
class Resistor "Resistor (Definition 25)"
  extends pHS.Dissipative.R(redeclare replaceable ElectricPort port);
end Resistor;

class LinearResistor
// Linear resistor v = R · i
  replaceable SI.Resistance RMatrix[port.dimPort,port.dimPort];
  extends Dissipative.Resistor;
equation
  PhiR = port.e − RMatrix ∗ port.f ∗ port.dir;
end LinearResistor;

class LinearConductor
// Linear conductor i = G · v
  replaceable SI.Conductance GMatrix[port.dimPort,port.dimPort];
  extends Dissipative.Resistor(label="G");
equation
   PhiR = GMatrix ∗ port.e − port.f ∗ port.dir;
end LinearConductor;
```

D. EXAMPLE: CHUA CIRCUIT

The *Chua circuit* [13] is a non-linear active circuit that exhibits chaotic behavior with a "double scroll" strange attractor in the phase space. A comprehensive analysis of its chaotic behavior can be found in [31] and [32]. The implementation presented in this section is similar to that in the Modelica Standard Library (MSL) example ChuaCircuit, in which the non-linear resistor (see Fig.17.b) is implemented like in [31, pp. 644 and 645].

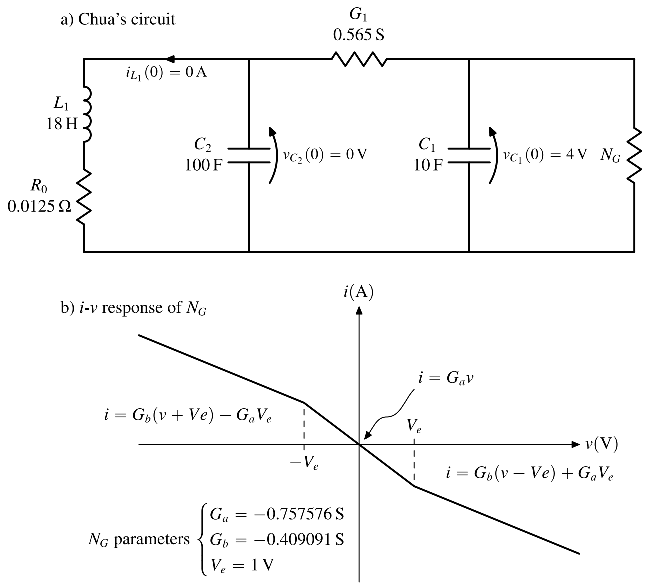


FIGURE 17. **Chua circuit with three storage elements (L_1, C_1 and C_2) and a non-linear resistor N_G (adapted from the MSL).**

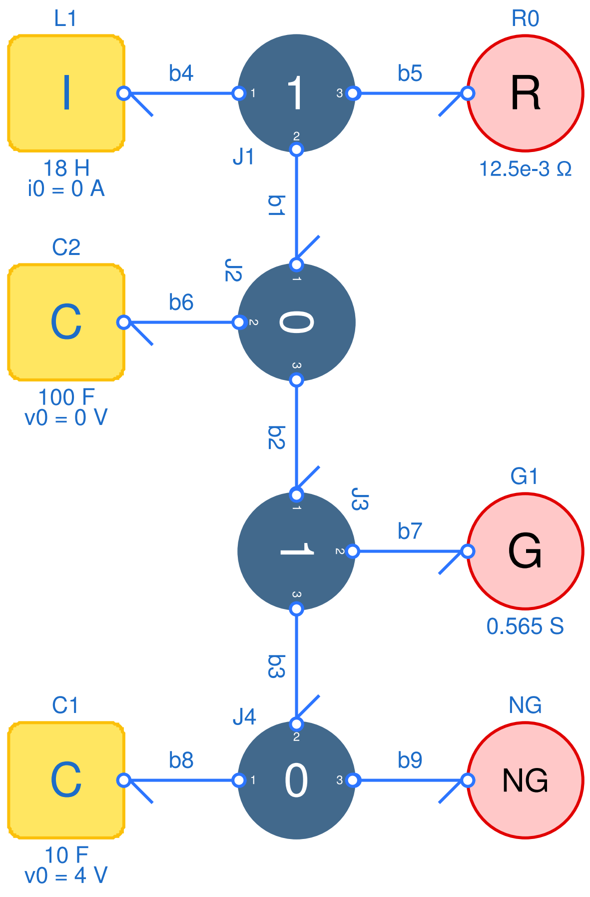


FIGURE 18. **Port-Hamiltonian model of the Chua circuit (Fig.17.a).**

Fig.18 shows the bond-graph model of the Chua circuit (Fig.17.b). The Modelica code of the non-linear component NG is:

```
class NonLinearConductor "Non−linear conductor (adapted from MSL)"
  SI.Voltage v = port.e[1];
  SI.Current i = port.f[1];
  parameter SI.Conductance Ga "Conductance in inner voltage range";
  parameter SI.Conductance Gb "Conductance in outer voltage range";
  parameter SI.Voltage Ve "Inner voltage range limit";
  extends Dissipative.Resistor(redeclare OneDimensionalElectricPort
    port, label="NG");
equation
  PhiR = {(if (v < −Ve) then Gb∗(v + Ve) − Ga∗Ve
           elseif (v > Ve) then Gb∗(v − Ve) + Ga∗Ve
           else Ga∗v) − i∗port.dir};
end NonLinearConductor;
```

The traditional electrical diagram in Fig.17.a is easier to understand than the graphical representation in Fig.18, but the proposed port-Hamiltonian representation (derived from the bond graphs) makes it easier to visualize the connection network for energy exchange between the different components of the system.

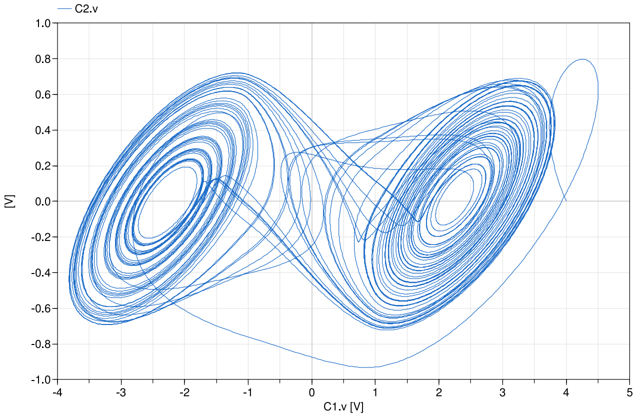


FIGURE 19. **Simulation of the Chua circuit for 5×10^4 s with a relative tolerance of 10^{-12}. The figure shows the projection on the plane v_{C_1}–v_{C_2} of the trajectory with initial state $(v_{C_1}, v_{C_2}, i_{L_1})|_{t=0} = (4\,\mathrm{V}, 0\,\mathrm{V}, 0\,\mathrm{A})$.**

The simulation results obtained by using the systematic port-Hamiltonian implementation (Fig.19) are the same as in the MSL implementation, and similar to those executed with *Spice3e2* in [32, p. 672], in which the chaotic behavior can be observed with a "double scroll" strange attractor.

VII. CONCLUSIONS

In this work we have presented a general purpose modeling library for modeling multiphysics systems with a port-Hamiltonian approach.

The capabilities of the library to carry out a general model design procedure have been illustrated with examples from two different physical domains: the N-body problem (Hamiltonian mechanics), and the Chua circuit (electrical networks). The object-oriented design of the library and its implementation with the Modelica language has allowed to systematically develop models belonging to two different physical domains in a unified manner. These examples have also been modeled with the Modelica Standard Library and we have obtained ODEs of the same order, with similar simulation times, and with identical numerical results

A future research line will be the extension of this library for the modeling of thermofluid systems and its assessment in the development of models for large-scale solar thermal power plants.

APPENDIX

NOTATION

Let $\xi = (E, \pi, B)$ be a constant rank real vector bundle [28] with dual $\xi^* = (E^*, \pi^*, B)$, and let $F = (f, g) : \xi \to \xi^*$ be a morphism in the category $\mathbf{VB}_{\mathbb{R}}$ of real vector bundles, i.e., for each $b \in B$ the function $f : \pi^{-1}(b) \to (\pi^*)^{-1}(b)$ is linear. The expression $w = F(v)$ for $v \in \xi$ and $w \in \xi^*$ is represented with Dirac's bra-ket notation $|w\rangle = F\,|v\rangle$. The *natural pairing* (or *duality product*) of $w \in \xi^*$ and $v' \in \xi$ is also represented with the bra-ket notation $w(v') = \left\langle w \,\middle|\, v' \right\rangle = \left\langle v' \,\middle|\, w \right\rangle = \left\langle v' \,\middle|\, F(v) \right\rangle = \left\langle v' \,\middle|\, F \,\middle|\, v \right\rangle$.

The fibre product (or Whitney sum) of two bundles $\xi_1 = (E_1, \pi_1, B)$ and $\xi_2 = (E_2, \pi_2, B)$ is the bundle $\xi_1 \oplus \xi_2 = (E_1 \oplus E_2, \pi, B)$ where $E_1 \oplus E_2 = \{(e_1, e_2) \in E_1 \times E_2 : \pi_1(e_1) = \pi_2(e_2)\}$ and $\pi(e_1, e_2) = \pi_1(e_1) = \pi_2(e_2)$. The fiber of $\xi_1 \oplus \xi_2$ over $b \in B$ is $\pi^{-1}(b) = \pi_1^{-1}(b) \times \pi_2^{-1}(b)$ [28, p.15].

The product of the bundle $\xi = (E, \pi, B)$ and the set C is the bundle $\xi \oplus C = (E \oplus C, \pi', B)$ where $\pi'(e, c) = \pi(e)$ and whose fibers are $\pi'^{-1}(b) = \pi^{-1}(b) \times C$.

The bilinear form $\Omega_F : \xi \oplus \xi \to \mathbb{R}$ associated to F is defined with the expression $\Omega_F(v', v) = \left\langle v' \,\middle|\, F \,\middle|\, v \right\rangle$ for all $v, v' \in \xi$, $F \in \mathrm{Hom}(\xi, \xi^*)$; and it is said to be *symmetric* provided $\left\langle v' \,\middle|\, F \,\middle|\, v \right\rangle = \left\langle v \,\middle|\, F \,\middle|\, v' \right\rangle$ for all $v, v' \in \xi$, *skew-symmetric* if $\left\langle v' \,\middle|\, F \,\middle|\, v \right\rangle = -\left\langle v \,\middle|\, F \,\middle|\, v' \right\rangle$ and *positive semidefinite* if $\langle v \,|\, F \,|\, v\rangle \geq 0$.

The mapping $T \equiv F \mapsto \Omega_F$ is a natural isomorphism $\mathrm{Hom}(\xi, \xi^*) \cong \mathrm{Hom}(\xi \oplus \xi, \mathbb{R})$ with inverse $T' \equiv \Omega \mapsto F_\Omega$ defined by $F_\Omega(v') = \Omega(v', -)$ for all $v' \in \xi$ (it is straightforward to prove that $T' \circ T = \mathrm{id}_{\mathrm{Hom}(\xi,\xi^*)}$ and $T \circ T' = \mathrm{id}_{\mathrm{Hom}(\xi \oplus \xi, \mathbb{R})}$), and let us say that F is a symmetric, skew-symmetric or positive semidefinite map provided its bilinear form Ω_F is so.

The dual map $F^* : \xi^{**} \to \xi^*$ has the same domain and codomain as F because $\xi^{**} = \xi$, and $w = F^*(v)$ is represented with the ket expression $|w\rangle = F^*\,|v\rangle$ or with the bra expression $\langle w| = \langle v|\,F$, and this means it is possible to represent the pairing $\left\langle F^*(v) \,\middle|\, v' \right\rangle = \left\langle v' \,\middle|\, F^*(v) \right\rangle$ as $\left\langle v' \,\middle|\, F^* \,\middle|\, v \right\rangle = \left\langle v \,\middle|\, F \,\middle|\, v' \right\rangle$. To prove the last equality, we use the definition of dual map (i.e., if $w = F(v)$ then $F^*(w(v)) = w(F(v))$ for all $w \in \xi$) and write,

$$\begin{aligned} \left\langle F^*(w) \,\middle|\, v \right\rangle &= \langle w \,|\, F(v)\rangle \\ \left\langle v \,\middle|\, F^*(w) \right\rangle &= \langle w \,|\, F(v)\rangle \\ \left\langle v \,\middle|\, F^* \,\middle|\, w \right\rangle &= \langle w \,|\, F \,|\, v\rangle\,. \end{aligned}$$

In the case of linear maps between vector spaces we use the bra-ket notation as follows:

1) The vector $v \in \mathsf{V}$ written as ket $|v\rangle$ is represented by a column matrix $v \in \mathbb{R}^{n\times 1}$.
2) If v is written as bra $\langle v|$, then it is represented by a row matrix $v^\intercal \in \mathbb{R}^{1\times n}$.
3) The duality product $\langle w \,|\, v\rangle$ is the product of the bra $\langle w|$ by the ket $|v\rangle$, and it is computed with the matrix product $w^\intercal v$.
4) The linear mapping $F : \mathsf{V} \to \mathsf{V}^*$ is represented by a matrix $F \in \mathbb{R}^{n\times n}$, and its action on bra and ket vectors is computed as Table 3 shows.

TABLE 3. Bra-ket notation and matrix computation.

bra-kets	matrices
$\lvert w\rangle = F\,\lvert v\rangle$	$w = Fv$
$\langle w\rvert = \langle v\rvert\,F^*$	$w^\intercal = (Fv)^\intercal = v^\intercal F^\intercal$
$\langle v \,\lvert\, F \,\rvert\, w\rangle = \langle w \,\lvert\, F^* \,\rvert\, v\rangle$	$v^\intercal F w = w^\intercal F^\intercal v$
$\lvert v\rangle\langle w\rvert$	$v w^\intercal$

REFERENCES

[1] R. Abraham and J. E. Marsden, *Foundations of Mechanics*, 2nd ed. Providence, RI, USA: AMS, 2008.

[2] I. V. Arnol'd, *Mathematical Methods of Classical Mechanics*, 2nd ed. New York, NY, USA: Springer-Verlag, 1989.

[3] *Modelica—A Unified Object-Oriented Language for Systems Modeling. Language Specification. Version 3.4*, Modelica Assoc., Linköping, Sweden, 2017.

[4] J. K. Åström, H. Elmqvist, and S. E. Mattsson, "Evolution of continuous-time modeling and simulation," in *Proc. 12th Eur. Simulation Multiconf. (EMS)*, 1998, pp. 9–18.

[5] C. Batlle, I. Massana, and E. Simo, "Representation of a general composition of dirac structures," in *Proc. IEEE Conf. Decis. Control Eur. Control Conf.*, Dec. 2011, pp. 5199–5204.

[6] W. Borutzky, *Bond Graph Methodology: Development and Analysis of Multidisciplinary Dynamic System Models*. New York, NY, USA: Springer-Verlag, 2010.

[7] P. C. Breedveld, "Multibond graph elements in physical systems theory," *J. Franklin Inst.*, vol. 319, nos. 1–2, pp. 1–36, Jan. 1985.

[8] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, vol. 14. Philadelphia, PA, USA: SIAM, 1996.

[9] J. W. Brewer, "Bond graphs of microeconomic systems," ASME, Paper 75-WA/Aut-8, 1977.

[10] F. E. Cellier, "Hierarchical non-linear bond graphs: A unified methodology for modeling complex physical systems," *Simulation*, vol. 58, no. 4, pp. 230–248, Apr. 1992.

[11] F. E. Cellier and À. Nebot, "The Modelica bond graph library," in *Proc. 4th Int. Modelica Conf.*, Hamburg, Germany, 2005, pp. 1–10.

[12] J. Cervera, A. J. van der Schaft, and A. Baños, "Interconnection of port-Hamiltonian systems and composition of dirac structures," *Automatica*, vol. 43, no. 2, pp. 212–225, Feb. 2007.

[13] L. O. Chua, "The genesis of Chua's circuit," *Int. J. Electron. Commun.*, vol. 46, no. 3, pp. 250–257, 1992.

[14] T. J. Courant, "Dirac manifolds," *Trans. Amer. Math. Soc.*, vol. 319, no. 2, pp. 631–661, 1990.

[15] S. Dai, "Compositional design of cyber-physical systems using port-Hamiltonian systems," Ph.D. dissertation, Graduate School, Vanderbilt Univ., Nashville, TN, USA, 2016. [Online]. Available: https://etd.library.vanderbilt.edu/available/etd-08042016-123306/unrestricted/Dai.pdf

[16] S. Dai, Z. Lattmann, and X. Koutsoukos, "Compositional design of cyber-physical systems using port-Hamiltonian systems," in *Cyber-Physical Systems: From Theory to Practice*, D. B. Rawart, J. J. P. C. Rodrigues, and I. Stojmenovic, Eds. Boca Raton, FL, USA: CRC Press, 2015, ch 2.

[17] M. Dalsmo and A. van der Schaft, "On representations and integrability of mathematical structures in energy-conserving physical systems," *SIAM J. Control Optim.*, vol. 37, no. 1, pp. 54–91, Jan. 1998.

[18] *Dymola–Dynamic Modeling Laboratory–User Manual 1B: Developing and Simulating a Model*, Dassault Systèmes AB, Vélizy-Villacoublay, France, 2019.

[19] A. de la Calle, F. E. Cellier, L. J. Yebra, and S. Dormido, "Improvements in BondLib, the modelica bond graph library," in *Proc. 8th EUROSIM Congr. Modeling Simulation*, Sep. 2013, pp. 282–287.

[20] J.-C. Delvenne and H. Sandberg, "Finite-time thermodynamics of port-Hamiltonian systems," *Phys. D, Nonlinear Phenomena*, vol. 267, pp. 123–132, Jan. 2014.

[21] A. Donaire and S. Junco, "Derivation of input-state-output port-Hamiltonian systems from bond graphs," *Simul. Model. Pract. Theory*, vol. 17, no. 1, pp. 137–151, Jan. 2009.
[22] I. Dorfman, *Dirac Structures and Integrability of Nonlinear Evolution Equations*. Hoboken, NJ, USA: Wiley, 1993.
[23] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, Eds., *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*. Berlin, Germany: Springer, 2009.
[24] H. Elmqvist, "A structured model language for large continuous systems," Ph.D. dissertation, Dept. Autom. Control, Lund Inst. Technol., Lund, Sweden, 1978.
[25] P. A. Fritzson, *Principles of Object-Oriented Modeling and Simulation With Modelica 3.3: A Cyber-Physical Approach*, 2nd ed. Hoboken, NJ, USA: Wiley, 2015.
[26] G. Golo, A. V. D. Schaft, C. P. Breedveld, and C. B. Maschke, "Hamiltonian formulation of bond graphs," in *Nonlinear and Hybrid Systems in Automotive Control*, R. Johansson and A. Rantzer, Eds. London, U.K.: Springer-Verlag, 2003, ch. 16, pp. 351–372.
[27] D. D. Holm, T. Schmah, and C. Stoica, *Geometric Mechanics and Symmetry: From Finite to Infinite Dimensions*. Oxford, U.K.: Oxford Univ. Press, 2009.
[28] D. Husemoller, *Fibre Bundles*, 3rd ed. New York, NY, USA: Springer-Verlag, 1994.
[29] International Astronomical Union (IAU), "RESOLUTION B2: On the re-definition of the astronomical unit of length," in *Proc. 28th Gen. Assem. Int. Astronomical Union (IAU)*, 2012, pp. 3–4. [Online]. Available: https://www.iau.org/static/resolutions/IAU2012_English.pdf
[30] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*, 5th ed. Hoboken, NJ, USA: Wiley, 2012.
[31] M. P. Kennedy, "Three steps to chaos—Part I: Evolution," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 40, no. 10, pp. 640–656, Oct. 1993.
[32] M. P. Kennedy, "Three steps to chaos—Part I: A Chua's circuit primer," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 40, no. 10, pp. 657–674, Oct. 1993.
[33] J. M. Lee, *Introduction to Smooth Manifolds*, 2nd ed. New York, NY, USA: Springer, 2012.
[34] E. H. Lieb and J. Yngvason, "The physics and mathematics of the second law of thermodynamics," *Phys. Rep.*, vol. 310, no. 1, pp. 1–96, Mar. 1999.
[35] B. Luzum, N. Capitaine, A. Fienga, W. Folkner, T. Fukushima, J. Hilton, C. Hohenkerk, G. Krasinsky, G. Petit, E. Pitjeva, M. Soffel, and P. Wallace, "The IAU 2009 system of astronomical constants: The report of the IAU working group on numerical standards for Fundamental Astronomy," *Celestial Mech. Dyn. Astron.*, vol. 110, no. 4, p. 293, Jul. 2011.
[36] J. Maks, "A spinor approach to port-Hamiltonian systems," in *Proc. 19th Int. Symp. Math. Theory Netw. Syst. (MTNS)*, Budapest, Hungary, Jul. 2010, pp. 1167–1171.
[37] H. M. Paynter, *Analysis and Design of Engineering Systems*. Cambridge, MA, USA: MIT Press, 1961.
[38] L. R. Petzold, "Description of DASSL: A differential/algebraic system solver," Sandia Nat. Labs., Livermore, CA, USA, Tech. Rep. SAND82-8637, 1982.
[39] M. Pfeifer, S. Caspart, S. Pfeiffer, C. Müller, S. Krebs, and S. Hohmann, "Automated generation of explicit port-Hamiltonian models from multi-bond graphs," 2019, *arXiv:1909.02848*. [Online]. Available: http://arxiv.org/abs/1909.02848
[40] D. H. Bernard Tellegen, "The network element," *Philips Res. Rep.*, vol. 3, pp. 81–101, Apr. 1948.
[41] J. U. Thoma, "Entropy and mass flow for energy conversion," *J. Franklin Inst.*, vol. 299, no. 2, pp. 89–96, Feb. 1975.
[42] A. van der Schaft, *L_2-Gain and Passivity Techniques in Nonlinear Control*. New York, NY, USA: Springer-Verlag, 2000.
[43] A. V. D. Schaft, "Port-Hamiltonian systems," in *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*, V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyninckx, Eds. Berlin, Germany: Springer, ch. 3, 2009.
[44] A. van der Schaft and D. Jeltsema, "Port-Hamiltonian systems theory: An introductory overview," *Found. Trends Syst. Control*, vol. 1, no. 2, pp. 173–378, 2014.
[45] J. C. Willems, "Paradigms and puzzles in the theory of dynamical systems," *IEEE Trans. Autom. Control*, vol. 36, no. 3, pp. 259–294, Mar. 1991.
[46] J. C. Willems, "The behavioral approach to open and interconnected systems," *IEEE Control Syst.*, vol. 27, no. 6, pp. 46–99, Dec. 2007.
[47] Y. K. Wong, "Application of bond graph models to economics," *Int. J. Model. Simul.*, vol. 21, no. 3, pp. 181–190, Jan. 2001.
[48] D. Zimmer and F. E. Cellier, "The Modelica multi-bond graph library," in *Proc. 5th Int. Modelica Conf.*, 2006, pp. 559–568.

FRANCISCO M. MÁRQUEZ received the B.S. degree in electronics engineering from the Universidad de Alcalá, Alcalá de Henares, Spain, in 1990, and the M.S. degree in telecommunication engineering and the M.S. degree in statistical and computational information processing from the Universidad Politécnica de Madrid, Madrid, Spain, in 2010 and 2011, respectively.

From 1989 to 1991, he was the Head of a laboratory with INDRA, Spain. Since 1991, he has been a Lecturer with the Departamento de Automática, UAH.

PEDRO J. ZUFIRIA received the Telecommunication Engineering degree from the Universidad Politécnica de Madrid (UPM), in 1986, the M.Sc. degree in ME, the M.Sc. degree in EE, the Ph.D. degree from the University of Southern California (USC), in 1989, the Doctor Ingeniero de Telecomunicación degree from the MEC, in 1991, and the Licenciado en Ciencias Matemáticas degree from the Universidad Complutense de Madrid, in 1997.

From 1987 to 1990, he was a Teaching and Research Assistant with the USC, in collaboration with different projects of the National Science Foundation and TRW/NASA. Since 1991, he holds a teaching position (currently as a full professor) at the Telecommunications Engineering School (ETSIT), UPM, where he was the Chairman of the Department of Applied Mathematics for Information and Communication Technologies, from 1999 to 2004. In the ETSIT, he was responsible for the Office of International Relations with other academic institutions, from 1993 to 1996. He was the Vice-Dean of research and graduate studies, in 1999, and the Co-Director of the Orange Chair, from 2009 to 2019. Since 2019, he has been the Co-Director of the Cabify Chair. He has also been the founder and the scientist responsible for the Neural Networks Group and currently the Dynamical Systems, Learning, and Control Research Group, where he has been involved in research projects of the European Union as well as several Spanish official and private institutions. He has authored over one hundred international publications in his research field. His research interests include analysis, control, and fault diagnosis of dynamical systems, and the applications of statistics, machine learning paradigms, and complex network theory on system modeling and data processing.

LUIS J. YEBRA was born in Almería, Spain, in 1971. He received the Telecommunications Technical Engineering degree from the Universidad de Alcalá, Alcalá de Henares, Spain, in June 1993, a Physics degree from the Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain, in June 1997, and the Ph.D. degree in automatic control and industrial computing from the Department of Automatic Control and Computer Science, UNED, in May 2006.

From 1999 to 2007, he was a Predoctoral Fellow of the National Research Center CIEMAT, and a Researcher with the Department of Energy of CIEMAT in Plataforma Solar de Almería (PSA-CIEMAT). Since 2007, he has been a Tenured Scientist in dynamic modeling and automatic control with PSA-CIEMAT, where he has been mainly focused on concentrated solar thermal plants (CST). He has coauthored four books, more than 30 scientific articles published in international journals (JCR), and more than 70 international conference proceeding papers. He has supervised three Ph.D. degree students. He has participated in more than 30 publicly funded research projects, two technology transfer agreements, and the creation of a spin-off company.

Dr. Yebra has received three prizes to the best academic curriculum for the Physics degree.