Compositional Modeling and Design of Cyber-Physical Systems Using Port-Hamiltonian Systems

By

Siyuan Dai

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical Engineering

December, 2016

Nashville, Tennessee

Approved:

Xenofon Koutsoukos, Ph.D.

Janos Sztipanovits, Ph.D.

Gabor Karsai, Ph.D.

Gautam Biswas, Ph.D.

Shige Wang, Ph.D.

Chipotle is tasty.

ACKNOWLEDGMENTS

During my time as a graduate student here at Vanderbilt University, I was able to work and collaborate with a number of great people. I would first like to thank my Ph.D. advisor, Professor Xenofon Koutsoukos for his tremendous help and guidance of my research. I would also like to thank Professor Janos Sztipanovits, Professor Gabor Karsai, and Professor Gautam Biswas from Vanderbilt for agreeing to serve on my thesis committee. Finally I would like to thank Dr. Joseph D'Ambrosio from General Motors for the opportunities of two summer internships at GM and Dr. Shige Wang from GM Research for his guidance of my internships and for serving on my thesis committee.

I like to thank others of the Vanderbilt community who have affected my time here at Vanderbilt in a positive way, whether it is work related or outside, in particular Zhenkai Zhang, Heath LeBlanc, Emeka Eyisi, Daniel Mack, Ashraf Tantawy, Zsolt Lattmann, Amin Ghafouri, Hamzah Abdel-aziz, Bradley Potteiger, and other members of the ISIS community.

I also want to thank my family for the continuous support throughout my time as a graduate student and for always believing in me.

Finally, I want to give a special thank you to the National Science Foundation (CNS-1035655). Without their financial support none of this work would have been possible.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Chapter 1

Introduction

1.1 Motivation

The advancement of technology over the last few decades has led to the ubiquity of embedded systems in everyday lives. Traditional embedded system technologies emphasize the computational elements and how to design these elements to perform a certain task or objective. Research over the past ten years has bridged the link between the computational elements and the physical dynamics into cyber-physical systems (CPS). CPS are complex engineering systems that integrate computational, communication, and control elements with physical dynamics [3]. Examples of CPS include automotive systems, aeronautical systems, industrial process control systems, electrical power grids, and environmental monitoring systems, etc. As the computational components in modern CPS increase in both number and complexity, great challenges arise for their integration with the physical domain [9]. Consequently, rigorous engineering methods are needed for the integration of computational components with the physical system in order to achieve predictable, correct behavior. Model-based design is an important method for CPS that efficiently connects all development phases of control software and ensures desirable system performance.

Composition is a fundamental idea in engineering that reduces system complexity, decreases cost, and allows for scalability and re-usability. The predominant challenge in the composition of CPS is heterogeneity caused by the different layers of CPS. Traditionally, composition is done in an *ad hoc* way in which components are individually designed and integrated into the system with the objective of "making it work". As the field of CPS continues to grow and evolve, more problems emerge from the interaction of heterogeneous domains which leads to a loss of compositionality, which significantly hampers the system integration process [3]. The physical and computational domains are distinguished by

the fundamental difference in how time is represented, i.e. continuous-time for physical domains and discrete-time for computational domains. In software systems, processes execute in sequential steps and are commonly represented by difference equations. In physical systems, processes exist in parallel which are represented by differential equations [10]. Careful formulation of a modeling framework is needed to integrate physical and computational elements.

The system integration problem has been a challenge for the automotive industry for many years. An automotive CPS typically consists of plants (vehicle dynamics components such as transmission and wheels), controllers (computational components where control algorithms are implemented), and sensors/actuators (links the control signals with the vehicle dynamics) integrating through a communication network [11]. The original equipment manufacturers (OEMs) purchase controllers and subsystems from a wide variety of suppliers; since suppliers design subsystems differently, OEMs treat these controllers and subsystems as "black boxes" and must test and verify that the final integrated product behaves properly. Although automobiles have existed in this capacity since electronics were first introduced to them, advancements in technology and the exponential growth in the number of computational elements within modern vehicles necessitate an extensive amount of prototyping which increases the cost of development [12].

1.2 Research Challenges

CPS contain a plethora of computational and physical components working together to achieve an objective. The design method of these CPS becomes important because of the ability to simulate, analyze, and identify design flaws in the system before prototyping. As the number of components and interactions increase with advancements in technology, the complexity of CPS also increases, which necessitates component-based design methods where components are individually defined and implemented. Component-based design and modeling is desirable because it allows for scalability, substitutablity, and reusability

when modeling large-scale systems. Several challenges arise for the compositional design and modeling of CPS which is addressed in this dissertation.

Heterogeneous domains present a significant challenge to the modeling of CPS. The composition of physical and computational domains is inherently heterogeneous due to the differences in how the variables change with respect to time (continuous-time vs. discrete-time). Furthermore, there are sub-domains within both physical and computational domains which are commonly modeled in different frameworks. In the physical domain there can be physical elements which are modeled using different modeling paradigms such as circuit diagrams or free-body diagrams. In the computational domain there can be computational elements which have different memory sizes, sampling times, processor speeds, and architectures. In this dissertation we address the modeling of heterogeneous domains by using the port-Hamiltonian system (PHS) modeling framework.

Interactions present a significant challenge to the simulation of CPS. Interactions are either causal, where one variable acts as the cause and the other variable acts as the effect, or acausal, where both variables are related via equation statements without any causality implications. In the computational domain, components interact through an input-output characteristic where causality is explicit in the model. In the physical domain, components interact through shared variables based on the physics of the system where the notion of causality is ambiguous [9]. In this dissertation we address the challenge of interactions by using PHS, which can be expressed as either causal or acausal models, to unify the physical and computational domains.

Hybrid dynamics and nonlinearities present significant challenges for the control design of CPS. A hybrid system exhibits both continuous and discrete behavior and is used to model a large class of CPS. Most physical dynamics are nonlinear by nature, and nonlinear systems do not obey the superposition principle. Problems that arise from nonlinear systems are difficult to solve and are commonly approximated using linear methods. The PHS framework contains specifications and rules for implementing hybrid dynamics and

nonlinearities.

Safety is important for CPS because of the benefits of keeping the system operating within an appropriate and safe region. In order to have safe and reliable CPS, safety analysis methods must be scalable and be able to address a large class of systems. Heterogeneous domains and hybrid dynamics create significant challenges for the safety of CPS; though there are many well-established methods for systems that are either purely discrete or purely continuous, more advanced techniques are needed for systems containing a mixture of discrete and continuous dynamics. Nonlinearities add an additional challenge to the safety of CPS; most nonlinear systems are solved by approximation using linear methods and many methods handle nonlinearities with similar linearization methods.

1.3 Organization

In this section we provide an overview of the dissertation and contributions towards addressing the challenges in the compositional design and modeling of CPS. The technical contributions of this dissertation are contained in Chapters 4 - 8.

- Chapter 2 presents various related work in the area of CPS research. We begin by providing an overview on the various modeling methods for CPS such as bond graphs, PHS, and Modelica. Simulation of both causal and acausal models of computation are presented with an emphasis on software tools. A brief description of passivity is presented. Safety of hybrid systems is also presented. Finally, we end the related work chapter with a discussion on automotive CPS.

- Chapter 3 presents detailed theoretical background on PHS, passivity, and safety analysis. Equations which describe PHS are detailed which show how systems interact. Passivity as an analysis tool is then presented in both continuous-time and discrete-time format. Passivity indexes and passivity-based control are introduced to characterize control systems. Finally, safety analysis of CPS is presented with an

emphasis on barrier certificates. The technical details presented in this chapter are used in the subsequent chapters of this dissertation.

- Chapter 4 presents a method for the component-based modeling and simulation of CPS using PHS. A domain-specific modeling language based on PHS developed using the principles of Model-Integrated Computing is presented. The unique structure of PHS allows for the modeling of CPS with nonlinearities, hybrid dynamics, and heterogeneous domains in a component-based way. Simulation methods for PHS using its Dirac structure is presented in both acausal and causal forms. Interactions between domains occur through ports which are captured by the Dirac structure and constituent equations.

- Chapter 5 presents a compositional model-based control design method for CPS using PHS with a specific emphasis on automotive applications. Hybrid dynamics and nonlinearities are modeled in the PHS framework, and controllers are designed to ensure passivity for the closed-loop system. We choose control gains which ensures that the system stabilizes to desired values. We apply the method to an in-depth case study of an autonomous vehicle and demonstrate the process of modeling of the control and vehicle dynamics.

- Chapter 6 presents a safety analysis method for PHS by using the Hamiltonian function as a barrier function. Using the Hamiltonian function, we are able to derive formal proofs of safety which ensures safe operation of the PHS. This method was motivated by the unsafe skidding behavior on autonomous vehicles when the speed control and steering control systems have conflicting objectives. We apply the method to the same case study from Chapter 5 and show how this safety method imposes additional constraints on the control parameters.

- Chapter 7 shows an extension to the work in Chapter 6 by applying a discretization scheme called conservative time-discretization to the safety analysis method which

allows us to guarantee that the system remains safe despite discretization effects. We show how the different control design scheme is able to offset the loss in passivity the PHS experienced during discretization.

- Chapter 8 shows the simulation results of both the continuous-time and discrete-time control designs deployed on the hardware-in-the-loop platform. We present an overview of the platform itself. Implementation effects such as discretization and quantization will affect the passivity of the closed-loop system and are discussed.

- Chapter 9 concludes this dissertation. A summary of the overall content of this dissertation is described succinctly. Finally, we introduces some ideas for future directions for the work presented in this dissertation.

Chapter 2

Related Work

Cyber-physical system (CPS) is a multi-disciplinary research topic involving heterogeneous domains/components spanning many fields of research. In this chapter we present related work that are relevant to the topics of this dissertation. In Section 2.1 we review the design and modeling of CPS, covering bond graph modeling (Section 2.1.1), port-Hamiltonian system (PHS) modeling (Section 2.1.2), and Modelica modeling (Section 2.1.3). In Section 2.2 we provide an overview of simulation in CPS. Simulation of causal models is presented in Section 2.2.1, detailing the concept of causality and how it converts bond graph models to signal-flow graphs and block diagrams. In Section 2.2.2 we discuss the simulation of acausal models with an emphasis on the Modelica simulation environment. In Section 2.3 we present background on the control design of CPS where passivity is introduced as a control design tool. In Section 2.4 we present background on the safety analysis of hybrid systems specifically discussing ellipsoidal techniques and barrier certificates. Finally, we end the related work chapter in Section 2.5 with CPS in the automotive context. First we present background for adaptive cruise control (ACC) systems in Section 2.5.1, which focuses on controlling the longitudinal dynamics of a vehicle. We then present background for lane keeping control (LKC) systems in Section 2.5.2, which focuses on controlling the lateral dynamics of a vehicle.

2.1 Modeling

Component-based modeling of CPS is a challenging problem because of the inherent heterogeneity within CPS [13]. Components are implemented as well-defined Models of Computation (MoC) and abstract semantics are used to define interactions between them [10]. Each MoC is converted into an abstraction; abstract algebra is used to formulate

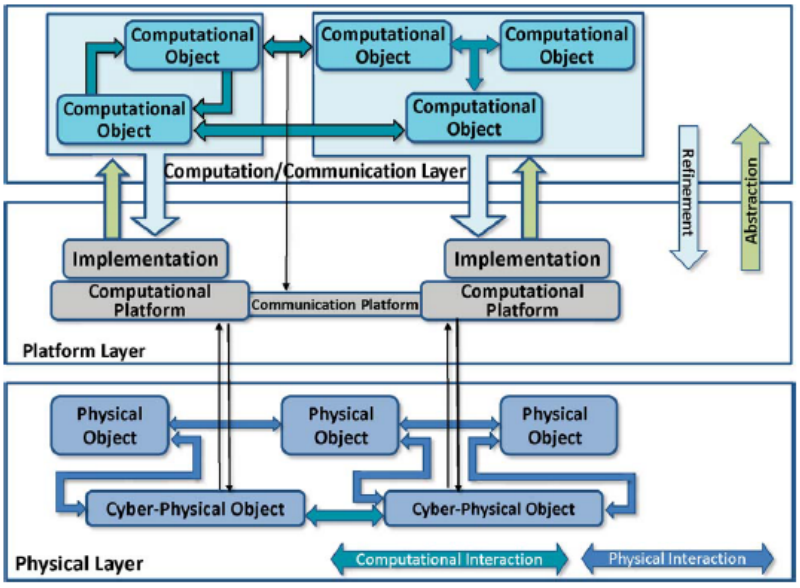

Figure 2.1: The three design layers in CPS [3]

their interconnections. As a result, a CPS is formed as an amalgamation of heterogeneous abstract MoCs held together with well-defined abstract connections.

The intricacies of both acausal and causal modeling can be captured using a Domain-Specific Modeling Language (DSML) [14]. Physical components interact with each other through an acausal flow of energy representing the underlying laws of physics. Computational components interact with each other through an input-output characteristic. Interactions between the physical and computational components are formalized through unidirectional physical signals and computational signals [15].

The difference in abstraction between physical dynamics and computational elements continues to pose a significant challenge [16]. The abstractions regarding computational and networking elements will need to be redefined in a way that merge the layers of CPS [17]; in order to accurately model the interactions of the computational and physical components in the systems, interfaces need to be well formulated [18]. Port-based modeling has garnered attention over the years since it captures the underlying physics of the system. Port-based modeling of lump-parameter complex physical systems leads to a class of systems called PHS, where the central concept is energy transfer [19]. The dy-

namics of the PHS is determined by its storage elements (the Hamiltonian function) and its resistive elements; The interconnection of these elements contributes to its geometric structure. PHS is derived from bond graph modeling [2].

2.1.1 Bond Graphs

A PHS is graphically represented using a network of bond graphs. A bond graph is domain-independent, and is a useful tool for the modeling of physical systems; different physical domains such as electrical, mechanical rotational, hydraulic, and thermodynamics are all described in the same way through a physical exchange of energy. Its major distinction from tools such as signal-flow graphs is that every arc (bond) in a bond graph model represents a bi-directional exchange of energy, rather than a uni-directional flow of signals. Bond graphs exhibit strong topological and graphical similarities to schematic models of the corresponding physical system (e.g. free-body diagrams, electrical circuit diagrams) [20].

A bond graph is a directed graph, where vertexes represent physical phenomena and edges (bonds) represent idealized energy connections. Bond graphs allow for the recursive decomposition of a system into subsystems, where each vertex is represented by a bond graph model [21]. Each bond is represented by a half arrow, where the direction of the arrow denotes the flow of positive energy [22]. Similar to the passive sign convention of electrical circuits, the positive direction of energy flow is arbitrary; the model is valid as long as its choice of energy flow direction is consistent. The transfer of energy between different subsystems and physical domains occur through real-world engineering interactions such as mechanical actuators, electrical wires, and pulleys, etc.

$$\text{Power} = \text{Effort} \times \text{Flow} \tag{2.1}$$

Power is a ubiquitous concept in all physical domains, and is the inner product of two

physical variables. Each bond in a bond graph consists of two power conjugate variables, called effort and flow, whose product is power, as shown in (2.1). Table 2.1 shows different physical domains and their corresponding power conjugate variables.

Table 2.1: Table of efforts and flows in different physical domains [1]

Domain	Effort (e)	Flow (f)
Electrical	Voltage (V)	Current (A)
Mechanical Translational	Force (N)	Velocity (m/s)
Mechanical Rotational	Torque (Nm)	Angular Velocity (rad/s)
Hydraulic	Pressure (N/m^2)	Volume Flow (m^3/s)
Thermodynamic	Temperature (K)	Entropy Flow (Js/K)

At the lowest level of hierarchy, vertexes are represented by bond graph elements which represent distinct physical processes [23]:

1. Distribution (one or zero junction): Junctions connect two or more bond graph elements in a power-continuous manner; power entering the junction must be equal to the power leaving the junction. The difference between the two types of junctions is the power conjugate variable that is held constant on all attached bonds; in a one junction the flow variable is held constant, while in a zero junction the effort variable is held constant. A common way of interpreting junctions is using circuit theory; a zero junction represents Kirchhoff Current Law, while a one junction represents Kirchhoff Voltage Law.

2. Transformation (transformer or gyrator): Transformation elements connect two bond graph elements in a power-continuous manner. In a transformer there is a modulation term which the efforts on the two bonds are related by a variable k and the flows on the two bonds are related by the inverse of k. Typically, transformers represent the transferring of energy inside a domain (e.g. hydraulic cylinders). In a gyrator there is a similar modulation term l in which the effort and flow on the two bonds are related by a variable l and the opposite variables are related by the inverse of l. Typically,

gyrators represent the transferring of energy between different domains (e.g. electric motor).

3. Conversion (resistive element): A linear resistive element relates the effort and flow variables of its corresponding bonds through a constant value. The power sign convention on these resistive elements imply that power flows into them are dissipated. A damper is a mechanical example of a resistive element that dissipate power into heat. If the relationship between the effort and flow variable is not a constant value, then the resistive element is classified as nonlinear.

4. Storage (storage element): Storage elements store energy. Energy is described in relation to either the effort or the flow, through a generalized momentum or displacement variable. The generalized momentum variable, p, is obtained through the integration of the effort variable; the generalized displacement variable, q, is obtained through the integration of the flow variable. Table 2.2 shows the generalized momentum and displacement variables in the various domains. Though most domains contain two types of energy storage elements (and consequently both a generalized momentum and displacement variable), some domains such as the thermodynamic and chemical only contain one type of energy storage element and are therefore omitted from the table.

The bond graph elements discussed so far are all passive elements which either store energy, dissipate energy, or conserve energy. Active elements are categorized into sources and sinks; they are not part of the bond graph model, but represent the boundary conditions in which the bond graph model interacts with the environment [24]. Sources either impose an effort or a flow value onto the rest of the bond graph model, depending on if it is a source of effort or source of flow. Sources can either be constant or time-dependent.

Table 2.2: Table of momentum and displacement in different physical domains [1]

Physical domain	Sub-domain	Flow f	Effort e	State x
Electric circuit	Electrical Magnetic	Current Voltage	Voltage Current	Electric charge Magnetic flux
Mech. translation	Potential Kinetic	Velocity Force	Force Velocity	Displacement Momentum
Mech. rotation	Potential Kinetic	Ang. velocity Torque	Torque Ang. velocity	Ang. displacement Ang. momentum
Hydraulic	Potential Kinetic	Vol. flow Pressure	Pressure Vol. flow	Volume Flow momentum

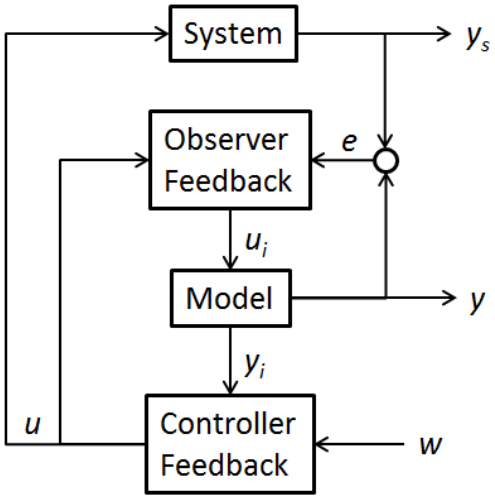

Figure 2.2: Model-based observer control scheme [2]

2.1.1.1 Control Design Using Bond Graphs

The use of bond graphs in physical domain control design leads to a method called Model-Based Observer (MBO) Control, in which the physical system being controlled as well as the controller and observer are all implemented as bond graphs [25]. MBO allows for the retention of physical properties of the system through the use of bond graphs, and can also be transformed into other representations (transfer functions, state-space representations, etc.) for simulation purposes. MBO has been shown to be robust to errors in physical parameters and state variables [26]

The MBO control architecture is shown in Figure 2.2 in which the *system* box represents the system being controlled and is treated as a black box; its output y_s feeds into the MBO controller and receives the control signal u. In the MBO controller, the *model*, *controller feedback*, and *observer feedback* blocks are all implemented as bond graph models:

1. Model: A dynamic simulation model of the *system* block implemented as a bond graph. It uses the same input, u, as the *system* block.

2. Controller Feedback: Control algorithm which generates the control signal u based on virtual sensor signals y_i from the model and the set-point w. The purpose of the controller feedback bond graph is to make the system behave in a certain way.

3. Observer Feedback: provides virtual sensor values, u_i, to the *model* block using $e = y_s - y$, as the model error. The purpose of the observer feedback bond graph is to drive the model error to zero, thereby ensuring that the *model* bond graph is a good representation of the *system*.

The idea of using bond graphs as a tool for control design is by formulating systems using the idea of power ports, allowing for the preservation of important physical properties.

2.1.2 Port-Hamiltonian Systems

In order to provide insight into the physical properties of the system, a mathematical framework is needed to describe the graphical model. The equations of a non-causal bond graph model extends to a PHS, which centers on a power-conserving interconnection structure called the Dirac structure. Generalized bond graph modeling is different from bond graph modeling in two ways, absence of causality and use of symplectic gyrators [27]. A symplectic gyrator is a unit gyrator that inverts the roles of the effort and flow [22]. In generalized bond graphs, symplectic gyrators are attached to I-storage elements, which unifies the treatment of C and I storage elements. Table 2.3 shows the various domains and their

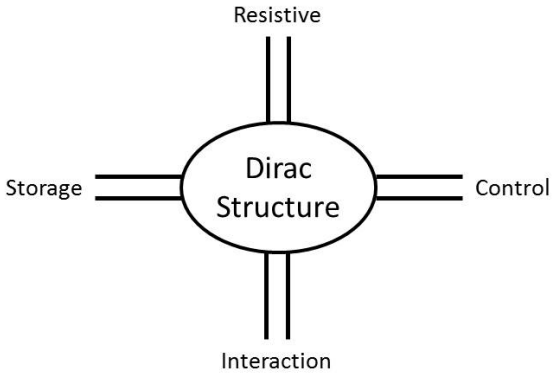

Figure 2.3: Port-Hamiltonian system [2]

corresponding characteristics from a PHS context. Energy-storage and resistive elements make up the internal ports; control and interaction elements make up the external ports. PHS are represented by Figure 2.3, where a set of ports (control, interaction, resistive, and storage) are interconnected through a Dirac structure [2]. Detailed technical details on PHS are shown in Section 3.1.

Table 2.3: Table of characteristics of domains in the PHS context [2]

Domain	Effort	Flow	Displacement	Momentum
Electro-Magnetic	Voltage	Current	Charge	Flux Linkage
Mech. Translational	Force	Velocity	Displacement	Momentum
Mech. Rotational	Torque	Ang. Velocity	Ang. Displacement	Ang. Momentum
Hydraulic	Pressure	Volume Flow	Volume	Tube Momentum
Thermodynamic	Temperature	Entropy Flow	Entropy	N/A

2.1.3 Modelica

Modelica can be used to model and simulate acausal bond graph models. Modelica was developed in 1996 with the goal of modeling physical systems by components through a standardized format. Modelica is an acausal object-oriented language; equations denote equality rather than assignments [5]. There are many Modelica software tools including OpenModelica [28], JModelica [29], and Dymola [4]. Modelica has become the standard

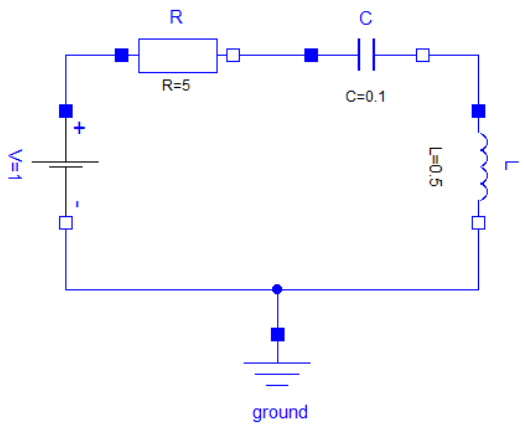

Figure 2.4: RLC example implemented in Dymola [4]

for acausal modeling of physical systems; the language and standard library continues to expand as it becomes widely adopted in both industry and academia [30].

System modeling in Modelica is done through the Modelica Standard Library (MSL) [31], which is an open-source library developed by the Modelica Association for the modeling of mechanical, electrical, thermal, fluid, and control systems. Models are decomposed into a set of interconnected components, and are represented graphically using the graphical editor of a Modelica modeling environment [32]. To demonstrate the graphical representation, we present a simple series-RLC circuit modeled in Dymola (Figure 2.4). The Modelica text (Figure 2.5) has two parts which describe the topology of the system: components and connections. Components are defined in the Modelica Standard Library and are referenced through the library hierarchy; for example, the statement "Modelica.Electrical.Analog.Basic.Resistor R(R=5)" instantiates a resistor *R* with resistive value 5 of the *Resistor* class in the Modelica Electrical Components library.

Connections specify the interconnection between components in a Modelica model; in the case of the series RLC circuit each connection is specified as an interaction of voltage and current, where the voltages obey Kirchhoff Voltage Law and the currents obey Kirchhoff Current Law. In order to assist with the integration with other software environments, variable type definitions in the Modelica Standard Library reflect the naming conventions

```
model MyCircuit
  Modelica.Electrical.Analog.Sources.ConstantVoltage V(V = 1);
  Modelica.Electrical.Analog.Basic.Resistor R(R = 5);
  Modelica.Electrical.Analog.Basic.Capacitor C(C = 0.1);
  Modelica.Electrical.Analog.Basic.Inductor L(L = 0.5);
  Modelica.Electrical.Analog.Basic.Ground ground;
equation
  connect(V.p, R.p);
  connect(R.n, C.p);
  connect(C.n, L.p);
  connect(V.n, L.n);
  connect(V.n, G.p);
end MyCircuit;
```

Figure 2.5: Modelica text of the RLC example implemented inside Dymola [4]

for physical quantities given by the International System of Units. In addition to using components and connections provided by the MSL, the designer can also create his/her own components and connections using the Modelica language specification [33]. The Modelica language also contains many advanced capabilities [34]:

1. *Matrix Equations*: Useful for multi-body modeling. Allows the designer to build equations based on matrices.

2. *Model Class Parameters*: defining a model class as a parameter using the "redeclare" construct so that only one version of the model class is needed.

3. *Discontinuous and Instantaneous Equations*: Useful for modeling hybrid and switched systems. These equations allow for the modeling of discontinuities, discrete events, and changes to the structure of the equations.

4. *Algorithms and Functions*: Useful for when the system contains parts pertaining to procedural programming. Because of the special semantic meaning for the equal sign in Modelica (strict equality as opposed to an assignment), assignments in procedural programming are implemented using ":=".

Each Modelica model begins as a .mo file containing a series of interconnected components. The translator *flattens* the equations of the .mo file, which includes processes such as type checking, inheritance matching, and import statements. The optimizer then applies Pantelide's Algorithm to the flatten model equations in an effort to reduce the index of

the differential and algebraic equations (DAE) system [35]. The optimized flatten equations are then used by the code generator to generate source code (e.g. C code) needed for simulation, which the compiler transforms into an executable [36].

2.2 Simulation

Causal models, consisting of hierarchical block diagrams, are commonly used for the computational domain because of the explicit input/output format; transmission of a signal occurs through a connection between the blocks, and the signal represents the value of the variable that is being transferred between the blocks. However, in the physical domain, acausal models are used to preserve the physical reality of the modeled system; using a set of declarative equations, physically intuitive relationships are defined among the variables.

As a result of the innate differences between causal and acausal models, simulation is handled differently. Causal models are simulated through a set of ordinary differential equations (ODE); Simulink is an example of a causal simulation environment that simulates systems using either a fixed-step or variable-step ODE solver. Acausal models are simulated through a set of DAEs [5]; Modelica is an example of an acausal simulation environment that simulates systems using DAE solvers [37]. A DAE is a more general form of an ODE, where variables are implicitly defined, and reducing its index involves the use of Pantelide's Algorithm [35].

Because CPS contains both physical and computational domains, co-simulation platforms have been developed. Extension of physical domain simulators to including computational elements has certain shortcomings which inhibits its ability to apply to a more general network scheme [38]. Extension of computational domain simulators to including physical dynamic elements is also problematic because physical models are characterized by DAEs, which are difficult to capture in a computational domain [39]. Recent research has been focused on the integration of a full computational domain simulator with a full physical domain simulator [40].

2.2.1 Simulation of Causal Models

MATLAB/Simulink is a popular graphical tool for the simulation of block diagrams [41]. Simulink is widely used for control design and digital signal processing and has become a powerful tool for the design of CPS. For the design of CPS, integration between components of different domains is essential. S-functions in the Simulink environment extends the capabilities of Simulink by allowing the interfacing with other software packages. For example, CarSim, a commercial software package which models and simulates a vehicle's performance given driver action and environment interfaces with Simulink through S-functions [42]. Previous work done on the hardware-in-the-loop simulation platform involves the design of control software using Simulink and simulating and verifying the system using CarSim S-functions; C-code is then generated from Simulink and implemented on the HIL platform which simulates the behavior of the vehicle in real time [7] [8].

Similar to S-functions, Functional Mockup Interface (FMI) was developed with the goal of defining a standardized interface for the development of complex CPS. The development of FMI was influenced by the idea of component-based design; a *virtual* system which is assembled from a set of components, each with its physical laws and computational dynamics captured [43]. Because CPS contain tightly integrated components of different domains, the use of different software environments for each domain is desirable; FMI allows the integration of these different software domain through model exchange or co-simulation [44]. Comparatively, FMI provides a few advantages over S-functions [43]:

1. S-function DLL is specific to the Simulink simulator, whereas for FMI the DLL is specific to the modeling environment.

2. S-function is not suitable for embedded systems because of the large memory overhead caused by storing all information in the DLL. For FMI only the minimum amount of information is stored in the DLL; the information not required for simulation is put on an XML file.

3. Complexity of the S-function definition dwarfs the complexity of FMI definition.

4. S-function is proprietary to Mathworks, Inc, which can cause legal problems. FMI, on the other hand, uses a BSD license which imposes minimal restrictions.

Bond graphs present a physical representation of systems, and while the representation is intuitive, generation of equation is not possible without determining the causal relationship among the variables. Each bond in the model is interpreted as a bi-directional signal flow. Through a process called causality analysis, the signal directions in each bond can be determined, which allows for the generation of equations [45]. Causality analysis uses causal strokes, which indicate the direction of the imposition of effort. Different bond graph elements have different constraints regarding the imposition of effort or flow, which results in the following causality rules:

1. Fixed Causality: the constituent equations of the bond graph element only allows for one of the two power conjugate variables to be the outgoing variable. There are two cases in which fixed causality can occur, either through source elements or nonlinear elements. A source element delivers either an effort or a flow into the system by definition, depending on whether it is a source of effort or source of flow. In nonlinear elements the constituent equations are singular, which forces one variable to be the output; nonlinearities commonly occur in resistive and storage elements.

2. Constrained Causality: causality of one of the bonds attached to the element imposes constraints on all other bonds attached to the element. Zero junctions, one junctions, linear transformers, and linear gyrators are examples of bond graph elements with constrained causality. In zero junctions, one bond must impose effort, whereas all other bonds must impose flow. The opposite occurs for one junctions, where one bond imposes flow, while all other bonds must impose effort. For transformers, one bond impose effort while the other bond impose flow. For gyrators, both bonds impose either effort or flow.

3. Indifferent Causality: there are no causality constraints on these bond graph elements. The best example for an indifferent causality element is the linear resistor, where either power conjugate variables can be the output variable.

4. Preferred Causality: there is a preference for the causality of the bond. Preferred causality occurs at linear storage elements, where causality determines either an integration or a differentiation for the constituent equations. Since integration with respect to time is a process that is realizable, integration is considered to be "preferred" compared to differentiation. Differentiation requires future information which results in an anti-causal system.

Once causality analysis is applied to the bond graph model, a series of equations can be derived from the bond graph model. In the case that there are no causality conflicts in the model, a set of ordinary-differential equations are derived. In the case that there are causality conflicts in the model, a set of differential and algebraic equations are then derived [46]. An ODE solver can then be used to simulate the bond graph model and obtain simulation results. The *Sequential Causal Assignment Procedure* is an algorithm that assigns causality to every bond in a bond graph model [21]; an application to hybrid bond graphs allows for the derivation of state equations [47].

2.2.2 Simulation of Acausal Models

The model translation and simulation process for acausal simulation in a Modelica environment is depicted in Figure 2.6. Each Modelica model begins as a .mo file containing a series of interconnected components. The translator *flattens* the equations of the .mo file, which includes processes such as type checking, inheritance matching, and import statements. The optimizer then applies Pantelide's Algorithm to the flatten model equations in an effort to reduce the index of the DAE system to an ODE [35]. The optimized flatten equations are then used by the code generator to generate C code needed for simulation,

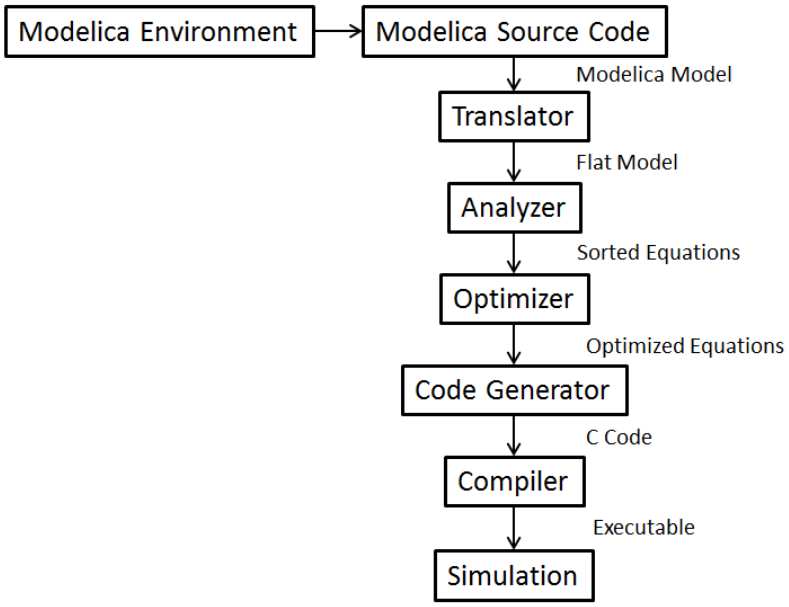

Figure 2.6: Modelica simulation process [5]

which the compiler transforms into an executable [36].

By modeling systems in Modelica, the designer is allowed to skip the causality assignment procedure. In a way, instead of needing to transform the physical models through a computational procedure involving causality, the designer models the physics of the system and lets the simulation environment do the rest.

2.3 Passivity

Control design is traditionally implemented as part of the computational domain, where signals have explicit input-output formats. However, many control design techniques for CPS are focused on the energy of the system, which introduces passivity [48]. The concept of passivity originated in circuit theory, where a filter consisting of only transformers, resistors, inductors, and capacitors does not generate energy; additionally, these filters are guaranteed to be stable and are robust to disturbances [49]. Consequently, passivity is a powerful tool used for control design and system analysis [48].

A passive system is guaranteed to be stable if a positive definite storage function is

used [50]. Therefore, passivity provides a sufficient condition for the stability of a system; a passive system, when unforced, is Lyapunov stable [51] [52]. In the case that the storage function used is positive semi-definite instead, additional conditions on zero-state detectability is needed to ensure that the passive system remains stable [53]. In addition to stability, passive systems also contain phase properties [54]. Passive systems have minimum-phase and possess a low relative degree [55] [56]. Technical details on passivity are shown in Section 3.2.

2.4 Safety of Cyber-Physical Systems

As a result of complex dynamics within CPS, safety analysis of CPS becomes an important problem [57]. When a system contains purely discrete dynamics, techniques such as temporal logic can be used to verify and ensure that these systems always operate within a safe region [58]. On the other hand, a system which contains purely continuous dynamics can be verified using theorems and principles developed in control theory (stability analysis) [59]. However, since CPS contains both discrete and continuous dynamics, new methods are required for its safety analysis extending beyond methods use for purely discrete or purely continuous systems.

Hybrid systems have been commonly used to model many kinds of CPS, and in this section we present several methods for safety analysis of hybrid systems. Analysis of hybrid systems revolves around the determination of the reachable sets for the continuous dynamics of the system [60]. Most early methods for determining safety requires the calculation of either the exact set of reachable states or an approximated set of reachable states, using a wide variety of techniques such as quantifier elimination [61], geometric programming relaxations [62], ellipsoidal calculus [63], and polygonal approximations [64] [65]. Safety analysis using ellipsoidal calculus is interesting because the Hamiltonian function of a PHS is quadratic, being the sum of squares of the state variables, and is inherently ellipsoidal [66] [67].

Barrier certificates, which are similar in structure to Lyapunov functions, were introduced for the purpose of verifying nonlinear systems with uncertainties [68]. The use of barrier certificates allows for the validation of a larger class of continuous-time nonlinear models, including differential-algebraic systems, models with uncertain inputs, and hybrid models [69]. Barrier certificates are functions of state-parameter-time which denote that there are no state trajectories starting from a given set of initial conditions that ends up in an unsafe region of operation; the sufficient condition states that if there exists a barrier certificate for a system, then the system is safe [70]. Technical details on barrier certificates are shown in Section 3.3.

2.5 Automotive Cyber-Physical Systems

The first automobiles were designed to transport people and supplies from one place to another; these automobiles were completely characterized by their physical dynamics, and the many advancements in this area has enabled automobiles to become a popular choice for meeting society's transportation demands [71]. However, problems such as traffic congestion and safety concerns have spurred the addition of autonomous controllers to automobiles, which changes automobiles from pure physical dynamics to complex CPS. As the number of controllers added to automobiles increase, automotive CPS becomes more complex and rigorous engineering methods are needed to ensure the safety and operations of modern vehicles [12].

Automotive Open System Architecture (AUTOSAR) is a worldwide development partnership with the goal establishing an open-source and standardized software architecture for automotive ECUs. AUTOSAR contains details which describe standardized software modules, define how application software components interface with one another, and build a universal design approach using a standardized exchange format. These details allow AUTOSAR to help innovate automotive electronic systems to further improve the performance, safety, and efficiencies of modern vehicles. The scalability of different software

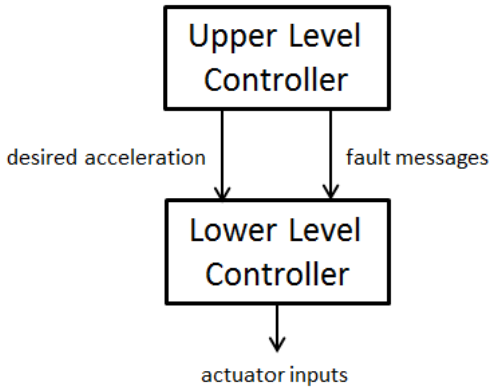

Figure 2.7: High level controller configuration for ACC [6]

and hardware components over the vehicle product life cycle is also facilitated by AUTOSAR. AUTOSAR strives to assist in the design and development of future vehicles and to improve the cost efficiency of integrating new technologies [72].

2.5.1 Adaptive Cruise Control

An Adaptive Cruise Control (ACC) system extends from the traditional cruise control system, where the vehicle maintains the user's set speed by automatically tracking the speed with respect to the throttle angle [73]. Studies have shown that ACC systems can improve traffic flow characteristics and lead to more efficient vehicles [74]. The ACC behaves just like the traditional cruise control system when the host vehicle does not detect a leading vehicle. However, if the radar system of the host vehicle detects a lead vehicle, the ACC determines a new set speed for which the host vehicle is allowed to travel. If the lead vehicle is traveling at a speed slower than the host vehicle, then the ACC will push the host vehicle's speed to match that of the lead vehicle speed while maintaining a safe distance. Doing so ensures that the host vehicle drives comfortably and also avoids potential collisions.

Early development of adaptive cruise control systems focused on extending the traditional cruise control system by adding hybrid dynamics in the form of throttling and

braking. Individual control algorithms within the ACC were implemented as either PID controllers with fixed gains or adaptive controllers where the gains are determined by gain scheduling [75]. As vehicle systems get more complicated, a more complicated control architecture is needed to ensure that the vehicle behaves properly under the influence of the ACC. Recent development in ACC has prompted a more hierarchical control architecture, where an upper level controller determines the desired acceleration of the vehicle, and the lower level controller computes the throttle or brake command based on the desired acceleration. The two controllers work together to achieve two stability conditions: individual stability, where the vehicle on its own will asymptotically reach a desired speed and spacing, and also platoon stability, where a legion of vehicles will all satisfy individual stability [76]. Control barrier functions have also been used in a control design approach demonstrated for ACC and place constraints on the host vehicle's acceleration and deceleration [77]. They balance the objectives of maintaining a desired host vehicle velocity and a relative distance above a minimum threshold.

2.5.2 Lane Keeping Control

The concept of Lane Keeping Control (LKC) first began as an extension of the lane departure warning system which have started to appear on high end vehicles in the early 2000's. The technology has matured since then and many car manufacturers have extended its functionality, resulting in a semi-autonomous vehicle when coupled with the ACC [78]. The objective of an LKC system is to control the steering of the vehicle in order to keep the vehicle in the center of the lane [79]. The LKC either steers the vehicle towards the center of the lane when it detects that the vehicle is drifting or it proactively keeps the vehicle in the center of the lane [80]. New systems are being developed which increases the functionality of LKC further by including lane changing maneuvers [81].

Lane keeping controllers are commonly designed using nested PID algorithms, see Figure 2.8 [7]. Through the use of sensors and cameras, the LKC is able to detect when the

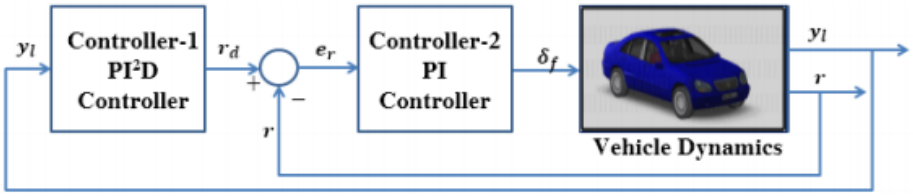

Figure 2.8: High level controller configuration for LKC [7]

vehicle is about to depart from the lane and adjust the steering wheel accordingly to keep the vehicle in lane [82]. A vehicle, not unlike an airplane, has three degrees of freedom in terms of rotational motion: pitch, roll, and yaw. Under most normal driving circumstances, the rotational motions of pitch and roll are ignored for obvious reasons. The LKC affects the yaw of the vehicle by adjusting the steering [83].

For the vehicle dynamics, the lateral dynamics and longitudinal dynamics are related. The lateral dynamics model is dependent on the speed of the vehicle. This can sometimes cause a problem which manifests through the interaction of LKC and ACC. The scenario in which the ACC drives the vehicle to approach a curve at a high speed may hamper the LKC's ability to keep the vehicle in the lane; this is potentially catastrophic as the vehicle can spin out of control. This problem is alleviated by the introduction of a supervisory controller, which will limit the vehicle's speed when taking a curve, ensuring that the LKC can satisfy its control objective [84].

2.6 Comparison with Dissertation Work

Heterogeneity is a major challenge for the component-based modeling and design of CPS. As CPS continue to increase in complexity and scale, traditional *ad hoc* ways of integration become more and more unreliable which results in the need for more rigorous methods of integration. The goal of the work in this dissertation is to address problems and challenges which are highlighted in Section 1.2. In particular, we developed methods and tools for the modeling, design, and analysis of CPS using PHS.

Modeling of CPS Most of the literature on this subject focuses on the heterogeneous

modeling of different computational and physical domain, then integrating them together through an abstraction of each layer. This form of modeling can become complex and requires well-defined interactions between the different layers of CPS. The modeling approach presented in this dissertation is inspired by the port-based modeling of physical systems, in which different domains are integrated together through a common notion of power and energy. By modeling the computational domain of CPS using PHS, we can ensure passivity when integrated with the physical domain.

Simulation of CPS Traditional bond graph simulation requires the causality assignment of the global model, which generates a large set of equations pertaining to the model. The simulation technique presented in this dissertation approaches the causality assignment procedure in a compositional way. By instantiating each component as a generalized bond graph, we derive its Dirac structure in the form of a kernel representation and and a hybrid-input-output representation. The kernel representation provides an acausal computational model which we simulate using Modelica, and the hybrid-input-output representation provides a causal computational model which we simulate using Simulink.

Control design in CPS The work presented in this dissertation is influenced by passivity-based control. We use the interconnection rules of PHS to ensure that the resulting controller, and the subsequent closed-loop system, is passive. We model hybrid and switching dynamics through PHS, which ensures that passivity is not affected. Control ports and Dirac structures allow for the composition of multiple controllers with the plant system. This control design method will serve as the first step of the greater model-design process. We also implemented the end-to-end process on a hardware-in-the-loop simulation platform.

Safety analysis of CPS (Specifically referring to the contents of Section 3.3)Safety analysis methods such as ellipsoidal approximations and barrier certificates contain complex mathematical definitions which precludes the system from beginning in a safe state and ending in an unsafe state. In the case of barrier certificates these bounds and conditions

are derived using sum of squares decomposition methods, which can more than adequately handle systems whose dynamics can be described as polynomials. However, many CPS contain components which are highly non-polynomial, which necessitates a different way of obtaining the barrier certificate. In this dissertation we propose to use the Hamiltonian function of the PHS as a barrier certificate in order to show safety.

Automotive CPS The automotive CPS that we consider in this dissertation are systems which contribute to autonomous driving, specifically ACC and LKC systems. By working together they eliminate the need for a driver to make sure that the vehicle stays on the road and moves at an appropriate speed. Composition of different automotive systems are typically done in an 'ad hoc' way in which individual components are designed separately and then composed together under the supervision of a supervisory controller. The work done in this dissertation uses PHS so that there is an actual composition rule, which makes it easier to introduce additional systems.

Chapter 3

Background

In this chapter we present some theoretical background on port-Hamiltonian systems (PHS), passivity, and barrier certificates. In Section 3.1 we show technical details on PHS, including an extension to multi-modal PHS used to described hybrid/switched systems. In Section 3.2 we present background on the basics of continuous-time (Section 3.2.1) and discrete-time (Section 3.2.2) passivity. In Section 3.2.3 we show how passivity is applied to hybrid/switched systems. In Section 3.2.4 we present the compositional property of passivity and passivity indexes. In Section 3.2.5 we show how passivity is used in PHS for the purpose of control design. We conclude the background chapter with a review of barrier certificates and how they are used to validate the safety of nonlinear and hybrid systems (Section 3.3).

3.1 Port-Hamiltonian Systems

Physical systems are generally described a set of differential and algebraic equations, since algebraic constraints between state variables are a possibility [19]. Though implicit equations create problems in simulation, they generally have no effect on mathematical formulations. In particular, PHS model both implicit and explicit systems through the Dirac structure [85].

Definition 1. *[2] Let F be an n-dimensional linear space of flow variables f, and F^* be its dual space of effort variables e, power is defined as:*

$$P = \langle e, f \rangle, (f, e) \in F \times F^*.$$

Definition 2. *[2] A Dirac structure on $F \times F^*$ is a subspace $D \subset F \times F^*$ if the following two conditions are true:*

1. $\langle e, f \rangle$ where $= 0$, for all $(f, e) \in D$,
2. $\dim[D] = \dim[F]$.

There are various ways to represent Dirac structures, with each representation using different structural matrices. The kernel representation is an acausal representation of the Dirac structure where there are no explicit input-output relationships between the effort and flow variables.

Definition 3. *[2] Every Dirac structure $D \subset F \times F^*$ can be represented in a kernel representation as:*

$$D(x) = \{(f, e) \in F \times F^* : F(x)f + E(x)e = 0\},$$

where the matrices $E(x)$ and $F(x)$ satisfy the following two conditions:

1. $\text{rank}[F(x)|E(x)] = \dim[F]$,
2. $E(x)F^{\mathsf{T}}(x) + F(x)E^{\mathsf{T}}(x) = 0$.

The constrained input-output representation is a causal representation of the Dirac structure. Using a matrix J a cause-effect relationship is defined between the effort and flow variables.

Definition 4. *[2] A Dirac structure $D \subset F \times F^*$ can be represented in a constrained input-output representation as:*

$$D(x) = \{(f, e) \in F \times F^* : f = J(x)e + G(x)\lambda, G^{\mathsf{T}}(x)e = 0\}.$$

In the case of algebraic constraints between the effort and flow variables, it may not be possible to derive a constrained input-output representation. This necessitates the formulation of the hybrid input-output representation, where certain effort and flow variables are interchanged. Suppose that the effort and flow variables are split as $\begin{bmatrix} e_1 & e_2 \end{bmatrix}^\mathsf{T}$ and $\begin{bmatrix} f_1 & f_2 \end{bmatrix}^\mathsf{T}$. Correspondingly, the matrices $E(x)$ and $F(x)$ are also split as $\begin{bmatrix} E_1(x) & E_2(x) \end{bmatrix}$ and $\begin{bmatrix} F_1(x) & F_2(x) \end{bmatrix}$.

Definition 5. *[2] A Dirac structure $D \subset F \times F^*$ can be represented in a hybrid input-output representation as:*

$$D(x) = \{(f,e) \in F \times F^* : \begin{bmatrix} e_1 & f_2 \end{bmatrix}^\mathsf{T} = J(x) \begin{bmatrix} f_1 & e_2 \end{bmatrix}^\mathsf{T}\},$$

where $J(x)$ is a skew-symmetric matrix given by:

$$J(x) = -J^\mathsf{T}(x) = \begin{bmatrix} E_1(x) & F_2(x) \end{bmatrix}^{-1} \begin{bmatrix} F_1(x) & E_2(x) \end{bmatrix}.$$

Regardless of the representation, Dirac structures establish the power balancing equation which is fundamental to PHS:

$$\frac{d}{dt}H = e_R^\mathsf{T} f_R + e_C^\mathsf{T} f_C + e_I^\mathsf{T} f_I. \tag{3.1}$$

H in (3.1) denotes the Hamiltonian function of the PHS. The Hamiltonian function represents the energy stored in the system; the flows of the energy storage are given by the rate $\dot{x}$ of the energy state variables x; the efforts are given by the co-energy variables $\frac{\partial H}{\partial x}$ [86]. e_R and f_R denote the effort and flow variables associated with internal energy dissipation, where the relationship between the power conjugate variables obey a static resistive relationship of the form:

$$R(f_R, e_R) = 0.$$

A special case of resistive relationship arises if the relationship between e_R and f_R can be expressed in an explicit input-output form:

$$f_R = -F(e_R), \tag{3.2}$$

where the function $F(*)$ relates the effort and flow variables. (3.2) is the general equation for a non-linear resistive element. For a linear resistive element (3.2) simplifies down to:

$$f_R = -\tilde{R}e_R,$$

where $\tilde{R}$ is a positive semi-definite symmetric matrix. e_C and f_C denote effort and flow variables associated with external control; these control port variables are accessible to controller action; one variable acts as the actuator and the conjugate variable acts as the sensor. The final terms in (3.1), e_I and f_I, denote effort and flow variables associated with interactions of the PHS with the environment.

A PHS with a Hamiltonian function H, energy storage ports S, resistive ports R, control ports C, interconnection ports I, and a Dirac structure D can be written in a formal model in (3.3),

$$\Sigma = (H,R,C,I,D), \tag{3.3}$$

which leads to the the implicitly defined PHS dynamics:

$$\Sigma : (-x'(t), \frac{\partial H}{\partial x}, f_R(t), e_R(t), f_C(t), e_C(t), f_I(t), e_I(t)) \in D. \tag{3.4}$$

(3.4) simplifies down to an input-state-output PHS when there are no algebraic constraints on the state variables [87]:

$$\Sigma : \begin{cases} \dot{x} = [J(x) - R(x)]\frac{\partial H}{\partial x} + G(x)u + K(x)d \\ y = G^{\mathsf{T}}(x)\frac{\partial H}{\partial x} \\ z = K^{\mathsf{T}}(x)\frac{\partial H}{\partial x} \end{cases} \tag{3.5}$$

Its Dirac structure is summarized in the following skew-symmetric matrix:

$$D : \begin{bmatrix} -J(x) & -G_R(x) & -G(x) & -K(x) \\ G_R^{\mathsf{T}}(x) & 0 & 0 & 0 \\ G^{\mathsf{T}}(x) & 0 & 0 & 0 \\ K^{\mathsf{T}}(x) & 0 & 0 & 0 \end{bmatrix},$$

where the term $G_R(x)$ denotes the resistive structure of the system. An important property of PHS is compositionality. The interconnection of PHS occurs through interaction ports of their respective Dirac structures.

Definition 6. *[88] Let D_A be a Dirac structure defined on the linear space $F_1 \times F_2$, and D_B be Dirac structure defined on the linear space $F_2 \times F_3$, where F_2 is the space of shared power conjugate variables between D_A and D_B. The composition of D_A and D_B, $D_C = D_A||D_B$, is a Dirac structure defined on the linear space $F_1 \times F_3$.*

Definition 6 shows that the composition of two Dirac structures yields another Dirac structure [89]. By extension, the composition of PHS is shown in Definition 7 [90].

Definition 7. *[2] Let $\Sigma_i = (H_i, F_i, D_i)$, where $i = 1, 2, \ldots, k$, be k PHS interconnected through a Dirac structure on the space of $F_1 \times F_2 \times \ldots \times F_k$. This results in a PHS $\Sigma = (H, F, D)$ where $H = H_1 + H_2 + \ldots + H_k$ and $D_I = D_1||D_2||\ldots||D_k$.*

An important generalization of PHS is the multi-modal PHS, which is useful for describing hybrid/switched system. The formulation of multi-modal PHS were inspired by the modeling of switching power converters [91]. Given a plant system with a Hamiltonian function $H_p(x_p)$, continuous states $x_p \in X_p \subseteq \mathbb{R}^{n_p}$, discrete states $s_p \in S_p$, disturbances $\delta \in$

$\mathbb{R}^o$, and a control system a Hamiltonian function $H_c(x_c)$, continuous states $x_c \in X_c \subseteq \mathbb{R}^{n_c}$, and discrete states $s_c \in S_c$, where $\{n_p, n_c, o\} \in \mathbb{N}^4$, we can write the set of dynamic equations of the closed-loop system as an input-state-output multi-modal PHS with Hamiltonian function $H(x) = H_p(x_p) + H_c(x_c)$, continuous states $x = \begin{bmatrix} x_p & x_c \end{bmatrix}^\mathsf{T} \in X = X_p \times X_c$, discrete states $s = \begin{bmatrix} s_p & s_c \end{bmatrix}^\mathsf{T} \in S = S_p \times S_c$, initial states $X_0 = X_{p0} \times S_{p0} \times X_{c0} \times S_{c0}$, and discrete transitions $\mathbb{T} \subseteq (X \times S) \to (X \times S)$:

$$\begin{cases} \dot{x} &= [J(x,s) - R(x,s)] \frac{\partial H}{\partial x} + \begin{bmatrix} L_p(x_p, s_p) \\ 0 \end{bmatrix} \delta \\ \zeta &= \begin{bmatrix} L_p^\mathsf{T}(x_p, s_p) & 0 \end{bmatrix} \frac{\partial H}{\partial x} \end{cases} \tag{3.6}$$

$$J(x,s) = \begin{bmatrix} J_p(x_p, s_p) & -G_p(x_p, s_p) G_c^\mathsf{T}(x_c, s_c) \\ G_c(x_c, s_c) G_p^\mathsf{T}(x_p, s_p) & J_c(x_c, s_c) \end{bmatrix},$$

$$R(x,s) = \begin{bmatrix} R_p(x_p, s_p) & 0 \\ 0 & R_c(x_c, s_c) \end{bmatrix},$$

where $J_p(x_p, s_p) \in \mathbb{R}^{n_p \times n_p}$ and $J_c(x_c, s_c) \in \mathbb{R}^{n_c \times n_c}$ are skew-symmetric interconnection matrices, $R_p(x_p, s_p) \in \mathbb{R}^{n_p \times n_p}$ and $R_c(x_c, s_c) \in \mathbb{R}^{n_c \times n_c}$ are symmetric positive semi-definite damping matrices, $G_p(x_p, s_p) \in \mathbb{R}^{n_p \times m}$, $G_c(x_c, s_c) \in \mathbb{R}^{n_c \times m}$, $L_p(x_p, s_p) \in \mathbb{R}^{n_p \times o}$, and (δ, ζ) are the input-output pairs corresponding to the disturbance port.

3.2 Passivity

Passivity is a mathematical concept which describes power and energy conservation of physical systems [50]. Furthermore, passivity applies in cases of abstract energy, where there is no physical meaning of the energy in the system; this property makes passivity useful in characterizing a larger class of systems [85].

3.2.1 Continuous-Time Passivity

Passivity is commonly defined in the state-space approach, where the energy stored in the system is related to the external energy coming into the system [92]. First we provide a formal definition of a continuous-time dynamical system.

Definition 8. *[93] A dynamical system, J, is described by the following state space realization, where $x \in X \subset R^n$, $u \in U \subset R^m$ and $y \in Y \subset R^m$ are the state, input, and output variables, respectively; X, U, and Y are the state, input, and output spaces, respectively:*

$$J : \begin{cases} \dot{x} = f(x,u) \\ y = h(x,u) \end{cases}$$

A supply function, W, is then defined as follows:

Definition 9. *[93] The supply rate $W(t) = W(u(t), y(t))$ is a function defined on $U \times Y$, such that $W(t)$ satisfies the following constraint:*

$$\int_{t_0}^{t} |W(t)| dt < \infty.$$

Using both Definitions 8 and 9, we can provide the following definition for a dissipative system:

Definition 10. *[93] A dynamical system, J, is dissipative with respect to its supply rate $W(t)$ if there exists a non-negative storage function $V(x)$ such that:*

$$\int_{t_0}^{t} W(\tau) d\tau \geq V(x) - V(x_0). \tag{3.7}$$

The inequality presented in (3.7) is called the dissipation inequality, which leads to several choices for the supply rate function, $V(x)$ [56]. This results in the formulation of QSR dissipativity supply rates, which assumes that the system is a black box and only relates the inputs to the system with the outputs of the system [50].

Definition 11. *[56] Given constant matrices Q, S, and R, with Q and R symmetric, a dynamical system, J, is QSR Dissipative if it is dissipative with respect to the supply rate, W, defined as:*

$$W(u,y) = y^{\mathsf{T}}Qy + 2y^{\mathsf{T}}Su + u^{\mathsf{T}}Ru.$$

Specific choices of the matrices Q, S, and R results in different cases of passivity, where ε, δ, are positive constants and γ is an arbitrary constant:

1. Passive if $Q = 0, S = \frac{1}{2}I, R = 0$.
2. Strictly Input Passive (SIP) if $Q = 0, S = \frac{1}{2}I, R = -\delta I$.
3. Strictly Output Passive (SOP) if $Q = -\varepsilon I, S = \frac{1}{2}I, R = 0$.
4. Very Strictly Passive (VSP) if $Q = -\varepsilon I, S = 0, R = -\delta I$.
5. Finite-Gain Stable (FGS) if $Q = -I, S = 0, R = \gamma^2 I$.

From these conditions, we can see that passivity is a specific form of QSR dissipativity when the system is constrained to being a square system (consisting of the same number of inputs and outputs). The applicability of passivity arises from the fact that its definition applies to a wide variety of systems: linear or nonlinear, time-invariant or time-variant, continuous time or discrete time. For the continuous-time definition of input-output passivity, we must first describe the L_2-space. A signal y is in the L_2-space of functions if it has has finite energy:

$$\int_0^\infty y^{\mathsf{T}}y(t) < \infty. \tag{3.8}$$

The only functions that satisfies the constraints of (3.8) are functions which asymptotically approach zero. In order to define the passivity using an input-output structure, first we must define an inner product:

Definition 12. *[50] The inner product of L_2-space signals u and y is defined as:*

$$\langle u, y \rangle = \int_0^\infty u^\mathsf{T}(t) y(t) dt.$$

Using definition 12, (3.8) can be rewritten as

$$\int_0^\infty y^\mathsf{T} y(t) = \langle y, y \rangle < \infty. \tag{3.9}$$

In the case of a non-∞ start time

$$\langle u, y \rangle_T = \int_0^T u^\mathsf{T}(t) y(t) dt.$$

(3.9) implies that the finite truncation of any signal that is part of the L_2-space must have finite energy.

$$\langle y, y \rangle_T = \int_0^T y^\mathsf{T}(t) y(t) dt = ||y||^2.$$

Using all the definitions above and the QSR dissipativity of definition 11, we can define passivity in terms of input u and output y as follows:

Definition 13. *[50] Given a dynamical system, J, with input signal u and output signal y in L_2-space:*

1. J is *passive* if there exists a constant β such that:

$$\langle Ju, u \rangle_T = \langle y, u \rangle_T \geq -\beta, \forall T \geq 0.$$

2. J is *strictly output passive* if there exist constants β and $\varepsilon \geq 0$ such that:

$$\langle Ju, u \rangle_T = \langle y, u \rangle_T \geq \varepsilon \langle y, y \rangle_T - \beta, \forall T \geq 0.$$

3. J is *strictly input passive* if there exist constants β and $\delta \geq 0$ such that:

$$\langle Ju, u\rangle_T = \langle y, u\rangle_T \geq \delta\langle u, u\rangle_T - \beta, \forall T \geq 0.$$

4. J is *very strictly passive* if there exist constants β and $\varepsilon, \delta \geq 0$ such that:

$$\langle Ju, u\rangle_T = \langle y, u\rangle_T \geq \delta\langle u, u\rangle_T + \varepsilon\langle y, y\rangle_T - \beta, \forall T \geq 0.$$

5. J is *finite gain stable* if there exist constants β and γ such that:

$$\langle Ju, Ju\rangle_T = \langle y, y\rangle_T \leq \gamma\langle u, u\rangle_T - \beta, \forall T \geq 0.$$

3.2.2 Discrete-Time Passivity

Similar to the definitions of continuous-time passivity of the previous section, discrete-time passivity is also defined in terms of the state space approach extended from the QSR dissipativity supply rates [94]. The definitions for passivity in discrete-time is similar in format to those in continuous-time. A discrete-time dynamical system is defined as follows:

Definition 14. *[94] A dynamical system, J, is described by the following state space realization, where x ε X $\subset R^n$, u ε U $\subset R^m$ and y ε Y $\subset R^m$ are the state, input, and output variables, respectively; X, U, and Y are the state, input, and output spaces, respectively:*

$$J : \begin{cases} x(k+1) = f(x(k), u(k)) \\ y(k) = h(x(k), u(k)) \end{cases}$$

A supply function, $W(k)$, is then defined as follows:

Definition 15. *[94] The supply rate $W(k) = W(u(k), y(k))$ is a function defined on $U \times Y$, such that $W(k)$ satisfies the following constraint:*

$$\sum_{k=0}^{\infty} W(k) < \infty.$$

Using both definitions 14 and 15, we can provide the following definition for a discrete-time dissipative system:

Definition 16. *[94] A dynamical system, J, is dissipative with respect to its supply rate $W(k)$ if there exists a non-negative storage function $V(k)$ such that:*

$$\sum_{k=0}^{N} W(u(k), y(k)) \geq V(k+1) - V(0).$$

The QSR Dissipativity Supply Rate, converted to discrete-time, provides the following extension to definition 16:

Definition 17. *[94] Given constant matrices Q, S, and R, with Q and R symmetric, a dynamical system, H, is QSR dissipative if it is dissipative with respect to the supply rate, W(k), as:*

$$W(u(k), y(k)) = y^{\mathsf{T}}(k)Qy(k) + 2y^{\mathsf{T}}(k)Su(k) + u^{\mathsf{T}}(k)Ru(k).$$

Similar conditions for Q, S, and R, such as those in definition 11 apply for the various forms of discrete-time passivity [95]. The formal discrete-time definition of input-output passivity has signal y is in the L_2-space of functions if it has has finite energy [96]:

$$\sum_{k=0}^{\infty} y^{\mathsf{T}}(k)y(k) < \infty. \tag{3.10}$$

The inner product in discrete time is similarly defined as:

Definition 18. *[50] The inner product of L_2-space signals $u(k)$ and $y(k)$ is defined as:*

$$\langle u(k), y(k) \rangle = \sum_{k=0}^{\infty} u^{\mathsf{T}}(k)y(k).$$

Using Definition 18, (3.10) can be rewritten as:

$$\sum_{k=0}^{\infty} y^{\mathrm{T}}(k)y(k) = \langle y(k), y(k)\rangle < \infty.$$

In the case of a non-∞ end time,

$$\langle u(k), y(k)\rangle_N = \sum_{k=0}^{N} u^{\mathrm{T}}(k)y(k).$$

$$\langle y(k), y(k)\rangle_N = \sum_{k=0}^{N} y^{\mathrm{T}}(k)y(k)dt = ||y(k)||^2.$$

Using all of the definitions above and the QSR dissipativity of Definition 17, we can define passivity in terms of input $u(k)$ and output $y(k)$ as follows [97]:

Definition 19. *[94] Given a dynamical system, J, with input signal $u(k)$ and output signal $y(k)$ in L_2-space:*

1. J is *passive* if there exists a constant β such that:

$$\langle Ju(k), u(k)\rangle_N = \langle y(k), u(k)\rangle_N \geq -\beta, \forall N \in \mathbb{N}.$$

2. J is *strictly output passive* if there exist constants β and $\varepsilon \geq 0$ such that:

$$\langle Ju(k), u(k)\rangle_N = \langle y(k), u(k)\rangle_N \geq \varepsilon\langle y(k), y(k)\rangle_N - \beta, \forall N \in \mathbb{N}.$$

3. J is *strictly input passive* if there exist constants β and $\delta \geq 0$ such that:

$$\langle Ju(k), u(k)\rangle_N = \langle y(k), u(k)\rangle_N \geq \delta\langle u(k), u(k)\rangle_N - \beta, \forall N \in \mathbb{N}.$$

4. J is *very strictly passive* if there exist constants β and $\varepsilon, \delta \geq 0$ such that:

$$\langle Ju(k), u(k)\rangle_N = \langle y(k), u(k)\rangle_N \geq \delta\langle u(k), u(k)\rangle_{\mathbf{N}} + \varepsilon\langle y(k), y(k)\rangle_N - \beta, \forall N \in \mathbb{N}.$$

5. J is *finite gain stable* if there exist constants β and γ such that:

$$\langle Ju(k), Ju(k)\rangle_N = \langle y(k), y(k)\rangle_N \leq \gamma\langle u(k), u(k)\rangle_N - \beta, \forall N \in \mathbb{N}.$$

3.2.3 Passivity of Hybrid/Switched Systems

Passivity of hybrid systems is based on the energy of the system in the presence of switching behaviors [98] [99] [100]. Instead of formulating a global Lyapunov or storage function, stability of hybrid systems is analyzed through the stability of each mode [101] [102]. Passivity for hybrid systems is defined in terms of the storage functions of each mode [103].

Definition 20. *[104] A hybrid system given by the tuple H is passive if for every mode i there exists a continuously differentiable storage function $V_i(x)$ such that the following conditions hold for non-negative δ and ε, and a positive semi-definite function $\phi(x)$:*

1. The system is instantaneously passive while each mode is active,

$$u^\mathsf{T} y \geq \frac{dV_i}{dt} + \delta u^\mathsf{T} u + \varepsilon y^\mathsf{T} y + \psi(x).$$

2. When switching to each mode, the energy that each subsystems accumulates while inactive is bounded by the energy supplied during the time interval,

$$\int_{t_0}^{t} u^\mathsf{T} y d\tau \geq V_i(x(t)) - V_i(x(t_0)).$$

Passivity of hybrid and switched systems is formulated in PHS through the use of switches [105]. From a bond graph point of view, a switch is defined as either a zero

source of effort or a zero source of flow [106]. Regardless of the mode of the switch, it always delivers zero power into the system, thus never affecting the total energy of the system [107]. As result, as long as modes of the system can be represented as passive PHS, even the introduction of switching behavior will not change the overall passivity of the system [108]. A more general definition for passivity is defined for switched systems, which has a more relaxed set of assumptions. Switched systems are a special class of hybrid system, where emphasis is placed on the continuous dynamics and the discrete dynamics as finite switching [99].

Definition 21. *[100] A switched system is passive if for every mode i there exists a storage function $V_i(x)$ such that the following conditions hold for non-negative δ and ε:*

1. The system is instantaneously passive while each mode is active,

$$u^{\mathsf{T}} y \geq \frac{dV_i}{dt} + \delta u^{\mathsf{T}} u + \varepsilon y^{\mathsf{T}} y.$$

2. Each mode is dissipative during its inactive time,

$$w(u,y) \geq \frac{dV_i}{dt} + \delta u^{\mathsf{T}} u + \varepsilon y^{\mathsf{T}} y.$$

3. There exist functions ϕ_j^i for each cross supply rate that are absolutely summable and such that,

$$w_j^i(u,y,x,t) \leq \phi_i^j, \forall j \neq i.$$

3.2.4 Passivity Indexes

There are three interconnection rules for the construction of large scale passive systems through passive subsystems: parallel interconnection, feedback interconnection, and symmetric input-output transformation [109]. By following these three simple rules, we can

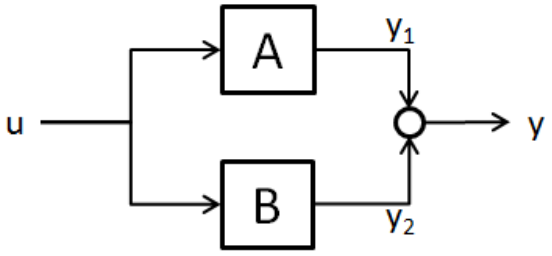

Figure 3.1: Parallel interconnection of two systems, A and B.

guaranteed that the overall system is passive as long as each subsystem is passive. These composition rules form the foundation of passivity-based control design [110].

Theorem 1. *[111] The parallel interconnection of two passive systems (Figure 3.1) results in a passive overall system.*

Proof. Since each system, A and B, is passive, there exists storage functions V_A and V_B such that

$$\int_{t_0}^{t} u^{\mathsf{T}} y_1 d\tau \geq V_A(x_A(t)) - V_A(x_A(t_0)), \tag{3.11}$$

$$\int_{t_0}^{t} u^{\mathsf{T}} y_2 d\tau \geq V_B(x_B(t)) - V_A(x_B(t_0)). \tag{3.12}$$

We then define a new storage function for the composed system, $V(x)$:

$$V(x) = V_A(x_A) + V_B(x_B).$$

Now summing the two inequalities in (3.11) and (3.12):

$$\begin{aligned}
\textstyle\int_{t_0}^{t} u^{\mathsf{T}} y_1 d\tau + \int_{t_0}^{t} u^{\mathsf{T}} y_2 d\tau &\geq V_A(x_A(t)) - V_A(x_A(t_0)) + V_B(x_B(t)) - V_A(x_B(t_0)) \\
\textstyle\int_{t_0}^{t} u^{\mathsf{T}} (y_1 + y_2) d\tau &\geq [V_A(x_A(t)) + V_B(x_B(t))] - [V_A(x_A(t_0)) + V_A(x_B(t_0))] \\
\textstyle\int_{t_0}^{t} u^{\mathsf{T}} y d\tau &\geq V(x(t)) - V(x(t_0))
\end{aligned}$$

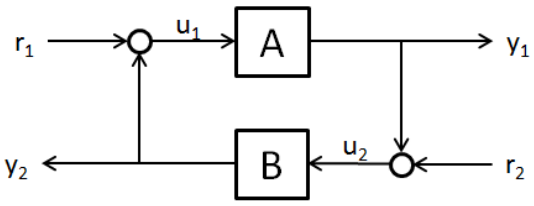

Figure 3.2: Feedback interconnection of two systems, A and B.

The input-output mapping of u to y is passive.

□

Theorem 2. *[111] The feedback interconnection of two passive systems (Figure 3.2) results in a passive overall system.*

Proof. Since each system, A and B, is passive, there exists storage functions V_A and V_B such that

$$\int_{t_0}^{t} u_1^{\mathsf{T}} y_1 d\tau \geq V_A(x_A(t)) - V_A(x_A(t_0)), \tag{3.13}$$

$$\int_{t_0}^{t} u_1^{\mathsf{T}} y_2 d\tau \geq V_B(x_B(t)) - V_A(x_B(t_0)). \tag{3.14}$$

Now summing the two inequalities in (3.13) and (3.14):

$$\begin{aligned}
\textstyle\int_{t_0}^{t} u_1^{\mathsf{T}} y_1 d\tau + \int_{t_0}^{t} u_2^{\mathsf{T}} y_2 d\tau &\geq V_A(x_A(t)) - V_A(x_A(t_0)) + V_B(x_B(t)) - V_A(x_B(t_0)) \\
\textstyle\int_{t_0}^{t} (r_1 - y_2)^{\mathsf{T}} y_1 d\tau + \int_{t_0}^{t} (r_2 - y_1)^{\mathsf{T}} y_2 d\tau &\geq [V_A(x_A(t)) + V_B(x_B(t))] - [V_A(x_A(t_0)) + V_A(x_B(t_0))] \\
\textstyle\int_{t_0}^{t} r^{\mathsf{T}} y d\tau &\geq V(x(t)) - V(x(t_0))
\end{aligned}$$

The input-output mapping of r to y is passive.

□

Figure 3.3: Symmetric input-output transformation of system A

Theorem 3. *[55] The system composition (Figure 3.3) where the inputs of a passive system A are pre-multipled by a matrix, M, and the outputs of the passive system are are post-multiplied by the transpose of the same matrix, M^{T}, results in a passive overall system.*

Proof. Since the system A is passive, there exist a storage function V

$$\int_{t_0}^{t} u_1^{\mathsf{T}} y_1 d\tau \geq V(x(t)) - V(x(t_0)).$$

Now transforming u_1 and y_1 to u and y through the matrices M and M^{T}, respectively:

$$\begin{aligned} \int_{t_0}^{t} (Mu)^{\mathsf{T}} ((M^{\mathsf{T}})^{-1} y) d\tau &\geq V(x(t)) - V(x(t_0)) \\ \int_{t_0}^{t} u^{\mathsf{T}} M^{\mathsf{T}} (M^{\mathsf{T}})^{-1} y d\tau &\geq V(x(t)) - V(x(t_0)) \\ \int_{t_0}^{t} u^{\mathsf{T}} y d\tau &\geq V(x(t)) - V(x(t_0)) \end{aligned}$$

The input-output mapping of u to y is passive.

□

These interconnection rules and the preservation of passivity leads to the concept of passivity index. Passivity as a property is inherently binary, where a system is either passive or not passive. Passivity indexes extends the passivity formulation by determining the degree of passivity or how far a system is from being passive [112]. A system that is not passive may be close to being passive in that a small feedback or feed-forward gain will passify it. A system which is excessively passive can be used to connect in parallel or feedback to a non-passive system in order to passify it. A positive value for an index means that the system has an *excess* of passivity; a negative value for an index means that the system has a *shortage* of passivity. There are two passivity indexes: input feed-forward

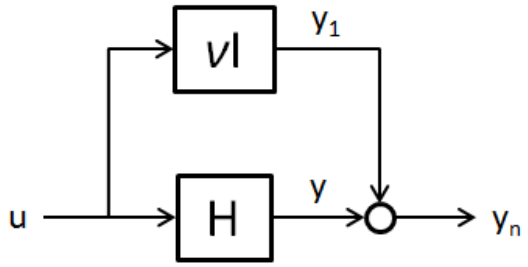

Figure 3.4: Input feed-forward passivity

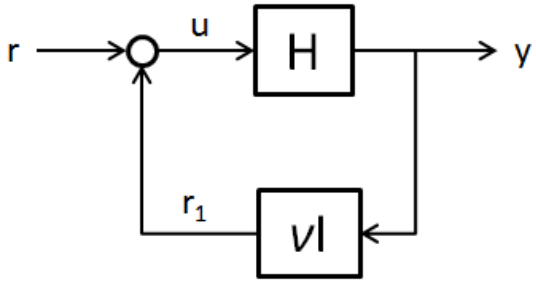

Figure 3.5: Output feedback passivity

passivity (IFP) and output feedback passivity (OFP) [113].

Definition 22. *[114] The input feed-forward passivity index (shown in Figure 3.4), IFP(ν), is the largest gain that can be put in a negative parallel interconnection with a system such that the interconnected system is passive.*

Definition 23. *[114] The output feedback passivity index (shown in Figure 3.5), OFP(ρ), is the largest gain that can be put in a positive feedback interconnection with a system such that the interconnected system is passive.*

The IFP index measures the minimum-phase property of the system, while the OFP index measures the stability of the system [115]. When a system lacks IFP it is non-minimum phase and can be made passive only if the system is stable; conversely, when a system lacks OFP it is unstable and can be made passive if the system has low relative degree and is minimum phase. Even though the two passivity indexes are independent of

each other, both indexes need to be determined in order to determine the passivity level of the system.

Definition 24. *[116] A system H has input feedforward passivity index ν and output feedback passivity index ρ if there exists a constant β such that the following inequality holds:*

$$\int_{t_0}^{t} [(1+\nu\rho)u^{\mathsf{T}}y - \rho y^{\mathsf{T}}y - \nu u^{\mathsf{T}}u]d\tau \geq -\beta.$$

There are two special cases of passivity that are considered. The system is input strictly passive when the system has an excess of IFP; the system is output strictly passive when the system has an excess of OFP. A system that is output strictly passive is also finite-gain L_2 stable. Applying passivity indexes to (3.7) results in:

$$\int_{t_0}^{t} u^{\mathsf{T}}yd\tau \geq V(x(t)) - V(x(t_0)) + \int_{t_0}^{t} \nu u^{\mathsf{T}}ud\tau + \int_{t_0}^{t} \rho y^{\mathsf{T}}yd\tau. \tag{3.15}$$

Definition 25. *[113] A system is input strictly passive if (3.15) holds for $\nu > 0$ and $\rho \geq 0$.*

Definition 26. *[113] A system is output strictly passive if (3.15) holds for $\nu \geq 0$ and $\rho > 0$.*

3.2.5 Passivity-Based Control Design Using Port-Hamiltonian Systems

A controlled system consists of a plant (typically passive) and a control algorithm interacting with the plant through sensors and actuators [117]. Traditional control design require that the system being controlled is represented in a format specific to the control design technique [118]. For example, the root-locus control design method requires that system be represented as a pole-zero plot and the control gains affect the placement of the poles and zeros. However, because the system is abstracted into a mathematical format, specific physical attributes of the system are lost. In order to preserve the physical representation of the system, the design and modeling of controllers with an emphasis on the physical model was developed [119]. By extension, control design using PHS is also

based on the idea of emphasizing the energy and power of the system through these power ports [120]. Consider the PHS in (3.5), ignoring the interaction port (d,z):

$$\dot{x} = [J(x) - R(x)]\frac{\partial H}{\partial x} + G(x)u.$$

The derivative of the Hamiltonian function is defined by [2]:

$$\dot{H}(x) = (\partial H(x))^{\mathsf{T}}[J(x) - R(x)]\partial H(x) + (\partial H(x))^{\mathsf{T}}G(x)u. \tag{3.16}$$

Through substitution and other mathematical manipulations, (3.16) simplifies down to:

$$\dot{H}(x) = -(\partial H(x))^{\mathsf{T}}R(x)\partial H(x) + y^{\mathsf{T}}u, \tag{3.17}$$

where $\dot{H}(x)$ denotes the stored power in the system, $(\partial H(x))^{\mathsf{T}}R(x)$ denotes the dissipated power, and $y^{\mathsf{T}}u$ denotes the supplied power. So long as the dissipation of the system is non-negative ($R(x) \geq 0$), the unforced system ($u = 0$) is asymptotically stable ($\dot{H}(x) \leq 0$), which indicates that the Hamiltonian function is a Lyapunov function [121]. Energy-balancing control using PHS extends (3.17) to the energy-balance equation of the system [2]:

$$H(x(t)) - H(x(0)) = -d(t) + \int_0^t y^{\mathsf{T}}(\tau)u(\tau)d\tau, \tag{3.18}$$

where $H(x(t)) - H(x(0))$ denotes the stored energy, $d(t)$ denotes the dissipated energy, and $\int_0^t y^{\mathsf{T}}(\tau)u(\tau)d\tau$ denotes the supplied energy. The control objective is to operate the system at a nonzero equilibrium point [122], $x*$, given a control input $u = \beta(x) + v$.

$$H_a(x(t)) - H_a(x(0)) = -\int_0^t y^{\mathsf{T}}(\tau)\beta(x(\tau))d\tau, \tag{3.19}$$

where H_a is the energy supplied by the controller. Using both (3.18) and (3.19) we can establish a new energy-balance equation for the closed-loop system,

$$H_d(x(t)) - H_d(x(0)) = -d_d(t) + \int_0^t y^{\mathsf{T}}(\tau)v(\tau)d\tau, \tag{3.20}$$

which leads to the following proposition:

Proposition 1. *[2] The closed loop system is a PHS of the form* $\dot{x} = [J(x) - R(x)]\frac{\partial H_d}{\partial x}(x)$*, with* H_d *given in (3.20) and a stable equilibrium point* $x*$*, if there exists a function* $\beta(x)$ *and a vector function* $K(x) = \frac{\partial H_a}{\partial x}(x)$ *satisfying* $[J(x) - R(x)]K(x) = G(x)\beta(x)$ *such that the following three conditions are true:*

1. $\frac{\partial K}{\partial x} = \frac{\partial K}{\partial^{\mathsf{T}} x}$
2. $K(x*) = -\frac{\partial H}{\partial x}(x*)$
3. $\frac{\partial K}{\partial x}(x*) > -\frac{\partial^2 H}{\partial x^2}(x*)$

Unfortunately, energy-balancing control is hampered the dissipation obstacle [2]. One of the requirements for Proposition 1 is that the natural dissipation of the PHS must satisfies $R(x)K(x) = 0$, and since $R(x)$ is commonly expressed as a diagonal matrix the proposition requires that there be no dissipation in the system [123]. In order to overcome this dissipation obstacle the *control by interconnection* method was developed [124].

CPSs are typically controlled by large-scale networked control systems, in which the overall behavior of the system depends on the interactions between the control software, communication network, and the physical dynamics [125]. When integrating the layers of CPS, implementation effects such as network delay and packet losses prohibits the use of traditional component-based design methods [126] [127]. Passivity-based control techniques can design CPS that are robust to these network uncertainties [128] [129].

In control by interconnection, control systems are designed as PHS, which interconnect to the plant PHS at designated control ports. This method treats the plant PHS as a black box in which the controller only has information regarding the effort and flow variables of the control port [130]. Given a plant PHS represented as:

$$P: \begin{cases} \dot{x} = [J(x) - R(x)]\frac{\partial H}{\partial x} + G(x)u \\ y = G^{\mathsf{T}}(x)\frac{\partial H}{\partial x} \end{cases}$$

and a controller PHS represented as:

$$C: \begin{cases} \dot{w} = [J_C(w) - R_C(w)]\frac{\partial H_C}{\partial w} + G_C(w)u_C \\ y_C = G_C^{\mathsf{T}}(w)\frac{\partial H_C}{\partial w} \end{cases}$$

A standard feedback interconnection between the plant and controller is shown in (3.21) where e and e_C are the external signals:

$$\begin{cases} u = -y_C + e \\ u_c = y + e_C \end{cases} \tag{3.21}$$

The closed loop system involving the plant and the controller PHS models, is also a PHS and has the following form:

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{w} \end{bmatrix} &= \left(\begin{bmatrix} J(x) & -G(x)G_C^{\mathsf{T}}(w) \\ G_C(w)G^{\mathsf{T}}(x) & J_C(w) \end{bmatrix} - \begin{bmatrix} R(x) & 0 \\ 0 & R_C(w) \end{bmatrix} \right) \begin{bmatrix} \frac{\partial H}{\partial x} \\ \frac{\partial H_C}{\partial w} \end{bmatrix} \\ &\quad + \begin{bmatrix} G(x) & 0 \\ 0 & G_C(w) \end{bmatrix} \begin{bmatrix} e \\ e_C \end{bmatrix} \\ \begin{bmatrix} y \\ y_C \end{bmatrix} &= \begin{bmatrix} G^{\mathsf{T}}(x) & 0 \\ 0 & G_C^{\mathsf{T}}w \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial x} \\ \frac{\partial H_C}{\partial w} \end{bmatrix} \end{aligned} \tag{3.22}$$

The energy of the closed-loop system satisfies $H_{cl}(x) = H(x) + H_C(x)$. From a PHS point of view, there are three characterization of the behavior of systems based on energy-balance of (3.18): interconnection structure, dissipation, and energy storage (energy shaping) [131]. This results in passivity-based control manipulating these three degrees of freedom to influence the behavior of the closed-loop system. Additionally, port-based control

methods can also be applied to PHS descriptions of hybrid systems [132].

The canonical coordinate transform method is used extensively in classical mechanics for analyzing the dynamical equations of physical systems [133]. These transformations preserve the Hamiltonian structure of the system and preserves important system properties such as losslessness and passivity. Consider a generic PHS written in the form of (3.5). We then consider a time-invariant coordinate transformation defined by the expression $\bar{x} = \Phi(x)$ and apply it to (3.5). The dynamical equations transform into the following form

$$\begin{aligned}
\dot{\bar{x}} &= \frac{\partial \Phi}{\partial x}^{\mathsf{T}} \dot{x} \\
&= \frac{\partial \Phi}{\partial x}^{\mathsf{T}} [J(x) - R(x)] \frac{\partial H}{\partial x} + \frac{\partial \Phi}{\partial x}^{\mathsf{T}} G(x) u \\
&= \frac{\partial \Phi}{\partial x}^{\mathsf{T}} [J(x) - R(x)] \frac{\partial \Phi}{\partial x} \frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}} + \frac{\partial \Phi}{\partial x}^{\mathsf{T}} G(x) u
\end{aligned}$$

and the output equation transforms into the following form

$$y = G^{\mathsf{T}}(x) \frac{\partial \Phi}{\partial x} \frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}}.$$

The new Hamiltonian function becomes $H(\Phi^{-1}(\bar{x}))$. The matrices $\frac{\partial \Phi}{\partial x}^{\mathsf{T}} J(x) \frac{\partial \Phi}{\partial x}$ and $\frac{\partial \Phi}{\partial x}^{\mathsf{T}} R(x) \frac{\partial \Phi}{\partial x}$ are skew-symmetric and positive symmetric, respectively, which means that the coordinate transformed system is also a PHS [134]. Generally, the coordinate transform is used whenever two PHS are composed [135].

Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC) is a control design technique which regulates the behavior of the closed loop system by assigning a desired PHS; through solving a matching equation involving interconnection assignment, damping assignment, and energy shaping, IDA-PBC achieves the desired control objective while maintaining passivity of the overall system [136]. The IDA-PBC method can be expressed in the Dirac structure framework. Consider a PHS with Hamiltonian H (corresponding to storage port S), resistive port R, and control port C, its kernel representation can be expressed as:

$$D : \{F_S f_S + E_S e_S + F_R f_R + E_R e_R + F_C f_C + E_C e_C\}. \tag{3.23}$$

The power balancing equation can then be derived from (3.23):

$$-F_S \dot{x}(t) + E_S \frac{\partial H}{\partial x} + F_R f_R(t) + E_R e_R(t) + F_C f_C(t) + E_C e_C(t). \tag{3.24}$$

Given a control input $u = \beta(x)$, the control objective is such that the desired closed-loop system's implicit PHS form is given by:

$$-F_S \dot{x}(t) + E_{S_d} \frac{\partial H_d}{\partial x} + F_{R_d} f_{R_d}(t) + E_{R_d} e_{R_d}(t). \tag{3.25}$$

Combining (3.24) and (3.25), eliminating the $\dot{x}(t)$ term, and simplifying the result yields the following matching equation:

$$E_S \frac{\partial H}{\partial x}(x(t)) - F_R f_R = E_{S_d} \frac{\partial H_d}{\partial x}(x(t)) - F_{R_d} f_{R_d}.$$

The three degrees of freedom are indicated by $E_S \to E_{S_d}$ (interconnection assignment), $F_R f_R \to F_{R_d} f_{R_d}$ (damping assignment), and $\frac{\partial H}{\partial x} \to \frac{\partial H_d}{\partial x}$ (energy shaping). There are three methods of solving the matching equation through the three degrees of freedom: Non-Parameterized IDA, Algebraic IDA, and Interlaced Algebraic-Parameterized IDA [2].

3.3 Barrier Certificates for Safety Analysis of CPS

Definition 27. *[68] Given a system $\dot{x} = f(x)$ with states x taking values from the set $X \subset \mathbb{R}$, a set of initial states $X_0 \subset X$ and a set of unsafe states $X_u \subset X$. The system is safe if there exists a continuously differentiable function B such that:*

1. $B(x) \leq 0, \forall x \in X_0$,

2. $B(x) > 0, \forall x \in X_u$,

3. $\frac{\partial B}{\partial x}(x)f(x) \leq 0, \forall x \in X$.

There is no trajectory $x(t)$ of the system such that $x(0) \in X_0$, $x(T) \in X_u$ for some $T \geq 0$, and $x(t) \in X$ [137]. The function $B(x)$ in Definition 27 is called a barrier certificate. Given that $f(x)$ is polynomial and the sets X, X_0, and X_u are semi-algebraic, the barrier certificate can be computed using a technique called sum of squares programming [138] [139].

Definition 28. *[138] A barrier certificate is a sum of squares if there exist polynomials $B_1(x)$, $B_2(x)$, ..., $B_m(x)$ such that $B(x) = \sum_{i=1}^{m} f_i^2(x)$. Using semidefinite programming, the sum of squares decomposition for $B(x)$ can be calculated.*

Extending from Definition 27, safety of hybrid systems can be formulated with barrier certificates as follows:

Definition 29. *[137] Given a hybrid system $H = (X, L, X_0, I, F, D)$ and an unsafe set X_u. Suppose there exists a collection of barrier certificates $B_l(x)$ for $l \in L$, each of which is differentiable with respect to its argument and satisfies:*

1. $B_l(x) \leq 0, \forall x \in \text{initial } l$,
2. $B_l(x) > 0, \forall x \in \text{unsafe } l$,
3. $\frac{\partial B_l}{\partial x}(x)f_l(x) \leq 0, \forall x \in I(l)$,
4. $B_l(x) \leq 0, \forall x \in \text{Guard}(s, s')$.

The safety of the hybrid system H is guaranteed as long as Definition 29 is valid [140]. These structure of these definitions is used in the safety analysis method presented in Chapter 6.

Chapter 4

Component-Based Modeling and Simulation Using Port-Hamiltonian Systems

The use of models as an initial step in the development of complex systems is prevalent in all fields of science and engineering. A model is an abstraction of occurrences in the physical world, and represents it through a set of concepts. In engineering, models provide a way of representing systems using a specific set of mathematical constructs. The difference in abstraction between physical dynamics and computational elements poses a significant challenge for the modeling of CPS [85]. The abstractions regarding computational and networking elements will need to be redefined in a way that merge the layers of CPS [19]. The main contribution of this chapter is a domain-specific modeling language (DSML) called Port-Hamiltonian Systems Modeling Language (PHSML); it is based on port-Hamiltonian systems (PHS) and uses the principles of Model-Integrated Computing [141]. Both acausal and causal simulation using Modelica and Simulink, respectively, is also discussed. This work has been presented in [142] and [143].

4.1 Port-Hamiltonian System Modeling Language

In order to accurately model the interactions of different components in the systems, interfaces need to be well formulated [18]. The concept of a port is a result of interaction between different elements and sub-models within a model. Port-based modeling has garnered attention over the years since it captures the underlying physics of the system. For physical systems, this interaction is depicted as an exchange of energy through the power bond. A power bond represents a bilateral relationship between the power ports of two interacting elements or sub-models. Each bond contains two power-conjugate variables (called effort and flow) which relate to power by a product relationship shown in Definition 1.

Port-based modeling of lump-parameter complex physical systems leads to a framework of generalized bond graphs, which are different from traditional bond graphs in two ways: use of symplectic gyrators and absence of causality. Conventional physical domains usually have two types of storage elements associated with two types of energy (for example, capacitors store electrical energy, while inductors store magnetic energy) [19]. In generalized bond graphs, symplectic gyrators are attached to I-storage elements, which inverts the roles of effort and flow and allow I-storage elements to be treated like C-storage elements [22]. The purpose of symplectic gyrators is to decompose the two types of energy of physical domains to a unified type of energy. The lack of causality within generalized bond graphs allow the preservation of the underlying physics of the model. In a generalized bond graph, storage, resistive, and source ports connect through a generalized junction structure, which includes junctions, transformers, and gyrators [27].

A generalized bond graph can be summarized by it energy storing elements, resistive elements, and power-conserving elements connected by bonds. Collectively, the power-conserving elements form the Dirac structure, and connects all other non-power-conserving elements in the model. This extends to the formulation of PHS, represented by figure 2.3, where a set of ports (control, interaction, resistive, and storage) are interconnected through a Dirac structure [2]. A PHS with a Hamiltonian function H, energy storage ports S, resistive ports R, control ports C, interconnection ports I, and a Dirac structure D can be written in a formal model in (3.3),

Energy-storage and resistive elements make up the internal ports of a PHS; control and interaction elements make up the external ports. The internal ports and the interaction ports make up the physical portion of CPS where the physical dynamics are captured through (3.4); the control ports make up the cyber portion of CPS and describe the interaction between a PHS and a controller; control software is developed to affect one of the control port's power conjugate variables in order to perform an action. The interaction through control ports forms the fundamental concept of compositionality; a PHS controller connects

with a PHS plant, resulting in another PHS. The Hamiltonian function of the composed system is the sum of the Hamiltonian functions of its components; the Dirac structure of the composed system is determined by the Dirac structure of its components as well.

4.1.1 Syntax and Semantics

The PHSML is developed using Generic Modeling Environment (GME), based on the model integrated computing tool suite developed at Institute for Software and Integrated Systems at Vanderbilt University [144]. GME provides a meta-modeling environment to create DSMLs, and allows for the construction of domain models through a collection of objects and connectors defined for a specific domain. GME ensures that the domain specific instance models comply with the specified meta-model. Within the meta-model there are different concepts to represent an object. An object can either be an atom, which is the basic indivisible object that cannot contain other objects; a connection, which can connect two objects; or a model, which can contain other objects such as atoms, connections or models. Hierarchical composition is supported through model elements. Within model elements objects can be connected to each other based on the rules defined in the meta-model. Each connection has a source object (denoted by *src*) and a destination object (denoted by *dst*).

PHSML encodes *all* modeling and connection rules specific for PHS. The model is developed in GME as a PHSML model; a model translator, i.e. interpreter, processes the structural information of the PHSML model and creates a PHS model characterized by Dirac structures and constituent equations of ports. The top level hierarchy of PHSML is called component assembly that consists of components, which serve as the objects, and bonds, which serve as the connections. A component is defined as an object consisting of a Dirac structure, a set of internal ports, and a set of external ports. PHSML defines two types of bonds (i.e. connections): internal bonds connect different sub-objects within a component, while external bonds connect external ports of different components. The PHSML model provides a set of structural data for each object, which are enumerated as

follows:

1. ID: The unique tag generated for each object in the PHSML model.
2. Name: The name of the object as given in the PHSML model.
3. Type: The type of the object.
4. Equation: The constituent equation of the object, defined by the user in the PHSML model.
5. Ratio: The constant ratio value of the object.
6. Bond: Lists the unique ID's of all bonds which are attached to the component.

4.1.2 Bonds and Components

The meta-model of PHSML is summarized in Figure 4.2. Bonds form the connection semantics of the PHSML. Each bond in PHSML, shown in Figure 4.1, contains a pair of power-conjugate variables, effort and flow, whose inner product yields power. Bonds are used to connect components and objects together. Each component is instantiated as a model, and contains a series of sub-components given by *DiracStructure*, *InternalPort*, and *ExternalPort*.

4.1.2.1 Dirac Structure

The most important modeling element to each component is the Dirac structure, which describes the power-conserving interconnection structure of each component. The Dirac structure is a collection of four different types of modeling elements: *Gyrator*, *Transformer*, *Zero Junction*, or *One Junction*. *Gyrators* and *Transformers* contain a *Ratio* attribute which denotes the transformer or gyrator ratio. *Zero Junction*s and *One Junction*s denote either a parallel or series interconnection between other elements.

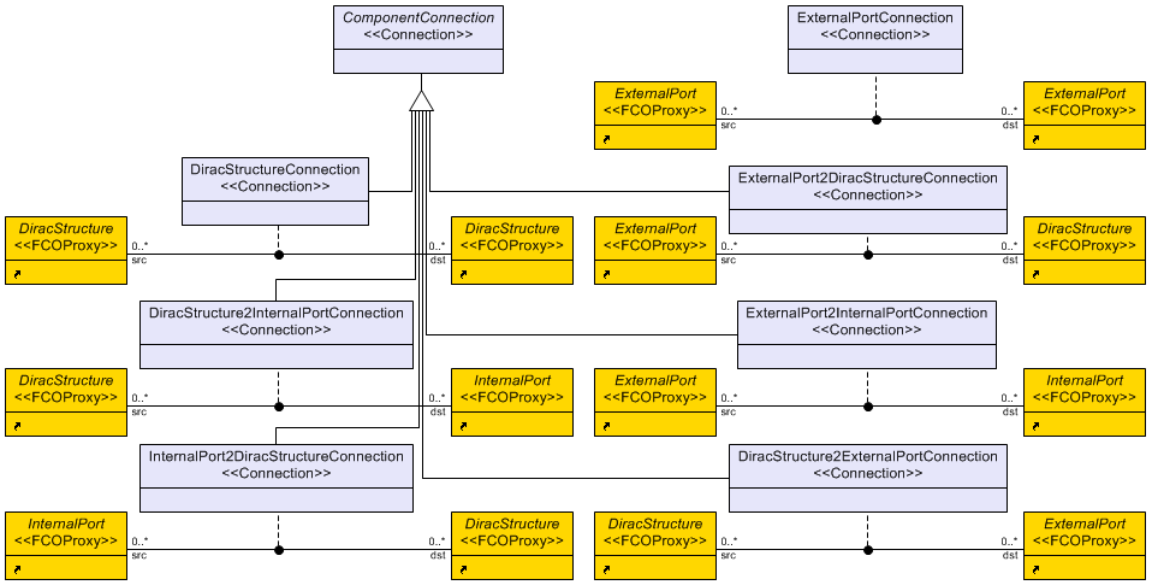

Figure 4.1: Meta-model of the connections in PHSML

4.1.2.2 Internal Ports

The two internal ports (which are also denoted as atoms) both have dynamics which relates their power-conjugate variables through constituent equations. The equation attribute allows the user to specify the constituent equations of the port with respect to the power conjugate variables, effort and flow. A linear resistive port atom relates the power conjugate variables through an equation in the form of $e = Rf$, where R is a constant; a linear storage port atom relates the power conjugate variables through an equation in the form of $\text{der}(e) = Sf$, where S is a constant. Nonlinear resistive and storage ports do not contain a formulaic expression, but still relate the effort and flow variables through a function.

4.1.2.3 External Ports

The two types of external ports are fundamentally similar. Control ports also contain the equation attribute, because we classify PHS elements such as sources and sinks in control ports. Through the *Equation* attribute, the constituent equations of sources and sinks can be specified. Interaction ports allow for different components to connect to each other and

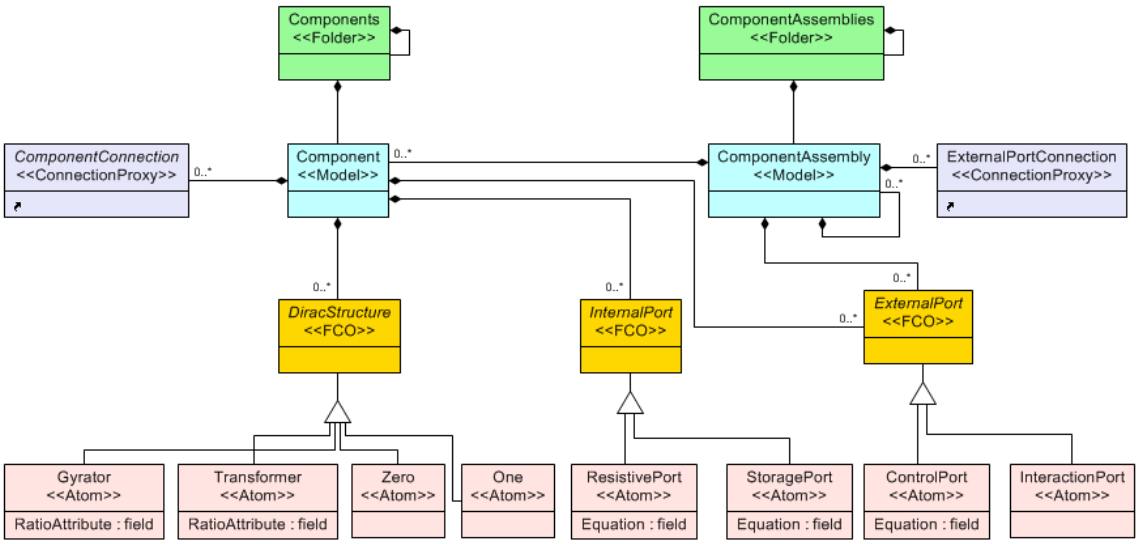

Figure 4.2: The meta-model of PHSML

contain no additional attributes.

4.1.3 Constraints

The most important goal of PHSML is to ensure that models are valid PHS. Constraints are implemented in the PHSML meta-model to ensure that certain interconnections between PHSML modeling elements, which will violate the underlying property of PHS, are not allowed. There are two types of constraints: (a) directional connections and connections between modeling types are implemented as UML diagram shown in Figure 4.1 and (b) a model transformation software component defines a set of rules, which require a specific number of connections for each object. A detailed listing of allowable connections between elements is detailed as follows:

1. Junctions (includes both one and zero junctions): There are no constraints to the number of bonds which connect to these objects. However, at least one bond must have this junction object as destination and at least one bond must have this junction object as source. Because of power conservation property of junctions, power has to come into the junction and also has to leave.

2. Transformers and Gyrators: There must be exactly two bonds connecting to this object. One bond must have this object as destination and the other bond must have this object as source.

3. Internal Ports (includes both resistive and storage ports): There must be exactly one bond connection to this object. The bond can have this object as source or destination. Power is generally dissipated or stored, which results in power generally flowing into a resistive or storage port. However, cases can arise in which the storage port or resistive port can deliver power. A discharging capacitor is a good example of a storage port providing power; an effort source that is dependent on flow is an example of a resistive port providing power. An Internal Port cannot be connected to another Internal Port (Figure 4.1).

4. External Ports (includes both control and interaction ports): There must be one bond connection to this object. The bond can have this object as either destination or source. These ports denote boundary conditions either between components or with the environment. External Ports can be connected to Internal Ports, but Internal Ports cannot be connected to External Ports directly (Figure 4.1).

4.1.4 Interpreter

The interpreter of PHSML uses MATLAB functions to transform a PHS to an acausal computational model by deriving the kernel representation of the Dirac structure. The GME model provides a set of structural data for each object, which are enumerated as follows:

1. ID: The unique tag generated for each object in the model.

2. Name: The name of the object as given inside GME.

3. Type: The type of the object.

4. Equation: The constituent equation of the object, defined by the user inside GME.

5. Ratio: Only applicable for either transformers or gyrators.

6. Bond: Lists the unique ID's of all bonds which are attached to the component.

Derivation of the Dirac structure focuses on the power conserving elements in the PHS, which consists of objects with "Type" of Zero, One, Transformer, or Gyrator. The interpreter sequentially combines all power-conserving elements to generate its kernel representation equations. The main idea in the method is recognizing that there is a pattern to the Dirac structure of every power conserving element. A transformer or a gyrator has only one possible kernel representation due to its one bond in and one bond out nature.

$$\begin{bmatrix} -1 & r \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e_i \\ e_o \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ -1 & \frac{1}{r} \end{bmatrix} \begin{bmatrix} f_i \\ f_o \end{bmatrix} = 0, \tag{4.1}$$

$$\begin{bmatrix} -1 & 0 \\ 0 & \frac{1}{k} \end{bmatrix} \begin{bmatrix} e_i \\ e_o \end{bmatrix} + \begin{bmatrix} 0 & k \\ -1 & 0 \end{bmatrix} \begin{bmatrix} f_i \\ f_o \end{bmatrix} = 0. \tag{4.2}$$

The kernel representations of a transformer and a gyrator is described by (4.1) and (4.2), respectively, where r is the transformer turning ratio and k is the gyrator ratio (the value which depends on the structural data "Ratio"). A zero or one junction, on the other hand, can have an infinite number of kernel representations due to its MIMO nature (multiple bonds in, multiple bonds out). However, there is a pattern to the matrices; for a zero and one junction with n bonds, their kernel representations are shown in (4.3) and (4.4):

$$\begin{bmatrix} 1 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & \ddots & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_{n-1} \\ e_n \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} = 0, \tag{4.3}$$

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_{n-1} \\ e_n \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & \cdots & \cdots & 0 \\ 1 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & 0 & \ddots & \cdots & \vdots \\ \vdots & \vdots & \vdots & \cdots & 1 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_{n-1} \\ f_n \end{bmatrix} = 0. \qquad (4.4)$$

The algorithm of the interpreter constructs the kernel representation of every power-conserving element inside the component using (4.1) − (4.4). Then it combines all individual kernel representations together using equation substitution methods. A summary of the algorithm is listed in Algorithm 1.

Algorithm 1 GBG2DS

-**for all** Components
- **for all** Power-conserving elements
- Find E and F based on pattern;

- **while** Interconnections left
- Substitute Dirac structure equations to form combined Dirac structure;

For all non-power-conserving elements in the PHS, constituent equations are denoted by "Equation". Compiling these equations with the Dirac structure derived by Algorithm 1 provides an acausal computational model of the PHS.

4.1.5 Challenges

The PHSML framework addresses four important challenges: heterogeneous domains, composition, hybrid dynamics, and nonlinearities.

Heterogeneous Domains: The most important characteristic of PHS is that they are domain neutral, with different domains interacting with each other through gyrators. A symplectic gyrator is a special kind of gyrator whose gyrator ratio is fixed at 1; it is used to invert the roles of the effort and the flow variables. In the PHS framework, symplectic gy-

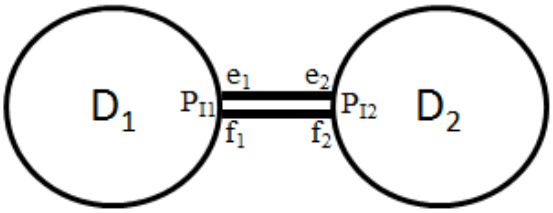

Figure 4.3: Interaction between two Dirac structures

rators are used to unify C and I storage elements into a single type of storage element [27]. The constituent equations of a gyrator indicates that it is power-conserving, and in the modeling process we include gyrators in the Dirac structure. Using PHS, we can model both the computational and physical domains. The two domains will interact through interaction ports, where the effort and flow variables act as sensors and actuators. The interfacing between the computational and physical domains allows for the modeling of complex CPS.

Composition: Composition of PHS is derived from the composition of Dirac structures, where it has been proven that the composition of two or more Dirac structures yield another Dirac structure (see Section 2.1.2). Components are composed together through a mutual interaction port; in each interaction four effort and flow variables are involved. From Figure 4.3 the two Dirac structures, D_1 and D_2, interact through the interaction ports P_{I1} and P_{I2}. The four effort and flow variables interact through the following two equations $e_1 = e_2$ and $f_1 = f_2$.

Hybrid Dynamics: The modeling of hybrid dynamics is influenced by hybrid bond graphs, which are an extension of bond graphs but with an emphasis on switching behavior [47]. Hybrid Dynamics are implemented using switches and model switching and discontinuous behavior. A switch is defined as a control port with two modes, and each mode either forces the switch to act as a zero source of effort or a zero source of flow. Using these switches do not violate the PHS formulation because switches are power neutral; at least one of the power conjugate variables has to be zero at all times, which results in zero power. Switches are implemented through junctions; which yields two types of

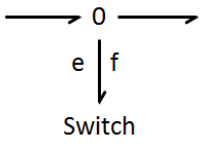

Figure 4.4: Zero junction switch

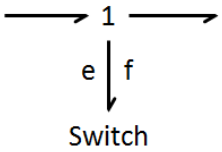

Figure 4.5: One junction switch

switches, zero junction switch (Figure 4.4) and one junction switch (Figure 4.5). For a zero junction switch, a zero effort indicates that the switch is off, as all efforts attached to the zero junction become zero. For a one junction switch, a zero flow indicates that the switch is off, as all flows attached to the one junction become zero.

Even though these switches do not affect the energy of the system, care must be exercised when implementing them. A switch is an alternating source, which means that the causality of that bond will change since it imposes either a zero flow or a zero effort onto the system. When designing hybrid systems using these switches careful placement is required to avoid negatively affecting the causality of the system, which will create problems in simulation by generating implicit equations.

Nonlinearities: PHSML is designed to handle nonlinear behavior. Nonlinearities occur in three forms: dissipative elements with nonlinear behavior, storage elements with nonlinear behavior, and transformation elements with nonlinear behavior. Nonlinear dissipative elements are treated similarly as linear ones, except that the constituent equations are no longer invertible. In the nonlinear ports the relationship between effort and flow is constrained as a function, which results in a fixed causality. Causality affects storage ports to a lesser degree because they already have preferred causality.

1. Generalized nonlinear resistor:

$$e = \Phi_R(f).$$

2. Generalized nonlinear storage, where q is the generalized displacement:

$$\begin{aligned} e &= \Phi_C(q), \\ q &= \int_{t_0}^{t} f d\tau. \end{aligned}$$

3. Generalized transformer, where k is a nonlinear function:

$$\begin{aligned} e_1 &= k e_2, \\ f_2 &= k f_1. \end{aligned}$$

4. Generalized gyrator, where l is a nonlinear function:

$$\begin{aligned} e_1 &= l f_2, \\ e_2 &= l f_1. \end{aligned}$$

4.2 Case Study: Engine Dynamometer Model

In this section we present an example which illustrates PHSML through a case study of an engine-dynamometer (dyno) model. The engine-dynamometer (dyno) model is inspired by PHS models of physical systems presented in [145]. The model consists of three interconnected components shown in Figure 4.6: engine, starter, and dynamometer. The engine-dynamometer model is a physical system described by three interacting PHS components. Control ports on each component describe the cyber portions which allow controllers to interface with the model.

The engine component is modeled by a one junction and a transformer connecting a throttle control port, interaction ports to the starter and dyno components, and a storage port which is the inertia of the engine. The engine either delivers a positive amount of

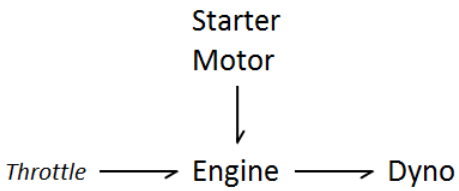

Figure 4.6: Engine-dynamometer high level model

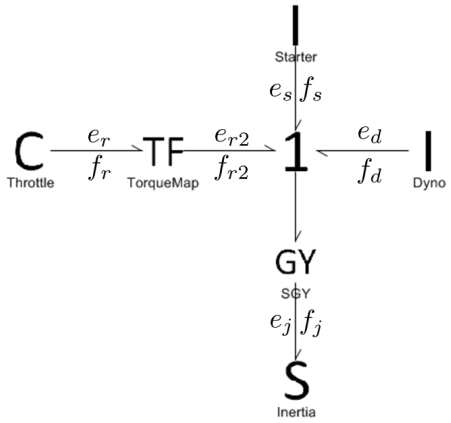

Figure 4.7: PHS model of the engine

calculated torque to the inertia load (when throttle is engaged), or it delivers a negative amount of calculated torque (negative throttle indicates braking). The throttle controlled torque map is described as a nonlinear transformer which links the throttle control port with the one junction. A PHS model of the engine built using PHSML is presented in Figure 4.7; variable names are labeled for bonds in order to describe the constituent equations. e_r and f_r are the power conjugate variables of the throttle control port; e_s and f_s are the power conjugate variables of the interaction port to the starter motor; e_j and f_j are the power conjugate variables of the engine inertia; e_d and f_d are the power conjugate variables of the interaction port to the dynamometer.

The nonlinearity of the engine component is caused by the nonlinear transformer which models the throttle control port. The variables e_r, f_r, e_{r2}, and f_{r2} are related by an interpolation mapping; the interpolation function uses a series of if-then-else statements, based off of real torque map data from a Cummins Inc. engine [146]. The transformer ratio, T, is

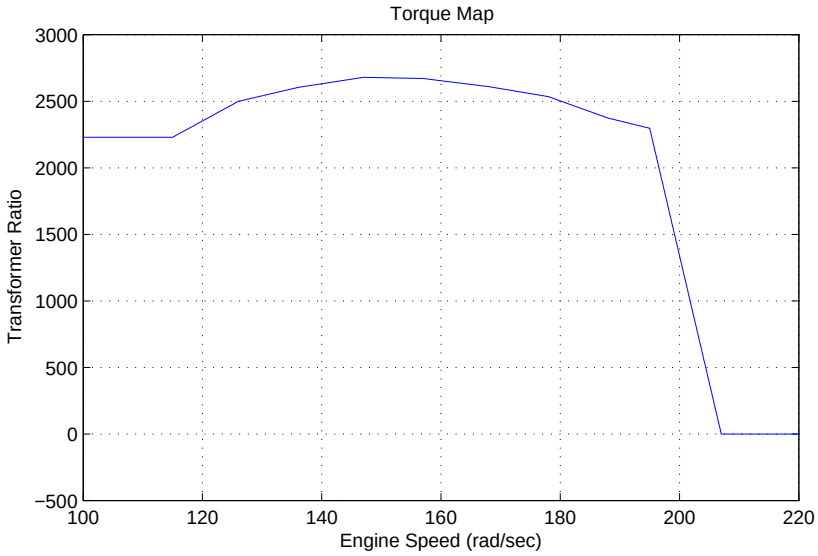

Figure 4.8: Torque map of the nonlinear transformer

a function of the engine speed; the mapping of f_{r2} (engine speed) to T is illustrated by the torque map in Figure 4.8:

$$\begin{aligned} e_{r2} &= Te_r \\ f_r &= Tf_{r2} \end{aligned}$$

The inertia from the engine has the following constituent equation:

$$\frac{d}{dt}e_j = \frac{f_j}{J}$$

where J is the moment of inertia value.

Hybrid dynamics is central to the operation of the starter motor. Hybrid dynamics are implemented as switch networks. Given that one of the constituent variables of a switch is always equal to zero, a switch will never contribute any power to the system. A PHS model of the starter motor is presented in figure 4.9. The amount of torque that the starter motor provides is governed by a Boolean signal s. If $s = 1$ is true, the switch $SW2$ becomes active and the switch $SW1$ becomes inactive; if $s = 0$ is true, the switch $SW1$ becomes active and the switch $SW2$ becomes inactive. The two torque values are modeled as constant sources

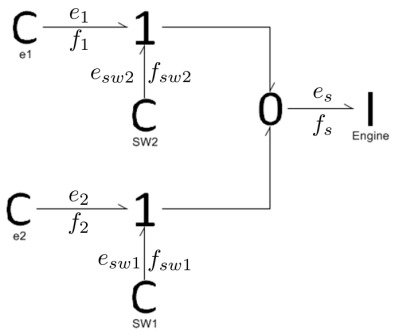

Figure 4.9: PHS model of the starter motor

of efforts e_1 and e_2. Their flow outputs are f_1 and f_2, respectively.

The hybrid constituent equation of $SW1$ is:

$$e_{sw1} = 0 \quad \text{if } s = 1$$
$$f_{sw1} = 0 \quad \text{if } s = 0.$$

The hybrid constituent equation of $SW2$ is:

$$e_{sw2} = 0 \quad \text{if } s = 0$$
$$f_{sw2} = 0 \quad \text{if } s = 1.$$

The dynamometer simulates road load to the engine and is a system without hybrid dynamics or nonlinearities. It is modeled as a proportional-integral system which takes the angular velocity of the engine as a measured signal and a step source as the set point signal. The error signal is then calculated by subtracting the set point signal and measured signal. The output of the PI controller is a simulated torque which goes back to the engine as a load. A PHS model of the dynamometer is presented in Figure 4.10. e_u and f_u are the power conjugate variables of the set point control port; e_R and f_R are the power conjugate variables of the proportional controller; e_I and f_I are the power conjugate variables of the integral controller.

The proportional component has the following constituent equation:

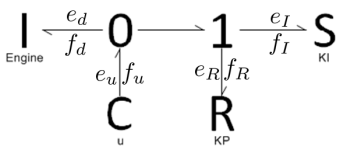

Figure 4.10: PHS model of the dynamometer

$$e_R = K_R f_R,$$

where K_R is the proportional gain value. The integral component has the following constituent equation:

$$e_I = K_I \int_{t_0}^{t} f_I d\tau,$$

where K_I is the integrator gain value.

4.3 Component-Based Simulation using Port-Hamiltonian Systems

Component-based simulation uses either acausal computational models or causal computational models. Acausal computational models are described by a set of differential algebraic equations (DAE) which requires a differential algebraic system solver (DASSL); the proposed work of the previous chapter intends to generate acausal computational models which simulates using a Modelica simulation environment. Causal computational models are described by a set of ordinary differential equations (ODE) which can be solved using an ODE solver. In this chapter we discuss the generation of causal computational models through a method called Compositional Causality Analysis (CCA), which are then simulated using Simulink.

The model interpreter generates the Dirac structure equations and the constituent equations of the power ports from a PHSML model. In particular, the Dirac structure is generated as a kernel representation, defined by the E and F matrices. The kernel representation

is an acausal model, and one of the advantages of acausal models is independent analysis; the kernel representation of each component can be derived without knowledge of its neighboring components or the global model. Equations defined inside the Modelica language have no inherent causality, and can model the kernel representation of Dirac structures. The transition from PHS to Modelica is intuitive because both frameworks define interactions in terms of power ports.

4.3.1 Acausal Simulation of Port-Hamiltonian Systems

Derivation of the Dirac structure focuses on the power conserving elements in the PHS, which consists of objects with *Type* of Zero, One, Transformer, or Gyrator. The method sequentially combines all power-conserving elements to generate its kernel representation equations by recognizing that there is a pattern to the Dirac structure of every power conserving element. A transformer or a gyrator has only one possible kernel representation due to its SISO (one bond in, one bond out) nature.

The kernel representations of a transformer and a gyrator is described by (4.1) and (4.2), respectively, where r is the transformer turning ratio and k is the gyrator ratio. A zero or one junction, on the other hand, can have an infinite number of kernel representations due to its MIMO nature (multiple bonds in, multiple bonds out). However, there is a pattern to the matrices; for a zero and one junction with n bonds, their kernel representations are shown in (4.3) and (4.4):

4.3.1.1 Implementation of Acausal Simulation in Modelica

The model interpreter reads structural data from a PHSML model. A model transformation method is then applied to the data which transforms a PHS into an acausal computational model (in the form of a MATLAB m-script). We developed an Modelica algorithm based on Algorithm 1 which constructs the kernel representation of every power-conserving element inside the component using $(4.1) - (4.4)$. The algorithm then combines all indi-

vidual kernel representations together using equation substitution methods.

For all non-power-conserving elements in the PHS, constituent equations are denoted by the "Equation" parameter of the data structure. Compiling these equations with the Dirac structure derived by Algorithm 1 provides an acausal computational model. The hierarchy section of the Modelica code contains system parameters and power conjugate variables. The equation section of the Modelica code contains the Dirac structure, written as a matrix equation, and the set of constituent equations for each power port. The process of writing the Modelica code is detailed by Algorithm 2.

Algorithm 2 Generalized Bond Graphs to Modelica

- Model Hierarchy Section;
- Write Dirac structure matrices;
-**for all** Power ports on the Dirac structure
- Write indexed effort and flow variables;
- Model Equation Section;
-**for all** Power ports on the Dirac structure
- Write Equation attribute;
- end model;

4.3.1.2 Acausal Simulation of the Engine-Dynamometer

Applying Algorithm 1 to the engine-dynamometer example, we obtain the kernel representation of the Dirac structure of the engine (4.5), starter motor (4.6), and dynamometer (4.7):

$$\begin{bmatrix} -1 & -\frac{1}{T} & 0 & -\frac{1}{T} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_r \\ e_s \\ e_j \\ e_d \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{1}{T} & 0 \\ 1 & -T & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -T \end{bmatrix} \begin{bmatrix} f_r \\ f_s \\ f_j \\ f_d \end{bmatrix} = 0, \tag{4.5}$$

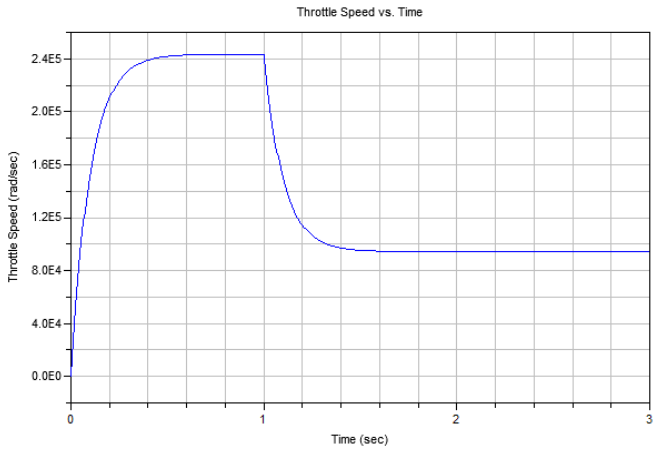

Figure 4.11: Modelica simulation results

$$\begin{bmatrix} -1 & -1 & -1 & -1 & 0 \\ -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_{sw2} \\ e_2 \\ e_{sw1} \\ e_s \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} f_1 \\ f_{sw2} \\ f_2 \\ f_{sw1} \\ f_s \end{bmatrix} = 0, \tag{4.6}$$

$$\begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_u \\ e_d \\ e_R \\ e_I \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_u \\ f_d \\ f_R \\ f_I \end{bmatrix} = 0. \tag{4.7}$$

The .mo file that is generated from Algorithm 2 can be used to simulate the PHS using a Modelica simulation environment. We simulate the engine-dyno model with parameters of $J = 20$, $K_R = 200$, $K_I = 100$. Using control inputs of $f_u = 20$, $e_1 = 600$, $e_2 = 0$, and e_r (8 for the first second and 2 for the rest), the simulation results of a system output f_r plotted with 3 seconds is shown in Figure 4.11.

4.3.2 Causal Simulation of Port-Hamiltonian Systems

Simulation of causal models necessitates the assignment of causality to the acausal models. Causality assignment on the PHS requires causal strokes to be assigned to every bond in the model. A PHS with no causality conflicts can be transformed into state space equations and block diagrams for simulation [20]. We developed a method which exploits the Dirac structure of the PHS to generate component block diagram subsystems, which can connect together to form the full system. Using the kernel representation of the Dirac structure, a second model transformation method called Compositional Causality Analysis derives its respective hybrid-input-output representations.

We have seen that acausal simulation is generated from the kernel representations of each component, which are derived independently of each other. Causal simulation of a component, on the other hand, requires knowledge of its interacting neighbor components in order to determine the causality of each interaction port. This difference makes causal computational models of components more difficult to generate as compared to acausal computational models. However, the causal method does have one significant advantage over the acausal method: identification of causality conflicts. Any conflicts in causality will result in an under-ranked $F(x)$ matrix, which will result in the inability to derive the hybrid-input-output representation. The acausal method, on the other hand, has no way of identifying causal conflicts; kernel representations of the Dirac structure do not give any indication of causality conflicts.

In the method presented in this section, we simulate systems and components using the hybrid-input-output representation of the Dirac structure; one of the key properties of that representation is that the $J(x)$ matrix is skew symmetric [27], which means that the entire diagonal line is zeros. We also constrained the modeling to only allow integral causality for storage elements. For these reasons this simulation technique does not generate any algebraic loops.

Even though the case study applied the methods to a linear system, the methods can also

be applied to nonlinear systems. In a physical systems sometime there can be nonlinear elements for which an inversion of its constitutive equations is difficult or even impossible. In order to address this problem of nonlinearity is to define various types of resistive elements (linear vs. nonlinear) and setting causality constraints, or lack of causality constraints, appropriately.

4.3.2.1 Compositional Causality Analysis

Compositional Causality Analysis was motivated by the idea of hierarchy: a (global) PHS decomposed into interacting components, where each component is also a PHS. Causality assignment on the PHS requires causal strokes to be assigned to every bond in the model. A PHS with no causal conflicts can be transformed into state space equations and block diagrams for simulation [20]. The causality of interaction bonds can be easily determined when looking at the global model; looking at a local model or component, there is no way of determining the causality of an interaction bond; without explicit knowledge of neighboring components, its causality is indifferent.

A system can be described by its components and their interactions with each other. We exploit the Dirac structure of the PHS, in which components generate component block diagram subsystems, which can connect together to form the full system. Due to the compositional nature of the Dirac structure, composition of components can also be characterized by a Dirac structure. The PHSML interpreter provides the constituent equations and the kernel representation of the Dirac structure, which Compositional Causality Analysis uses to derive its respective hybrid-input-output representations.

The hybrid-input-output representation of a PHS, which is a causal computational model, arises from causality assignment of the kernel representation. In causal analysis, there can be no causal conflict, and the causality of every bond must be determined. We developed a method based on elementary column operations of $E(x)$ and $F(x)$ in order obtain the hybrid-input-output representation. We illustrate the elementary column operations with a

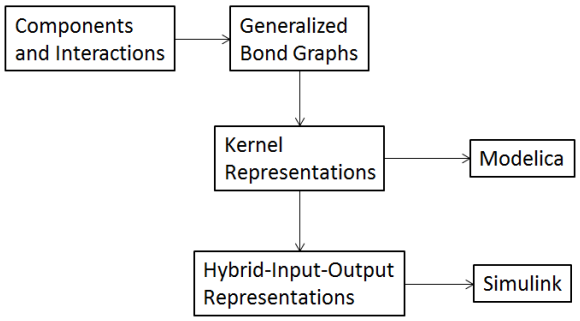

Figure 4.12: Flow chart of the simulation methodology

simple example. Consider the following kernel representation of a system with only two power ports:

$$\begin{bmatrix} E_1 & E_2 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \end{bmatrix} + \begin{bmatrix} F_1 & F_2 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = 0. \tag{4.8}$$

Mathematically, it is equivalent to the following representation:

$$\begin{bmatrix} E_1 & F_2 \end{bmatrix} \begin{bmatrix} e_1 \\ f_2 \end{bmatrix} + \begin{bmatrix} F_1 & E_2 \end{bmatrix} \begin{bmatrix} f_1 \\ e_2 \end{bmatrix} = 0. \tag{4.9}$$

The difference between (4.8) and (4.9) lies in the fact that the positions of E_2 and F_2 are interchanged. This elementary column operation interchanges a column in $E(x)$ with a corresponding column in $F(x)$. From bond graph theory we know that source elements have fixed causality; sources of effort have effort-out-flow-in causality, while sources of flow have flow-out-effort-in causality. As a result, the Dirac structure must solve the flow of a source of effort given its effort (backwards for sources of flow). In order to do so we need to apply the elementary column operation to all source of flow power ports.

Although storage elements have preferred causality, we still want to fix their causality to integral causality, due to the issues that arise with implementing differentiators. As a result of this constraint, we do not need to apply the elementary column operation to storage power ports. The difficult part of this process lies in determining column swap for linear

resistive and interaction power ports. Linear resistive power ports can be either effort-out-flow-in or effort-in-flow-out; there is no *a priori* way of determining how each resistor will behave. Similarly, there are two possible causality configurations for interaction power ports. In order to determine the causality of an interaction power port on a component, we must examine the component along with all neighboring components.

Consider the two Dirac structures in Figure 4.3, D_1 and D_2, connected through an interaction. From a structural perspective, if the interaction power port of D_1 is effort-out-flow-in, the interaction power port of D_2 has to be flow-out-effort-in, and vice versa. Therefore, we can declare that for two Dirac structures that share the same interaction, the causality of their interaction ports must be opposite.

Similar to resistive power ports, there are two possible causality configurations for interaction power ports. In order to determine the causality of an interaction power port on a component, we must examine the component along with all neighboring components. In this section we propose an interaction port propagation assignment (IPPA) algorithm which iterates through each component sequentially and determines the causality of every interaction power port.

4.3.2.2 Implementation of Causal Simulation

The implementation of causal simulation takes advantage of the indifferent causality nature of linear resistors. Because of its indifferent causality, a linear resistor can never be the source of a causal conflict, which allows for the analysis of each component with its resistive columns intentionally left out. With linear resistive power ports ignored for the time being, the only power ports with questionable causality are interaction power ports. We propose an IPPA algorithm which iterates through each component sequentially and determines the causality of every interaction power port. Our algorithm takes advantage of the indifferent causality nature of linear resistors. Because of its indifferent causality, a resistor can never be the source of a causal conflict, which allows for the analysis of

each component with its resistive columns intentionally left out. With resistive power ports ignored for the time being, the only power ports with questionable causality are interaction power ports.

Similar to assigning causal strokes to a PHS model, we select a component with the least number of interaction ports and most number of constraint ports (source and storage) with which to begin propagation of interaction port assignments. We analyze the ranks of the $F(x)$ matrix of the starting component and perform combinations of elementary column operations until it has full rank; additionally, do the same thing for the neighboring components. The process gets repeated until every interaction port is assigned. A summary of the algorithm is listed in Algorithm 3:

Algorithm 3 IPPA

-**for all** Components
- Swap all source of flow columns;
- Remove all resistive columns;

- **while** There are still unassigned interactions
- Find component with greatest interaction to constraint ratio;
- Propagate interaction assignment;

- Put back all resistive columns;

Now that all interaction ports are assigned, we must determine the causality of every linear resistive power port. In order to do so we analyze the rank of $F(x)$ of individual components; we then apply the elementary column operation to the appropriate resistive power ports. Note that for components with multiple resistors, there can be more than one unique solution of column swapped resistive ports, but any of the solutions will work in simulation. A summary is listed in Algorithm 4:

Algorithm 4 Kernel2Hybrid

-**for all** Components
- Determine which resistive columns to interchange;
- Swap said resistive columns to obtain full ranked F;
- $J \leftarrow -F^{-1}E$;

In Simulink, every block has a clearly defined input-output relationship. With each component's hybrid-input-output representation, we can derive their corresponding block diagrams and simulate them inside Simulink. The interpreter uses Simulink to create the block diagrams, where each component is modeled with the subsystem block. Inside each component subsystem, there is another subsystem (the Dirac structure of the component) along with its attached ports; the Dirac structure subsystem consists of a series of sums and gains, depicting the dynamics of the Dirac structure equations. Power ports are modeled as follows:

1. Source of effort: effort modeled as a step input (i.e. a constant voltage source); flow sunk to a scope.

2. Source of flow: flow modeled as a step input (i.e. a constant current source); effort sunk to a scope.

3. Storage port: flow goes into an integrator and a gain (inverse of the storage value) to obtain effort.

4. Resistive port: effort and flow have a linear relationship, which can be represented by a gain, whose value depends on the orientation of the resistive element.

5. Interaction port: effort and flow modeled as Inport and Outport.

Algorithm 5 creates block diagrams using the hybrid-input-output representation of the Dirac structure and the constituent equations of power ports.

4.3.2.3 Causal Simulation of Engine-Dynamometer

Applying Algorithm 4 to the engine-dynamometer example, we obtain the hybrid input-output representation of the Dirac structure of the engine (4.10):

Algorithm 5 DS2BD

-**for all** Components
- **for all** Power ports
- **if** Non-interaction port
- Model according to rules;
- **if** Interaction port
- Model as Simulink inports/outports;

-Connect all components;

$$\begin{bmatrix} f_r \\ f_s \\ f_j \\ f_d \end{bmatrix} = \begin{bmatrix} 0 & 0 & -T & 0 \\ 0 & 0 & -1 & 0 \\ T & 1 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} e_r \\ e_s \\ e_j \\ e_d \end{bmatrix}, \tag{4.10}$$

Integrating the switch signal s into the Dirac structure results in the following hybrid input-output representation of the starter motor. The $J(x)$ matrix in (4.11) is modulated by the Boolean signal s, which shows the hybrid dynamics of the starter motor system.

$$\begin{bmatrix} e_s \\ f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 0 & s & 1-s \\ -s & 0 & 0 \\ s-1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_s \\ e_1 \\ e_2 \end{bmatrix}. \tag{4.11}$$

Finally, the hybrid input-output representation for the dynamometer is shown in (4.12):

$$\begin{bmatrix} e_u \\ e_d \\ f_R \\ f_I \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ -1 & -1 & 0 & 0 \\ -1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} f_u \\ f_d \\ e_R \\ e_I \end{bmatrix}. \tag{4.12}$$

Simulating the Simulink model of the engine-dyno with the same control inputs and parameters, we can view the results of the system output f_r plotted with 3 seconds shown

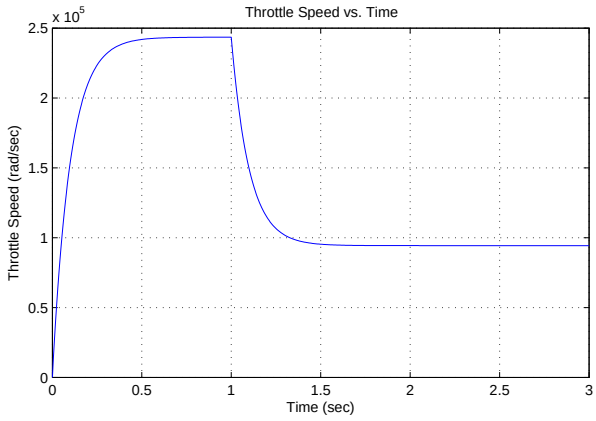

Figure 4.13: Simulink simulation results

in Figure 4.13.

4.4 Summary

Using concepts in Model Integrated Computing, we developed PHSML, a domain specific modeling language for compositional modeling using PHS. By defining constraints within GME, we are able to enforce interconnection rules which satisfies the theory of PHS, thus guaranteeing that the resulting models retain certain properties such as passivity. We have implemented an interpreter which exports the structural data of the model into a MATLAB script. The MATLAB functions that we designed read these scripts and generate the Dirac structure, constituent equations, and Hamiltonian function of each component.

We present a case study of an engine-dynamometer model, which is built using PHSML. The model contains nonlinearities in the engine component and hybrid dynamics in the starter motor component. We developed a component-based method of simulating systems built using PHSML by deriving acausal models of computation, in which the constituent equations of the acausal kernel representation is derived from the model; Modelica code is generated from the equations and simulated using Dymola's differential algebraic equations solver. We extended the component-based method of simulating systems by deriving causal models of computation, in which the acausal kernel representation is converted into

the causal hybrid input-output representation, then instantiated as block diagrams and simulated using Simulink's ordinary differential equations solver.

Chapter 5

Compositional Model-Based Control Design Using Port-Hamiltonian Systems

Heterogeneous domains create challenges for the composition and integration of controllers with the physical system. Many methods of integration are based on the ad hoc idea of "making it work" and though these methods have worked well in the past, complexities of modern automotive systems have made these methods implausible. Problems from the integration of components are often discovered late in the development cycle, leading to costly ramifications of the designs. As modern CPS expand in complexity, a modeling framework is needed to effectively generate reliable models which are compositional and have the ability to handle complex interactions. An autonomous vehicle is an example of a complex cyber-physical system (CPS) containing physical dynamics and controllers controlling the speed and steering of the vehicle [3]. A speed system controls the speed of the vehicle and is a hybrid system operating in two modes, throttle control mode where the throttle angle is determined and brake control mode where the brake pressure is determined. A steering system controls the angle of the steering wheel in order to maintain a desired position on the road.

The main contribution of this chapter is a passive control design method for the model-based design of CPS which shows that by designing controllers as port-Hamiltonian systems (PHS), the interactions between the computational and physical domains occur through power ports, allowing conditions for passivity to be formalized. Through PHS, we unify the formulation of the computational and physical domain by emphasizing the idea of power/energy flow. The methodology begins with the physical system which is modeled as a PHS. Passivity-based control methods are used to design the controllers which are also modeled as PHS [147]. We present a case study where the model-based design methodology is applied to an automotive control system. The physical dynamics of a vehicle system

is decomposed into its longitudinal and lateral dynamics interacting through a Dirac structure. We then design a speed and steering control system that interact with each other to achieve autonomous driving. This work has been presented in [148] and [149].

5.1 Model-Based Design Process

Model-based design of CPS is an active research area where a large amount of work is being done to address the various challenges caused by the heterogeneity of the different layers of CPS [3]. We created a tool-chain to integrate the components of control design from MATLAB/Simulink with the aspects of software deployment such as scheduling, discretization, and quantization with a primary application to automotive control systems [8]. The tool-chain is designed using an embedded software design environment called Embedded Systems Modeling Language (ESMoL) [150] which enables a software development process unifying the control design stage of development with code generation and deployment. The tool-chain is evaluated over a hardware-in-the-loop experimental platform over a time-triggered communication network which guarantees that the whole process is reliable, predictable, and robust to disturbances [151].

The first step of the model-based design tool-chain is designing and modeling the controller for a particular function in MATLAB/Simulink and using simulations to test and verify the correctness of the system. The methodology we developed in this paper focuses on this step of the model-based design process by modeling both the controller and plant as PHS, and using passivity-based design methods to ensure correct behavior of the overall system. PHS provide an effective way of characterizing the interactions presented in the CPS and is also able to model nonlinearities and hybrid dynamics. The resulting control design model from this approach can then be imported into ESMoL and used for subsequent phases of the model-based design process. The remaining design steps consist of importing the control designs into ESMoL, specifying the logical software architecture, defining the hardware platform, deploying the model, specifying the timing behavior of the

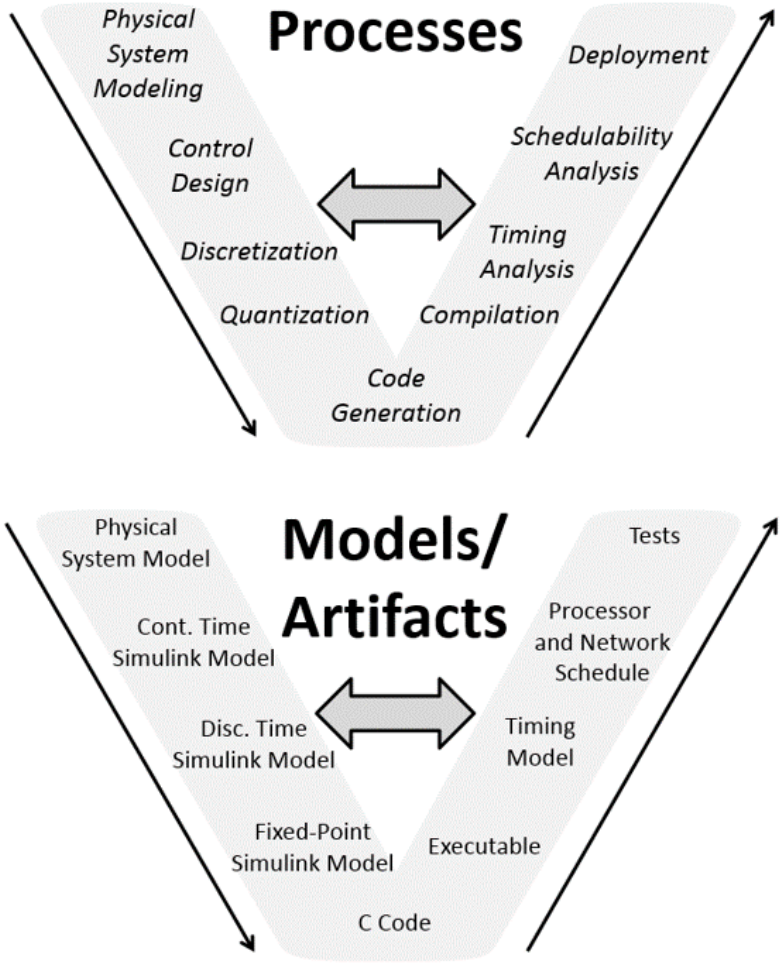

Figure 5.1: Diagram of the processes and artifacts

system, flattening the model for analysis, scheduling the tasks, and implementing on the platform for testing [8].

Figure 5.1 shows a diagram of the flow of processes of the model-based design process and the corresponding flow of models/artifacts. The left branch of each diagram focuses on design using models, while the right branch of each diagram focuses on the implementation on a HIL simulation platform. There are several dependencies between the left and right branches of the processes and models/artifacts. For example, schedulability analysis may impose constraints on the sampling rate that affect the discretization of the controller and its stability analysis. Additionally, the model and software artifact generated from each step of the process must be analyzed in order to validate their correctness. The rest of the

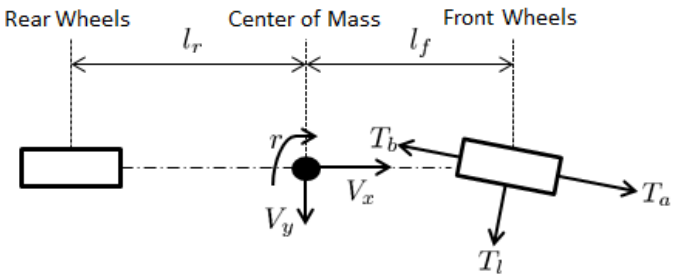

Figure 5.2: Free-body diagram of the vehicle dynamics

chapter is organized based on the flow of Figure 5.1.

5.2 Physical System Modeling

The objective of physical system modeling is to represent the free-body diagram of a physical system as a PHS model implemented in Simulink. The Simulink model is then used for control design. This necessitates the use of PHSML (developed in Chapter 4) which contains the rules and constraints of PHS. In this chapter we demonstrate the model-based design methodology using an automotive system as an example. Figure 5.2 shows a free-body diagram of the vehicle dynamics. The vehicle is front-wheel drive, resulting in the input forces from throttle and brake being applied to the front wheels shown in the diagram.

The longitudinal input force from the throttle, $T_a = C_a\theta_a$, is a function of the throttle valve angle θ_a and the experimental throttle constant C_a. The longitudinal input force from the brakes, $T_b = C_bP_b$, is a function of the braking pressure P_b and the experimental braking constant C_b. The lateral input force from the steering, $T_l = 2C_f\delta$, is a function of the steering angle δ and the cornering stiffness of the front wheels C_f. The longitudinal velocity, lateral velocity, and yaw rate, are represented by V_x, V_y, and r, respectively. Interactions between the longitudinal and lateral dynamics can be derived by analysis of the free-body diagram [79].

A PHS representation of the longitudinal dynamics, the lateral dynamics, and their

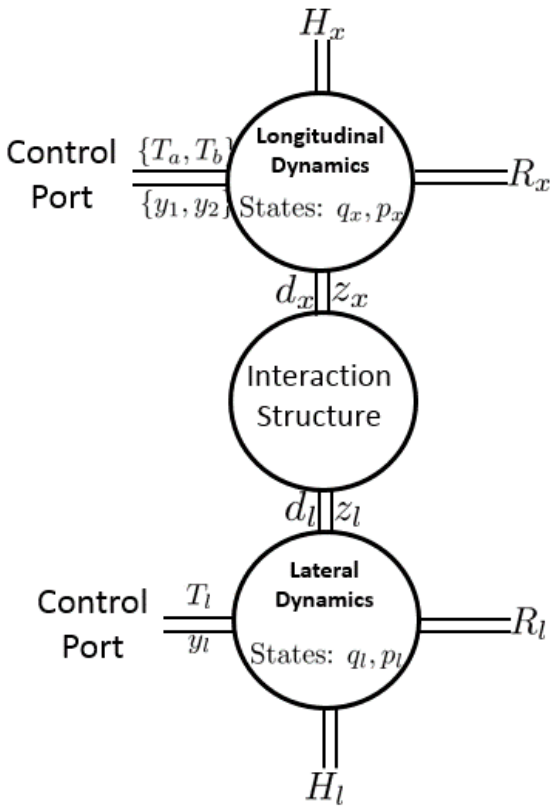

Figure 5.3: PHS representation of the vehicle dynamics

interactions is shown in Figure 5.3. The vehicle dynamics model is decomposed into a longitudinal dynamics component, a lateral dynamics component, and an interaction structure. The longitudinal dynamics contain two control ports (T_a, y_1) and (T_b, y_2) and an interaction port (d_x, z_x). The state variables are the longitudinal momentum p_x and the longitudinal displacement q_x. The outputs of the control ports y_1 and y_2 are V_x and $-V_x$, respectively. The x-component of the lateral force affecting longitudinal motion is represented by d_x and its power-conjugate is represented by z_x. The lateral dynamics contain a control port (T_l, y_l) and an interaction port (d_l, z_l). The state variables are $q_l = \begin{bmatrix} q_y & q_r \end{bmatrix}^\mathsf{T}$ and $p_l = \begin{bmatrix} p_y & p_r \end{bmatrix}^\mathsf{T}$, where p_y is the lateral momentum, p_r is the angular momentum, q_y is the lateral displacement, and q_r is the angular displacement. The output of the control port y_l is $V_y + l_f r$. The y-component of the longitudinal force applied to the center of mass is represented by d_l and its power-conjugate is represented by z_l.

5.2.1 Vehicle Longitudinal Dynamics

The longitudinal dynamics is a nonlinear system where the vehicle longitudinal velocity, V_x, is affected by the acceleration and deceleration force T_a and T_b. The longitudinal dynamics has the following Hamiltonian function:

$$H_x(q_x, p_x) = \frac{1}{2m} p_x^2 + U_x(q_x),$$

where m represents the mass of the vehicle and $U_x(q_x)$ represents the potential energy. The longitudinal dynamics is modeled in the form of (3.5):

$$\begin{cases} \begin{bmatrix} \dot{q}_x \\ \dot{p}_x \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -1 & -R_x \end{bmatrix} \begin{bmatrix} \frac{\partial H_x}{\partial q_x} \\ \frac{\partial H_x}{\partial p_x} \end{bmatrix} + \begin{bmatrix} 0 \\ G_x \end{bmatrix} u_x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d_x \\ y_x &= \begin{bmatrix} 0 & G_x^\mathsf{T} \end{bmatrix} \begin{bmatrix} \frac{\partial H_x}{\partial q_x} & \frac{\partial H_x}{\partial p_x} \end{bmatrix}^\mathsf{T} \\ z_x &= \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial H_x}{\partial q_x} & \frac{\partial H_x}{\partial p_x} \end{bmatrix}^\mathsf{T}, \end{cases} \tag{5.1}$$

where $u_x = \begin{bmatrix} T_a & T_b \end{bmatrix}^\mathsf{T}$, $G_x = \begin{bmatrix} 1 & -1 \end{bmatrix}$, $R_x = am + bp_x + \frac{cm^2}{p_x}$, a represents the tire rolling friction constant, b represents the air resistance constant, and c represents the static friction force.

5.2.2 Vehicle Lateral Dynamics

The lateral dynamics is a nonlinear system where the sum of the vehicle lateral velocity and modified vehicle yaw rate, $V_y + l_f r$, is affected by the lateral force T_l. The lateral dynamics has the following Hamiltonian function:

$$H_l(q_y, q_r, p_y, p_r) = \frac{1}{2m} p_y^2 + \frac{1}{2I} p_r^2 + U_l(q_y, q_r),$$

where I represents the moment of inertia of the vehicle and $U_l(q_y, q_r)$ represents the potential energy. The lateral dynamics is modeled in the form of (3.5):

$$\begin{cases} \begin{bmatrix} \dot{q}_l \\ \dot{p}_l \end{bmatrix} &= \begin{bmatrix} 0 & E \\ -E & -R_l \end{bmatrix} \begin{bmatrix} \frac{\partial H_l}{\partial q_l} \\ \frac{\partial H_l}{\partial p_l} \end{bmatrix} + \begin{bmatrix} 0 \\ G_l \end{bmatrix} T_l + \begin{bmatrix} 0 \\ K_l \end{bmatrix} d_l \\ y_l &= \begin{bmatrix} 0 & G_l^\mathsf{T} \end{bmatrix} \begin{bmatrix} \frac{\partial H_l}{\partial q_l} & \frac{\partial H_l}{\partial p_l} \end{bmatrix}^\mathsf{T} \\ z_l &= \begin{bmatrix} 0 & K_l^\mathsf{T} \end{bmatrix} \begin{bmatrix} \frac{\partial H_l}{\partial q_l} & \frac{\partial H_l}{\partial p_l} \end{bmatrix}^\mathsf{T}, \end{cases} \tag{5.2}$$

$$R_l = \begin{bmatrix} \frac{W_1}{V_x} & \frac{W_2}{V_x} \\ \frac{W_2}{V_x} & \frac{W_3}{V_x} \end{bmatrix},$$

where E is the identity matrix, $G_l = \begin{bmatrix} 1 & l_f \end{bmatrix}^\mathsf{T}$ and $K_l = \begin{bmatrix} 1 & 0 \end{bmatrix}^\mathsf{T}$. The parameter constants of R_l are $W_1 = 2C_f + 2C_r$, $W_2 = 2C_f l_f - 2C_r l_r$, and $W_3 = 2C_f l_f^2 + 2C_r l_r^2$, where C_r is the cornering stiffness of the rear wheels.

5.2.3 Vehicle Interaction Dynamics

Composition of the longitudinal and lateral dynamics through the interaction structure results in a nonlinear Dirac structure for the composed dynamics. The interaction between the longitudinal and lateral dynamics is a mapping of velocity to force, which indicates a gyrator relationship. The gyrator ratio must have units of kg/s which is represented by multiplying the mass of the vehicle with the yaw rate. The interaction structure is modeled as a Dirac structure modulated by the angular momentum p_r:

$$\begin{bmatrix} d_x \\ d_l \end{bmatrix} = \begin{bmatrix} 0 & -\frac{mp_r}{I} \\ -\frac{mp_r}{I} & 0 \end{bmatrix} \begin{bmatrix} z_x \\ z_l \end{bmatrix}. \tag{5.3}$$

The Hamiltonian function of the composed longitudinal and lateral dynamics is $H(q,p) = H_x + H_l$. Composition of (5.1) and (5.2) through (5.3) results in the following nonlinear PHS:

$$\begin{cases} \begin{bmatrix} \dot{q} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & E \\ -E & -R \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \\ y = \begin{bmatrix} 0 & G^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix}, \end{cases} \tag{5.4}$$

$$R = \begin{bmatrix} R_x & \frac{mp_r}{I} & 0 \\ \frac{mp_r}{I} & \frac{mW_1}{p_x} & \frac{mW_2}{p_x} \\ 0 & \frac{mW_2}{p_x} & \frac{mW_3}{p_x} \end{bmatrix}, G = \begin{bmatrix} G_x & 0 \\ 0 & G_l \end{bmatrix},$$

where $q = \begin{bmatrix} q_x & q_l \end{bmatrix}^{\mathsf{T}}$, $p = \begin{bmatrix} p_x & p_l \end{bmatrix}^{\mathsf{T}}$, $u = \begin{bmatrix} u_x & T_l \end{bmatrix}^{\mathsf{T}}$, and $y = \begin{bmatrix} y_x & y_l \end{bmatrix}^{\mathsf{T}}$. Interactions between the longitudinal and lateral dynamics are captured in R.

5.2.4 Model Validation

Validation of the analytical vehicle dynamics model is important for subsequent steps of the model-based design methodology. We use the CarSim S-function of a mid-size sedan for the HIL platform as the actual model [42]. Passivity indexes allow a way to characterize a system by determining its excess or shortage of passivity [112]. By selecting parameters so that the passivity indexes of the analytical models are similar to that of the CarSim model, we can conclude that the analytical models are reasonable approximations of the actual vehicle dynamics. The CarSim model has inherent bounds on its inputs [42]. The throttle angle valve (θ_f) has a lower bound of 0 and an upper bound of 1.5. The brake pressure (P_b) has a lower bound of 0 and upper bound of 10. The steering angle (δ) has a lower bound of -480 and an upper bound of 480 [42]. Using these CarSim variable bound values we mathematically determine that T_a has a lower bound of 0 N and an upper bound of 3104 N, T_b has a lower bound of 0 N and an upper bound of 3715 N, and T_l has a lower bound of -1200 N and an upper bound of 1200 N.

Table 5.1: Table of vehicle parameter values

a	b	c	C_r	l_r	C_f	l_f
0.1	0.006	10	200	1.4	300	1.4

We experimentally selected values for the vehicle model (Table 5.1) so that the passivity indexes of the analytical model closely match that of the CarSim model by running both models through twenty scenarios and optimizing the passivity index values using the method in [152]. The CarSim model gives the mass ($m = 1650$) and the inertia ($I = 3234$). Using the techniques demonstrated in [153], we determine that the passivity indexes of the CarSim model (ν_c, ρ_c) are (181, 0.6). We determined that the passivity indexes of the analytical model (ν_a, ρ_a) are (177, 0.6), indicating that the analytical model is a reasonable approximation of the CarSim model.

5.3 Continuous-Time Control Design

A control system is designed to regulate the behavior of other systems. Given the model of a physical system implemented as a PHS, we can design controllers which enable the closed-loop system to behave correctly. Our objective is to design controllers modeled as PHS which interface with the physical system through designated power ports, thereby regulating the behavior of the closed-loop system. Continuous-time control design is midway between physical system modeling and discretization within the overall model-based design process. We use passivity analysis in order to ensure that the closed-loop system remains stable, minimum-phased, and have a low relative degree [154]. The Simulink model generated from the controller PHS is used as the initial control design for the HIL platform.

A high level system model of the vehicle dynamics interacting with the controllers (consisting of a speed control and steering control) is shown in Figure 5.4. The controllers are implemented as PHS, and they interact with the vehicle dynamics through the power ports of T_a, T_b, and T_l, which were previously defined in the beginning of Section 5.2. Transformation of PHS into Simulink is a relatively simple procedure because the PHS equations are written in a format similar to state-space representation. State variables and subsequent computations are linked together through integrators and adders.

The objective of the controllers is to maintain a desired speed V_d and lateral displacement q_d. The controller model consists of a speed control component, a steering control component, and an interaction structure. The speed control shares the two control ports with the longitudinal dynamics and contains two interaction ports (d_{a1}, z_{a1}) and (d_{a2}, z_{a2}). Its state variables $x_a = \begin{bmatrix} x_{at} & x_{ab} \end{bmatrix}^{\mathsf{T}}$ are derived using the desired speed, where $x_{at} = \int_{t_0}^{t} (V_x - V_d) d\tau$ and $x_{ab} = \int_{t_0}^{t} (V_d - V_x) d\tau$. The steering control shares the control port with the lateral dynamics and contains two interaction ports (d_{b1}, z_{b1}) and (d_{b2}, z_{b2}). Its state variable $x_b = q_y - q_d$ is derived using the desired lateral displacement.

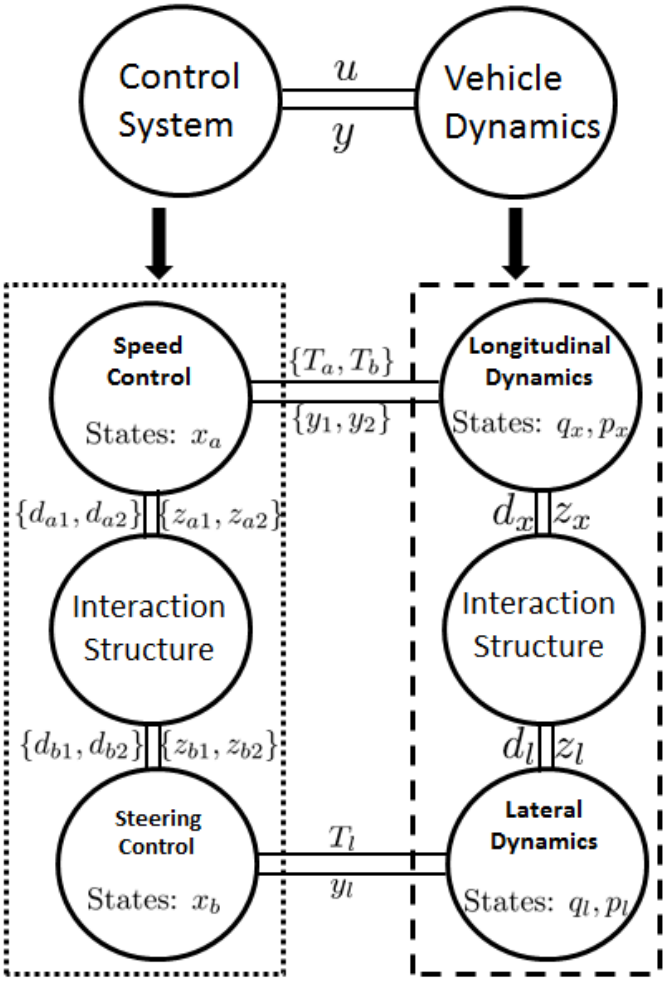

Figure 5.4: PHS representation of the controllers interacting with vehicle dynamics

5.3.1 Speed Control

The control objective of the speed control system is to prevent the host car from colliding into the lead car by maintaining a safe distance between the vehicles and a desired speed. For simplicity we consider the case shown in Figure 5.5 in which the vehicles are driving on a straight road, which allows us to disregard the lateral dynamics. We design the speed control to have the following Hamiltonian function:

$$H_a(x_a) = \frac{1}{2}(s_t k_{ti} x_{at}^2 + s_b k_{bi} x_{ab}^2),$$

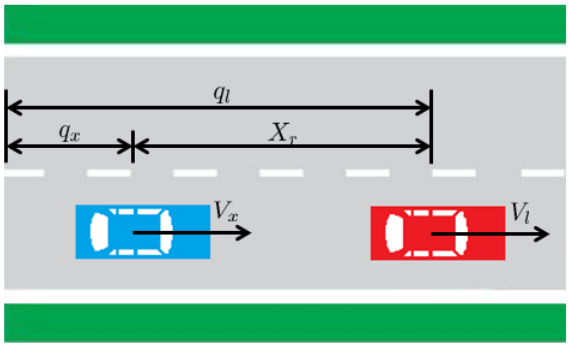

Figure 5.5: Lead vehicle and host vehicle on a straight road

where k_{ti} and k_{bi} are the gains of the Hamiltonian. The discrete variables $s = (s_t, s_b) \in \{0,1\}$ are used to model the hybrid dynamics of throttling and braking. s_t is the switch associated with the throttle control and similarly, s_b is the switch associated with the brake control. The switching dynamics are defined in (5.5), where h_+ and h_- are hysteresis constants introduced to prevent the system from rapidly alternating between accelerating and decelerating, and X_r and X_d are the relative distance between the two vehicles and the desired distance, respectively:

$$\begin{cases} (s_t, s_b) = (1,0) \text{ for } V_d - y_1 \geq 0, X_r \geq h_+ X_d, \\ (s_t, s_b) = (0,1) \text{ for } V_d - y_1 < 0, X_r < h_- X_d. \end{cases} \tag{5.5}$$

We design the speed control as an input-state-output PHS with direct-feed-through [89]:

$$\begin{cases} \dot{x} &= [J(x) - R(x)]\frac{\partial H}{\partial x} + G(x)u + K_1(x)d_1 \\ y &= G^{\mathsf{T}}(x)\frac{\partial H}{\partial x} + [M(x) + S(x)]u + K_2(x)d_2 \\ z &= \begin{bmatrix} K_1(x)^{\mathsf{T}} & 0 \\ 0 & K_2(x)^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial x} \\ u \end{bmatrix}, \end{cases} \tag{5.6}$$

where $M(x) \in \mathbb{R}^{m \times m}$ is a skew symmetric interconnection matrix and $S(x) \in \mathbb{R}^{m \times m}$ is a symmetric positive-semi-definite damping matrix. We design the speed control in the form of (5.6) because the feed-through term helps to ensure zero steady-state error [2]:

$$\begin{cases} \dot{x}_a & = -R_a \frac{\partial H_a}{\partial x_a} + G_a y_x + K_{a1} d_{a1} \\ u_x & = G_a^{\mathsf{T}} \frac{\partial H_a}{\partial x_a} + S_a y_x + K_{a2} d_{a2} \\ \begin{bmatrix} z_{a1} \\ z_{a2} \end{bmatrix} & = \begin{bmatrix} K_{a1}^{\mathsf{T}} & 0 \\ 0 & K_{a2}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \frac{\partial H_a}{\partial x_a} \\ y_x \end{bmatrix}, \end{cases} \tag{5.7}$$

where $K_{a1} = \begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}}$ and $K_{a2} = \begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}}$. The parameter matrices are denoted as:

$$R_a = \begin{bmatrix} s_t k_t & 0 \\ 0 & s_b k_b \end{bmatrix}, G_a = \begin{bmatrix} s_t P & 0 \\ 0 & s_b \end{bmatrix},$$

$$S_a = \begin{bmatrix} s_t k_{td} & 0 \\ 0 & s_b k_{bd} \end{bmatrix},$$

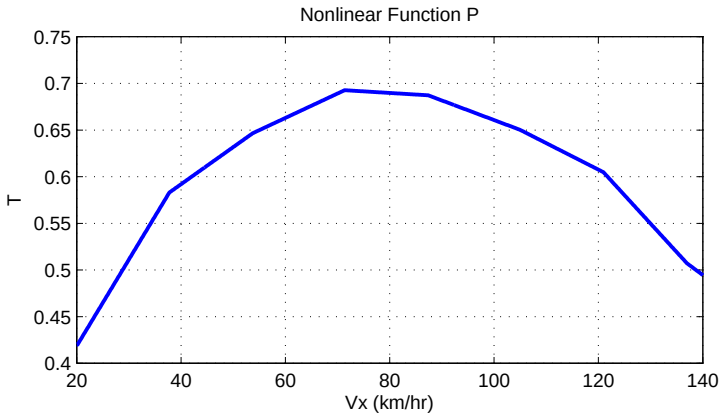

Figure 5.6: Nonlinear function P

where k_t and k_{td} are throttle control gains. k_b and k_{bd} are brake control gains. P is derived from the inverse engine map for the vehicle and is a mapping of the ratio of the acceleration force to V_x (Figure 5.6) [8].

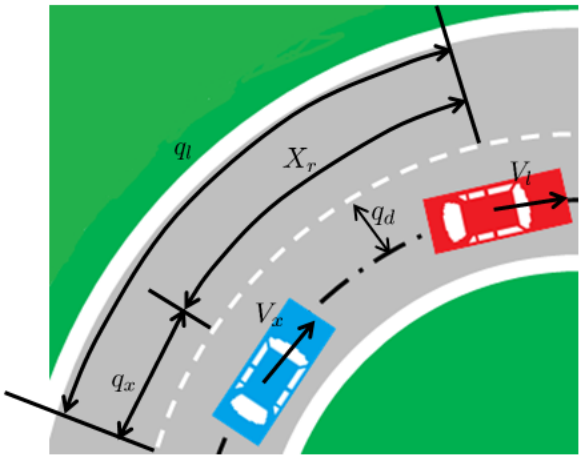

Figure 5.7: Diagram of lead vehicle and host vehicle on a curved road

5.3.2 Steering Control

In this section we consider the control objectives of a vehicle with both speed and steering control. In addition to maintaining a safe distance between it and the lead vehicle, the host car must also maintain a reasonable lateral acceleration as to not spin off the road (see Figure 5.7). This is a result of interactions between the lateral and longitudinal dynamics, characterized in (5.3). We consider the case shown in Figure 5.7 in which the lead vehicle and host vehicle are driving on a curved road. We design the steering control to have the following Hamiltonian function:

$$H_b(x_b) = \frac{1}{2} k_{si} x_b^2,$$

where k_{si} is the gain of the Hamiltonian. We design the steering control in the form of (5.6):

$$\begin{cases} \dot{x}_b &= y_l + d_{b1} \\ T_l &= \frac{\partial H_b}{\partial x_b} + k_{sd} y_l + d_{b2} \\ \begin{bmatrix} z_{b1} \\ z_{b2} \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial H_b}{\partial x_b} \\ y_l \end{bmatrix}, \end{cases} \tag{5.8}$$

where k_{sd} is the steering control gain.

5.3.3 Integrated Speed and Steering Control

It can be seen from (5.4) that the inputs to the longitudinal dynamics (T_a and T_b) affect the lateral dynamics. Similarly, the input to the lateral dynamics (T_l) affects the longitudinal dynamics. This can create problems such that at high speeds, actions by the speed control may interfere with the objective of the steering control. In order to alleviate this problem, we introduce an interaction structure so that the state variables and outputs of the speed control are affected by the state variable of the steering control, and vice versa. Similar to (5.3), the interaction structure of the control system is represented with the following Dirac structure:

$$\begin{bmatrix} d_{a1} \\ d_{a2} \\ d_{b1} \\ d_{b2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & J_c & 0 \\ 0 & 0 & 0 & M_c \\ -J_c^{\mathsf{T}} & 0 & 0 & 0 \\ 0 & -M_c^{\mathsf{T}} & 0 & 0 \end{bmatrix} \begin{bmatrix} z_{a1} \\ z_{a2} \\ z_{b1} \\ z_{b2} \end{bmatrix}. \tag{5.9}$$

The parameters J_c and M_c define how the speed control and the steering control interact. The Hamiltonian function of the composed control system is denoted as $H_c = H_a + H_b$. Composition of (5.7) and (5.8) through (5.9) results in the following PHS representation:

$$\begin{cases} \dot{x} &= \begin{bmatrix} -R_a & J_s \\ -J_s^{\mathsf{T}} & 0 \end{bmatrix} \frac{\partial H_c}{\partial x} + \begin{bmatrix} G_a & 0 \\ 0 & 1 \end{bmatrix} y \\ u &= \begin{bmatrix} G_a^{\mathsf{T}} & 0 \\ 0 & 1 \end{bmatrix} \frac{\partial H_c}{\partial x} + \begin{bmatrix} S_a & M_s \\ -M_s^{\mathsf{T}} & k_{sd} \end{bmatrix} y, \end{cases} \tag{5.10}$$

where $J_s = \begin{bmatrix} J_c & 0 \end{bmatrix}^{\mathsf{T}}$, $M_s = \begin{bmatrix} M_c & 0 \end{bmatrix}^{\mathsf{T}}$, and $x = \begin{bmatrix} x_{at} & x_{ab} & x_b \end{bmatrix}^{\mathsf{T}}$. The desired velocity V_d and the desired displacement q_d are included in the state variables x. Inclusion of J_c and M_c

in (5.10) is a result of the interaction between the lateral and longitudinal dynamics in the vehicle model. Figure 5.8 shows the implementation of the continuous-time control design in Simulink.

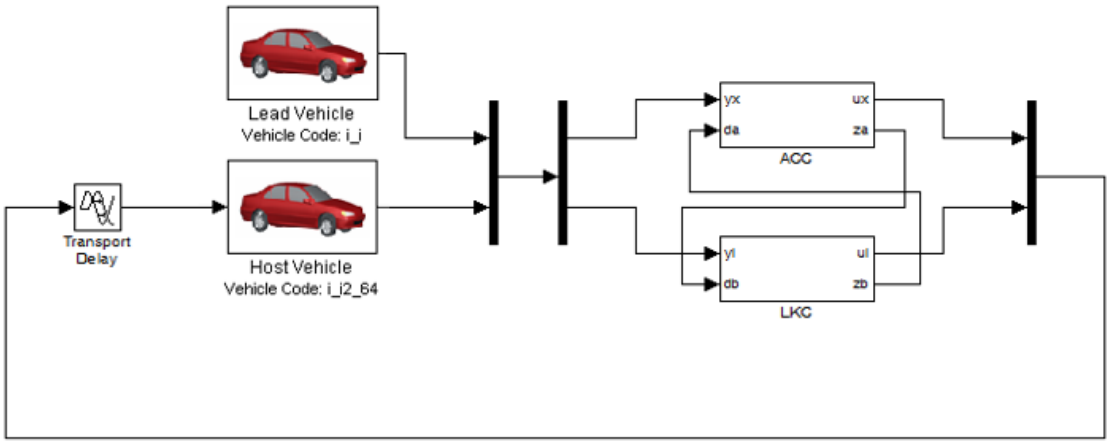

Figure 5.8: Simulink block diagram of the continuous-time control design

5.4 Passivity Analysis

Passivity is commonly defined in the state-space approach, where the energy stored in the system is related to the external energy coming into the system [154]. Passivity provides a sufficient condition for the stability of a system; a passive system, when unforced, is Lyapunov stable. In the case that the Hamiltonian function used is positive semi-definite, additional conditions on zero-state detectability is needed to ensure that the passive system remains stable. In addition to stability, passive systems also contain phase properties; passive systems have minimum-phase and possess a low relative degree [53]. Regardless of the representation, Dirac structures establish the power balancing equation which is fundamental to PHS.

H in (3.1) denotes the Hamiltonian function of the PHS. The Hamiltonian function represents the energy stored in the system. The flows of the energy storage are given by the rate x' of the energy state variables x; the efforts are given by the co-energy variables $\frac{\partial H}{\partial x}$ [86]. e_R and f_R denote the effort and flow variables associated with internal energy

dissipation, where the relationship between the power conjugate variables obey a static resistive relationship of the form of $R(f_R, e_R) = 0$. e_C and f_C denote effort and flow variables associated with external control. These control port variables are accessible to controller action, where one variable acts as the actuator and the conjugate variable acts as the sensor. The final terms in (3.1), e_I and f_I, denote effort and flow variables associated with interactions of the PHS with the environment. In PHSML, the Hamiltonian function of a component is associated with its storage elements, described by $H(x) = \sum_{i=1}^{n} \frac{1}{2k_i} x_i^2$, where n is the total number of storage elements in the component, k_i is the parameter constant of the ith storage element, and x_i is the state variable of the ith storage element. Passivity of the system can be stated as:

$$\int_{t_0}^{t} e_I^{\mathsf{T}} f_I d\tau + \int_{t_0}^{t} e_C^{\mathsf{T}} f_C d\tau = H(x) - \int_{t_0}^{t} e_R^{\mathsf{T}} f_R d\tau, \tag{5.11}$$

(5.11) implies that the component is passive with respect to a supply energy of $\int_{t_0}^{t} e_I^{\mathsf{T}} f_I d\tau + \int_{t_0}^{t} e_C^{\mathsf{T}} f_C d\tau$ and dissipated energy of $\int_{t_0}^{t} e_R^{\mathsf{T}} f_R d\tau$ as long as the Hamiltonian function is positive-definite.

Theorem 4. *The closed-loop system (5.10) is passive with respect to inputs y, outputs u, and Hamiltonian function $H_c = H_a + H_b$ if k_{ti}, k_{bi}, k_t, k_{td}, k_b, k_{bd}, k_{si}, $k_{sd} \geq 0$. Additionally, (5.10) will asymptotically stabilize the velocity V_x and the lateral position q_y to the desired velocity V_d and lateral position q_d, respectively.*

Proof. Passivity of the composed control system is proven using the energy-balancing equation:

$$\begin{aligned} \frac{dH_c}{dt} &= \frac{\partial H_c}{\partial x}^{\mathsf{T}} \begin{bmatrix} -R_a & J_c \\ -J_c^{\mathsf{T}} & J_l \end{bmatrix} \frac{\partial H_c}{\partial x} + \frac{\partial H_c}{\partial x}^{\mathsf{T}} \begin{bmatrix} G_a & 0 \\ 0 & G_l \end{bmatrix} y \\ &\leq u^{\mathsf{T}} y - y^{\mathsf{T}} \begin{bmatrix} S_a & 0 \\ 0^{\mathsf{T}} & S_l \end{bmatrix} y. \end{aligned}$$

Passivity of the system is shown by the inequality $\frac{dH_c}{dt} \leq u^{\mathsf{T}} y$. Asymptotic stability of the closed-loop system is shown by combining (5.4) with (5.10). The PHS representation of the closed-loop system:

$$\left\{ \begin{bmatrix} \dot{q} \\ \dot{p} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} -\tilde{M} & \tilde{M} & 0 \\ -\tilde{M}^{\mathsf{T}} & \tilde{J} & \tilde{K} \\ 0 & -\tilde{K}^{\mathsf{T}} & -\tilde{Q} \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{H}}{\partial q} \\ \frac{\partial \tilde{H}}{\partial p} \\ \frac{\partial \tilde{H}}{\partial x} \end{bmatrix}, \right.$$

where $\tilde{M}$, $\tilde{J}$, $\tilde{K}$, and $\tilde{Q}$ are defined as:

$$\tilde{M} = \mathrm{diag}\left(\frac{1}{m}, \frac{1}{m}, \frac{1}{I}\right), \tilde{J} = \begin{bmatrix} 0 & \frac{M_c}{m} & 0 \\ -\frac{M_c}{m}^{\mathsf{T}} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{K} = \begin{bmatrix} s_t k_{ti} P & 0 & 0 \\ 0 & s_b k_{bi} & 0 \\ 0 & 0 & \frac{k_{si}}{l_f} \end{bmatrix}, \tilde{Q} = \begin{bmatrix} s_t k_t & -\frac{J_c}{m} & 0 \\ \frac{J_c}{m}^{\mathsf{T}} & s_b k_b & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

with a coordinate-transformed Hamiltonian function, $\tilde{H}(q, p, z)$:

$$\begin{aligned} \tilde{H} = \; & \tfrac{s_t k_{td}}{2m}(mV_d - p_x)^2 + \tfrac{s_b k_{bd}}{2m}(p_x - mV_d)^2 \\ & + \tfrac{1}{2m} p_y^2 + \tfrac{1}{2I} p_r^2 + \tfrac{k_{sd}}{2}(q_y - q_d)^2 \\ & + \tfrac{s_t k_{ti}}{2} x_{at}^2 + \tfrac{s_b k_{bi}}{2} x_{ab}^2 + \tfrac{k_{si}}{2} x_b^2, \end{aligned}$$

which we can use to verify that $\dot{\tilde{H}}$ is always less than or equal to zero and that $p_x = mV_d$ and $q_y = q_d$ at the boundary. □

5.5 Simulation Results

In this section, we present simulation results which test and verify the closed-loop system. We validate the control design using simulations to show that the vehicle behaves correctly given changes in lead vehicle speed, road curvature, and slope of the road. Simulation of the closed-loop system consists of two minutes of running time in which the host vehicle follows a lead vehicle on a road with a trajectory shown in Figure 5.9. This trajectory is comprehensive because it contains many curves and straight segments which will test the effectiveness of the control design. The lead vehicle starts at a speed of 60 km/hr and runs for forty seconds, before speeding up linearly to 85 km/hr for twenty seconds; the lead vehicle maintains 85 km/hr for ten seconds, before linearly slowing down to 50 km/hr. Table 5.2 shows the gain values of the controllers, which are computed using power-shaping stabilization [155].

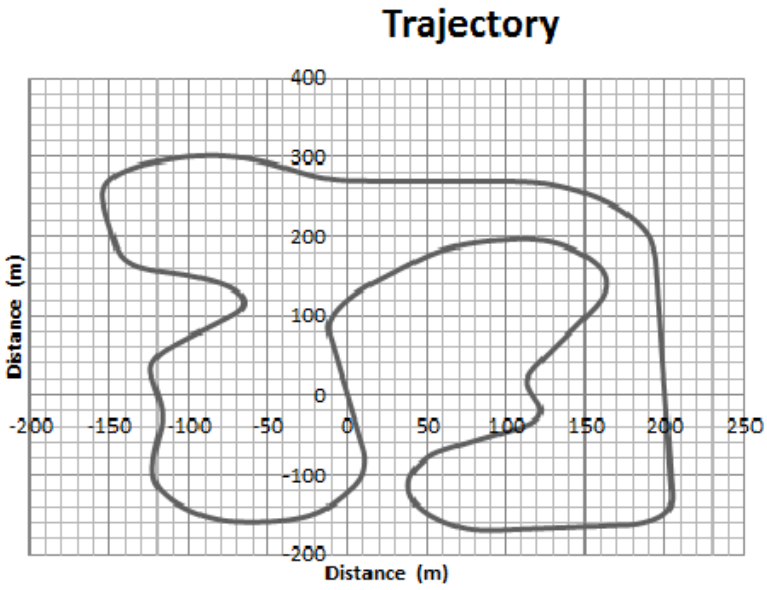

Figure 5.9: Course trajectory for vehicle

We present simulation results for two control systems; in the first control system there are no interactions between the speed control and steering control, $J_c, M_c = 0$; in the second control system, the interaction gains are $J_c = 0.2$ and $M_c = -0.5$. Figure 5.10 shows the

velocity of the lead vehicle and the host vehicle on the top subplot and the relative distance between the two vehicles on the bottom subplot; Figure 5.11 shows the lateral displacement on the top subplot and the lateral acceleration on the bottom subplot. Table 5.3 shows the various scenarios that appear during the simulation.

Table 5.2: Table of controller gains

k_{ti}	k_{bi}	k_t	k_{td}	k_b	k_{bd}	k_{si}	k_{sd}
0.05	0.01	0.1	0.02	0.2	0.02	40	15

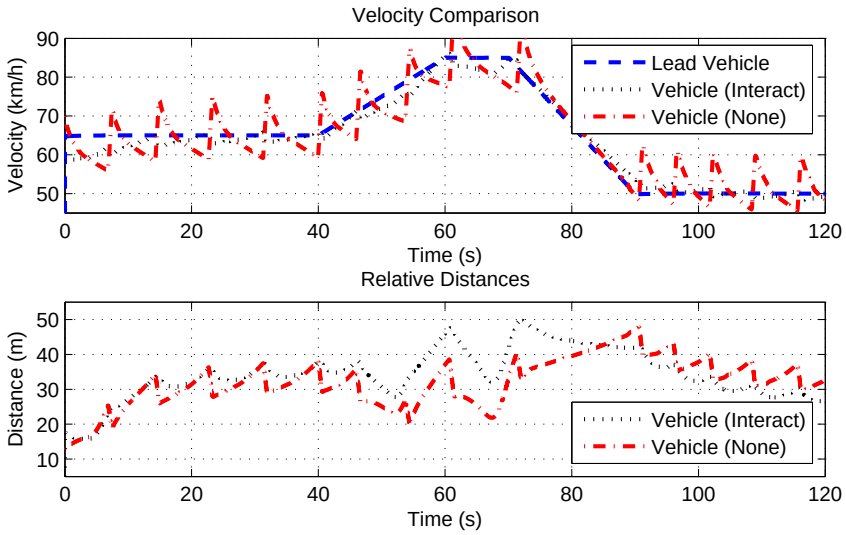

Figure 5.10: Vehicle velocities and relative distances

The simulation results indicate that the control system is able to keep to the objectives of speed control and steering control despite changes in the lead vehicle speed, slopes, and turns encountered by the host vehicle. For the results shown in Figure 5.10, the lack of interactions between the speed and steering control results in relatively large changes in speed; the addition of interactions results in a smoother velocity profile. Figure 5.11 also indicate better results for the case with interactions. Based on the simulation results, we determine that the maximum lateral acceleration $A_m \approx 1.2\ \mathrm{m/s^2}$; marking cyan dashed lines for A_m and $-A_m$ on the second subplots of Figure 5.11, we show that the vehicle operates

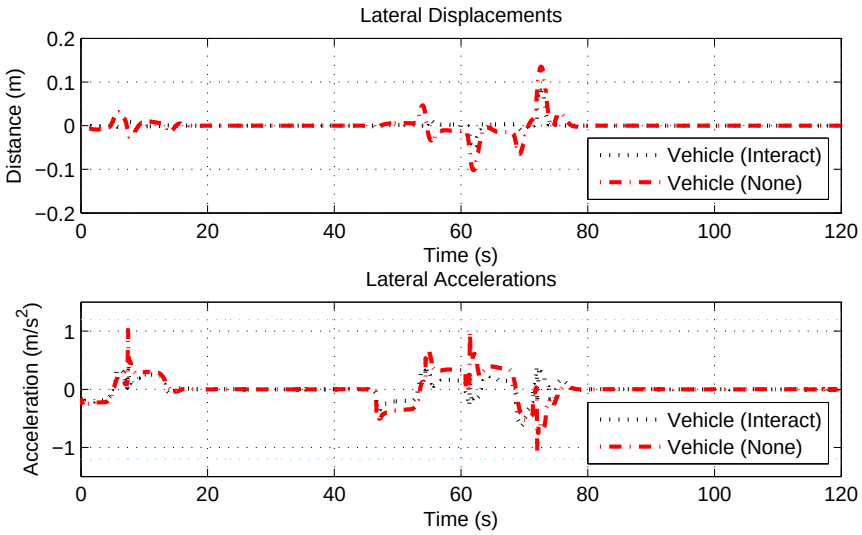

Figure 5.11: Vehicle lateral accelerations and displacements

Table 5.3: Table of simulation scenarios

Scenario	Time (s)	V_l (km/hr)	Slope (^{o})	turns
1	$0-40$	65	0	3
2	$40-52$	$65-77$	0	1
3	$52-60$	$77-85$	-15	0
4	$60-70$	85	-15	1
5	$70-90$	$85-50$	-15	1
6	$90-94$	50	-15	0
7	$94-103$	50	0	1
8	$103-120$	50	15	1

within a safe lateral acceleration range. Figures 5.12 and 5.13 show the stability of the vehicle velocity and lateral displacements.

5.6 Summary

We developed a compositional control design framework based on PHS which allows for the integration of multiple controllers. The key to this framework is the use of interaction structures which follow the rules of PHS which allows the closed-loop system to preserve passivity. We presented simulation results which demonstrate the effectiveness

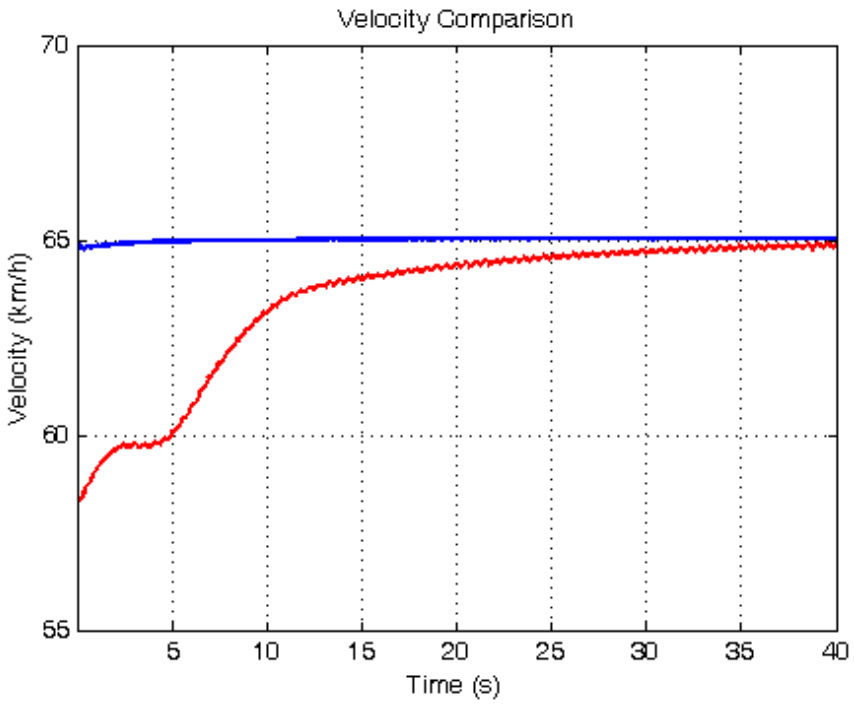

Figure 5.12: Stability of vehicle velocity

of the framework, which inspired us to use the framework as the beginning stage of our model-based design methodology.

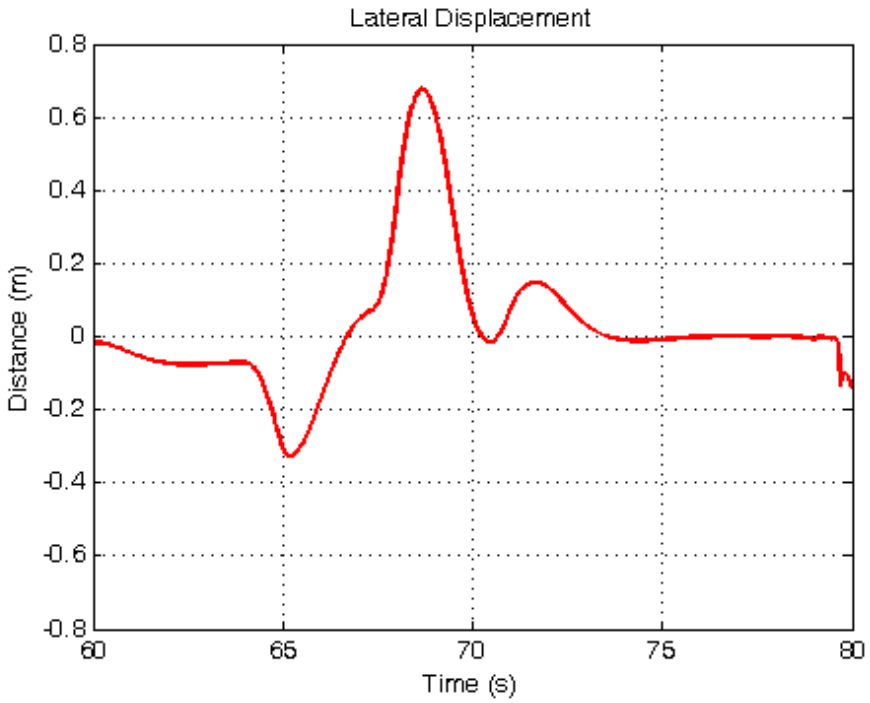

Figure 5.13: Stability of lateral displacements

Chapter 6

Safety Analysis of Automotive Control Systems Using Port-Hamiltonian Systems

Safe operation is an important requirement for a vehicle equipped with an adaptive cruise control (ACC) and lane keeping control (LKC) system. The design of the ACC and LKC systems must ensure that the host vehicle can safely navigate roads. The appearance of a lead vehicle provides an additional constraint for the ACC in that the host vehicle maintains a desired speed depending on the behavior of the lead vehicle. A lead vehicle which suddenly decelerates creates a safety problem for the host vehicle. The ACC design on the host vehicle must guarantee that the distance between the lead and host vehicle stay above a minimum threshold. Turns and curves provide constraints for the LKC in that the host vehicle must maintain its position in the center of the road. Large road curvatures create skidding problems for the host vehicle. The ACC and LKC design on the host vehicle must guarantee that the lateral acceleration does not exceed a maximum threshold. The challenge considered in this chapter is to prove the safety of an automotive control system consisting of ACC and LKC despite the nonlinearities, hybrid dynamics, and disturbances present in the system.

The main contribution of this chapter is an approach for the safety analysis of cyber-physical systems (CPS) such as automotive control systems. The dynamics of the vehicle and the control systems are described using port-Hamiltonian systems (PHS) which gives the approach the benefit of compositionality. Hybrid behavior is characterized using multi-modal PHS. The approach represents the safe states of the system using a bounded from above energy level of the Hamiltonian function. Similarly, the unsafe states of the system are represented using a bounded from below energy level of the Hamiltonian function. Passivity is used to prove that as long as the safe and unsafe energy regions do not overlap, trajectories that begin within a lower energy level (safe states) cannot terminate within a

higher energy level (unsafe states). The approach can be applied to any system described as a multi-modal PHS.

We evaluate the approach by analyzing the safety conditions for two systems. First, we assume a straight road and consider the longitudinal dynamics and the ACC. We derive safety conditions for the ACC which ensure that the host vehicle does not collide with a lead vehicle. Second, we assume a curved road and consider the interactions between the longitudinal dynamics, lateral dynamics, ACC, and LKC. We derive safety conditions for the ACC and LKC which ensure that the host vehicle does not collide with a lead vehicle or skid off of the road. We use the vehicle parameters, disturbances, and safety conditions to select control parameters so that the closed-loop system is safe. In order to validate the approach, we present simulation results by implementing the closed-loop system using Simulink and CarSim [42]. This work has been presented in [156].

6.1 Safety Analysis Approach

The idea of the approach is to use the energy of the system as conditions and constraints in order to show the safety property of the system. We consider the plant and controller dynamics described by multi-modal PHS. We use the dynamic equations and Hamiltonian functions to derive the dynamic equations and Hamiltonian function of the closed-loop system. We characterize the initial and unsafe regions using the energy of the Hamiltonian function and show that the system trajectory cannot enter the unsafe region.

6.1.1 Generic Safety Problem Statement

Given a hybrid system represented as (3.6) with Hamiltonian function $H(x)$ and bounded disturbances, the safety problem is to show that there are no trajectories of the closed-loop system that reach an unsafe region of the state space.

Definition 30. *Given a multi-modal PHS (3.6) and $H(x)$ with continuous states $X = X_p \times$*

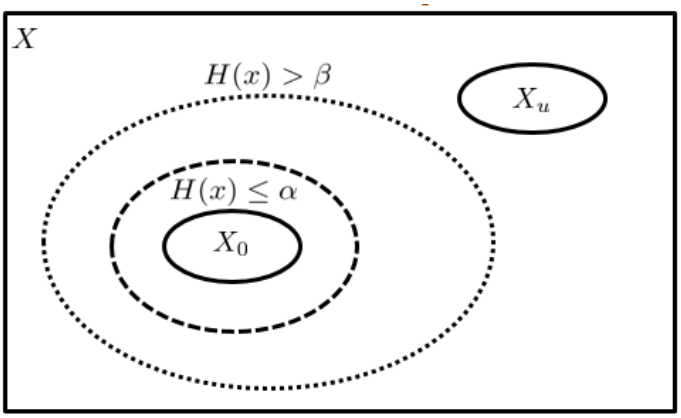

Figure 6.1: The Hamiltonian function prevents the trajectory from reaching the unsafe set X_u.

$X_c \subseteq \mathbb{R}^{n_p+n_c}$, discrete states $S = S_p \times S_c$, initial states $X_{p0} \times X_{c0} \times S_{p0} \times S_{c0} \subseteq X \times S$, unsafe states $X_{pu} \times X_{cu} \times S_{pu} \times S_{cu} \subseteq X \times S$, and disturbances $\Delta \subset \mathbb{R}^o$, a system trajectory $\Gamma(x(t),s(t)) : [0,T] \to X \times S$ is unsafe if there exists a positive time instant T and a finite sequence of discrete transition times $0 \leq t_1 \leq \cdots \leq t_N \leq T$ such that $\Gamma(x(0),s(0)) \in X_{p0} \times X_{c0} \times S_{p0} \times S_{c0}$ and $\Gamma(x(T),s(T)) \in X_{pu} \times X_{cu} \times S_{pu} \times S_{cu}$. The system is safe if there are no unsafe state trajectories.

6.1.2 Safety Analysis of Generic Multi-Modal Port-Hamiltonian System

We consider the following definitions for initial states, unsafe states, and guard conditions that specify discrete mode transitions. For each discrete state $s \in S$, the initial continuous states are defined as $\text{Init}(s) = \{x \in X : (x,s) \in X_{p0} \times X_{c0} \times S_{p0} \times S_{c0}\}$ and the unsafe continuous states are defined as $\text{Unsafe}(s) = \{x \in X : (x,s) \in X_{pu} \times X_{cu} \times S_{pu} \times S_{cu}\}$. Each transition of discrete states from $s \in S$ to $s' \in S$ is associated with the guard condition $\text{Guard}(s,s') = \{x,x' \in X : \{x,s\} \to \{x',s'\} \in \mathbb{T}\}$.

Similar to safety analysis using barrier certificates, the method in this paper shows that trajectories beginning from the safe region cannot reach the unsafe region. However, the barrier certificate typically separates the initial and unsafe states using its zero level set,

while the Hamiltonian function characterizes the initial and unsafe states using two energy levels. A canonical coordinate transform Φ is needed to convert the dynamic equations and Hamiltonian function of the system into a form which shows the actual minimum energy. Technical details regarding canonical coordinate transformation of PHS can be found in [133]. The passivity condition prevents trajectories starting in the safe region from reaching the unsafe region. Figure 6.1 provides a visual illustration of the method.

Theorem 5. *A multi-modal PHS described by (3.6) and $H(x)$, with continuous states $x \in X$, discrete states $s \in S$, initial states Init(s), unsafe states Unsafe(s), and bounded disturbances $\delta \in \Delta$ is safe if the canonical coordinate transformation $\bar{x} = \Phi(x)$ and transformed Hamiltonian function $H(\Phi^{-1}(\bar{x}))$ satisfy the following four conditions with $\alpha \leq \beta$*

1. $H(\Phi^{-1}(\bar{x})) \leq \alpha, \forall x \in \text{Init}(s)$
2. $H(\Phi^{-1}(\bar{x})) > \beta, \forall x \in \text{Unsafe}(s)$
3. $\zeta^{\mathsf{T}}\delta \leq \frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}}^{\mathsf{T}} \overline{R}(\bar{x},s) \frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}}, \forall \{x, \delta\} \in X \times \Delta$
4. $H(\Phi^{-1}(\bar{x})) \leq \alpha, \forall x \in \text{Guard}(s,s')$

Proof. Assuming that the Hamiltonian function $H(x)$ satisfy the four conditions in Theorem 5, yet there exists a time $T \geq 0$, an input δ, and initial states Init(s), and a trajectory $\Gamma(x(t),s(t))$ such that $\Gamma(x(T),s(T)) \in \text{Unsafe}(s)$. We show that the Hamiltonian function cannot simultaneously satisfy the four condition and reach the unsafe region, thus proving safety by contradiction. The time derivative of the Hamiltonian functions $\frac{dH}{dt}$ can be written as:

$$\begin{aligned}
\frac{\partial H(x)}{\partial x}^{\mathsf{T}} \dot{x} &= \frac{\partial H(x)}{\partial x}^{\mathsf{T}} [J(x,s) - R(x,s)] \frac{\partial H(x)}{\partial x} \\
&\quad + \frac{\partial H(x)}{\partial x}^{\mathsf{T}} L(x,s)\delta \\
&= \frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}}^{\mathsf{T}} [\overline{J}(\bar{x},s) - \overline{R}(\bar{x},s)] \frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}} \\
&\quad + \frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}}^{\mathsf{T}} \overline{L}(\bar{x},s)\delta \\
&= -\frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}}^{\mathsf{T}} \overline{R}(\bar{x},s) \frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}} + \zeta\delta
\end{aligned}$$

$$\overline{J}(\overline{x},s) = \frac{\partial \Phi}{\partial x} J(x,s) \frac{\partial \Phi}{\partial x}^{\mathsf{T}} \Bigg|_{x=\Phi^{-1}(\overline{x})}$$

$$\overline{R}(\overline{x},s) = \frac{\partial \Phi}{\partial x} R(x,s) \frac{\partial \Phi}{\partial x}^{\mathsf{T}} \Bigg|_{x=\Phi^{-1}(\overline{x})}$$

$$\overline{L}(\overline{x},s) = \frac{\partial \Phi}{\partial x} L(x,s) \Bigg|_{x=\Phi^{-1}(\overline{x})}$$

Condition (3) shows that the system trajectory on the time interval of $[0,T]$ is non-increasing, which indicates that $H(x(T)) \leq H(x(0))$. Additionally, condition (4) asserts that during a discrete transition, the Hamiltonian function will not jump to an increasing value. These statements, however, contradict the original assumption that the system states start at Init(s) and end at Unsafe(s). As a result, we can conclude that the system is safe.

□

6.2 Collision Avoidance

In this section, we consider the safety analysis of a vehicle with ACC following a lead car and maintaining a safe distance between the vehicles. The goal for the ACC is to prevent the host car from colliding into the lead car in the event of rapid deceleration. For simplicity, we consider the case shown in Figure 5.5 in which the vehicles are driving on a straight road, which allows us to omit the lateral dynamics. The longitudinal dynamics (5.1) contain the continuous states $\{q_x, p_x\} \in X_k \subseteq \mathbb{R}^2$, initial states $X_{k0} \subseteq X_k$, inputs $\{T_a, T_b\}$, and disturbances $\{\delta_g, \delta_{wx}\}$. The ACC (5.7) is connected to the longitudinal vehicle dynamics through the control ports and allows for autonomous driving by controlling T_a and T_b. The ACC system has continuous states $x_a \in X_a \subseteq \mathbb{R}^2$, discrete states $\{s_t, s_b\} \in S_a$, initial states $X_{a0} \times S_{a0} \subseteq X_a \times S_a$, and discrete transitions $\mathbb{T}_a \subseteq (X_a \times S_a) \rightarrow (X_a \times S_a)$.

The closed-loop system has a Hamiltonian function $H_k = H_a(x_a,s) + H_x(q_x,p_x)$, initial states $X_0 = X_{k0} \times X_{a0} \times S_{a0}$, discrete transitions $\mathbb{T}_k \subseteq (X \times S_a) \rightarrow (X \times S_a)$, and disturbances $\{\delta_g, \delta_{wx}\} \in \Delta_g \times \Delta_{wx}$. Its input-state-output PHS is described by:

$$
\begin{cases}
\begin{bmatrix} \dot{q}_x \\ \dot{p}_x \\ \dot{x}_{at} \\ \dot{x}_{ab} \end{bmatrix} = [\tilde{J}_x - \tilde{R}_x] \begin{bmatrix} \frac{\partial \tilde{H}_x}{\partial q_x} \\ \frac{\partial \tilde{H}_x}{\partial \hat{p}_x} \\ \frac{\partial \tilde{H}_x}{\partial x_{at}} \\ \frac{\partial \tilde{H}_x}{\partial x_{ab}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_g \\ \delta_{wx} \end{bmatrix} \\
\begin{bmatrix} \zeta_g \\ \zeta_{wx} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{H}_x}{\partial q_x} & \frac{\partial \tilde{H}_x}{\partial \hat{p}_x} & \frac{\partial \tilde{H}_x}{\partial x_{at}} & \frac{\partial \tilde{H}_x}{\partial x_{ab}} \end{bmatrix}^{\mathsf{T}}
\end{cases} \tag{6.1}
$$

$$
\tilde{J}_x = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & -s_t P & s_b \\ 0 & s_t P & 0 & 0 \\ 0 & -s_b & 0 & 0 \end{bmatrix},
$$

$$
\tilde{R}_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & R_x + s_t k_{td} + s_b k_{bd} & 0 & 0 \\ 0 & 0 & s_t k_t & 0 \\ 0 & 0 & 0 & s_b k_b \end{bmatrix}.
$$

6.2.1 Collision Avoidance Safety Problem Statement

The control gains can be selected to stabilize the host vehicle velocity to $V_l + \gamma \frac{(X_r - X_d)V_l}{X_d}$. However, stability does not imply safety. We do not consider the scenario in which a lead vehicle appears in front of the host vehicle driving faster than or equal to the host vehicle set speed because the safety property is trivial since the controller stabilizes the host vehicle velocity to the set speed indicating that the relative distance between the two vehicles will not be less than the initial relative distance. We consider the scenario in which a lead vehicle appears in front of the host vehicle driving slower than the host vehicle set speed. In this scenario, the safety property needs to be validated because if the ACC does not react

accordingly and slow the host vehicle to a reasonable speed, a collision may occur. The safety condition for the longitudinal dynamics asserts that the relative distance between the two vehicles will never reach a minimum distance q_m. We can represent the unsafe host vehicle displacement as the set of:

$$X_{ku} = \left\{ q_x \in \mathbb{R} : q_x \geq \int_0^t V_l d\tau + q_l(0) + q_m \right\}, \tag{6.2}$$

where $q_l(0)$ is the initial displacement value of the lead vehicle. Given (6.1), the safety condition for the longitudinal vehicle dynamics and ACC system states that that all possible trajectories cannot reach the unsafe region described by (6.2).

6.2.2 Safety Analysis of Collision Avoidance

In order to show safety, we make some assumptions regarding the parameters of the lead and host vehicle. The first assumption is that the initial velocity of the lead vehicle is greater than a minimum velocity which depends on the deceleration of the lead vehicle (a_l) and the relative distance between the vehicles. The second assumption is that the initial relative distance between the vehicles is greater than a minimum distance which depends on the deceleration and velocity of the lead vehicle. If the initial velocity of the vehicle is high compared to the host vehicle velocity, then the initial relative displacement can be low because the host vehicle does not need a large distance to react to the lead vehicle velocity. However, if the initial velocity of the vehicle is low compared to the host vehicle velocity, then the initial relative displacement must be high because the host vehicle needs a larger distance to react to the low lead vehicle velocity. The relationship between the initial relative distance and the initial vehicle velocities is described in (6.3).

$$X_r(0) = \frac{V_l^2(0)}{2a_l} - \frac{V_x^2(0)}{2\dot{V}_x}. \tag{6.3}$$

We need the following definitions for initial states, unsafe states, and guard sets. For

each discrete state $s_a \in S_a$, the initial continuous states are defined as $\text{Init}(s_a) = \{ (q_x, p_x, x_a) \in X : (q_x, p_x, x_a, s_a) \in X_0 \}$ and the unsafe continuous states are defined as $\text{Unsafe}(s_a) = \{ (q_x, p_x, x_a) \in X : q_x \in X_{ku} \}$. Each transition of discrete states from $s_a \in S_a$ to $s'_a \in S_a$ is defined using the guard condition $\text{Guard}(s_a, s'_a) = \{ (q_x, p_x, x_a), (q_x, p_x, x_a)' \in X : (q_x, p_x, x_a, s_a) \rightarrow (q'_x, p'_x, x'_a, s'_a) \}$. Safety analysis of the longitudinal dynamics uses $\overline{p}_x = \Phi(p_x) = p_x - m(1 + \gamma \frac{X_r - X_d}{X_d}) V_l$ as the canonical coordinate transformation on the longitudinal momentum.

We apply Theorem 5 to the composed longitudinal dynamics and ACC system. Given initial conditions $\text{Init}(s_a)$, we derive the energy bound α as a function of the initial host vehicle velocity $V_x(0)$, initial relative distance $X_r(0)$, and initial lead vehicle velocity $V_l(0)$. The initial relative distance must be greater than or equal to $\frac{V_l^2(0)}{2a_l} - \frac{V_x^2(0)}{2a_l}$ where a_l is the bounded lead vehicle deceleration. Consequently, we restate the first condition of Theorem 5 as $H_k(\Phi^{-1}(\overline{p}_x)) \leq \alpha, \forall x \in \text{Init}(s_a)$, where

$$\alpha = m \frac{k_{td} + k_{bd}}{2} (V_x(0) - (1 + \gamma \frac{X_r(0) - hV_l(0) - S_0}{hV_l(0) + S_0}) V_l(0))^2.$$

Given the unsafe states $\text{Unsafe}(s_a)$, we derive the energy bound β as a function of host vehicle velocity V_x and lead vehicle velocity V_l. The energy of the transformed Hamiltonian function has a maximum value which indicates that the minimum relative distance has been reached. Consequently, we restate the second condition of Theorem 5 as $H_k(\Phi^{-1}(\overline{p}_x)) > \beta, \forall x \in \text{Unsafe}(s_a)$, where

$$\beta = m \frac{k_{td} + k_{bd}}{2} (V_x - (1 - \gamma) V_l)^2.$$

Given an initial relative distance greater than q_m, α is less than β, which validates the first two conditions. Given the disturbances $\{\delta_g, \delta_{wx}\} \in \Delta$, we must guarantee that the system trajectory will never begin in $\text{Init}(s_a)$ and end in $\text{Unsafe}(s_a)$. Consequently, we restate the third condition of Theorem 5 as

$$\zeta_g \delta_g + \zeta_{wx} \delta_{wx} \leq$$

$$\frac{\partial H_k(\Phi^{-1}(\overline{p}_x))}{\partial \overline{p}_x}^{\mathsf{T}} \frac{\partial \Phi}{\partial p_x} R_x(\Phi^{-1}(\overline{p}_x)) \frac{\partial \Phi}{\partial p_x}^{\mathsf{T}} \frac{\partial H_k(\Phi^{-1}(\overline{p}_x))}{\partial \overline{p}_x}$$

$$\forall (q_x, p_x, x_a, \delta_g, \delta_{wx}) \in X \times \Delta.$$

Discrete transitions between the throttle and brake control mode must also be taken into account in order to guarantee that the system will not transition into Unsafe(s_a). We restate the fourth condition of Theorem 5 as $H_k(\Phi^{-1}(\overline{p}_x)) \leq \alpha$, $\forall (q_x, p_x, x_a) \in \text{Guard}(s_t, s_b)$. In Section 5.3, the ACC is designed by selecting control parameters that satisfy these safety conditions.

6.3 Skidding Avoidance

In this section, we consider the safety problem of a vehicle with both ACC and LKC following a lead car around a curved road (Figure 5.7). In addition to maintaining a safe distance between the vehicles, the host car must also maintain a lateral acceleration as to not skid off the road. Interactions between the lateral and longitudinal dynamics, which can be characterized as an interaction structure, contribute to the lateral acceleration. The lateral dynamics (5.2) contain the continuous states $\{q_l, p_l\} \in X_l \subseteq \mathbb{R}^4$, initial states $X_{l0} \subseteq X_l$, input T_l, and disturbance δ_{wy}. The LKC system (5.8) has continuous states $x_b \in X_b \subseteq \mathbb{R}$ and initial states X_{b0}.

In order to derive the closed-loop system, we define the variables $q = \begin{bmatrix} q_x & q_l \end{bmatrix}^{\mathsf{T}}$, $p = \begin{bmatrix} p_x & p_l \end{bmatrix}^{\mathsf{T}}$, $x = \begin{bmatrix} x_{at} & x_{ab} & x_b \end{bmatrix}^{\mathsf{T}}$, $\delta = \begin{bmatrix} \delta_g & \delta_{wx} & \delta_l \end{bmatrix}^{\mathsf{T}}$, and $\zeta = \begin{bmatrix} \zeta_g & \zeta_{wx} & \zeta_l \end{bmatrix}^{\mathsf{T}}$. The closed-loop system has a Hamiltonian function $\tilde{H}(q,p,z) = H_x + H_l + H_a + H_b$, continuous states $\{q,p,x\} \in \tilde{X}$, initial states $\tilde{X}_0 = \tilde{X}_{p0} \times \tilde{X}_{c0} \times S_a$, discrete transitions $\tilde{\mathbb{T}} \subseteq$

$(\tilde{X} \times S_a) \rightarrow (\tilde{X} \times S_a)$, and disturbances $\delta = \{\delta_g, \delta_{wx}, \delta_{wy}\} \in \Delta_g \times \Delta_{wx} \times \Delta_{wy}$.

$$\begin{cases} \begin{bmatrix} \dot{q} \\ \dot{p} \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ -I & \tilde{J}-\tilde{R} & \tilde{K} \\ 0 & -\tilde{K}^{\mathsf{T}} & -\tilde{Q} \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{H}}{\partial q} \\ \frac{\partial \tilde{H}}{\partial p} \\ \frac{\partial \tilde{H}}{\partial x} \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{L} \\ 0 \end{bmatrix} \delta \\ \zeta = \begin{bmatrix} 0 & \tilde{L} & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{H}}{\partial q} & \frac{\partial \tilde{H}}{\partial p} & \frac{\partial \tilde{H}}{\partial x} \end{bmatrix}^{\mathsf{T}} \end{cases} \tag{6.4}$$

where $\tilde{J}$, $\tilde{L}$, $\tilde{R}$, $\tilde{K}$, and $\tilde{Q}$ are defined as:

$$\tilde{J} = \begin{bmatrix} 0 & \frac{mp_r}{I} - M_c & -l_f M_c \\ -\frac{mp_r}{I} + M_c & 0 & 0 \\ l_f M_c & 0 & 0 \end{bmatrix}, \tilde{L} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{R} = \begin{bmatrix} R_x + s_t k_{td} + s_b k_{bd} & 0 & 0 \\ 0 & \frac{mW_1}{p_x} + k_{sd} & \frac{mW_2}{p_x} + l_f k_{sd} \\ 0 & \frac{mW_2}{p_x} + l_f k_{sd} & \frac{mW_3}{p_x} + l_f^2 k_{sd} \end{bmatrix},$$

$$\tilde{K} = \begin{bmatrix} s_t P & s_b & 0 \\ 0 & 0 & -1 \\ 0 & 0 & -l_f \end{bmatrix}, \tilde{Q} = \begin{bmatrix} s_t k_t & 0 & -J_c \\ 0 & s_b k_b & 0 \\ J_c & 0 & 0 \end{bmatrix}.$$

6.3.1 Skidding Avoidance Safety Problem Statement

The control gains can be selected to stabilize the host vehicle velocity to $V_l + \gamma \frac{(X_r - X_d)V_l}{X_d}$ and the lateral displacement to q_d. However, stability does not imply safety. The unsafe states for the lateral momentum are related to that of the longitudinal momentum because of the interactions between the longitudinal and lateral dynamics. The inputs to the longitudinal dynamics (T_a and T_b) affect the lateral dynamics. Similarly, the input to the lateral

dynamics (T_l) affects the longitudinal dynamics. This introduces an additional safety constraint on the system. In order for the vehicle to operate safely on the road, its lateral acceleration must not exceed a maximum value A_m. If the lateral acceleration exceeds A_m, the vehicle will skid. This lateral acceleration value of the vehicle is affected by the yaw rate and longitudinal velocity of the vehicle. This interaction between lateral and longitudinal motion results in an unsafe region characterized by a set defined as:

$$X_{lu} = \{p_x \in \mathbb{R}, p_r \in \mathbb{R} : p_x p_r \geq m^2 I A_m\}. \tag{6.5}$$

This safety condition indicates that longitudinal and lateral motion are bounded by a hyperbolic relationship. A large longitudinal momentum results in a lower bound for the lateral and yaw momentum, and a large lateral and yaw momentum results in a lower bound for the longitudinal momentum. Using this safety constraint we must verify that the product of longitudinal momentum and yaw rate does not exceed a maximum threshold. Given (6.4) and $\tilde{H}(q,p,z)$, the safety condition for the vehicle dynamics, ACC system, and LKC system states that that all possible trajectories cannot reach the unsafe region described by (6.2) and (6.5).

6.3.2 Safety Analysis of Skidding Avoidance

A road can be divided into segments consisting of four types of road profiles: straight road, decreasing curvature, constant curvature, and increasing curvature. Of the four cases the lateral acceleration safety problem is trivial for the straight road and decreasing curvature cases. A straight road nullifies the unsafe state set X_{lu} and a decreasing road curvature relaxes the safety condition. In order to safely navigate a curved section of the road, the vehicle must avoid the unsafe regions of X_{ku} and X_{lu}. Given a road curvature of ρ, the yaw momentum required is calculated as $p_r = \frac{I p_x}{m}\rho$, which shows the direct relationship between the yaw momentum and the longitudinal momentum. Additionally, the road cur-

vature is related to the vehicle slip angle ω and steering angle θ_s:

$$\rho = \frac{\cos(\omega)\tan(\theta_s)}{l_f + l_r},$$

$$\omega = \arctan(\frac{l_r}{l_f + l_r}\tan(\theta_s)).$$

The lateral momentum depends on the longitudinal momentum, the yaw momentum, and the vehicle slip angle:

$$p_y = p_x \sin(\frac{p_r}{I} + \omega).$$

Given that ω and p_r are directly proportional to ρ, we can represent the state variable p_y as a function directly proportional to p_x and ρ. We need the following definitions for initial states, unsafe states, and guard sets. For each discrete state $s_a \in S_a$, the initial continuous states are defined as $\overline{\text{Init}}(s_a) = \{(q,p,x) \in \tilde{X} : (q,p,x,s_a) \in \tilde{X}_0\}$ and the unsafe continuous states are defined as $\overline{\text{Unsafe}}(s_a) = \{(q,p,x) \in \tilde{X} : (q_x, p_x, p_r) \in X_{ku} \times X_{lu}$. Each transition of discrete states from $s_a \in S_a$ to $s'_a \in S_a$ is associated with the guard set $\overline{\text{Guard}}(s_a, s'_a) = \{(q,p,x),(q,p,x)' \in \tilde{X} : (q,p,x,s_a) \to (q',p',x',s'_a)\}$. Safety analysis of the vehicle dynamics uses $\tilde{\Phi}$ as the canonical coordinate transformation for the momentum variables.

$$\begin{bmatrix} \overline{p}_x \\ \overline{p}_y \\ \overline{p}_r \end{bmatrix} = \begin{bmatrix} \tilde{\Phi}_x(p_x) \\ \tilde{\Phi}_y(p_y) \\ \tilde{\Phi}_r(p_r) \end{bmatrix} = \begin{bmatrix} p_x - m(1+\gamma\frac{X_r - X_d}{X_d})V_l - M_c x_b \\ p_y + k_{si}(q_y - q_d) + M_c(x_{at} + x_{ab}) \\ p_r + k_{si}(q_r - \frac{q_d}{l_f}) + M_c\frac{x_{at}+x_{ab}}{l_f} \end{bmatrix}.$$

We apply Theorem 5 to the composed longitudinal dynamics, lateral dynamics, ACC, and LKC system. Given initial conditions $\overline{\text{Init}}(s_a)$, we derive the energy bound $\tilde{\alpha}$ as a function of the initial host vehicle velocity $V_x(0)$, initial relative distance $X_r(0)$, initial lead vehicle velocity $V_l(0)$, and initial road curvature $\rho(0)$. Consequently, we restate the first condition of Theorem 5 as $\tilde{H}(\tilde{\Phi}^{-1}(\overline{p})) \leq \tilde{\alpha}, \forall (q,p,x) \in \overline{\text{Init}}(s_a)$, where

$$\begin{aligned}\tilde{\alpha} &= m\tfrac{k_{td}+k_{bd}}{2}(V_x(0)-(1+\gamma\tfrac{X_r(0)-hV_l(0)-S_0}{hV_l(0)+S_0})V_l(0))^2 \\ &\quad +\tfrac{m}{2}V_x^2(0)\sin^2(\rho(0)V_x(0)+\omega(0))+\tfrac{I}{2}\rho^2(0)V_x^2(0).\end{aligned}$$

Given the unsafe states $\overline{\text{Unsafe}}(s_a)$, we derive the energy bound $\tilde{\beta}$ as a function of host vehicle velocity V_x, relative distance X_r, lead vehicle velocity V_l, and road curvature ρ. The energy of the transformed Hamiltonian function has a maximum value which indicates that the maximum lateral acceleration has been reached. Consequently, we restate the second condition of Theorem 5 as $\tilde{H}(\tilde{\Phi}^{-1}(\overline{p})) > \tilde{\beta}, \forall (q,p,x) \in \overline{\text{Unsafe}}(s_a)$, where

$$\begin{aligned}\tilde{\beta} &= m\tfrac{k_{td}+k_{bd}}{2}(V_x-(1-\gamma)V_l-\tfrac{M_c}{m}(q_y-q_d))^2 \\ &\quad +\tfrac{m}{2}(V_x\sin(\rho V_x+\omega)+k_{si}(q_y-q_d))^2 \\ &\quad +\tfrac{I}{2}(\rho V_x+k_{si}(q_y-\tfrac{q_d}{l_f}))^2.\end{aligned}$$

Given the disturbances $\{\delta_g, \delta_{wx}, \delta_{wy}\} \in \Delta$, we must guarantee that the system trajectory will never begin in $\overline{\text{Init}}(s_a)$ and end in $\overline{\text{Unsafe}}(s_a)$. Consequently, we restate the third condition of Theorem 5 as

$$\begin{gathered}\zeta_g\delta_g+\zeta_{wx}\delta_{wx}+\zeta_{wy}\delta_{wy} \leq \\ \frac{\partial\tilde{H}(\tilde{\Phi}^{-1}(\overline{p}))}{\partial(q,\overline{p})}^{\mathsf{T}}\frac{\partial\tilde{\Phi}}{\partial p}\tilde{R}(\tilde{\Phi}^{-1}(\overline{p}))\frac{\partial\tilde{\Phi}}{\partial p}^{\mathsf{T}}\frac{\partial\tilde{H}(\tilde{\Phi}^{-1}(\overline{p}))}{\partial(q,\overline{p})}, \\ \forall(q,p,x,\delta_g,\delta_{wx},\delta_{wy}) \in \tilde{X}\times\tilde{\Delta}.\end{gathered}$$

Discrete transitions between throttle and brake control mode must also be taken into account in order to guarantee that the system will not transition into $\overline{\text{Unsafe}}(s_a)$. Consequently, we restate the fourth condition of Theorem 5 as $\tilde{H}(\tilde{\Phi}^{-1}(\overline{p})) \leq \tilde{\alpha}, \forall (q,p,x) \in \overline{\text{Guard}}(s_t,s_b) \cup \overline{\text{Guard}}(s_b,s_t)$. In Section 5.3, the ACC and LKC are designed by selecting control parameters that satisfy these safety conditions.

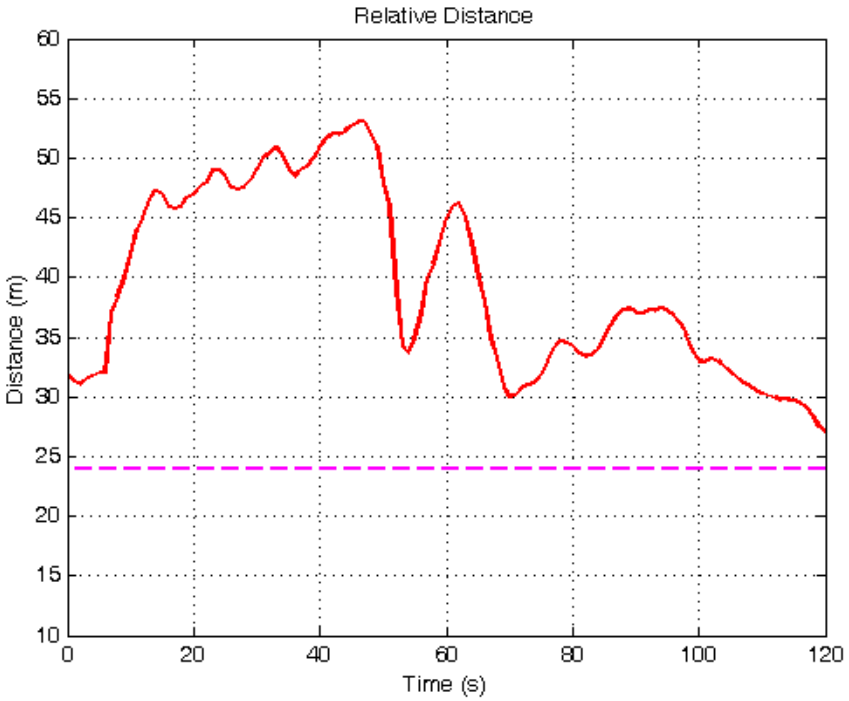

Figure 6.2: Safety of relative distances

6.4 Simulation Results

In this section, we present simulation results to illustrate the approach. We use the same validated PHS model of a standard E-class sedan model as in Section 5. Passivity-based control is used to select the control parameters of the ACC and LKC by considering the total energy of the closed-loop system, specifically considering the dissipation of the system being the difference between the stored energy and the incoming energy [130]. After selecting the control parameter values, we verify that those values obey the safety conditions listed in Section 6.3. Using the experimental passivity index methods and the experimental data of the controllers, we compute the passivity indexes and finite L^2 gain of the controllers as $\mu = 0.8$, $\rho = 5.6$, and $\gamma = 2$. The safety conditions derived in Sections 6.2 and 6.3 are valid for vehicle velocities given a maximum road decline angle of 15 degrees which corresponds to $\delta_g = 4200$ N and a maximum lead vehicle deceleration of 5 m/s^2 which corresponds to a braking distance of 50 m from 80 km/hr to 0 km/hr.

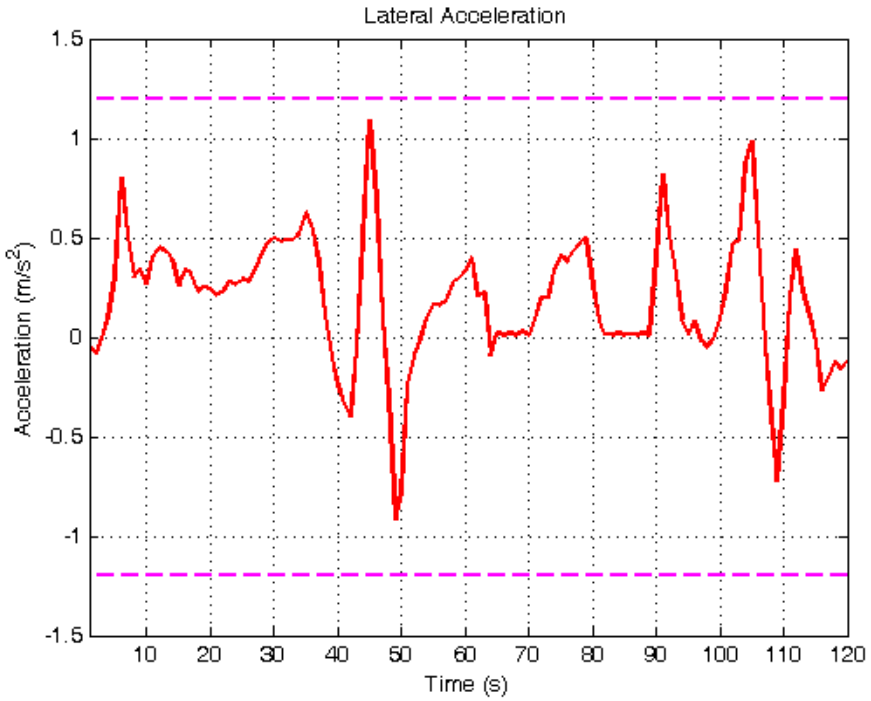

Figure 6.3: Safety of lateral accelerations

Simulation of the closed-loop system consists of two minutes of running time in which the host vehicle follows a lead vehicle on the road featured in Figure 5.9. Figures 6.2 and 6.3 show the full two minutes, where we can see that they are both safe. Figure 6.4 shows the time range of 0 to 5 s, which is a straight segment of the road with a zero degree decline. The simulation results show that the system is safe since the relative distance is greater than $q_m = 24$ m. The curve radius is large because the road is relatively straight, so the lateral acceleration is near zero. Figure 6.5 shows the time range of 46.5 to 51.5 s, which is a curved segment of the road with a zero degree decline. The curve radius during this time period decreases, which corresponds to a non-zero lateral acceleration value. Safety is ensured because the lateral acceleration is bounded by $A_m = 1.2$ m/s^2.

Figure 6.6 shows the time range of 54 to 58 s, which is a straight segment of the road with a fifteen degree decline. The control parameters of the ACC system are designed to compensate for disturbances such as road decline, and the system is safe since the relative

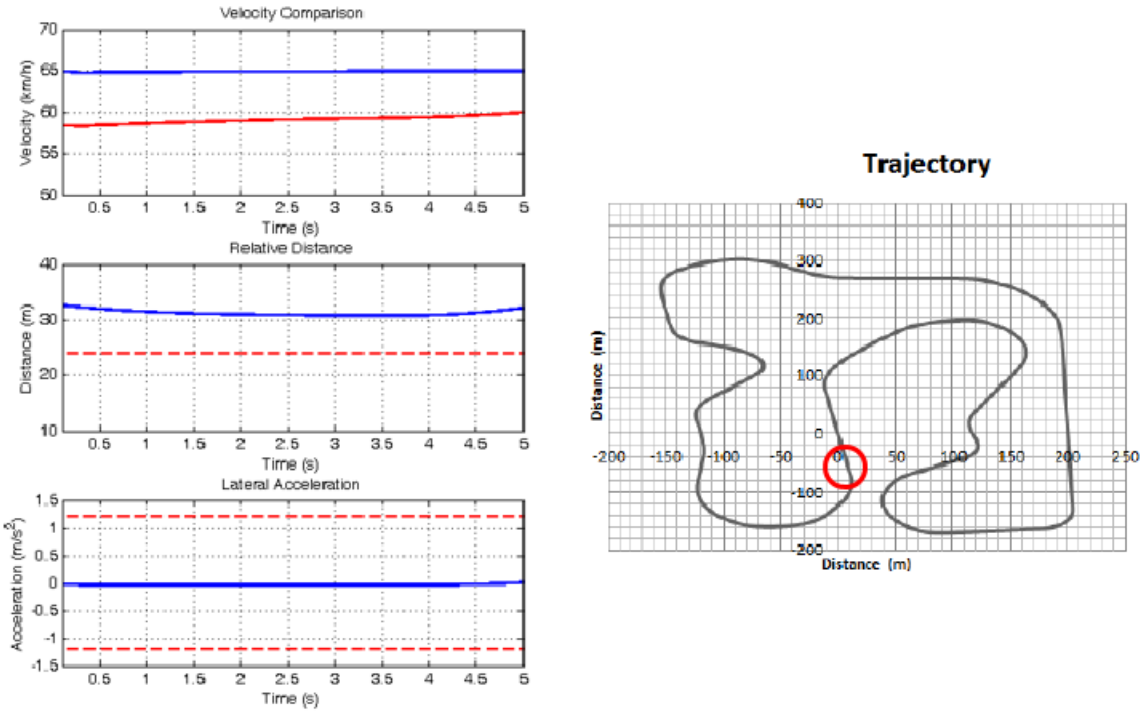

Figure 6.4: Zero degree decline and straight road for continuous-time case

distance is greater than q_m. Similar to the time range of 0 to 5 s, the curve radius is large because the road is relatively straight, so the lateral acceleration is near zero. Figure 6.7 shows the time range of 70 to 75 s, which is a curved segment of the road with a fifteen degree decline. The simulation results show that safety conditions are satisfied.

6.5 Summary

We developed a safety analysis method based on multi-modal PHS which utilizes the Hamiltonian function as a barrier certificate and shows safety by proving that trajectories cannot begin within safe regions and end in unsafe regions. By using the energy levels of the Hamiltonian functions, we can derive conditions on model and control parameters which lead to the system's safe operation. This method is useful and comprehensive because it is able to address the safety problem of PHS given complex interactions, nonlinearities, disturbances, and hybrid dynamics. We presented two case studies using automotive

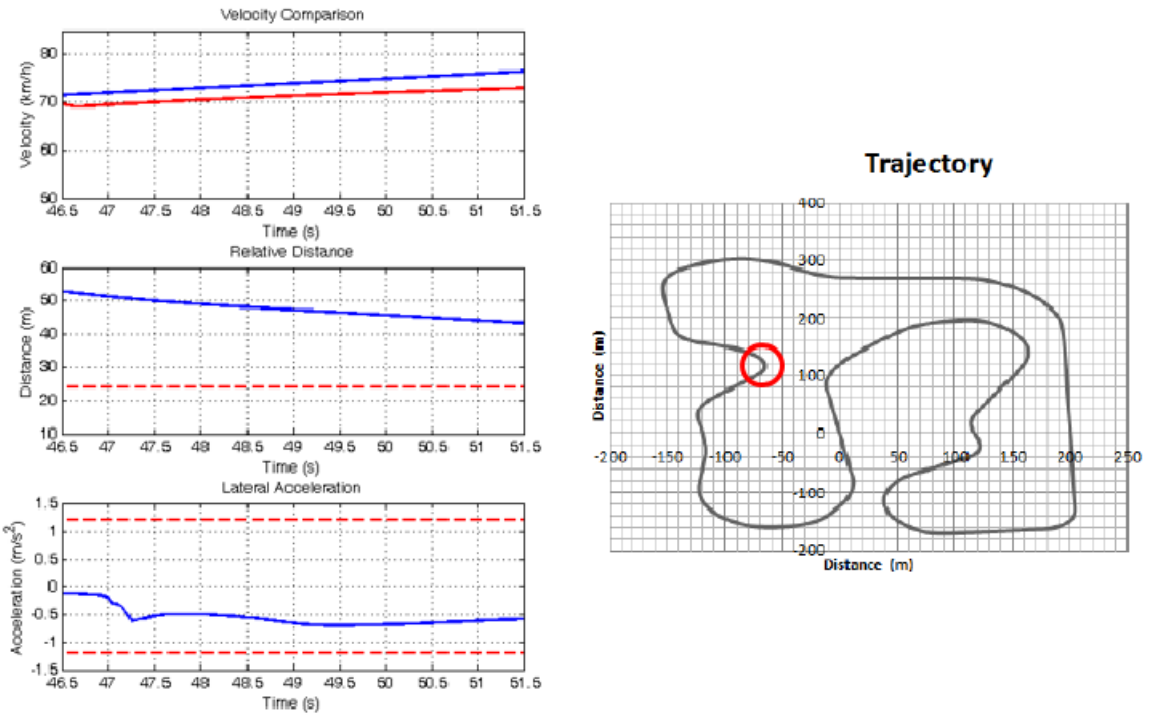

Figure 6.5: Zero degree decline and curved road for continuous-time case

control systems and demonstrate safety conditions which leads to the prevention of unsafe behavior such as collision and skidding. We recorded simulation results which show the effectiveness of the safety analysis approach.

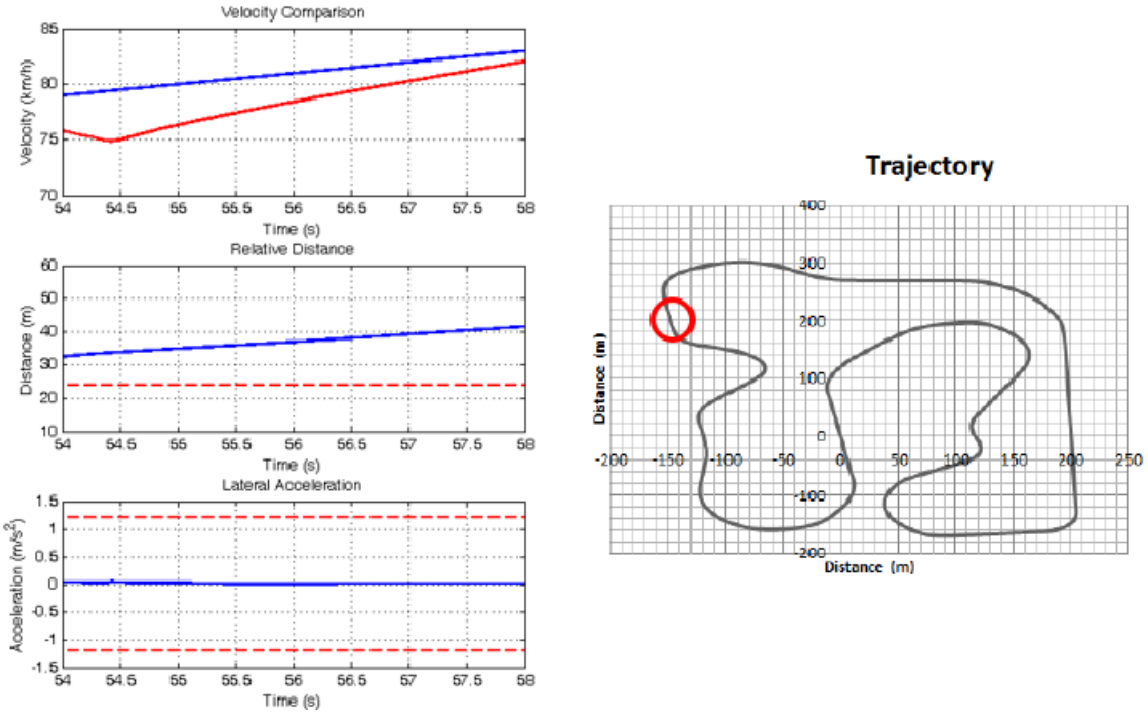

Figure 6.6: Fifteen degree decline and straight road for continuous-time case

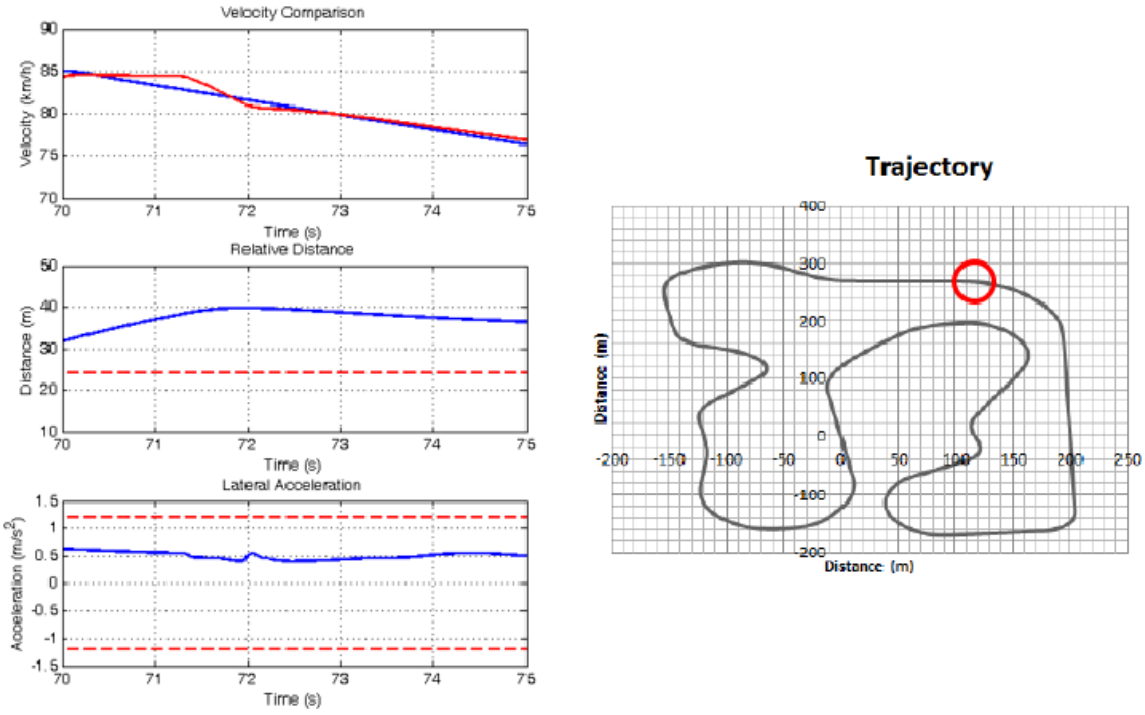

Figure 6.7: Fifteen degree decline and curved road for continuous-time case

Chapter 7

Discrete-Time Safety Analysis of Automotive Control Systems

In Chapter 6 we described a method that verifies the safety of a port-Hamiltonian system (PHS) by using its Hamiltonian function as a barrier. However, an important disadvantage of the safety analysis method is that it is only applicable to PHS described in continuous-time. A well known fact in passivity-based control is that when a passive system is discretized, its passivity is no longer guaranteed. As a result, we need to manually verify that the PHS remains passive in spite of implementation effects such as discretization and quantization in order to guarantee its safety property. The challenge considered in this chapter is to prove the safety of an automotive control system consisting of adaptive cruise control (ACC) and lane keeping control (LKC) despite discretization effects.

The main contribution of this chapter is a alternate approach to the method presented in Chapter 6 which utilizes conservative time-discretization for PHS. The dynamics of the vehicle and the control systems are described using discrete-time PHS models [157]. Conservative time-discretization enables the system to retain passivity even when the system is subjected to discretization. By retaining passivity, the system can be proven that as long as the safe and unsafe energy regions do not overlap, trajectories that begin within a lower energy level (safe states) cannot terminate within a higher energy level (unsafe states). We evaluate the approach by analyzing the safety conditions for a vehicle equipped with an ACC and LKC. We derive discrete-time safety conditions for the ACC and LKC which ensure that the host vehicle does not collide with a lead vehicle or skid off of the road.

7.1 Discrete-Time Safety Analysis Approach

In the event of discretization, certain essential properties of PHS such as Hamiltonian structure and conservation are inevitably lost. Therefore, an appropriate discrete-time

model is required for the discretization of (3.6). We denote the discrete samples of the continuous state variables as $x_k \equiv x(kt_s)$ where k is a non-negative integer and t_s is the sampling period. For space consideration, we introduce the notation of $\hat{x}_k = \frac{x_{k+1}+x_k}{2}$ to denote the sampled mid-point of the k-th sample. We use the method described in [157] called conservative time-discretization to discretize (3.6):

$$\begin{cases} \frac{x_{k+1}-x_k}{t_s} &= [J(\hat{x}_k,s)-R(\hat{x}_k,s)]\frac{\partial H}{\partial x}(\hat{x}_k) - J(\hat{x}_k,s)H^g(k)\frac{x_{k+1}-x_k}{Q(k)} + \begin{bmatrix} L(\hat{x}_k,s) \\ 0 \end{bmatrix} \delta(k) \\ \zeta(k) &= \begin{bmatrix} L^\mathsf{T}(\hat{x}_k,s_p) & 0 \end{bmatrix} \frac{\partial H}{\partial x}(\hat{x}_k) \end{cases} \tag{7.1}$$

$$H^g(k) = H(x_{k+1}) - H(x_k) + \langle \frac{\partial H}{\partial x}(\hat{x}_k), J^g(k) \rangle$$

$$J^g(k) = J^+(\hat{x}_k,s)J(\hat{x}_k,s)(x_{k+1}-x_k)$$

$$Q(k) = (x_{k+1}-x_k)^\mathsf{T} J^g(k)$$

where J^+ is the Moore-Penrose pseudo-inverse matrix of J and $(\delta(k),\zeta(k))$ are the input-output sampled pairs corresponding to the disturbance port. The advantage of the discrete-time representation of (7.1) is that the Hamiltonian structure of the system is preserved and the Dirac structure matrix J can be non-invertible.

7.1.1 Discrete-Time Safety Problem Statement

Figure 7.1 shows a visual representation of the parameter space for the control system being progressively restricted as the system has continuous-time passivity then discrete-time passivity imposed on itself as constraints. The restriction of parameter space implies that there are similar restrictions on the safety property. Given a hybrid system represented as (7.1) with Hamiltonian function H and bounded disturbances, the safety problem is to show that there are no trajectories of the closed-loop system that reach an unsafe region of

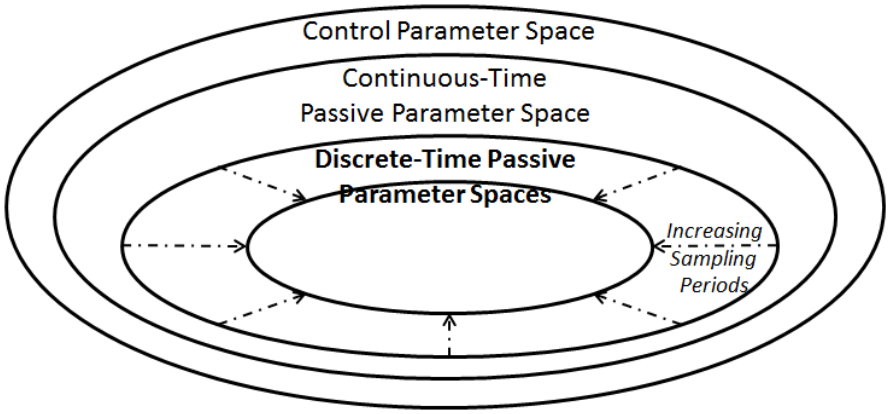

Figure 7.1: Constraints on the control parameter space

the state space.

Definition 31. *Given a discrete-time multi-modal PHS (3.6) and $H(x_k)$ with sampled states $x_k \in X$, initial states $x_0 \in X_0 \subseteq X$, unsafe state space $X_u \subseteq X$, and sampled disturbances $\Delta(k)$, a system trajectory, where k is an integer, $\Gamma(x(kt_s), s(kt_s)) : [0, Nt_s] \rightarrow X$ is unsafe if there exists a positive sample N and a finite sequence of discrete transition samples $0 \leq k_1 t_s \leq \cdots \leq k_{N-1} t_s \leq Nt_s$ such that $\Gamma(x_0) \in X_0$ and $\Gamma(x_k) \in X_u$. The system is safe if there are no unsafe state trajectories.*

A canonical coordinate transform Φ is needed to convert the continuous dynamic equations and Hamiltonian function of the system into a form which shows the actual minimum energy [133]. The traditional canonical coordinate transform method is defined for continuous-time PHS and is used for analyzing the dynamics of physical systems. However, canonical coordinate transform is not defined for discrete-time systems, which necessitates the application of coordinate transform to discrete-time PHS. We consider a coordinate transformation $\bar{x} = \Phi(x)$ for a continuous-time PHS described by (3.6). The dynamic equations of the system is described in the new coordinate system $\bar{x}$:

$$
\begin{aligned}
\dot{\bar{x}} &= \frac{\partial \Phi}{\partial x}\frac{dx}{dt} \\
&= \frac{\partial \Phi}{\partial x}^{\mathsf{T}}\{[J(x,s)-R(x,s)]\frac{\partial H(x)}{\partial x}+^{\mathsf{T}} L(x,s)\delta\} \\
&= \{\frac{\partial \Phi}{\partial x}^{\mathsf{T}}[J(x,s)-R(x,s)]\frac{\partial \Phi}{\partial x}\}\frac{\partial H(\Phi^{-1}(\bar{x}))}{\partial \bar{x}}+\frac{\partial \Phi}{\partial x}^{\mathsf{T}} L(x,s)\delta
\end{aligned} \tag{7.2}
$$

Defining $\overline{J}(\bar{x},s)=\frac{\partial \Phi}{\partial x}^{\mathsf{T}} J(x,s)\frac{\partial \Phi}{\partial x}$, $\overline{R}(\bar{x},s)\frac{\partial \Phi}{\partial x}^{\mathsf{T}} R(x,s)\frac{\partial \Phi}{\partial x}$, and $\overline{L}(\bar{x},s)=\frac{\partial \Phi}{\partial x}^{\mathsf{T}} L(x,s)$, we discretize the results of (7.2):

$$
\begin{cases}
\frac{\bar{x}_{k+1}-\bar{x}_k}{t_s} &= [\overline{J}(\hat{\bar{x}}_k,s))-\overline{R}(\hat{\bar{x}}_k,s)]\frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k)-\overline{J}(\hat{\bar{x}}_k,s))\overline{H}^g(k)\frac{\bar{x}_{k+1}-\bar{x}_k}{\overline{Q}(k)}+\begin{bmatrix}\overline{L}(\hat{\bar{x}}_k,s)\\ 0\end{bmatrix}\delta(k) \\
\zeta(k) &= \begin{bmatrix}\overline{L}^{\mathsf{T}}(\hat{\bar{x}}_k,s) & 0\end{bmatrix}\frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k)
\end{cases} \tag{7.3}
$$

$$
\overline{H}^g(k)=H(\Phi^{-1}(\bar{x}_{k+1}))-H(\Phi^{-1}(\bar{x}_k))+\langle\frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k),\overline{J}^g(k)\rangle
$$

$$
\overline{J}^g(k)=\overline{J}^+(\bar{x}_k,s)\overline{J}(\bar{x}_k,s)(x_{k+1}-x_k)
$$

$$
\overline{Q}(k)=(x_{k+1}-x_k)^{\mathsf{T}}\overline{J}^g(k)
$$

7.1.2 Safety Analysis of Discrete-Time Multi-Modal Port-Hamiltonian System

We consider the following definitions for initial states, unsafe states, and guard conditions that specify discrete mode transitions. For each discrete mode $s \in S$, the initial states are defined as $\text{Init}(s)=\{x_k \in X : (x_k,s) \in X_0 \times S_0\}$ and the unsafe states are defined as $\text{Unsafe}(s)=\{x_k \in X : (x_k,s) \in X_u \times S_u\}$. Each transition of discrete modes from $s \in S$ to $s' \in S$ is associated with the guard condition $\text{Guard}(s,s')=\{x_k,x_k' \in X : \{x,s\} \to \{x',s'\} \in \mathbb{T}\}$. Similar to the safety analysis using continuous-time Hamiltonian functions in Section 6.1, the method in this paper shows that trajectories beginning from the safe region cannot reach the unsafe region.

Theorem 6. *A multi-modal PHS described by (7.3) and $H(\Phi^{-1}(\bar{x}_k))$, with states $x_k \in X$,*

initial states Init(s), unsafe states Unsafe(s), and bounded disturbances $\delta(k) \in \Delta(k)$ is safe if the transformed Hamiltonian function $H(\Phi^{-1}(\bar{x}_k))$ satisfy the following four conditions with $\alpha \leq \beta$

1. $H(\Phi^{-1}(\bar{x}_k)) \leq \alpha, \forall x \in \text{Init}(s)$
2. $H(\Phi^{-1}(\bar{x}_k)) > \beta, \forall x \in \text{Unsafe}(s)$
3. $\zeta(k)^{\mathsf{T}}\delta(k) \leq \frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}^{\mathsf{T}} \overline{R}(\bar{x}_k, s) \frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}, \forall \{x_k, \delta\} \in X \times \Delta$
4. $H(\Phi^{-1}(\bar{x}_k)) \leq \alpha, \forall x_k \in \text{Guard}(s, s')$

Proof. Assuming that the Hamiltonian function $H(\Phi^{-1}(\bar{x}_k))$ satisfy the four conditions in Theorem 5, yet there exists a time $T \geq 0$, an input δ, and initial states Init(s), and a trajectory $\Gamma(x_0))$ such that $\Gamma(x_T) \in \text{Unsafe}(s)$. We show that the Hamiltonian function cannot simultaneously satisfy the four condition and reach the unsafe region, thus proving safety by contradiction. The time difference of the Hamiltonian functions can be written as:

$$\begin{aligned}
\frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k)^{\mathsf{T}} \frac{\bar{x}_{k+1} - \bar{x}_k}{t_s} &= \frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k)^{\mathsf{T}} [\overline{J}(\hat{\bar{x}}_k, s) - \overline{R}(\hat{\bar{x}}_k, s)] \frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k) \\
&\quad - \frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k)^{\mathsf{T}} \overline{J}(\hat{\bar{x}}_k, s) \overline{H}^g(k) \frac{\bar{x}_{k+1} - \bar{x}_k}{\overline{Q}(k)} \\
&\quad + \frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k)^{\mathsf{T}} \begin{bmatrix} \overline{L}(\hat{\bar{x}}_k, s) \\ 0 \end{bmatrix} \delta(k)
\end{aligned}$$

The important part of the proof is showing that the interaction structure $\overline{J}$ and the conservative Hamiltonian structure $\overline{J}(\hat{\bar{x}}_k, s))\overline{H}^g(k) \frac{\bar{x}_{k+1} - \bar{x}_k}{\overline{Q}(k)}$ contribute zero energy. As a result of the skew symmetric nature of $\overline{J}$ and the third term of the conservative Hamiltonian structure, we can conclude that:

$$\frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k)^{\mathsf{T}} \overline{J}(\hat{\bar{x}}_k, s) \frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k) = 0$$

$$\frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k)^{\mathsf{T}}\bar{J}(\hat{\bar{x}}_k,s)\langle\frac{\partial H(\Phi^{-1}(\bar{x}_k))}{\partial \bar{x}_k}(\hat{\bar{x}}_k),\bar{J}^g(k)\rangle\frac{\bar{x}_{k+1}-\bar{x}_k}{\bar{Q}(k)}=0$$

The sum of the remaining two terms of the conservative Hamiltonian structure can be simplified to

$$\frac{H(\Phi^{-1}(\bar{x}_{k+1}))-H(\Phi^{-1}(\bar{x}_k))}{t_s}^{\mathsf{T}}\bar{J}^{+}(\hat{\bar{x}}_k,s)\frac{H(\Phi^{-1}(\bar{x}_{k+1}))-H(\Phi^{-1}(\bar{x}_k))}{t_s}$$

Given that the matrix $\bar{J}^{+}$ is skew-symmetric, we can conclude that this term is also equal to zero. Condition (3) shows that the system trajectory on the interval of $[0,T]$ is non-increasing, which indicates that $H(x_T)\leq H(x_0)$. Additionally, condition (4) asserts that during a discrete transition, the Hamiltonian function will not jump to an increasing value. These statements, however, contradict the original assumption that the system states start at Init(s) and end at Unsafe(s). As a result, we can conclude that the system is safe. □

7.2 Discrete-Time Collision and Skidding Avoidance

According to Theorem 6, the discrete-time safety problem is the same as the continuous-time safety problem if the system's interconnection matrix, $J(x)$, is zero. In both control system that we designed, we included zero interconnection matrices, so the safety problem when applied to the discrete-time version of the system should be the same as the continuous-time case, which we described in detail in Chapter 6. This problem thus becomes trivial when we only account for safety problems when each system is by itself. On the other hand, we included a non-zero interconnection matrix in the composed system design, so the difference in safety problem applies in that case. In order to showcase the updated discrete-time design, we consider the safety problem of a vehicle with both an ACC and a LKC system following a lead car. The longitudinal dynamics has the following Hamiltonian function:

$$H_x(q_{x,k}, p_{x,k}) = \frac{1}{2m} p_{x,k}^2 + U_x(q_{x,k}),$$

where $q_{x,k}$ represents the sampled longitudinal displacement, $p_{x,k}$ represents the sampled longitudinal momentum, and $U_x(q_{x,k})$ represents the sampled longitudinal potential energy. The longitudinal dynamics is modeled in the form of (7.3):

$$\begin{cases} \begin{bmatrix} \frac{q_{x,k+1}-q_{x,k}}{t_s} \\ \frac{p_{x,k+1}-p_{x,k}}{t_s} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -R_x \end{bmatrix} \begin{bmatrix} \frac{\partial H_x}{\partial q_{x,k}} \\ \frac{\partial H_x}{\partial p_{x,k}} \end{bmatrix} + Q_x(p_{x,k}) + \begin{bmatrix} 0 \\ G_x \end{bmatrix} u_x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} d_x \\ y_x = \begin{bmatrix} 0 & G_x^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \frac{\partial H_x}{\partial q_{x,k}} & \frac{\partial H_x}{\partial p_{x,k}} \end{bmatrix}^{\mathsf{T}} \\ z_x = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial H_x}{\partial q_{x,k}} & \frac{\partial H_x}{\partial p_{x,k}} \end{bmatrix}^{\mathsf{T}} \end{cases} \tag{7.4}$$

$$Q_x(p_{x,k}) = \begin{bmatrix} 0 & \frac{1}{2m} p_{x,k+1}^2 - \frac{1}{2m} p_{x,k}^2 \\ \frac{1}{2m} p_{x,k}^2 - \frac{1}{2m} p_{x,k+1}^2 & 0 \end{bmatrix}$$

where $u_x = \begin{bmatrix} T_a & T_b \end{bmatrix}^{\mathsf{T}}$, $G_x = \begin{bmatrix} 1 & -1 \end{bmatrix}$, and $R_x = am + bp_{x,k} + \frac{cm^2}{p_{x,k}}$.

The lateral dynamics has the following Hamiltonian function:

$$H_l(q_{y,k}, q_{r,k}, p_{y,k}, p_{r,k}) = \frac{1}{2m} p_{y,k}^2 + \frac{1}{2I} p_{r,k}^2 + U_l(q_{y,k}, q_{r,k}),$$

where $q_{y,k}$ represents the sampled lateral displacement, $p_{y,k}$ represents the sampled lateral momentum, $q_{r,k}$ represents the sampled yaw displacement, $p_{r,k}$ represents the sampled yaw momentum, and $U_l(q_{y,k}, q_{r,k})$ represents the sampled lateral potential energy. The lateral dynamics is modeled in the form of (7.3):

$$
\begin{cases}
\begin{bmatrix} \frac{q_{l,k+1}-q_{l,k}}{t_s} \\ \frac{p_{l,k+1}-p_{l,k}}{t_s} \end{bmatrix} & = \begin{bmatrix} 0 & E \\ -E & -R_l \end{bmatrix} \begin{bmatrix} \frac{\partial H_l}{\partial q_{l,k}} \\ \frac{\partial H_l}{\partial p_{l,k}} \end{bmatrix} + Q_l(p_{l,k}) + \begin{bmatrix} 0 \\ G_l \end{bmatrix} T_l + \begin{bmatrix} 0 \\ K_l \end{bmatrix} d_l \\
y_l & = \begin{bmatrix} 0 & G_l^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \frac{\partial H_l}{\partial q_{l,k}} & \frac{\partial H_l}{\partial p_{l,k}} \end{bmatrix}^{\mathsf{T}} \\
z_l & = \begin{bmatrix} 0 & K_l^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \frac{\partial H_l}{\partial q_{l,k}} & \frac{\partial H_l}{\partial p_{l,k}} \end{bmatrix}^{\mathsf{T}},
\end{cases} \tag{7.5}
$$

$$
R_l = \begin{bmatrix} \frac{W_1}{V_x} & \frac{W_2}{V_x} \\ \frac{W_2}{V_x} & \frac{W_3}{V_x} \end{bmatrix},
$$

$$
Q_l(p_{l,k}) = \begin{bmatrix} 0 & 0 & \frac{1}{2m}p_{y,k+1}^2 - \frac{1}{2m}p_{y,k}^2 & 0 \\ 0 & 0 & 0 & \frac{1}{2I}p_{r,k+1}^2 - \frac{1}{2I}p_{r,k}^2 \\ \frac{1}{2m}p_{y,k}^2 - \frac{1}{2m}p_{y,k+1}^2 & 0 & 0 & 0 \\ 0 & \frac{1}{2I}p_{r,k}^2 - \frac{1}{2I}p_{r,k+1}^2 & 0 & 0 \end{bmatrix}
$$

where E is the identity matrix, $G_l = \begin{bmatrix} 1 & l_f \end{bmatrix}^{\mathsf{T}}$ and $K_l = \begin{bmatrix} 1 & 0 \end{bmatrix}^{\mathsf{T}}$. The parameter constants of R_l are $W_1 = 2C_f + 2C_r$, $W_2 = 2C_f l_f - 2C_r l_r$, and $W_3 = 2C_f l_f^2 + 2C_r l_r^2$, where C_r is the cornering stiffness of the rear wheels. Composition of the longitudinal and lateral dynamics occurs through the interaction structure of (7.6) modulated by the sampled angular momentum $p_{r,k}$:

$$
\begin{bmatrix} d_x \\ d_l \end{bmatrix} = \begin{bmatrix} 0 & -\frac{mp_{r,k}}{I} \\ -\frac{mp_{r,k}}{I} & 0 \end{bmatrix} \begin{bmatrix} z_x \\ z_l \end{bmatrix}. \tag{7.6}
$$

The Hamiltonian function of the composed longitudinal and lateral dynamics is $H(q_k, p_k) = H_x(q_{x,k}, p_{x,k}) + H_l(q_{y,k}, q_{r,k}, p_{y,k}, p_{r,k})$. Composition of (7.4) and (7.5) through (7.6) results in the following nonlinear PHS:

$$\begin{cases} \begin{bmatrix} \frac{q_{k+1}-q_k}{t_s} \\ \frac{p_{k+1}-p_k}{t_s} \end{bmatrix} & = \begin{bmatrix} 0 & E \\ -E & -R \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q_k} \\ \frac{\partial H}{\partial p_k} \end{bmatrix} + \begin{bmatrix} Q_x(p_{x,k}) & 0 \\ 0 & Q_l(p_{l,k}) \end{bmatrix} + \begin{bmatrix} 0 \\ G \end{bmatrix} u \\ y & = \begin{bmatrix} 0 & G^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \frac{\partial H}{\partial q_k} \\ \frac{\partial H}{\partial p_k} \end{bmatrix}, \end{cases} \tag{7.7}$$

$$R = \begin{bmatrix} R_x & \frac{mp_{r,k}}{I} & 0 \\ \frac{mp_{r,k}}{I} & \frac{mW_1}{p_{x_k}} & \frac{mW_2}{p_{x_k}} \\ 0 & \frac{mW_2}{p_{x_k}} & \frac{mW_3}{p_{x_k}} \end{bmatrix}, G = \begin{bmatrix} G_x & 0 \\ 0 & G_l \end{bmatrix},$$

where $q_k = \begin{bmatrix} q_{x,k} & q_{l,k} \end{bmatrix}^{\mathsf{T}}$, $p_k = \begin{bmatrix} p_{x,k} & p_{l,k} \end{bmatrix}^{\mathsf{T}}$, $u = \begin{bmatrix} u_x & T_l \end{bmatrix}^{\mathsf{T}}$, and $y = \begin{bmatrix} y_x & y_l \end{bmatrix}^{\mathsf{T}}$. The speed control system has the following Hamiltonian function:

$$H_a(x_{a,k}) = \frac{1}{2}(s_t k_{ti} x_{at,k}^2 + s_b k_{bi} x_{ab,k}^2),$$

with the following discrete PHS representation:

$$\begin{cases} \frac{x_{a,k+1}-x_{a,k}}{t_s} & = -R_a \frac{\partial H_a}{\partial x_{a,k}} + G_a y_x + K_{a1} d_{a1} \\ u_x & = G_a^{\mathsf{T}} \frac{\partial H_a}{\partial x_{a,k}} + S_a y_x + K_{a2} d_{a2} \\ \begin{bmatrix} z_{a1} \\ z_{a2} \end{bmatrix} & = \begin{bmatrix} K_{a1}^{\mathsf{T}} & 0 \\ 0 & K_{a2}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \frac{\partial H_a}{\partial x_{a,k}} \\ y_x \end{bmatrix}. \end{cases} \tag{7.8}$$

The parameter matrices are the same as the ones for (5.7). The steering control system has the following Hamiltonian function:

$$H_b(x_{b,k}) = \frac{1}{2} k_{si} x_{b,k}^2,$$

with the following discrete PHS representation:

$$\begin{cases} \frac{x_{b,k+1}-x_{b,k}}{t_s} & = y_l + d_{b1} \\ T_l & = \frac{\partial H_b}{\partial x_{b,k}} + k_{sd} y_l + d_{b2} \\ \begin{bmatrix} z_{b1} \\ z_{b2} \end{bmatrix} & = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial H_b}{\partial x_{b,k}} \\ y_l \end{bmatrix}. \end{cases} \tag{7.9}$$

Composition of the controllers via (5.9) introduces an interaction structure, which will also introduce an additional term in the equations for the composed control systems. The Hamiltonian function of the composed control system is denoted as $H_c = H_a(x_{a,k}) + H_b(x_{b,k})$. Composition of (7.8) and (7.9) through (5.9) results in the following PHS representation:

$$\begin{cases} \frac{x_{k+1}-x_k}{t_s} & = \begin{bmatrix} -R_a & J_s \\ -J_s^\mathsf{T} & 0 \end{bmatrix} \frac{\partial H_c}{\partial x_k} + Q_c(x_{at,k}, x_{ab,k}, x_{b,k}) + \begin{bmatrix} G_a & 0 \\ 0 & 1 \end{bmatrix} y \\ u & = \begin{bmatrix} G_a^\mathsf{T} & 0 \\ 0 & 1 \end{bmatrix} \frac{\partial H_c}{\partial x_k} + \begin{bmatrix} S_a & M_s \\ -M_s^\mathsf{T} & k_{sd} \end{bmatrix} y, \end{cases} \tag{7.10}$$

where $J_s = \begin{bmatrix} J_c & 0 \end{bmatrix}^\mathsf{T}$, $M_s = \begin{bmatrix} M_c & 0 \end{bmatrix}^\mathsf{T}$, and $x = \begin{bmatrix} x_{at,k} & x_{ab,k} & x_{b,k} \end{bmatrix}^\mathsf{T}$. The closed-loop system has a Hamiltonian function $\tilde{H}(q_k, p_k, z_k) = H_x + H_l + H_a + H_b$, sampled continuous states $\{q_k, p_k, x_k\} \in \tilde{X}$, initial states $\tilde{X}_0 = \tilde{X}_{p0} \times \tilde{X}_{c0} \times S_a$, discrete transitions $\tilde{\mathbb{T}} \subseteq (\tilde{X} \times S_a) \to (\tilde{X} \times S_a)$, and disturbances $\delta = \{\delta_g, \delta_{wx}, \delta_{wy}\} \in \Delta_g \times \Delta_{wx} \times \Delta_{wy}$.

$$\begin{cases} \begin{bmatrix} \frac{q_{k+1}-q_k}{t_s} \\ \frac{p_{k+1}-p_k}{t_s} \\ \frac{x_{k+1}-x_k}{t_s} \end{bmatrix} & = \begin{bmatrix} 0 & I & 0 \\ -I & \tilde{J}-\tilde{R} & \tilde{K} \\ 0 & -\tilde{K}^\mathsf{T} & -\tilde{Q} \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{H}}{\partial q_k} \\ \frac{\partial \tilde{H}}{\partial p_k} \\ \frac{\partial \tilde{H}}{\partial x_k} \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{L} \\ 0 \end{bmatrix} \delta \\ \zeta & = \begin{bmatrix} 0 & \tilde{L} & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{H}}{\partial q_k} & \frac{\partial \tilde{H}}{\partial p_k} & \frac{\partial \tilde{H}}{\partial x_k} \end{bmatrix}^\mathsf{T} \end{cases} \tag{7.11}$$

where $\tilde{J}$, $\tilde{L}$, $\tilde{R}$, $\tilde{K}$, and $\tilde{Q}$ are defined as:

$$\tilde{J} = \begin{bmatrix} 0 & \frac{mp_r}{I} - M_c & -l_f M_c \\ -\frac{mp_r}{I} + M_c & 0 & 0 \\ l_f M_c & 0 & 0 \end{bmatrix}, \tilde{L} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\tilde{R} = \begin{bmatrix} R_x + s_t k_{td} + s_b k_{bd} & 0 & 0 \\ 0 & \frac{mW_1}{p_x} + k_{sd} & \frac{mW_2}{p_x} + l_f k_{sd} \\ 0 & \frac{mW_2}{p_x} + l_f k_{sd} & \frac{mW_3}{p_x} + l_f^2 k_{sd} \end{bmatrix},$$

$$\tilde{K} = \begin{bmatrix} s_t P & s_b & 0 \\ 0 & 0 & -1 \\ 0 & 0 & -l_f \end{bmatrix}, \tilde{Q} = \begin{bmatrix} s_t k_t & 0 & -J_c \\ 0 & s_b k_b & 0 \\ J_c & 0 & 0 \end{bmatrix}.$$

Unsafe regions of state space are described by (7.12) and (7.13)

$$X_{ku} = \left\{ q_{x,k} \in \mathbb{R} : q_{x,k} \geq \sum_{i=0}^{k} t_s V_{l,i} + q_l(0) + q_m \right\}, \tag{7.12}$$

$$X_{lu} = \{ p_{x_k} \in \mathbb{R}, p_{r_k} \in \mathbb{R} : p_x p_r \geq m^2 I A_m \}. \tag{7.13}$$

Using the boundary conditions derived for the continuous time case and the assumptions, as shown in Chapter 6, we can prove that the discrete-time representation of (7.11) is guaranteed to be safe, using the following coordinate transform:

$$\begin{bmatrix} \overline{p}_{x,k} \\ \overline{p}_{y,k} \\ \overline{p}_{r,k} \end{bmatrix} = \begin{bmatrix} \tilde{\Phi}_{x,k}(p_{x,k}) \\ \tilde{\Phi}_{y,k}(p_{y,k}) \\ \tilde{\Phi}_{r,k}(p_{r,k}) \end{bmatrix} = \begin{bmatrix} p_{x,k} - m(1 + \gamma \frac{X_r - X_d}{X_d}) V_l - M_c x_{b,k} \\ p_{y,k} + k_{si}(q_{y,k} - q_d) + M_c(x_{at,k} + x_{ab,k}) \\ p_{r,k} + k_{si}(q_{r,k} - \frac{q_d}{l_f}) + M_c \frac{x_{at,k} + x_{ab,k}}{l_f} \end{bmatrix}.$$

With the coordinate transform defined, we can rewrite condition 3 as:

$$\zeta_g \delta_g + \zeta_{wx} \delta_{wx} + \zeta_{wy} \delta_{wy} \leq \frac{\partial \tilde{H}(\tilde{\Phi}_k^{-1}(\overline{p}_k))}{\partial (q_k, \overline{p}_k)}^{\mathsf{T}} \frac{\partial \tilde{\Phi}_k}{\partial p_k} \tilde{R}(\tilde{\Phi}_k^{-1}(\overline{p}_k)) \frac{\partial \tilde{\Phi}_k}{\partial p_k}^{\mathsf{T}} \frac{\partial \tilde{H}(\tilde{\Phi}_k^{-1}(\overline{p}_k))}{\partial (q_k, \overline{p}_k)}.$$

7.3 Summary

The motivating issue is that passivity degrades during discretization, which invalidates the safety analysis method from continuous-time when discretization is needed. As a result, we developed a safety analysis method based on conservative time-discretized multi-modal PHS which utilizes the Hamiltonian function as a barrier certificate; the work in this chapter is a direct extension to the work in Chapter 6. This method is useful because it allows the analysis method from Chapter 6 to be used because passivity does not degrade in conservative time-discretization. We presented a case study in which the discrete-time safety of an automotive control systems shows that unsafe behaviors such as collision and skidding are avoided.

Chapter 8

Evaluation and Validation Using Hardware-in-the-Loop Platform

The main contribution of this chapter is the generation of simulation results for the validation and evaluation of the continuous-time and discrete-time port-Hamiltonian systems (PHS) designed in Chapters 5 and 7. In order to implement the controllers on the HIL platform, implementation effects such as discretization and quantization are analyzed to ensure that passivity is retained. Code is generated from the discretized and quantized PHS models, analyzed for timing and scheduling, and deployed on the hardware platform. The HIL simulation platform consisting of the vehicle dynamics using a CarSim deployment on a Real Time(RT)-Target PC, the time-triggered communication network using a TTTech PCIe-XMC card, and the controllers using IBX-530W box with an Intel Atom processor running a RT-Linux operating system. This work has been partially presented in [149].

8.1 Hardware-in-the-Loop Platform

Figure 8.1 shows a high level diagram of the HIL simulation platform used [8]. The physical dynamics modeled in CarSim is deployed as a RT-Target so that it acts as a real vehicle. The RT-target is also integrated with a TTTech PCIe-XMC card which enables the seamless integration and communication with the ECUs on a time-triggered network. The HIL simulator has three ECUs which are connected to an 8-port 100Mbps TTEthernet development switch from TTTech [158]. Each ECU is an IBX-530W box with an Intel Atom processor running a RT-Linux operating system and is integrated with a TTEthernet Linux driver, which is a software-based implementation of the TTEthernet protocol in order to enable communication with the other systems in a TTEthernet network. The automotive control software is distributed over the ECUs and the tasks execute in the kernel

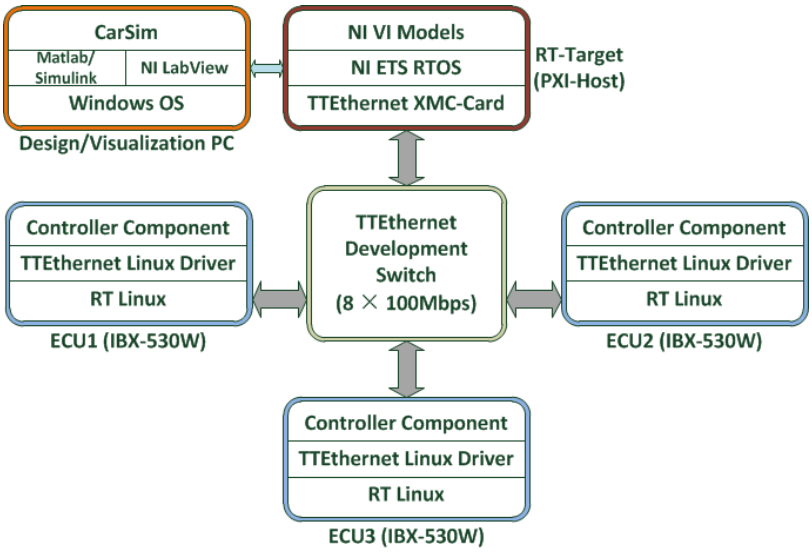

Figure 8.1: HIL simulator architecture [8]

space of RT-Linux which can utilize the synchronized time base off of the TTEthernet communication. The controllers are deployed on ECU1 and communicate with the RT-Target via the TTEthernet network which provides a synchronized time base for computation and communication.

8.2 Implementation Effects

After verifying the correctness of the control design and vehicle dynamics, we discretize, quantize, and account for other implementation effects while preserving passivity in preparation for deployment [8]. Although the continuous-time control systems are useful for early stages of design and validation, it is not feasible for the HIL simulation platform [159]. Given a continuous-time controller PHS, processes such as discretization and quantization are required to generate the C code for deployment on the ECUs in the platform. The objective of discretization is to convert the controllers from continuous-time PHS to discrete-time PHS [160]. The objective of quantization is to convert the floating-point discrete-time PHS into fixed-point discrete-time PHS.

8.2.1 Discretization

Prior to discretization, the continuous-time PHS is represented using block diagrams in a continuous-time Simulink model. Transformation of a continuous-time Simulink model into a discrete-time Simulink model is a procedure involving bi-linear transformations, up-samplers, and down-samplers. State variables and subsequent computations inside the controllers are linked together through delays and adders [161] [162]. We discretize the PHS controllers using sampling periods of 10 ms, 30 ms, and 50 ms. We employ discrete-time passivity in order to ensure that the closed-loop system will remain passive.

Discretization does not affect the high level system model of the vehicle dynamics interacting with the controllers [163]. However, passivity is a property that degrades under discretization [94] [164]. Intuitively, the larger the sampling period, the greater the degradation. The standard discretization relates the discrete-time input $u_d(k)$ to the continuous time input $u(t)$ using zero-order holds in that the continuous time t is bounded by kt_s and $(k+1)t_s$ where k is a non-negative integer and t_s is the sampling period [165] [166]. The discrete-time output $y_d(k)$ relates to the continuous time output $y(t)$ by a sampler $y_d(k) = y(kt_s)$ A crucial information here is that even if the original continuous time system is a passive PHS, its discretization is not necessarily passive [167] [168]. To circumvent this problem, [169] developed a different discretization approach in which the discrete-time output is modified as

$$y_d(k) = \frac{1}{t_s} \int_{kt_s}^{(k+1)t_s} y(t)dt.$$

This different discretization approach is interesting because it guarantees that the resulting discrete-time system is passive and thus able to be subjected to passivity-based control methods [170]. However, the approach has a shortcoming in that it requires a future output value of $y(t)$ at $(k+1)t_s$ which may not be possible to obtain if the system is highly nonlinear. Given this fact, we discretize (5.10) using the standard method and choose a sampling

period so that the system satisfies the discrete-time passivity inequality:

$$t_s \sum_{k=0}^{N} u_d(k)^{\top} y_d(k) \geq \mu_d t_s \sum_{k=0}^{N} \left\| u_d(k) \right\|^2 + \rho_d t_s \sum_{k=0}^{N} \left\| y_d(k) \right\|^2 \tag{8.1}$$

where N is a positive integer, μ_d is a real number, and ρ_d is a real number. In order to guarantee that the inequality in (8.1) is satisfied, we have to ensure that the sampling period is chosen so that the discrete-time passivity indexes are larger than zero given $\mu_d = \mu - t_s\gamma - t_s\gamma\left|\rho\right| - t_s^2\gamma^2\left|\rho\right|$ and $\rho_d = \rho - t_s\gamma\left|\rho\right|$ [169]. Using the experimental passivity index methods and the experimental data of the controllers from Section 5.2.4, we compute the passivity indexes and finite L^2 gain of the controllers as $\mu = 0.8$, $\rho = 5.6$, and $\gamma = 2$. We find that the discretized system will be passive given a sampling period smaller than $t_s \approx 55$ ms.

8.2.2 Quantization

Quantization is the mathematical process in which a large set of input values is rounded and truncated down to a smaller set, and is needed as a result of hardware limitations on the ECUs, leading to the fact that floating-point data cannot be processed [171]. This procedure involves using the MATLAB data command fixedt() and data conversion blocks between the CarSim S-functions and controllers. State variables and other variables are rounded and truncated into fixed-point data values. The ECUs that we use to implement the control system require either 16 or 32-bit fixed point data types in its operation and computation, which necessitates a concern for passive quantization.

In Simulink the quantization process is done using MATLAB's Fixed-Point Toolbox, in which the word lengths for all data are set as fixdt(1, 16, 8) and fixdt(1, 32, 16) for 16 and 32-bit data types, respectively [41]. Simulink's quantizer is a uniform mid-tread quantizer and is considered to be a passive quantizer, which is a concept introduced in [172] where the input v and output u mappings are bounded by two lines of slopes a and b,

$av^2 \leq uv \leq bv^2$. However, even though the quantizer is passive does not necessarily mean that the quantized system is passive [173]. In order to ensure passivity for the quantized system we implemented the transformation block M from [172] using the values of $m_{11} = 2$, $m_{12} = -0.36$, $m_{21} = 0$, and $m_{22} = 1$, which are computed using the passivity indexes of the controllers.

8.3 Deployment

The deployment stage consists of several steps: code generation, compilation, timing analysis, scheduling, and testing of the HIL simulation. The objective of the code generation is to convert the discretized and quantized Simulink models into C code that will be executed in the HIL platform. This procedure involves the Simulink Coder (previously called Real-Time Workshop) which automatically generates the necessary C code using the the available bits in a word and the value ranges (Q format) [174]. The Simulink Coder generates code with proper computation according to the chosen fixdt. The code generated from the Simulink models is in C, which needs to be compiled in order for deployment on the platform. In this procedure, we use gcc-4.2.4 together with the TTEthernet configuration file generated by the TTTech tool-chain to compile the generated C code, which is linked with the provided TTEthernet driver to become a kernel module. This kernel module does the computation and drives the Ethernet port of the ECU in a time-triggered fashion, namely that there is a static schedule table specifying when to compute and communicate based on a synchronized global time.

The objective of timing analysis is to compute the worst-case execution time (WCET) of the control tasks. In our case, the procedure involves empirically measuring the execution times of the designed control task (i.e. recording the time difference between each start and end of the execution of the task) through multiple runs under multiple circumstances. In our experiment, there are 12,000 measurements recorded, and we can plot them to observe the distribution, as shown in Fig. 8.2 (in order to show the frequencies of the execution

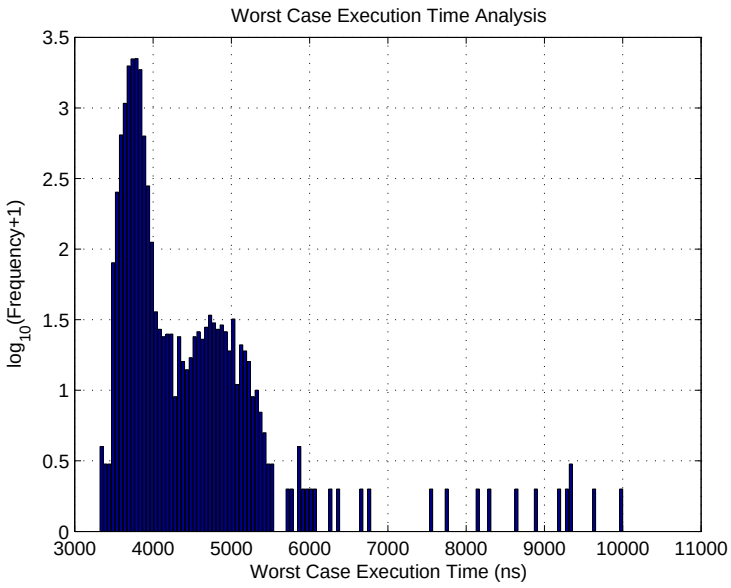

Figure 8.2: Histogram of worst-case execution times

times clearly, we use $\log_{10}(\text{number of times} + 1)$ on the y-axis).

Scheduling is the method by which processes are allocated to resources that complete the processes. The underlying execution model of our testbed platform is a time-triggered architecture (TTA) which maintains a synchronized global time base and requires a static schedule table for both computation and communication. The objective of scheduling is to generate a feasible static schedule for the control tasks and messages. In general, there are many scheduling approaches for TTA combining computation and communication [175]. In our case, we have one control task whose WCET is almost negligible thus needing only two time-triggered messages (one is for sensing and the other is for actuation). The measured WCET reveals that the computation is almost negligible (only about 30 μs) compared to the sampling period (which is on the millisecond level), therefore the schedulability problem becomes trivial in our experiments.

8.4 Evaluation of Continuous-Time Design

The objective of the testing is to verify the correctness of the design and to record simulation results that demonstrate its effectiveness. Our HIL simulation of the closed-loop system consists of the same two minutes of running time as the continuous-time case. We find that the discretized system will be passive given a sampling period smaller than $t_s \approx 65$ ms by using the methods described in Section 7.1. We are able to guarantee that the system will remain safe given the control parameters and the implementation effects. The gain values of the controllers from Table 5.2 are verified to retain passivity given sampling periods of 10 ms, 30 ms, and 50 ms. Information from the trajectory is encoded into the vehicle model provided by CarSim, which is then given to the controllers via the communication network. This procedure involves the implementation of the control systems on the HIL architecture.

Using the constraints provided by passivity, discretization, and quantization, we select control parameters so that the overall system remains passive. The gain values of the controllers from Table 5.2 are verified to retain passivity given sampling periods of 10 ms, 30 ms, and 50 ms. Additionally, by implementing a transformation matrix M we are able to guarantee that the control system retains passivity given quantization effects. After generating the C code from the six controllers, we computed the WCET to being 12 μs. The safety conditions derived in Sections 6.2 and 6.3 are valid for vehicle velocities given a maximum road decline angle of 15 degrees which corresponds to $\delta_g = 4200$ N and a maximum lead vehicle deceleration of 5 $\mathrm{m/s^2}$ which corresponds to a braking distance of 50 m from 80 km/hr to 0 km/hr.

Figure 8.3 shows the relative distance between the two vehicles under various sampling periods on the top left subplot and the lateral acceleration of the host vehicle under various sampling periods on the bottom left subplot. The figure also contains a legend on the bottom right and the trajectory with approximate location circled using a purple circle on the top right. The simulation results show that despite keeping to the objectives of speed

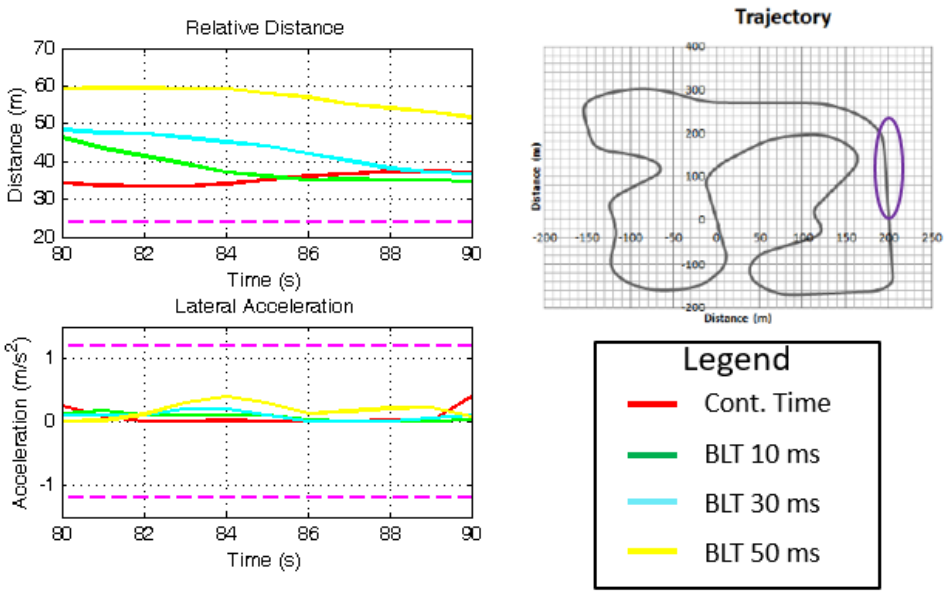

Figure 8.3: Relative distances and lateral accelerations for BLT cases

and steering control, there is a noticeable difference between the different sampling periods. As the sampling period increases the controller reacts more slowly to the behavior of the lead vehicle. Passivity also degrades with greater sampling periods. We experimentally computed the passivity indexes for the 10 ms case as $(0.7, 5.6)$, the 30 ms case as $(0.4, 5.4)$, and the 50 ms case as $(0.1, 5.4)$.

8.5 Evaluation of Discrete-Time Design

In this section, we present the simulation results to illustrate the discrete-time safety analysis approach and to show that the system remains safe. Simulation of the closed-loop system consists of also two minutes of running time in which the host vehicle follows a lead vehicle on the road. In all of the simulation results, the red curves pertain to the vehicle in continuous-time, the blue curves pertain to the host vehicle under 10 ms conservative time-discretization (CTD), the green curves pertain to the host vehicle under 10 ms bi-linear transformation (BLT), the teal curves pertain to the host vehicle under 30 ms bi-linear transformation, the yellow curves pertain to the host vehicle under 50 ms bi-linear

transformation, and the magenta lines pertain to safety bounds.

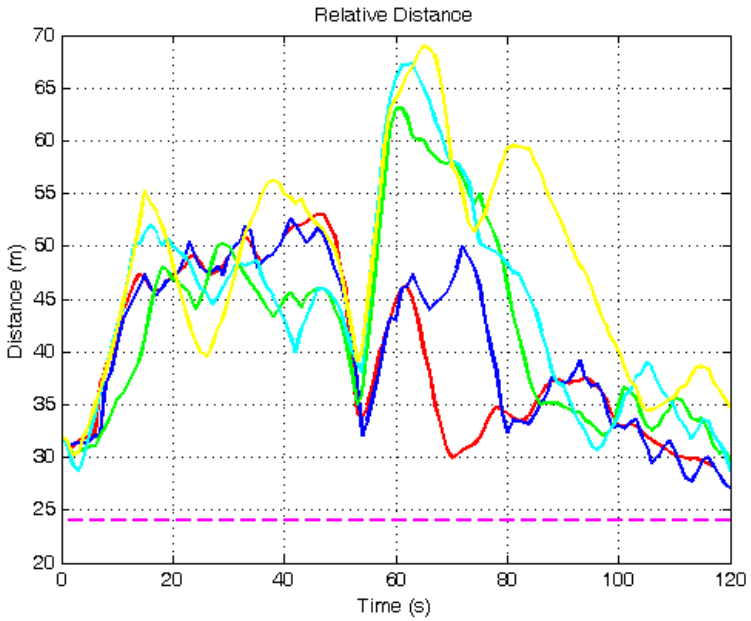

Figure 8.4: Relative distance for all cases

Figure 8.4 shows the relative distance between the two vehicles for the continuous-time case, CTD case, and the three BLT cases. Figure 8.5 shows the lateral acceleration of the host vehicles for the continuous-time case, CTD case, and the three BLT cases. Ideally, we want the results of the discretized cases to match that of the continuous-time case as much as possible. The results indicate that although all of the systems behave in a safe manner, only the CTD case is reasonably close to that of the continuous-time case. For the BLT cases, the performance of the controller gets progressively worse as the sampling period is increased.

Figure 8.6 contains a zoomed time-frame showing the relative distance between the two vehicles in the CTD case on the top left subplot and the lateral acceleration of the host vehicle under various sampling periods on the bottom left subplot. The figure also contains a legend on the bottom right and the trajectory with approximate location circled using a purple circle on the top right. The simulation results show that the results of the CTD case matches more closely to the continuous-time case, especially when compared to the results

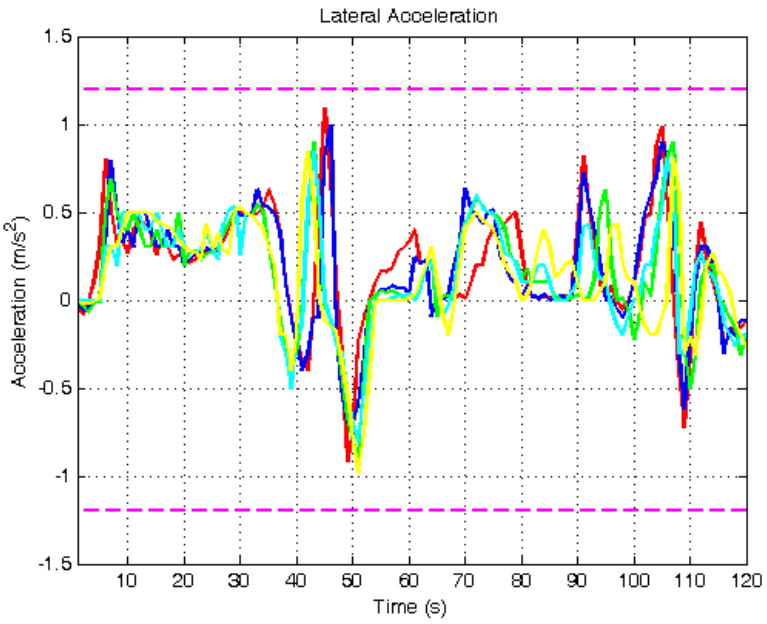

Figure 8.5: Lateral acceleration for all cases

of Figure 8.3. We also include some results from other time frames to show the difference between CTD and BLT schemes (Figures 8.7 and 8.8).

8.6 Summary

This chapter focuses on the HIL simulation platform, which provides a realistic environment for the testing of automotive control systems. We analyzed the impact of implementation effects of discretization/quantization and how they affect the passive PHS framework. We evaluated the continuous-time control design framework where the controllers are discretized using bi-linear transformation and implemented on the ECUs with varying levels of discretization and quantization. We also evaluated the discrete-time control design framework where the controllers are discretized using conservative time-discretization and implemented on the ECUs with varying levels of discretization and quantization.

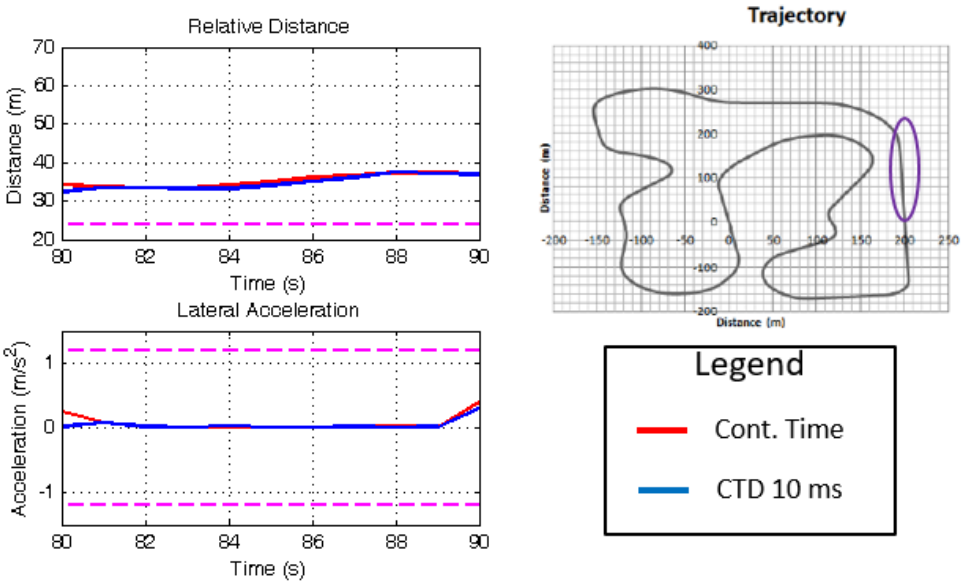

Figure 8.6: Relative distances and lateral accelerations for CTD case

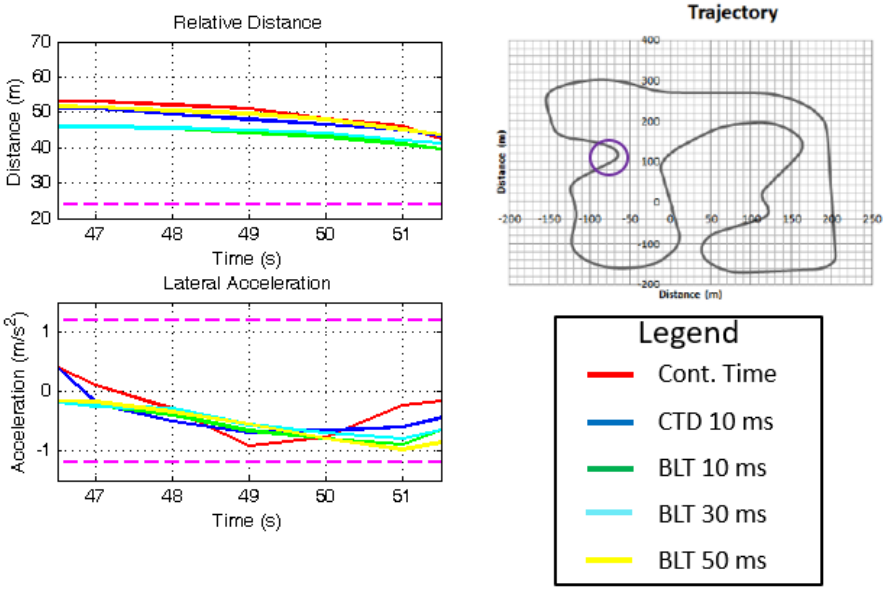

Figure 8.7: Relative distances and lateral accelerations between 46.5 - 51.5 s

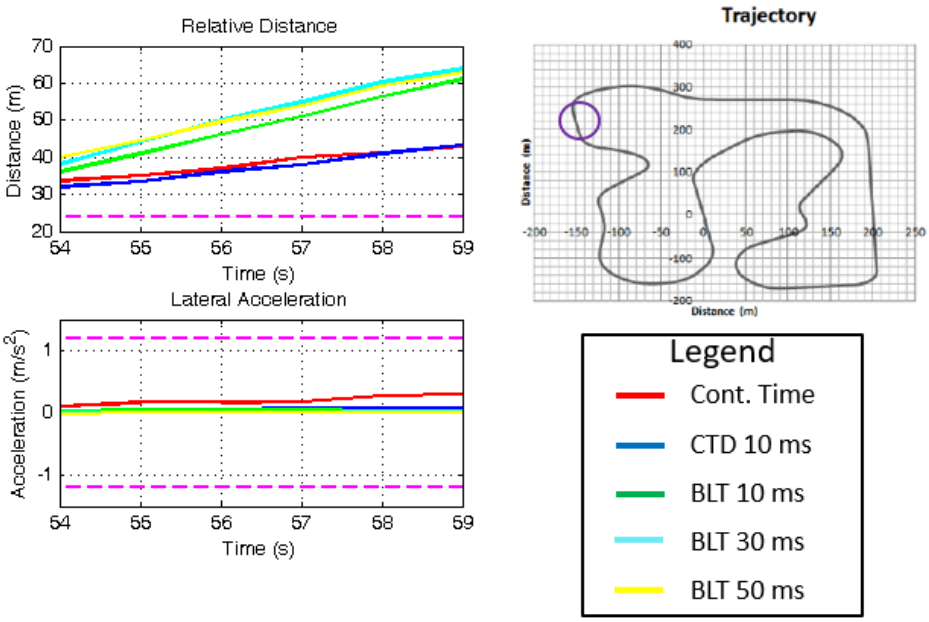

Figure 8.8: Relative distances and lateral accelerations between 54 - 59 s

Chapter 9

Conclusions

The increased growth and advancements in the area of cyber-physical systems (CPS) has resulted in many research topics in the modeling, simulation, and control design of CPS in order to ensure properties such as safety and stability. The goal of this dissertation is to address the various challenges facing the development of CPS. Research in CPS will continue to grow and expand as new techniques and methods are introduced.

9.1 Summary of Contributions

The pervasive theme of the work presented in this dissertation is the use of port-Hamiltonian systems (PHS) as a tool/framework for the modeling, designing, and analyzing of CPS. The main advantage of PHS is that they are compositional and hierarchical; its passivity property enables the use of passivity-based control design methods. Additionally, we can analyze the safety property of the system by using the Hamiltonian function as a barrier. We summarize our contributions as follows:

- *Component-Based Modeling and Simulation using Port-Hamiltonian Systems:* We developed a domain specific modeling language called PHSML for the compositional modeling of CPS. PHSML is integrated with constraint definitions which enforce connection rules and interpreters which exports structural data of the model into MATLAB scripts. The MATLAB functions generate the Dirac structure, constituent equations, and Hamiltonian functions of each PHSML component. A case study of an engine-dynamometer model is presented. We developed a component-based method of simulating systems built using PHSML by deriving acausal models of computation and implementing Modelica code. We also derived causal models of computation and implemented them in Simulink.

- *Compositional Model-Based Control Design Using Port-Hamiltonian Systems:* We developed a compositional control design framework based on PHS. The main idea of this control design framework is the use of the Dirac structures of PHS which allows the closed-loop system to preserve passivity. We presented simulation results of an automotive control system integrating speed control and steering control, which inspired us to use the framework as the initial stage of a model-based design methodology. We recorded simulation results which show the effectiveness of the design framework.

- *Safety Analysis of Automotive Control Systems Using Port-Hamiltonian Systems:* We developed a safety analysis method based on multi-modal PHS which utilizes the Hamiltonian function as a barrier certificate. Safety of the system is proven by isolating the safe and unsafe regions of state space and showing that trajectories cannot connect them. Consequently, we derive conditions on the model and control parameters which lead to safe behavior. We presented two case studies using automotive control systems and demonstrated both collision avoidance and skidding avoidance. We recorded simulation results which show the effectiveness of the safety analysis approach.

- *Discrete-Time Safety Analysis of Automotive Control Systems:* Motivated from the fact that passivity degrades during discretization, which may invalidate the continuous-time safety analysis method when implementing the system on a hardware-in-the-loop platform. As a result, we alter the discretization scheme using conservative time-discretization on multi-modal PHS which is able to preserve passivity in the event of discretization. We presented a case study in which the discrete-time safety of an automotive control systems shows that unsafe behaviors such as collision and skidding are also avoided.

- *Evaluation and Validation Using Hardware-in-the-Loop Platform:* We implement

the control design on a hardware-in-the-loop simulation platform. We analyzed the impact of implementation effects such as discretization and quantization. We evaluate the model-based design process where the controllers are implemented on ECUs with varying levels of discretization and quantization. These controllers are connected through a time-triggered communication network with the physical model of a vehicle provided by CarSim. We recorded results from the continuous-time implementation and the discrete-time implementation which show the effectiveness of the platform.

9.2 Future Work

The work presented in this dissertation can lead to many opportunities for future work. Passivity index is a concept commonly utilized to demonstrate the degree of passivity of a system and a lot of research has been done to use passivity index to design compensators dealing with delays. Given how PHS is useful for trajectory control (specifically using canonical coordinate transformations), and the relationship between PHS and passivity indexes, we believe that there may be some potential for research to bridge the concepts, allowing passivity indexes to be effectively used for trajectory control.

We also consider some future work for extending the safety analysis method from Chapters 6 and 7 further. The main motivation for extension results from obstacle avoidance, specifically obstacles which move either predictably or randomly. Currently the methods are implemented so that the energy bounds for safe and unsafe states do not change, which leads us to believe that the methods cannot be applied to moving obstacles. Given energy bounds which change with respect to time, we can assume that as long as these bounds do not overlap and the system retains passivity, the safety property should also hold. A reasonable amount of alteration may be required to prove the safety property when the bounds change.

List of Publications

Conference Papers

S. Dai and X. Koutsoukos. *Automated Generation and Simulation of Component-Based Generalized Bond Graphs*. International Conference on Bond Graph Modeling and Simulation, Genoa, Italy, 2012.

S. Dai and X. Koutsoukos. *Model-Based Automotive Control Design Using Port-Hamiltonian Systems*. International Conference on Complex Systems Engineering (ICCSE 2015), University of Connecticut, November 9-10, 2015.

S. and X. Koutsoukos. *Safety Analysis of Automotive Control Systems Using Multi-Modal Port-Hamiltonian Systems*. 19th ACM International Conference on Hybrid Systems: Computation and Control (HSCC 2016), Vienna, Austria, April 12-14, 2016.

Preparation of the discrete-time safety analysis method for conference publication.

Book Chapters

S. Dai, Z. Lattmann, and X. Koutsoukos. *Compositional Control Design of Cyber-Physical Systems*. Cyber-Physical Systems: From Theory to Practice, CRC Press, October 22, 2015, ISBN 9781482263329.

Journal Papers

S. Dai, Z. Zhang, and X. Koutsoukos. *Simulation and Virtual Prototyping of Automotive Control Systems Using Port-Hamiltonian Systems*. Submitted to the Journal of Software and Systems Modeling.

Preparation of the journal paper extension of the safety analysis work from HSCC 2016.

BIBLIOGRAPHY

[1] W. Borutzky. Bond graph modelling and simulation of multidisciplinary systems - an introduction. *Simulation Modelling Practice and Theory*, 17(1):3 – 21, 2009.

[2] V. Duindam, A. Macchelli, A. Stramigioli, and H. Bruyninckx. *Modeling and Control of Complex Physical Systems: The Port-Hamiltonian Approach*. Springer, 2009.

[3] J. Sztipanovits, X. Koutsoukos, G. Karsai, N. Kottenstette, P. Antsaklis, V. Gupta, B. Goodwine, J. Baras, and S. Wang. Toward a science of cyber-physical system integration. *Proceedings of IEEE*, 100:29–44, January 2012.

[4] Dymola. http://www.modelon.com/dymola.

[5] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. John Wiley and Sons, January 2004.

[6] R. Rajamani and C. Zhu. Semi-autonomous adaptive cruise control systems. *IEEE Transactions on Vehicular Technology*, 51, September 2002.

[7] D. Shang, E. Eyisi, Z. Zhang, X. Koutsoukos, J. Porter, G. Karsai, and J. Sztipanovits. A case study on the model-based design and integration of automotive cyber-physical systems. *21st Mediterranean Conference on Control and Automation*, June 2013.

[8] E. Eyisi, Z. Zhang, X. Koutsoukos, J. Porter, G. Karsai, and J. Sztipanovits. Model-based design and integration of cyber-physical systems: An adaptive cruise control case study. *Journal of Control Science and Engineering, Special Issue on Embedded Model-Based Control*, 2013.

[9] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic. Cyber-physical systems: The next

computing revolution. *Proceedings of the 47th Design Automation Conference*, pages 731–736, 2010.

[10] P. Derler and et. al. Modeling cyber-physical systems. *Proceedings of IEEE*, 100:13–28, January 2012.

[11] Y. Fallah, C. Huang, R. Sengupta, and H. Krishnan. Design of cooperative vehicle safety systems based on tight coupling of communication, computing, and physical vehicle dynamics. *International Conference on Cyber-Physical Systems 2010*, April 2010.

[12] R. Rajamani. *Vehicle Dynamics and Control*. Mechanical Engineering Series, 2012.

[13] J. Sztipanovits. Composition of cyber-physical systems. *Proceedings of the 14th Annual IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, 2007.

[14] G. Simko, T. Levendovsky, M. Maroti, and J. Sztipanovits. Towards a theory of cyber-physical systems modeling. *CyPhy*, 2013.

[15] S. Feng and L. Zhang. Integrated approach for modeling cyber-physical systems. *Advanced Technologies, Embedded and Multimedia for Human-centric Computing*, 2014.

[16] D. Jeltsema and J. M. A. Scherpen. Multidomain modeling of nonlinear networks and systems. *IEEE Control Systems Magazine*, 29:28–59, 2009.

[17] L. Lee. Cyber-physical systems: Design challenges. *International Symposium on Object, Component, and Service-Oriented Real-Time Distributed Computing (ISORC)*, May 2008.

[18] J. Lin, S. Sedigh, and A. Miller. Towards integrated simulation of cyber-physical

systems: A case study on intelligent water distribution. *8th International Conference on Dependable, Autonomic and Secure Computing*, 2009.

[19] S. Sakai and S. Stramigioli. Port-hamiltonian approaches to motion generation for mechanical systems. *IEEE International Conference on Robotics and Automation*, April 2007.

[20] J. Broenink. Introduction to physical systems modeling with bond graphs. In *in the SiE whitebook on Simulation Methodologies*, pages 18–27, 1999.

[21] D. Karnopp, D. Margolis, and R. Rosenberg. *System Dynamics: Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons, 2000.

[22] P. Breedveld. Multibond graph elements in physical systems theory. *Journal of the Franklin Institute*, 319(1-2):1–36, 1985.

[23] D. Margolis and T. Shim. Bond graph modelling for non-linear hydro-mechanical systems. *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, 219(371), December 2005.

[24] F. Cellier. Hierarchical nonlinear bond graphs: A unified methodology for modeling complex physical systems. *SIMULATION*, 58, April 1992.

[25] D. Karnopp. Bond graphs in control: Physical state variables and observers. *Journal of the Franklin Institute*, 308:221–234, 1979.

[26] P. Gawthrop. Physical model-based control: A bond graph approach. *Journal of the Franklin Institute*, 332(3):285–305, May 1995.

[27] G. Golo, P.C. Breedveld, B.M. Maschke, and A.J. Schaft van der. Geometric formulation of generalized bond graph models - part i: Generalized junction structures, 2000.

[28] OpenModelica. http://www.openmodelica.org/.

[29] JModelica. http://www.jmodelica.org/.

[30] Modelica Association. Modelica - a unified object oriented language for systems modeling - language specification. http://www.Modelica.org/.

[31] Modelica Association. Modelica standard library. https://github.com/modelica/Modelica.

[32] H. Elmqvist, S. Mattsson, and M. Otter. Modelica - the new object-oriented modeling language. *12th European Simulation Multiconference*, June 1998.

[33] H. Elmqvist and S. Mattsson. An introduction to the physical modeling language modelica. *Proceedings of the 9th European Simulation Symposium*, October 1997.

[34] S. Mattsson, H. Elmqvist, and M. Otter. Physical system modeling with modelica. *Control Engineering Practice*, 6:501–510, February 1998.

[35] C. Pantelides. The consistent initialization of differential-algebraic systems. *Journal of Science and Statistical Computing*, 9(2), March 1987.

[36] E. Carpanzano. Order reduction of general nonlinear dae systems by automatic tearing. *Mathematical and Computer modeling of Dynamical Systems*, 6(2):145–168, 2000.

[37] D. Henriksson and H. Elmqvist. Cyber-physical systems modeling and simulation with modelica. *International Modelica Conference*, 2011.

[38] A. Cervin, M. Ohlin, and D. Henriksson. Simulation of networked control systems using truetime. *Proceedings to the International Workshop on Networked Control Systems: Tolerant to Faults*, 2012.

[39] M. Branicky, V. Liberatore, and S. Phillips. Networked control system co-simulation for co-design. *American Control Conference*, 2003.

[40] A. Al-Hammouri. A comprehensive co-simulation platform for cyber-physical systems. *Computer Communications*, 36(1), 2012.

[41] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010.

[42] CarSim. http://www.carsim.com/.

[43] T. Blochwitz, M. Otter, M. Arnold, C. Bausch, C. Claus, H. Elmqvist, A. Junghanns, J. Mauss, M. Monteiro, T. Neidhold, D. Neumerkel, H. Olsson, J. Peetz, and S. Wolf. The functional mockup interface for tool independent exchange of simulation models. *8th International Modelica Conference*, 2011.

[44] H. Neema, J. Gohl, Z. Lattmann, J. Sztipanovits, G. Karsai, S. Neema, T. Bapty, J. Batteh, H. Tummescheit, and C. Sureshkumar. Model-based integration platform for fmi co-simulation and heterogeneous simulations of cyber-physical systems. *10th International Modelica Conference*, 2014.

[45] P. Breedveld. Modeling and simulation of dynamic systems using bond graphs. In *Control Systems, Robotics and Automation - Modeling and System Identification I*, volume 18-4-1 of *Encyclopedia of Life Support Systems*, pages 128–173. EOLSS Publishers Co. Ltd., Oxford, UK, 2008.

[46] L. Yu and X. Qi. Bond-graph modeling in system engineering. *2012 International Conference on Systems and Informatics*, May 2012.

[47] B. Podgursky, G. Biswas, and X. Koutsoukos. Efficient tracking of behavior in complex hybrid systems via hybrid bond graphs. In *Annual Conference of the Prognostics and Health Management Society 2010*, Portland, OR, US, 10/2010 2010.

[48] R. Ortega, A. van der Schaft, I. Mareels, and B. Maschke. Putting energy back in control. *IEEE Control Systems Magazine*, April 2001.

[49] E. Guilemin. *Synthesis of Passive Networks*. Wiley New York, Inc., 1957.

[50] A. van der Schaft. *L2-Gain and Passivity in Nonlinear Control*. Springer-Verlag New York, Inc., Secaucus, NJ, 1999.

[51] B. M. J. Maschke, R. Ortega, A. J. van der Schaft, and G. Escobar. An energy-based derivation of lyapunov functions for forced systems with application to stabilizing control. *Proceedings of the IFAC World Congress*, 1999.

[52] D. Hill and P. Moylan. The stability of nonlinear dissipative systems. *IEEE Transactions on Automatic Control*, 21:708–711, 1976.

[53] C. Byrnes, A. Isidori, and J. Willems. Passivity, feedback equivalence, and global stabilization of minimum phase nonlinear systems. *IEEE Transactions on Automatic Control*, 36, 1991.

[54] N. Kottenstette and P. Antsaklis. Relationships between positive real, passive dissipative, and positive systems. *American Control Conference*, 2010.

[55] C. Desoer and M. Vidyasagar. *Feedback Systems: Input-Output Properties*. Academic Press, Inc., 1975.

[56] H. Khalil. *Nonlinear Systems*. Prentice-Hall, Inc., Upper Saddle River, NJ, 2002.

[57] R. Thacker, K. Jones, C. Myers, and H. Zheng. Automatic abstraction for verification of cyber-physical systems. *International Conference on Cyber-Physical Systems 2010*, April 2010.

[58] E. Clarke and R. Kurshan. Computer-aided verification. *IEEE Spectrum*, 33(6):61–67, 1996.

[59] K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1996.

[60] et. al P. Kumar. A hybrid approach to cyber-physical systems verification. *Design Automation Conference 2012*, June 2012.

[61] G. Lafferriere, G. Pappas, and S. Yovine. Symbolic reachability computations for families of linear vector fields. *Journal of Symbolic Computation*, 32(3):231–253, 2001.

[62] H. Yazarel and G. Pappas. Geometric programming relaxations for linear systems reachability. *American Control Conference*, 2004.

[63] O. Botchkarev and S. Tripakis. Verification of hybrid systems with linear differential inclusions using ellipsoidal approximations. *In Hybrid Systems: Computation and Control, LNCS 1790*, pages 73–88, 2000.

[64] A. Bemporad, F. Torrisi, and M. Morari. Optimization-based verification and stability characterization of piecewise affine and hybrid systems. *In Hybrid Systems: Computation and Control, LNCS 1790*, pages 45–58, 2000.

[65] H. Hansen, G. Schneider, and M. Steffen. Reachability analysis of complex planar hybrid systems. *Science of Computer Programming*, 78:2511–2536, 2013.

[66] T. Hu and Z. Lin. Composite quadratic lyapunov functions for constrained control systems. *IEEE Transactions on Automatic Control*, 48(3), March 2003.

[67] F. Chernousko. Ellipsoidal state estimation for dynamical systems. *Nonlinear Analysis*, 63:872–879, 2005.

[68] S. Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42:117–126, 2006.

[69] S. Prajna and A. Rantzer. Primal-dual tests for safety and reachability. In *In: Hybrid Systems Computation and Control*, pages 542–556. Springer-Verlag, 2005.

[70] S. Prajna and A. Rantzer. Convex programs for temporal verification of nonlinear dynamical systems. *Journal of Control Optimization*, 46(3):999–1021, 2007.

[71] D. Work, A. Bayen, and Q. Jacobson. Automotive cyber physical systems in the context of human mobility. *National Workshop on High-Confidence Automotive Cyber-Physical Systems*, April 2008.

[72] F. Kirschke-Biller. Autosar - a global standard. *4th AUTOSAR Open Conference*, June 2012.

[73] G. Marsden, M. McDonald, and M. Brackstone. Towards an understanding of adaptive cruise control. *Transportation Research Part C*, May 2000.

[74] B. van Arem, C. van Driel, and R. Visser. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 7, December 2006.

[75] P. Ioannou, X. Xu, S. Eckert, D. Clemons, and T. Sieja. Intelligent cruise control: Theory and experiment. *32nd IEEE Conference on Decision and Control*, December 1993.

[76] D. Swaroop and J. Hedrick. String stability of interconnected dynamic systems. *IEEE Transactions on Automatic Control*, March 1996.

[77] A. D. Ames, J. W. Grizzle, and P. Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. *Proceedings of the 53rd IEEE Conference on Decision and Control*, December 2014.

[78] J.-W. Lee, B. Litkouhi, and H.-H. Huang. Two-stage lane keeping control algorithm

for lane sensing inaccuracy handling. *Proceedings of the ASME 2013 International Mechanical Engineering Congress and Exposition*, November 2013.

[79] J. Liu, T. Hu, and T. Hsu. Design of an automotive lane keeping system based on the structure of electric power steering. *Automotive Research Testing Center Technical Report*, 2005.

[80] J.-W. Lee and B. Litkouhi. A unified framework of the automated lane centering/changing control for motion smoothness adaptation. *15th International IEEE Conference on Intelligent Transportation Systems*, 2012.

[81] J.-W. Lee and B. Litkouhi. Control and validation of automated lane centering and lane changing maneuver. *Proceedings of the ASME 2009 Dynamic Systems and Control Conference*, October 2009.

[82] V. Cerone, M. Milanese, and D. Regruto. Combined automatic lane-keeping and driver's steering through a 2-dof control strategy. *IEEE Transactions on Control Systems Technology*, 17, January 2009.

[83] S. Wu, H. Chiang, J. Perng, T. Lee, and C. Chen. The automated lane-keeping design for an intelligent vehicle. *IEEE Intelligent Vehicles Symposium*, June 2005.

[84] R. Rajamani, H. Tan, B. Law, and W. Zhang. Demonstration of integrated lateral and longitudinal control for the operation of automated vehicles in platoons. *IEEE Transactions on Control Systems Technology*, 8(4):695–708, July 2000.

[85] A. van der Schaft. Port-hamiltonian systems: An introductory survey. *Proceedings of the International Congress of Mathematicians*, 2006.

[86] G. Golo, A. van der Schaft, P. Beedveld, and B. Mascheke. Hamiltonian formulation of bond graphs. *In: Nonlinear and Hybrid Systems in Automotive Control*, pages 351–372, 2003.

[87] A. Donaire and S. Junco. Derivation of input-state-output port-hamiltonian systems from bond graphs. *Simulation Modelling Practice and Theory*, 17:137–151, 2009.

[88] J. Cervera, A. J. van der Schaft, and A. Baños. Interconnection of port-hamiltonian systems and composition of dirac structures. *Automatica*, 43:214–217, February 2007.

[89] A. van der Schaft. Port-hamiltonian systems: Network modeling and control of nonlinear physical systems. *Advanced Dynamics and Control of Structures*, 2004.

[90] A. Macchelli and C. Melchiorri. Control by interconnection of mixed port-hamiltonian systems. *IEEE Transactions on Automatic Control*, 50, November 2005.

[91] G. Escobar, A. J. van der Schaft, and R. Ortega. A hamiltonian viewpoint in the modeling of switching power converters. *Automatica*, 35:445–452, 1999.

[92] J. Willems. Dissipative dynamical systems part ii: Linear systems with quadratic supply rates. *Archive for Rational Mechanics and Analysis*, 45:352–393, 1972.

[93] J. Willems. Dissipative dynamical systems part i: General theory. *Archive for Rational Mechanics and Analysis*, 45:321–351, 1972.

[94] C. Byrnes and W. Lin. Losslessness, feedback equivalence, and the global stabilization of discrete-time nonlinear systems. *IEEE Transactions on Automatic Control*, 39:83–98, 1994.

[95] O. J. Staffans. Passive linear discrete time-invariant systems. *Proceedings of the International Congress of Mathematicians*, 2006.

[96] J. Bao, K. Chan, W. Zhang, and P. Lee. An experimental pairing method for multi-loop control based on passivity. *Journal of Process Control*, 17:787–798, 2007.

[97] S. Stramigioli, C. Secchi, A. J. van der Schaft, and C. Fantuzzi. A novel theory for sample data systems passivity. *Proceedings of the IEEE Conference on Intelligent Robotic Systems*, 2002.

[98] A. Pogromsky, M. Jirstrand, and P. Spangeus. On stability and passivity of a class of hybrid systems. *IEEE Conference on Decision and Control*, 1998.

[99] J. Zhao and D. Hill. Dissipativity theory for switched systems. *IEEE Transactions on Automatic Control*, 53, May 2008.

[100] J. Zhao and D. Hill. A notion of passivity for switched systems with state-dependent switching. *Journal of Control Theory and Applications*, 4, 2006.

[101] M. Branicky. Stability of switched and hybrid systems. *IEEE Conference on Decision and Control*, 1994.

[102] M. Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

[103] M. Zefran. Passivity of hybrid systems based on multiple storage functions. *Allerton Conference*, 2001.

[104] M. Zefran, F. Bullo, and M. Stein. A notion of passivity for hybrid systems. *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001.

[105] L. Zhu. Passivity and stability of switched hamiltonian systems. *International Conference on Computer Application and System Modeling*, 2010.

[106] J. Buisson. Analysis of switching devices with bond graphs. *Journal of the Franklin Institute*, 300(6):1165–1175, 1993.

[107] K. Gerritsen, A. van der Schaft, and W. Heemels. On switched hamiltonian systems. *Proceedings MTNS*, 2007.

[108] C. Valentin, M. Magos, and B. Maschke. A port-hamiltonian formulation of physical switching systems with varying constraints. *Automatica*, 43, 2005.

[109] V. Popov. *Hyperstability of Control Systems*. Springer-Verlag New York, Inc., 1973.

[110] F. Kerber and A. van der Schaft. Compositional properties of passivity. *50th IEEE Conference on Decision and Control and European Control Conference*, December 2011.

[111] R. Sepulchre, M. Jankovic, and P. Kokotovic. *Constructive Nonlinear Control*. Springer-Verlag New York, Inc, 1997.

[112] H. Yu and P. Antsaklis. A passivity measure of systems in cascade based on passivity indices. *49th IEEE Conference on Decision and Control*, December 2010.

[113] J. Wen. Robustness analysis based on passivity. *American Control Conference*, pages 1207–1213, 1988.

[114] A. Kelkar and S. Joshi. Robust control of non-passive systems via passification. *American Control Conference*, 1997.

[115] A. Kelkar and S. Joshi. Robust passification and control of non-passive systems. *American Control Conference*, 1998.

[116] J. Bao and P. Lee. *Process Control: The Passive Systems Approach - Advances in Industrial Control*. Springer, 2007.

[117] G. Dauphin-Tanguy, A. Rahmani, and C. Sueur. Bond graph aided design of controlled systems. *Simulation Practice and Theory*, 7:493–513, December 1999.

[118] N. Hogan. Beyond regulators: Modeling control systems as physical systems. *Proceedings to the 1987 American Control Conference*, 1987.

[119] A. Sharon, N. Hogan, and D. Hardt. Controller design in the physical domain. *Journal of the Franklin Institute*, 328:697–721, 1991.

[120] D. A. Dirksz and J. M. A. Scherpen. Adaptive control of port-hamiltonian systems. *Proceedings of the 19th International Symposium on Mathematical Theory of Networks and Systems*, July 2010.

[121] G. Stacey and R. Mahony. A port-hamiltonian approach to formation control using bearing measurements and range observers. *Proceedings of the 52nd IEEE Conference on Decision and Control*, December 2013.

[122] B. M. J. Maschke and A. J. van der Schaft. Port-controlled hamiltonian systems: Modeling origins and system-theoretic properties. *IFAC Symposium on Nonlinear Control Systems*, pages 282–288, 1992.

[123] D. C. Tarraf. *Control of Cyber-Physical Systems*. Springer, 2013.

[124] E. Garcia-Canseco, D. Jeltsema, R. Ortega, and J. Scherpen. Power-based control of physical systems. *Automatica*, 46:127–132, 2010.

[125] N. Kottenstette, J. Hall, X. Koutsoukos, J. Sztipanovits, and P. Antsaklis. Design of networked control systems using passivity. *IEEE Transactions on Control Systems Technology*, 2012.

[126] J.-P. Richard. Time-delay systems: An overview of some recent advances and open problems. *Automatica*, 39:1667–1694, 2003.

[127] J.-H. Kim. Delay and its time-derivative dependant robust stability of time-delayed linear systems with uncertainty. *IEEE Transactions on Automatic Control*, 46(5):789–792, 2001.

[128] X. Koutsoukos, N. Kottenstette, J. Hall, P. Antsaklis, and J. Sztipanovits. Passivity-based control design for cyber-physical systems. *International Workshop on Cyber-Physical Systems Challenges and Applications*, June 2008.

[129] R. Pasumarthy and C.-Y. Kao. On stability of time-delay hamiltonian systems. *2009 American Control Conference*, pages 4909–4914, June 2009.

[130] R. Ortega, A. van der Schaft, F. Castanos, and A. Astolfi. Control by interconnection and standard passivity-based control of port-hamiltonian systems. *IEEE Transactions on Automatic Control*, 53, December 2008.

[131] R. Ortega, A. Loria, P. J. Nicklasson, and H. Sira-Ramirez. Passivity-based control of euler-lagrange systems. *Springer*, 1998.

[132] W. M. Haddad, S. G. Nersesov, and V. Chellaboina. Energy-based control for hybrid port-controlled hamiltonian systems. *Automatica*, 39:1425–1435, 2003.

[133] K. Fujimoto and T. Sugie. Canonical transformation and stabilization of generalized hamiltonian systems. *Systems and Control Letters*, 42:217–227, 2001.

[134] A. J. van der Schaft and B. M. J. Maschke. Mathematical modeling of constrained hamiltonian systems. *Proceedings of the Third IFAC Symposium on Nonlinear Control Systems*, 1995.

[135] K. Fujimoto and T. Sugie. Freedom in coordinate transformation for exact linearization and its application to transient behavior improvement. *Proceedings of the 35th IEEE Conference on Decision and Control*, pages 84–89, 1996.

[136] R. Ortega and E. Garcia-Canseco. Interconnection and damping assignment passivity-based control: A survey. *European Journal of Control*, 10:432–450, 2004.

[137] S. Prajna and A. Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *In: Hybrid Systems Computation and Control*, pages 477–492. Springer, 2004.

[138] S. Prajna, A. Papachristodoulou, and P. Parrilo. Introducing sostools: A general purpose sum of squares programming solver. In *In: Proceedings of the IEEE Conference on Decision and Control*, 2002.

[139] C. Sloth, G. J. Pappas, and R. Wisniewski. Compositional safety analysis using barrier certificates. *2012 Conference on Hybrid Systems Computation and Control*, April 2012.

[140] S. Prajna, A. Jadbabaie, and G. J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, August 2007.

[141] G. Karsai, J. Sztipanovits, A. Ledeczi, and T. Bapty. Model-integrated development of embedded software. *Proceedings of the IEEE*, 91(1), 2003.

[142] Siyuan Dai, Zsolt Lattmann, and Xenofon Koutsoukos. Compositional design of cyber-physical systems using port-hamiltonian systems. In *Cyber Physical Systems: From Theory to Practice*, pages 33–59. CRC Press, October 2015. doi:10.1201/b19290-4.

[143] S. Dai and X. Koutsoukos. Automated generation and simulation of component-based generalized bond graphs. *12th International Conference on Bond Graph Modeling and Simulation*, July 2012.

[144] Generic Modeling Environment. http://repo.isis.vanderbilt.edu/downloads?tool=GME.

[145] S. Das. *Mechatronic Modeling and Simulation Using Bond Graphs*. CRC Press, 2009.

[146] Cummins Inc. Basic engine model kta19-m3, engine configuration d193080mx02, September 2009.

[147] R. Ortega, Z. Jiang, and D. Hill. Passivity-based control of nonlinear systems: A tutorial. *Proceedings of the American Control Conference*, June 1997.

[148] S. Dai and X. Koutsoukos. Model-based automotive control design using port-hamiltonian systems. *International Conference on Complex Systems Engineering (ICCSE 2015)*, November 2015.

[149] S. Dai, Z. Zhang, and X. Koutsoukos. Simulation and virtual prototyping of automotive control systems using port-hamiltonian systems. *Submitted to the Journal of Software and Systems Modeling*, 2016.

[150] J. Porter, G. Hemingway, and H. Nine. The esmol language and tools for high-confidence distributed control systems design - part 1: Language, framework, and analysis. *Technical Report ISIS-10-109, Vanderbilt University*, 2010.

[151] Z. Zhang, E. Eyisi, X. Koutsoukos, J. Porter, G. Karsai, and J. Sztipanovits. Co-simulation framework for design of time-triggered cyber physical systems. *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, 2013.

[152] R. Hooke and T. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the Association of Computing Machinery*, 7:212–229, 1969.

[153] P. Wu, M. McCourt, and P. J. Antsaklis. Experimentally determining passivity indices: Theory and simulation. *Technical Report of the ISIS Group*, April 2013.

[154] The Network Simulator ns 2. http://www.isi.edu/nsnam/ns/doc/index.html.

[155] D. Dirksz and J. Scherpen. Port-hamiltonian and power-based integral type control of a manipulator system. *18th International Federation of Automatic Control World Congress*, August 2011.

[156] S. Dai and X. Koutsoukos. Safety analysis of automotive control systems using multi-modal port-hamiltonian systems. *19th ACM International Conference on Hybrid Systems: Computation and Control (HSCC 2016)*, April 2016.

[157] D. S. Laila and A. Astolfi. Construction of discrete-time models for port-hamiltonian systems with applications. *Systems and Control Letters*, 55:673–680, 2006.

[158] TTEthernet. http://www.tttech.com/en/products/ttethernet/.

[159] C. S. Ravichandran, S. Subha Rani, and T. Manikandan. Designing of pid controller for discrete time linear system using balanced approach reduced order model. *American Journal of Applied Sciences*, 4(3):155–159, 2007.

[160] V. Talasila, J. Clemente-Gallardo, and A. J. van der Schaft. Discrete port-hamiltonian systems: Mixed interconnections. *Proceedings of the 44th IEEE Conference on Decision and Control*, December 2005.

[161] D. S. Laila and D. Nesic. Changing supply rates for input-output to state stable discrete-time nonlinear systems with applications. *Automatica*, 39:821–835, 2003.

[162] D. Nesic, A. R. Teel, and P. Kokotovic. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Systems Control Letters*, 38:259–270, 1999.

[163] S. Aoues and W. Marquis-Favre. Canonical interconnection of discrete linear port-hamiltonian systems. *Proceedings of the 52nd IEEE Conference on Decision and Control*, December 2013.

[164] Y. Oishi. Passivity degradation under the discretization with the zero-order hold and the ideal sampler. *49th IEEE Conference on Decision and Control*, December 2010.

[165] O. Gonzalez. Time integration and discrete hamiltonian systems. *Journal of Nonlinear Science*, 6:449–467, 1996.

[166] M. de la Sen. Preserving positive realness through discretization. *Positivity*, 6(1):31–45, 2002.

[167] S. Stramigioli, C. Secchi, A. J. van der Schaft, and C. Fantuzzi. Sampled data systems passivity and discrete port-hamiltonian systems. *IEEE Transactions on Robotics*, 21(4):574–587, 2005.

[168] D. Nesic and A. R. Teel. A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. *IEEE Transactions on Automatic Control*, 49:1103–1122, 2004.

[169] R. Costa-Castello and E. Fossas. On preserving passivity in sampled-data linear systems. *Proceedings of the 2006 American Control Conference*, pages 4373–4378, June 2006.

[170] S. Kawakami and H. Fujioka. Stability analysis for a class of hamiltonian systems with digital control. *Proceedings of the 2010 SICE Annual Conference*, August 2002.

[171] F. Ceragioli and C. De Persis. Discontinuous stabilization of nonlinear systems: Quantized and switching controls. *IEEE Conference on Decision and Control*, pages 783–788, 2006.

[172] F. Zhu, H. Yu, M. J. McCourt, and P. J. Antsaklis. Passivity and stability of switched systems under quantization. *Conference on Hybrid Systems Computation and Control*, April 2012.

[173] D. F. Delchamps. Stabilizing a linear system with quantized state feedback. *IEEE Transactions on Automatic Control*, 35:916–924, 1990.

[174] K. K. Jiyang, K. Kum, J. Kang, and Sung W. A floating-point to fixed-point c converter for fixed-point digital signal processors. *Second SUIF Compiler Workshop*, 1997.

[175] J. Porter, G. Karsai, and Sztipanovits J. Towards a time-triggered schedule calculation tool to support model-based embedded software design. *Proceedings of the Seventh ACM International Conference on Embedded Software*, pages 167–176, October 2009.