Position Paper: Passivity-Based Architecture for Design of Cyber-Physical Systems

Xenofon Koutsoukos, Nicholas Kottenstette, Panos Antsaklis, Janos Sztipanovits
Vanderbilt University/ISIS and University of Notre Dame

The integration of physical systems through computing and networking has become the most pervasive application of Networking and Information Technology (NIT), a trend now known as Cyber-Physical Systems (CPS). Transportation systems represent an exciting area of CPS applications where successful design and implementations will affect our everyday lives. In CPS, the overall system dynamics emerge from the interaction among physical dynamics, computational dynamics, and communication networks. Physical functionality emerges from the interaction of networked computational and physical objects. Therefore, requirements related to physical dynamics, power, fault tolerance, safety, security, physical size and other system characteristics motivate and inspire designs both in the physical and computational components.

By ensuring that embedded computing and communication is used as the universal system integrator across all types and scales, NIT has fundamentally transformed the way we engineer systems. While this pervasive use of NIT for integrating systems offers exceptional opportunities, it also creates fundamental challenges by introducing heterogeneity and severe complexity.

- Many CPS are mission-critical and must be designed to live forever (with addition and replacement of individual components). Mechanical and civil engineering are much more successful in creating long-lived systems; the Brooklyn Bridge, for example, recently celebrated its 125th anniversary, even though its cables and other components are regularly replaced. In contrast, computer-based systems such as the National Air Traffic Control System have proven to be brittle and unreliable.
- The complexity of both the components and the integrated systems is steadily increasing. “System of Systems” integration has become a common term characterizing modern system development.
- Large systems evolve along “spiral-outs” that makes integration inherently incremental: deployed systems need to be integrated with components on different levels of maturity from prototypes to the simulated and to the real.

The current approach for system integration is in clear conflict with the trends mentioned above. Through its reliance on ad-hoc methods after all the components have been designed and manufactured, the existing integration methods simply aim at “making it work somehow.” Moreover, as the complexity of engineered systems continues to increase, the challenge of a systematic theory for systems integration gets increasingly worse. Finding a solution is hard because system integration is the phase where essential design concerns – usually separated into software, systems and control engineering – are coming together and the hidden, poorly understood interactions and conflicts suddenly surface. This makes system integration particularly challenging in CPS where fundamentally different physical and computational design concerns intersect.

Orthogonalization Between Design Concerns

Building systems from components is a basic method in all engineering disciplines to manage complexity, decrease time-to-market and contain cost. The feasibility of component-based system design depends on two key conditions: *compositionality* – meaning that system-level properties can be computed from local properties of components – and *composability* – meaning that component properties do not change as a result of interactions with other components. Compositionality is widely used in both computer science and engineering, but has been most successfully applied to restricted properties, small system size and limited interactions. Existing compositional frameworks address separate design concerns and neglect their interactions. The result is weakened or lost composability with effects turning up during system integration.

CPS are inherently heterogeneous not only in terms of their components but also in terms of essential design requirements. Besides functional properties, CPS are subject to a wide range of physical requirements, such as dynamics, power, physical size, and fault tolerance in addition to system-level requirements, such as safety and security. This heterogeneity does not go well with current methods of compositional design for several reasons. The most important principle used in achieving multi-objective compositionality is *separation* *of concerns* (in other words, defining design viewpoints). Separation of concerns works if the design views are orthogonal, i.e. design decisions in one view does not influence design decisions in other views. Unfortunately, achieving compositionality for multiple physical and functional properties *simultaneously* is a very hard problem because of the lack of orthogonality among the design views.

Effectiveness of the platform-based design largely depends on how much the design concerns (captured in the abstraction layers) are orthogonal, i.e., how much the design decisions in the different layers are independent. Heterogeneity causes major difficulties in this regard. The controller dynamics is typically designed without considering implementation side effects (e.g. numeric accuracy of computational components, timing accuracy caused by shared resource and schedulers, time varying delays caused by network effects, etc.). Timing characteristics of the implementation emerge at the confluence of design decisions in software componentization, system architecture, coding and HW/network design choices. Compositionality in one layer depends on a web of assumptions to be satisfied by other layers. For example, compositionality on the controller design layer depends on assumptions that the effects of quantization and finite word-length can be neglected and the discrete-time model is accurate. Since these assumptions are not satisfied by the implementation layer, the overall design needs to be verified after implementation – even worst – changes in any layer may require re-verification of the full system.

An increasingly accepted way to address the problems is to enrich abstractions in each layer with implementation concepts. While this is a major step in improving designers' understanding of implementation effects, it does not help in decoupling design layers and improving orthogonality across the design concerns. A controller designer can now factor in implementation effects (e.g., network delays), but still, if the implementation changes, the controller may need to be redesigned. Decoupling the design layers is a very hard problem and typically introduces significant restrictions and/or overdesign. For example, the Timed Triggered Architecture (TTA) orthogonalizes timing, fault tolerance, and functionality, but it comes on the cost of strict synchrony, and static structure.

Exploiting orthogonalization for CPS design requires novel composable cross-domain abstractions that capture properties from multiple domains. For example, in the design of networked controllers for UAVs, the functional domain focuses on the design of controller dynamics for executing e.g. tracking operations. When designing controller dynamics, both the physical platform related abstractions and safety/security related abstractions need to be explicitly considered. Physical platform related abstractions that are relevant for controller dynamics include timing uncertainties caused by the communication network, CPU dynamic power management that affects performance, jitter caused by the schedulers, value uncertainties caused by quantization, and finite precision arithmetic inaccuracies. Safety and security related abstractions that are relevant for controller dynamics include timing uncertainties caused by fault management, architecture adaptation, intrusion detection, encryption/decryption, separation kernels and virtualization. In addition, multi-level security, access control and confidentiality result in structural constraints in designing control laws.

Passivity-Based Design

This paper proposes a new design platform, called *Passivity-Based Architecture* (*PBA*) inspired by the rapidly increasing use of Networked Control System (NCS) architectures in constructing real-world CPS. PBA can contribute to: (1) providing a scalable computational method of composition for large-scale, real-world networked CPS, (2) controlling real-world system behavior and interactions in dynamic, ever-changing conditions, and (3) improving the safety of interaction between human operators and CPS.

PBA aims at addressing fundamental problems caused by networks effects, such as time-varying delay, jitter, limited bandwidth, and packet loss. To deal with these implementation uncertainties, we propose a model-design flow on top of passivity, a very significant concept from system theory. A precise mathematical definition requires many technical details, but the main idea is that a passive system cannot apply an infinite amount of energy to its environment. The inherent safety that passive systems provide is fundamental in building systems that are insensitive to implementation uncertainties. Passive systems have been exploited for the design of diverse systems such as smart exercise machines, teleoperators, digital filters, and networked control systems.

Our approach advocates a concrete and important transformation of model-based methods that can improve orthogonality across the design layers and facilitate compositional component-based design of CPS.

By imposing passivity constraints on the component dynamics, the design becomes insensitive to network effects, thus establishing orthogonality (with respect to network effects) across the controller design and implementation design layers. This separation of concerns empowers the model-based design process to be applied for networked control systems. Detailed information about the network effects needs not to be considered at the controller design layer and the theoretical guarantees about stability and performance are independent of the implementation uncertainties. Further, stability is maintained even in the presence of disturbance traffic in the network.

In our initial work, we have implemented PBA for a system consisting of a 6-DOF robotic system controlled by a digital controller over a wireless network and we have proven the stability of the networked control system. We have evaluated the system using simulations and experimental results validating the significant advantages of the architecture especially in the presence of time-varying delays. Currently, our work focuses on methods that provide an effective way to interconnect multiple passive systems and controllers as well as an integrated end-to-end tool chain for the model-based design of CPS based on passivity.

Passive systems have a unique property that when connected in either a parallel or negative feedback manner the overall system remains passive. Large scale and open systems organized as suitable interconnections of passive structures will be also passive and therefore robust to uncertainties. Further, the interconnection topology can be dynamically adapted or reconfigured in order to ensure system robustness. Novel scalable analysis and design methods are necessary for understanding and eventually controlling the impact of dynamic topologies on stability and robustness.

Characterizing robust CPS behavior requires new notions of uncertainty that cross-cut the physical, computation, and communication domains. These diverse notions of uncertainty require novel assessment metrics that capture different views of system stability and robustness. Extensions of passivity to discrete event and hybrid systems are required in order to characterize robust composition of software and physical components. Imposing such restrictions on the component dynamics will enable compositional modeling and reasoning for computing, sensing, and acting on the physical world.

Compositional properties such as passivity offer tremendous advantages for designing robust large scale systems but stability and robustness cannot be studied in isolation from other design concerns. Understanding fundamental tradeoffs between stability and robustness, system performance, safety and security, and properties of the physical platform should be a very critical research endeavor. This requires, for example, understanding how composition based on passivity affects performance, platform properties and other design concerns.

Biographical Information

Xenofon Koutsoukos is an Assistant Professor in the Department of Electrical Engineering and Computer Science at Vanderbilt University and a Senior Research Scientist in the Institute for Software Integrated Systems. His research interests include hybrid systems, real-time embedded systems, and sensor networks. Contact information: 615 322-8283; xenofon.koutsoukos@vanderbilt.edu.

Nicholas Kottenstette is a Research Scientist in the Institute for Software- Integrated Systems at Vanderbilt University. His research interests focus on digital control networks, fault tolerant systems, and telemanipulation systems. Contact information: 615 322-3162, nkottens@isis.vanderbilt.edu.

Janos Sztipanovits is the E. Bronson Ingram Distinguished Professor of Engineering in the Department of Electrical Engineering and Computer Science at Vanderbilt University. He is founding director of the Institute for Software- Integrated Systems. His research interests include model-integrated computing, structurally adaptive systems, and embedded software and systems. Contact information: 615 343-7472, janos.sztipanovits@vanderbilt.edu.

Panos Antsaklis is the H. Clifford and Evelyn A. Brosey Professor of Electrical Engineering and Concurrent Professor of Computer Science and Engineering at the University of Notre Dame. His recent research focuses on networked embedded systems and addresses problems in the interdisciplinary research area of control, computing and communication networks, and on hybrid and discrete event dynamical systems. Contact information: 574 631-5792; antsaklis.1@nd.edu.