Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 28 (2014) 285 – 293

Procedia Computer Science

Conference on Systems Engineering Research (CSER 2014)

Eds.: Azad M. Madni, University of Southern California; Barry Boehm, University of Southern California; Michael Sievers, Jet Propulsion Laboratory; Marilee Wheaton, The Aerospace Corporation
Redondo Beach, CA, March 21-22, 2014

Resilience in Large Scale Distributed Systems

Nikolai Matni*, Yoke Peng Leong, Yuh Shyang Wang, Seungil You, Matanya B. Horowitz, John C. Doyle[a]

[a]*All authors are with the California Institute of Technology, 1200 E. California Blvd., Pasadena 91125, USA*

Abstract

Distributed systems are comprised of multiple subsystems that interact in two distinct ways: (1) physical interactions and (2) cyber interactions; i.e. sensors, actuators and computers controlling these subsystems, and the network over which they communicate. A broad class of cyber-physical systems (CPS) are described by such interactions, such as the smart grid, platoons of autonomous vehicles and the sensorimotor system. This paper will survey recent progress in developing a coherent mathematical framework that describes the rich CPS "design space" of fundamental limits and tradeoffs between efficiency, robustness, adaptation, verification and scalability. Whereas most research treats at most one of these issues, we attempt a holistic approach in examining these metrics. In particular, we will argue that a control architecture that emphasizes scalability leads to improvements in robustness, adaptation, and verification, all the while having only minor effects on efficiency – i.e. through the choice of a new architecture, we believe that we are able to bring a system closer to the true fundamental hard limits of this complex design space.

Keywords: large scale; distributed; resilient; control theory; convex optimization; tradeoffs; fundamental limits; layered architecture;

* Corresponding author. Tel.: 626-590-5852; fax: 626-792-4257.
E-mail address: nmatni@caltech.edu

doi:10.1016/j.procs.2014.03.036

1. Introduction

A holistic systems theory perspective looks at the design of large-scale distributed systems through the lens of a "design space" involving fundamental tradeoffs between: (E) *Efficiency*: meeting quantitative objectives with minimal use of resources, (R) *Robustness*: maintaining efficiency despite disturbances and component uncertainty, (A) *Adaptation*: the ability to adapt to changing system components and objectives, (V) *Verification*: providing *a priori* guarantees, as formally as possible, of performance, and (S) *Scalability*: the ability to scale the design and implementation to (arbitrarily) large systems.

As it stands, although these metrics are fairly well understood individually, their combination is usually done in a serial manner, often through layering; this is especially true in aerospace and information technology related applications. Many of the great 20th century system theories in computing, control, and communications support, in limited ways, the layered separation of these objectives, but the overall process is still largely ad hoc. The goal of our work is to develop fundamental theories that aim to explain these tradeoffs between performance metrics, and in particular, that allow the designer to distinguish between which bounds are true hard limits on performance, and which are simply symptomatic of a poorly chosen architecture. In the context of biological systems, we have been able to provide a deep understanding of how R and E interact by leveraging tools from robust control theory[1], and we believe these insights to be fundamental.

In this paper, we seek to extend these insights to a particular class of distributed systems known as *cyber-physical systems* (CPS) (see Fig. 1). Such distributed systems are comprised of several subsystems that interact in two distinct but interdependent ways: (1) *physical interactions*; i.e. how each local subsystem affects its neighbors, and (2) *cyber interactions*; i.e. the sensors, actuators and computers controlling these subsystems, and the network over which they communicate and coordinate. Before delving in to how the five performance metrics that we have mentioned interact in a distributed setting, we take some time to point out some of the inherent challenges of addressing these metrics individually, and highlight some of the progress that has been made in the past decade in this respect.

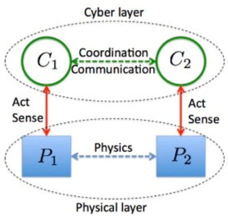

Fig. 1. A cyber-physical system is comprised of two distinct interactions: physical and cyber.

We begin with a focus on S, which is the subject of *distributed optimal control theory*. Traditional, or centralized, control theory is in general no longer applicable to large scale systems, as it assumes a central control unit that collects *all* sensor data, computes a global control action, and then broadcasts this control action out to *all* actuators. This quickly becomes both computationally intractable (the computational cost of these solutions scales poorly with the dimension of the system), and undesirable in terms of performance, as the collection of measurements/broadcasting of control actions invariably introduces *delay* in the controller. The natural solution is then to have a distributed control system, in which each local subsystem is equipped with its own controller, and in addition, delay constrained communication is allowed between these controllers: it is from this distribution of control authority that CPS emerge.

Unfortunately, in general, optimizing the performance of a distributed system can be very difficult (i.e. non-convex, and hence computationally intractable); c.f. the Witsenhausen counter-example for a canonical illustration[2]. It is only recently that a large class of systems for which the optimal synthesis problem admits a convex and hence tractable formulation, has been identified: these systems are said to satisfy a *quadratic invariance* (QI) property[3]. In the class of CPS that we consider, this condition takes a particularly simple form: it requires controllers to be able to