Research Statement

Nikolai Matni

Modern cyber-physical systems such as the Internet of Things (IoT), software defined networks (SDN) and the smart-grid are large-scale, distributed, interconnected, and dynamic. While layered control architectures have become ubiquitous and arguably necessary in achieving predictable and desirable behavior, there is no general theory that offers a principled approach to their design. My research aims to address this gap by developing a theory that incorporates layering, dynamics, optimization and control into a unified framework.

To concretely illustrate an effective use of layering, consider a familiar “live demo” where you fill a glass of water and take a drink. Any conscious planning and execution is typically done with simple arm and glass positions and velocities, and not in terms of muscle torques and forces needed to move the glass or compensate for the changing water weight. Remarkably, there is near perfect agreement between the actual trajectory of your arm, glass, and water and that planned using a highly simplified model of them. Even more remarkable is that this match between the physics of your body and the virtualized model in your mind is achieved using a distributed, resource constrained control system using communication with axon bundles (nerves) that is itself a layered system – actions at the muscular level are effected by changes at the cellular level, and in turn actions at the cellular level are effected by changes at the macromolecular level. This use of a recursive layered architecture to coordinate high-level planning and low-level actions across different scales, despite distributed and resource constrained control, is an example of the kind of system that my research seeks to design and understand.

Indeed this need to map high-level goals expressed in a simplified virtual space to low-level commands in a complex physical space is universal in cyber-physical systems. For example, in

- software defined networking high-level goals are specified by solutions to network resource allocation problems, and these solutions must be implemented using packet-level control via routers, admission controllers and local caching;

- robotics and sensorimotor control high-level objectives (e.g., raise arm to 90°, walk from point $A$ to point $B$) are easily stated, but must be implemented using low-level control of forces and torques;

- control of the smart grid high-level goals are stated in terms of economic or functional utilities (e.g., economic dispatch or optimal power flow), and these must be implemented by controlling the physics of the grid.

This broad range of applications underscores the need to understand how the behavior of an underlying complex physical system can match a plan generated using a simple virtual model of the system. Perhaps just as pressing is the need to understand how this can be done using resource constrained distributed controllers in dynamic and uncertain environments.

1 Integrating Dynamics, Control and Optimization in Layered Architectures

I believe that this understanding lies in an integrated theory of system architecture that incorporates layering, dynamics, control and optimization into a unified framework. The starting point for this theory is the observation that many cyber-physical systems must accomplish two complementary tasks: (i) identifying an optimal trajectory with respect to a functional or economic utility function, and (ii) efficiently making the state of the system follow this optimal trajectory despite model uncertainty, disturbances, sensor noise, and distributed information sharing constraints. While traditional approaches to layered architectures treat these two tasks in a fairly independent manner (i.e., static set-point planning is done using little to no modeling of the dynamics of the underlying system), I argue that in order to develop a truly integrated theory, these two tasks must be considered together.
To that end, in recent work I developed the foundations of a new theoretical framework, firmly rooted in distributed optimization and control, that informs when and how to use layering in the design of a dynamical cyber-physical system. In particular, in [1] I generalize the Layering as Optimization (LAO) framework to incorporate not only optimization, but dynamics and control as well. I show that by suitably relaxing an optimal control problem that jointly addresses determining and following an optimal trajectory, one can naturally recover a layered architecture composed of a low-level tracking layer and a top-level planning layer (cf. Fig 1). The tracking layer consists of a distributed optimal controller that takes as an input a reference trajectory generated by the top-level layer, where this top-level layer consists of a trajectory planning problem that optimizes the weighted sum of a utility function and a “tracking penalty” regularizer. This latter term can be viewed as the planning layer’s “virtual model” of the underlying physics of the system, and serves as a balance between the two by ensuring that the planned trajectory can indeed be efficiently followed by the tracking layer.

Figure 1: A functional schematic of the layered architecture derived in [1].

2 Sparse Controller Architectures

The tracking penalty $f_{\text{track}}(r_{0:N})$ added to the planning problem in Fig. 1 measures the ability of the lower layer to follow commands issued by the planning layer. Ideally this term should be made as small as possible: if it is too large, it can significantly bias the solution of the planning problem away from trajectories that are optimal with respect to the functional utility $C$. It is easy to see that a system’s ability to track a given reference trajectory is determined by its actuation, sensing and communication architecture: the more of these resources are present, the better we expect the performance of an optimal controller to be. However such architectural components can be expensive to build, install and maintain, and thus from an economic or energetic perspective there is a real motivation to use as few actuators, sensors and communication links as possible. The result is a tradeoff between architectural complexity and closed loop performance that, naively explored, would require enumerating a combinatorial design space. In order to explore this design space in a computationally tractable manner, I developed the [Regularization for Design (RFD)] framework [2, 3, 4] which provides a control system designer with tractable algorithmic tools (based on convex optimization and atomic norm regularization) for the co-design of a distributed optimal controller and the actuation, sensing and communication architecture used to implement it. A surprising outcome of this line of work is the identification of systems for which near centralized performance can be obtained using sparse controller architectures, thus suggesting that judiciously placed architectural resources can have a significant impact on closed loop performance. The importance of this work was recognized by the control community, which awarded the sole-author paper [4] the [Best Student Paper Prize of the 2013 IEEE Conference on Decision and Control], the premier conference in the field.

3 Scalability through Locality

Finally, as the scale of cyber-physical systems continues to grow, it is important that all of the aforementioned tools (distributed optimization, distributed control, layering and RFD) scale gracefully to systems that are potentially arbitrarily large. The [Localized Optimal Control] framework (cf. [5, 6, 7] and references therein) allows for optimal control and RFD methods to scale to arbitrarily large heterogeneous distributed systems, despite realistic communication delays between sensors and actuators. The intuition
behind this approach is that if a controller’s actuation, sensing and communication architecture is dense enough to localize the effects of individual disturbances, then the synthesis and implementation of an optimal controller decomposes into local subproblems of fixed size. This intuitive approach has deep and far reaching consequences: if certain physical architectural constraints are met by the reflexive tracking-layer of a cyber-physical system then analysis, design, implementation and optimization (i.e. planning) can be made arbitrarily scalable. Further, we have leveraged the RFD framework to design actuation architectures in a scalable manner such that the resulting system satisfies these localization properties [7].

4 Future Plans

My goal is to develop a research program that emphasizes the interplay between theory and application. I am a firm believer that good theory is driven by practical problems and case studies, and likewise, that new application areas can be opened up by developing fundamental theoretical tools that allow us to pose and answer heretofore unasked questions. Described below is an outline of my plan for building such a research program.

4.1 Theory

Layering In §1 I outlined a minimal version of a broader theoretical framework. Going forward, I intend to pursue the theoretical questions opened up by this novel perspective on layering, dynamics, control and optimization. Immediate objectives include (i) extending these techniques to incorporate explicit virtual dynamics in the planning layer, (ii) understanding how actuation, sensing, and communication density, control effort cost and system dynamics constrain achievable virtual dynamics, (iii) exploring the use of distributed optimization and game theoretic tools to solve the tracking penalty augmented utility maximization problem of the planning layer, and (iv) developing a method to recursively apply this approach to forward/reverse engineer multi-tiered hierarchical architectures for large-scale complex systems. The overarching goal is to develop a theoretical framework that will inform the reverse/forward engineering of all layers (and their interactions) of a complex system, ranging from strategic planning (e.g., drink a glass of water) down to the fundamental physical processes used to accomplish these goals (e.g., the cellular mechanisms behind muscle contraction). This in turn will open up a broad range of application areas, some of which I describe below.

Learning Layered architectures allow for “classical” learning tasks such as system identification and statistical inference to be integrated in a natural way. Taking a layered view of dynamical systems also allows us to study a different type of learning than that considered in the inference literature, namely that of learning how to move tasks from the slower but more flexible upper layers to faster specialized lower layers. This is precisely how humans learn to play a new sport or instrument: by practicing basic movements that are fundamental to the skill (e.g., playing chords on a guitar or passing a ball to a teammate in soccer), they become second nature and allow the player to focus on high-level objectives such as playing a new song or scoring a goal. This type of learning is found in many biological systems, and yet has few analogues in engineered systems. I believe that there is a tremendous opportunity to reverse engineer this learning process by integrating adaptive control and reinforcement learning into our theory of layered architectures, and to leverage these insights to design self-tuning and adaptive cyber-physical systems.

4.2 Application

I plan to apply these novel theoretical tools to a broad set of application areas. Immediate areas of focus include applying my theory of layering, dynamics, control and optimization to (i) reverse/forward engineer control architectures for software defined networks and (ii) to reverse engineer aspects of the
human sensorimotor control system. I also eventually hope to explore applications of these ideas to the smart-grid and to reverse engineering other biological systems and processes.

**Software defined networking** Software defined networking (SDN) is a huge paradigm shift in the networking community. A defining feature of SDN is the abstraction introduced between the traditional forwarding (data) plane and the control plane. This abstraction allows for an explicit separation between data forwarding and data control, and provides an interface through which network applications (such as traffic engineering, congestion control and caching) can programmatically control the network. This in turn allows for diverse, distributed application software to be run using diverse, distributed hardware in a seamless way: in essence, SDN enables the implementation of a network operating system. This added flexibility leads to new architectural and algorithmic design challenges, such as deciding which aspects of network functionality should be implemented in a centralized fashion in the application plane, which components of network structure should be virtualized by the control plane, and which elements of network control should remain in the data plane. In principle any combination of centralized, virtualized and decentralized functionality can be implemented via SDN.

The Layering as Optimization (LAO) decomposition approach to Network Utility Maximization (NUM) problems is widely regarded as the “theory of architecture” for networking. However it is important to note that the LAO/NUM approach to architecture design was developed in the pre-SDN era, and hence does not incorporate the added flexibility and elasticity that SDN affords to network control applications. In particular, LAO/NUM problems focus exclusively on solving static network resource allocation problems that do not explicitly incorporate transient performance or fast-time scale dynamics. In [8], we show how our new dynamic theory of layered architectures [1] can be used to combine the LAO/NUM framework with distributed optimal control, and argued that it should be viewed as a natural generalization of NUM to the SDN paradigm. We applied this novel framework to a novel joint traffic engineering/admission control problem, and showed that a hybrid SDN approach in which a modified traffic engineering problem is solved in the application plane and a distributed admission controller is implemented in the data plane leads to robust and efficient network behavior that outperforms both traditional distributed and fully centralized SDN approaches.

I am currently aggressively pursuing this application area, and in doing so have fostered collaborations with industry and academic experts. David Meyer, CTO, Chief Scientist and Brocade Fellow at Brocade Communication Systems, has been a strong supporter and eager collaborator on this project. Kevin (Ao) Tang and his group at Cornell University are currently building and testing both packet-level simulations and hardware testbeds to implement and evaluate our novel network control schemes. Finally, this line of work was a main component in establishing a collaboration between Huawei and John Doyle – in addition to the funding provided by Huawei, they also offer the possibility to rapidly implement and test network control schemes on large-scale experimental testbeds.

**Sensorimotor control** The human sensorimotor control system is remarkable in its effectiveness. It seamlessly maps high-level motion commands (such as drink a glass of water) to low-level physiology commands (such as flex bicep with force $X$) in such a way that the executed and planned motions are in near perfect agreement. It does so in a manner that is remarkably robust to disturbances using a highly constrained and distributed control system built from axon bundles (nerves) that are very heterogeneous (cf. Fig 2) in their physiology – this physiology in turn imposes heterogeneous communication delay and rate constraints on control signals. In [9] we use a toy model (a scalar unstable system with a quantizer and delay in the control loop) to explore hard limits on robust control that arise due to these physiologically imposed communication constraints, and are able to infer system design principles that are consistent with human physiology. When advanced warning of a disturbance is available (e.g., via vision), delay is less important allowing for resolution to be maximized, leading to nerves with many small but slow axons (with each axon representing a bit). In contrast, when little advanced warning is available (e.g., via touch), speed becomes crucial and resolution less so, leading to nerves with a few large and fast axons.
Of course, the human body is not described by a scalar system and a single controller acting over a quantized communication channel with delay: it is composed of a network of axons and muscles. Further, as we illustrated with our “live demo,” much of our conscious planning is done using simple virtual dynamics. I am currently pursuing the integration of the insights gained in [9] with our novel theory of layered architecture to understand why and how our virtual model plans are converted with such fidelity to physical actions. An immediate goal of this project is to provide a physiologically grounded first principles explanation for Fitts’ Law, which states that the time required to rapidly move to a target area is a function of the ratio between the distance to the target and the width of the target.

Select References


