Research Statement

Nikolai Matni

I believe that a successful and impactful research program in the information and decision sciences is built around four elements: (1) foundational theory applicable to a range of application areas, (2) computational tools allowing the insight and impact of this theory to translate to practice, (3) the use of these theoretical and computational tools to provide novel insight into and methodologies for timely application areas, and (4) outreach and pedagogy. This research statement describes my current progress and future plans in developing elements (1) - (3); my efforts and philosophy on outreach and pedagogy can be found in my teaching statement.

My work aims to develop novel foundational theory, rooted in optimization and control, for the design of large-scale cyber-physical systems (CPS) such as the power-grid, the internet of things and smart transportation systems. In contrast to the systems traditionally studied in control theory, modern day CPS are composed of a large number of interconnected, interacting and distributed subsystems. My work addresses this complexity by unifying, generalizing and extending foundational results of modern optimal control theory to the large-scale distributed setting. In particular, I developed theoretical and computational tools to address the system level challenges that arise when designing a large-scale CPS: these challenges and their solutions are described in Sections 1-3. Further, my theoretical framework has reached a level of maturity that allows it to be applicable to a wide range of large-scale systems, such as the power-grid and software-defined networks (recent and ongoing work in this respect is described below). The resulting theoretical and computational tools provide the first scalable, holistic and principled approach to the design of large scale CPS control systems.

1 A System Level Approach to Controller Synthesis

As modern CPS become increasingly dynamic and decentralized, feedback control systems will be essential in guaranteeing their reliable and efficient behavior. However, even seemingly simple decentralized optimal control problems are intractable (e.g., Witsenhausen’s counterexample), making it a challenge to extend foundational results of the field (e.g., the Youla parameterization and DGKF state-space solutions) to the distributed setting. A recent breakthrough was the identification of a broad class of tractable (convex) distributed optimal control problems by Rotkowitz and Lall in 2006. These systems satisfy a quadratic invariance (QI) property, and allow classical (Youla) based synthesis methods to be used to compute structured controllers. Although QI was an important step forward, due to its dependence on the Youla framework, the resulting controllers were not scalable to synthesize or implement. This lack of scalability has been a major deterrent preventing the adoption of these methods in practice, as it limits their scope to systems with at most hundreds of states.

We address this issue by developing the System Level Approach to Controller Synthesis [1, 2], in which we present an alternative to the Youla parameterization that is naturally suited to distributed (structured) controller synthesis. By rethinking and generalizing this foundational pillar of modern control theory, we are able to identify the broadest known class of convex constrained linear optimal control problems, of which QI distributed optimal control problems are a special instance. Perhaps the most important consequence of this approach is the ability to synthesize localized system responses and corresponding controllers, in which local control policies depend on local subsets of the global system model and state (cf. [3, 4, 5] and references therein). The notion of locality, which could not be captured by existing theory, allows us to

![Figure 1: Computational time needed for the synthesis of centralized, distributed (QI) and localized (SLA) LQR controllers as a function of system size.](image-url)
scale structured optimal controller synthesis methods to systems of arbitrary size: an example illustrating the benefits of our approach on controller synthesis computation time are shown in Figure 1. In Figure 2, we provide a graphical illustration of locality via a heatmap showing the map from a single disturbance to control action in centralized, QI-distributed and localized controllers for a 1-D chain.

In addition to the aforementioned locality constraints, the system designer can simultaneously impose multiple performance objectives, structural constraints on the controller, robustness to quantization in the communication and internal computation of the controller, and as described in the next section, limits on the actuation, sensing and communication complexity of the resulting controller. Thus, by generalizing the foundational results of QI and the Youla parameterization, the system level approach provides, for the first time, a unified framework for customized controller synthesis that is applicable to large-scale distributed systems.

2 Regularization for Design of Control Architectures

A system’s ability to provide robustness and/or performance is directly 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 [6, 7, 8] 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. This work was the first to provide a computational framework for the simultaneous design of actuation, sensing and communication architectures for dynamic output-feedback controllers, unifying and generalizing existing results in controller architecture design. We were also the first to observe that designing a controller architecture can be cast as finding structured solutions to a particular linear inverse problem, allowing us to draw on the rich theory developed in the structured inference literature to provide conditions under which optimally structured controllers are identified using finite dimensional convex optimization.

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 [8] the Best Student Paper Prize of the 2013 IEEE Conference on Decision and Control, the premier conference in the field. Finally, although this line of work was initially developed under the Youla/QI framework, we have since integrated RFD into the SLA to controller synthesis (cf., [1, 2, 3]), thus allowing controller architectures to be designed for large-scale systems.

By integrating the architecture co-design capabilities of RFD with the generality and scalability of the
SLA to controller synthesis, it appears that we now have, for the first time, a scalable, holistic and principled approach to the design of large-scale CPS control systems. As a proof of concept, we have applied these tools to a 200-state model of linearized swing dynamics (synthetic topology and parameter values) in [2], and are currently applying them to a larger and more realistic power system model. Our aim in this latter work is to provide a systematic study of the fundamental limits imposed on system performance by communication delay, actuation and sensing density, and sensor noise in the context of large-scale power systems: this is now possible thanks to our novel framework because we can compute and quantify the performance of optimal output-feedback distributed controllers at scale.

3 Layering to integrate optimization and control

The focus thus far has been on theory to inform the design of feedback control systems, i.e., systems that ensure robustness around a pre-specified operating point or trajectory. In modern CPS an equally important challenge is that of determining an optimal trajectory (or set-point), as measured by a user-specified utility function. Traditionally, the complementary tasks of (i) identifying an optimal trajectory with respect to a 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, are dealt with in a fairly independent manner. This ad-hoc design procedure results in a layered system in which there are little to no guarantees on overall system performance: for example, it is often the case that a static set-point is computed in a planning layer that has little to no information about the transient dynamics of the underlying system, and this set-point is then tracked by an independently designed feedback controller in the tracking layer.

I argue that in order to develop a truly integrated theory for the design of large-scale CPS, 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 [9] I generalize and unify distributed optimal control, model predictive control and the Layering as Optimization (LAO) framework to simultaneously incorporate optimization, dynamics and control. I show that by suitably relaxing a “master” control problem that jointly addresses determining and following an optimal trajectory, one can naturally recover a two-degree-of-freedom layered architecture composed of a low-level tracking layer and a top-level planning layer (cf. Fig 3). 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 “protocol” between the two layers that ensures that the planned trajectory can indeed be efficiently followed by the tracking layer (which is constrained by the closed loop system dynamics).

4 Future Plans

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.

With this increased design freedom comes the opportunity to develop a rich theory of network architecture that informs the design of more sophisticated, dynamic and reactive network control algorithms. To do so, we take inspiration from the LAO framework, which can be viewed as the first theoretical framework for network architectures that was able to both reverse engineer existing network protocols (e.g., TCP Reno and Vegas), as well as inform the design and forward engineering of novel protocols that outperformed the current state of the art (e.g., FAST TCP). Our aim is now to extend this approach to network architecture design to an environment in which dynamic, local and fast time-scale control can be integrated with global, slower-time scale control, as enabled by SDN.

My approach to advancing this research agenda is centered around three components: (1) the introduction of novel "reflex" layers to network control, wherein fast-time scale (e.g., order round trip time) fluctuations are explicitly addressed; (2) the integration of these novel "reflex" layers with more traditional network control tasks, such as traffic engineering, by exploiting my novel approach to layering; and (3) collaborations with domain experts for the implementation and evaluation of practical algorithms.

As a concrete example of progress made in this respect, I have developed and implemented (in collaboration with Ao (Kevin) Tang and his group at Cornell University) a novel approach to High Frequency Traffic Control (HFTraC) [10], which operates at the network layer at the timescale of round-trip time. By only controlling router service rates, HFTraC does not change routes and can easily work with any traffic engineering (TE) method which sets routing at a slower timescale. HFTraC’s objective is to minimize a weighted sum of queue length and flow rate fluctuation by utilizing available buffer space in routers network-wide, and therefore leads to significant performance improvement in terms of the tradeoff between bandwidth utilization, packet loss % and queueing delay. Another key component of this work is quantifying how the achievable performance of HFTraC is determined by the network architecture used to implement it (e.g., whether router service rate decisions are computed in a decentralized, distributed or centralized manner). This is because in such a high-frequency setting, the controller’s responsiveness and latency play an important role in determining the system performance. Therefore, in addition to introducing a novel network control algorithm, we also provide a systematic study and quantitative comparison of different network architectures that vary from fully centralized to completely decentralized. Finally, in order to validate the effectiveness of HFTraC, we implement and extensively evaluate its performance using several tools, including a custom designed experimental testbed, a Mininet emulator, a production wide area network (WAN) and a large-scale backbone network simulator.

Building on this success, I am currently exploring the integration of HFTraC with more traditional network tasks, such as traffic engineering (TE). Although this work is still in a preliminary stage, it has shown enough promise to secure funding from the Huawei Future Network Theory Lab for my advisor John Doyle. They are very enthusiastic about the proposed approach, and in addition to providing funding, are eager to implement and evaluate any algorithms that we develop on their testbeds.

**Additional application areas** Although I claim no expertise in the following areas, I have been or currently am involved in projects related to the (i) synthesis of treatment strategies for HIV and Cancer, (ii) reverse engineering of the human sensorimotor control system, and (iii) understanding the architectural design principles of cellular biology, and specifically, those of the heat shock response of E. Coli.
System Level Thinking  My research shifts focus from specific components (such as the controller) to the system as a whole: this approach has proved fruitful in developing novel approaches to controller and system synthesis, as described above. Going forward, I plan to explore the exciting research frontier opened up by the SLA to controller synthesis: I highlight here two particularly exciting directions. The first is to develop a theory of robustness, akin to that from the 90s centered around $H_{\infty}$ optimal control and $\mu$-synthesis, that is applicable to large-scale distributed systems. Such classical methods are standard in industries such as aerospace and process control, and by extending them to large-scale distributed settings, are poised to have significant impact on emerging areas such as smart-grids and intelligent transportation systems. The second direction is to integrate these ideas into a model predictive control (MPC) like framework, allowing them to be applied to nonlinear systems. MPC is another industry standard that has failed to gain widespread use in large-scale settings, something that I aim to change.

Finally, in order for theoretical advances to have broad impact across both academia and industry, we must provide easy to use and practical computational tools. To that end, an immediate priority is to develop a computational toolbox (e.g., in Matlab and/or Python) that allows for the system level approach to controller synthesis to be easily experimented with. Fortunately, the compositional and modular properties of the SLA makes it naturally amenable to software implementation.

Layering  In §3 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) understanding how actuation, sensing, and communication density, control effort cost and system dynamics constrain the high-level functional behavior of a CPS, (ii) exploring the use of distributed optimization and game theoretic tools (thus introducing economics and market-mechanisms) to solve the tracking penalty augmented utility maximization problem of the planning layer, and (iii) developing a method to recursively apply this approach to forward/reverse engineer multi-tiered layered 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, such as those described above.

Select References


