Making the Internet-of-Things a Reality: From Smart Models, Sensing and Actuation to Energy-Efficient Architectures

Paul Bogdan
Univ. of Southern California
pbogdan@usc.edu

Miroslav Pajic
Duke University
miroslav.pajic@duke.edu

Partha Pratim Pande
Washington State University
pande@eecs.wsu.edu

Vijay Raghunathan
Purdue University
vr@purdue.edu

ABSTRACT
Advances in the physical sciences and engineering enable the development of internet-of-things (IoT) to understand, interface / interact and engineer physical world (systems). However, the deployment of multitude of wireless sensors and agents spanning many application domains (e.g., environmental, healthcare, smart interconnected vehicles and trucks, smart buildings) leads not only to tremendous requirements for communicating massive amounts of multiscale heterogeneous data, but also calls for developing efficient strategies for mathematical modeling of the sensed data while overcoming the energy wall. To address these outstanding challenges, out-of-the-box approaches need to be explored. In this tutorial paper, we discuss these outstanding problems and describe a few far-reaching design methodologies that achieve the most sought after features in autonomous IoT, viz. intelligent mining, smart trustworthy sensing, closed-loop (networked) control, and energy efficiency.

CCS Concepts
C.1.2 [Multiple Data Stream Architectures (Multiprocessors)]: Interconnection architectures, G.2.2 [Graph Theory]: Graph algorithms, network problems, B.4.4 [Performance Analysis and Design Aids]: Simulation, C.2.1 [Network Architecture and Design]: Wireless communication.

Keywords
Internet-of-things, autonomy, compact yet accurate models, network control, networks-on-chip, manycore.

1. INTRODUCTION
Advances in the physical sciences and engineering enable the development of the internet-of-things (IoT) to understand, interface, interact and engineer the physical world (systems). Simply speaking, the living interconnection of billions of smart devices allows us to monitor and extract information about both the physical and the cyber communities.

High dimensionality, heterogeneity and nonetheless the intrinsic uncertainty of both the IoT structure and the interactions of the IoT with the physical world call for new breakthroughs in a number of disciplines of the entire stack (from logic and models, algorithms and control decisions to architecture). Embracing a distributed intelligence paradigm, in Section 2, we describe a mathematical framework and algorithmic tools for the IoT that allow autonomic data-driven discovery of models of the complex systems in order to provide both trajectory forecasting and control. Targeting to construct robust empirical models about the complex environments / communities in an autonomic fashion, in section 3, we discuss the main design considerations for architecting the energy-efficient IoT edge devices that can process continuous incoming streams of metadata and communicate only essential features. Deciphering the mathematical models of complex systems with ultra-low power IoT edge-devices enables also the development of a new breed of network control solutions (see Section 4) that span multiple smart cities domains (e.g., transportation, energy distribution, healthcare, water resources, environmental monitoring, emergency response, urban planning, and policy). To provide real-time closed-loop control decisions, we discuss in Section 5 how to overcome the memory bandwidth limitation, power inefficiency of large supercomputers and better exploit the parallelization capabilities dictated by the characteristics of the application domain and the structural features of control algorithms. Thus, we present a design methodology for massive many cores solving efficiently large-scale control problems. Section 6 concludes the paper with a brief discussion of future challenges.

2. Compact yet Accurate Mathematical Modeling of Complex Systems
The IoT paradigm refers to a “living” interconnection of uniquely identifiable smart embedded systems or things (e.g., sensors, actuators, mobile devices) with deeply embedded cyber capabilities for sensing, processing and communicating the accumulated large amounts of heterogeneous data about the world to computational nodes for real-time analysis, interpretation and determination of closed-loop control strategies. One fundamental promising advantage the IoT brings is the capability to autonomously and trustworthy process high volume, velocity, variety, veracity, variability and multi-value (heterogeneous) data for deciphering not only the structural and dynamical patterns of complex systems, but also for providing informed and robust control decisions. In the smart city domain, the IoT monitors the cross-dependencies between transportation, energy, and human mobility networks, constructs dynamical coupled equations for sensitivity analysis and determines the best control decisions (e.g., controls the human mobility by allocating the transportation vehicles under a partial car-sharing system paradigm such that the energy consumption and pollution are minimized). In the healthcare domain, the IoT infrastructure monitors the genomic, proteomic, metabolic and physiological dynamics [7], constructs mathematical models of the intrinsic multiscale interdependency and determines when and what interventions are needed to avoid frailty or the onset of diseases [10]. In the renewable energy and environmental impact management domains, the IoT can monitor and transmit information about the water flow and water quality of river networks, measure the energy usage of residential communities, construct mathematical models of the inter-dependency between the amount of residential...
energy required and the forecasted generated energy from river streams by controlling the run-of-the-river hydrokinetic power plants.

To achieve such goals, the IoT must possess a built-in intelligence / analytics capable of: (1) extracting prescriptive information in terms of patterns and statistical characteristics of the observed processes that lead to compact mathematical models (the term compact refers to models with minimum number of parameters / mathematical terms given the complexity of the investigated complex system); and (2) provide predictive power by constructing causal compact mathematical models that can explain what might happen to a set of variables or things of the whole if specific changes / perturbations are applied to other parts of the system. The synergetic coupling of physical and cyber worlds raises a few grand challenges [35]: (1) What are the appropriate compact model that captures the characteristics of cyber and physical processes and facilitates the analysis, design and optimization? (2) What compact yet accurate models should be developed to enable the design of large-scale autonomous IoT systems-of-systems? While answering these questions in entire generality is extremely difficult, in what follows, we will embrace a bottom-up approach by scrutinizing the mathematical characteristics of a single process, identifying the essential mathematical constituents and proposing a general master equation approach for constructing dynamical models [8]. This single process framework is used to propose a class of mathematical models when dealing with a collection of interdependent process.

To obtain reliable mathematical modeling reconstructions of processes involved in complex interdependent networks [35], we have to investigate and encapsulate in compact mathematical expressions the information about their main characteristics:

- Many processes display a rich spatial and temporal multi-fractal behavior [6][9]. The causal interactions between state variables of the complex system are heterogeneous and exhibit a varying degree of nonlinearity across space and time. Spatial and temporal multi-fractality is overlooked in mathematical modeling and parameter identification steps.

- Some processes exhibit an asymmetric non-Gaussian dynamical behavior [8][9]. More precisely, the dynamics of a state variable is predominantly characterized by two components: (1) The statistics of the magnitude of positive and negative increments of a process encodes information about the degree of asymmetry and nonlinearity, as well as, the influence of other state variables on overall dynamics of the state variable under scrutiny. Many complex processes exhibit positive (growth, birth) and negative (death/loss, death) bursts characterized by different distributions (e.g., exponential, Gaussian, α-stable) with significant implications on the mathematical expressions characterizing the form and structural couplings between state variables. The statistics of the magnitude increments dictates the mathematical expressions (linear, quadratic, power law) describing the rate of change in the state variable. (2) The statistics of the time intervals between which the stochastic process changes its value (inter-event times) dictates the short- or long-range dependence nature of the dynamics.

- State variables of complex systems can exhibit non-smooth (non-differentiable) dynamics [5][4].

To avoid postulating complicated mathematical expressions, that cannot be justified by the true dynamics, and account for the above-mentioned characteristics that are not fully considered in current modeling methods, we propose a mathematical framework consisting of the following strategy: (1) It analyzes the increments in the magnitude of a stochastic process and the increments in time / evolution of the stochastic process to determine if these two quantities are independent of each other. (2) It determines the main statistical properties of the increments in the magnitude and the increments in time / evolution of the stochastic process. More precisely, this analysis finds whether the increments in the magnitude of the stochastic process are well modeled by exponential, Gaussian, lognormal or α-stable distribution. Concomitantly, it determines whether the increments or the inter-event times in the evolution of the stochastic process exhibit strong variation and are well modeled by exponential, power-law or other probability distribution. (3) It incorporates the mathematical characteristics learned in the previous steps into a generalized master equation that is derived using the total probability theorem. A particular advantage of this approach is that the resulting mathematical (dynamical) equation governing the process is not obtained by postulation, but rather dictated by its intrinsic mathematical characteristics.

In many practical situations, besides capturing the long-range dependency (memory) of a process, it is also important to determine the degree of interaction / interdependence among a set of processes. Consequently, we developed a data-driven multivariate fractal model [35] and a wavelet-based algorithm for estimating its parameters. This new mathematical model improves the modeling accuracy of the dynamics of biological processes and is validated against medical observations.

To make the discussion more concrete, we consider the mathematical modeling within a brain-computer interface (BCI) and present a set of investigations on long-range memory and inter-process dependencies of a neuron-muscular process under the similar experimental settings of [35]. More precisely, i) we first collect the intra-muscular EMG (iEMG) signals from a healthy subject. The subject is asked to perform forearm movements. The iEMG signals are recorded at different sites of the forearm muscles. ii) Secondly, we identify three different models that (a) consider the long-range memory and their inter-couplings (i.e., spatio-temporal fractal or STF model), (b) capture only inter-process dependencies (i.e., no long-range memory) and (c) consider only long-range dependencies.
memory effect (i.e., assuming the coupling matrix as an identity matrix), respectively. iii) Next, we fit the models to actual measurements to gauge their capability of capturing the dynamics of the neuron-muscular process evolution.

Figure 1 shows the comparison between the actual measurements and the model output. The actual muscular activities recordings are collected from three different muscles involving in the forearm movements: a) extensor digitorum (ED), b) abductor pollicis longus, and c) supinator. All measurements (the blue lines in all three subplots) are made when the subject is asked to perform finger flexion at a consistent strength. To show the impact of long-range memory and spatial inter-coupling across different muscular processes that are collaboratively involved in the forearm movements, we first investigate the capability of models that capture only either one of factors, i.e., spatial coupling or long-range memory. The output of the model is generated by stimulating the identified models and retrieved parameters. We show the results associated with muscle ED, APL and SUP in Figure 1(a-c), respectively. The green line and the red line shows the predicted process dynamics considering long-range memory and inter-processing coupling, respectively. We can make the following key observations from the results: i) the predicted output is largely deviated from the actual measurements, suggesting the inability of both models to accurately capture how the muscular system evolves over time to sustain a specific forearm movement. ii) the model that captures only the spatial inter-dependencies has a tendency to smooth the predicted muscular system state transition compared to that of a fractal model. However, as one can notice in the collected measurements from all three muscles, the muscular process is non-smooth and rich in fierce state transitions (i.e., peaks and valleys). The failure to capture such statistical signature of the physiological processes, in some circumstances, might lead to severe medical consequences that impose irreversible damages to the patients because it is usually the rare but extreme events in physiological process that are introducing life-threatening events like sudden drop/rise of blood pressure or blood glucose. Smoothing the output might mask such events and cause malfunction of medical devices monitoring and controlling critical bio-markers. iii) In contrast, the output of fractal model, even without spatial coupling, behaves similarly as the actual process does in the following sense: it preserves the non-smoothness and all the extreme events of the processes. Interpreted in a practical setting, such model might introduce extra false alarm situations, but no ignorance of critical system transitions. Combined in the two models, the STF model capture both the spatial inter-dependencies across the processed involved and the long-term memory effects (i.e., how the past system states have an impact on current system evolution). We report the model output (the magenta lines) for all three muscles considered in Figure 1. (a-c). We can observe that the predicted output preserves the extreme events and fit to measurements well in all three cases, suggesting i) the evolution of muscular processes involved in forearm movements are driven not only by its past system states (i.e. memory effect) but also the system transition of synergic processes (i.e., spatial inter-coupling), thus calling for models that are able to capture both and ii) the proposed STF model is exactly one of such models. In summary this mathematical approach and results demonstrate an autonomic data-driven discovery of the true model.

3. Design Considerations for Architecting Ultra-Low Power IoT Edge Devices

Industry forecasts project that, by 2020, there will be around 50 billion smart devices connected to the IoT. A majority of these devices will be embedded systems that bridge the physical world with the world of computing. We refer to these devices as IoT edge devices [52,53]. A major challenge in realizing the vision of the IoT is powering these billions of edge devices. Due to the expense, inconvenience, or sheer infeasibility of wiring them, most of these edge devices are expected to be untethered and powered using batteries and/or energy harvesting. Further, stringent constraints on the device form-factor (and hence, the on-board energy storage capacity) exacerbate the problem as most of these IoT edge devices are required to have long operational lifetimes, from a few days to several years.

Fig. 2 shows a taxonomy [50] of IoT edge devices according to the power available to them and the longevity requirements. The first class of devices represents wearable devices (e.g., smart watches, fitness monitors, etc.), which have a longevity requirement spanning a few days to weeks. The next group comprises the dozens of set-and-forget (SAF) devices (e.g., home automation gadgets, water leak sensors, etc.), which are expected to last a few years without any user intervention. The next group of devices represents infrastructure and geophysical monitoring systems or IGMS (e.g., wireless sensors that monitor public infrastructure such as bridges, highways, and parking structures), which are semi-permanent devices whose longevity requirement exceeds several years. The last category consists of transiently powered computers or TPCs (e.g., RFID tags, smartcards, etc.), which are battery-less devices that depend solely on remote or ambient power supplies. As Fig. 2 shows, there is considerable heterogeneity in the power availability and longevity requirements of different IoT edge device types, and as a result, a one-size-fits-all approach to energy optimization will not work for all devices. This section presents an overview of the different aspects pertaining to the energy-efficient system design of different classes of IoT edge devices and highlights recent research that has proposed promising solutions to address these challenges.

3.1 Energy Harvesting for Efficient Energy Supply

The sheer number of IoT edge devices expected to be deployed in the near future presents a daunting task, i.e., reducing their post-deployment maintenance cost. This maintenance cost includes the cost of new batteries as well as the cost of disposing depleted batteries, labor, and the cost incurred due to system downtime. Addressing this challenge is paramount in order to enable the continued adoption of IoT devices at a brisk pace. The problem is exacerbated in SAF and IGMS devices due to their scale of deployment and operational environments. To address this challenge, energy harvesting has emerged as an attractive and increasingly feasible solution that can enable significantly prolonged operational lifetime and in some cases, even self-sustained, near-perpetual operation. Energy harvesters convert power from ambient sources into electrical power. Harvesting energy using photovoltaics (e.g., from sunlight) is the most widely adopted technique among all the different energy harvesting techniques, in part because of its relatively high power-density (e.g., outdoor environment monitoring [54]). Energy can also be harvested from other ambient energy sources such as RF waves (e.g., WiFi, television broadcast signals, etc.) [55], motion or

Figure 2: IoT edge device classification
vibration induced kinetic energy [56], thermoelectric generators (e.g., hot water pipes) [57], and even by scavenging the energy output from idle sensors [58]. Note that the choice of harvesting modality for a particular IoT device is dependent on its operating environment, form-factor constraints, as well as its power budget.

In addition to the energy harvester, an energy storage unit is an indispensable component in IoT edge devices. Many ambient energy sources have highly dynamic, uncontrollable energy output, making energy storage a key design consideration in the power supply architecture of the device. The properties of the energy storage unit are dictated by both the IoT application and the energy source of the particular IoT device. For example, most wearable devices are periodically recharged by the user. Additionally, these devices typically have a small form-factor. Therefore, high energy density is the most critical requirement for ensuring long lifetime and lithium-ion batteries are currently the dominant form of energy storage in these devices. On the other hand, in ambiently-powered IoT devices, where even a minuscule amount of energy must be harvested, high cycle efficiency or roundtrip efficiency is the critical trait. Super capacitors are promising energy storage components for such devices undergoing frequent charge-discharge cycles [55,56]. Despite these efforts to improve the efficiency and reliability of the power supply in IoT devices, it is not always practical or feasible to provide continuous power to the device under limited ambient energy availability and stringent form-factor constraints. Therefore, additional system-level techniques need to be developed for these devices to make them resilient and reliable in the face of power loss.

3.2 Leveraging Memory for Energy-Efficient Operation

As mentioned previously, SAF and IGMS devices are expected to have battery lifetimes of several years and, therefore, need to operate at extreme levels of energy efficiency. TPCs have a similar requirement of extreme energy-efficiency due to the fact that they are battery-less. This issue is further aggravated due to constraints on form-factor (which limits the capacity of on-board energy storage) and deployment location (which limits the type and amount of energy available for harvesting). A key observation in addressing this challenge for SAF and IGMS devices is that they typically operate in heavily duty-cycled mode. This results in the idle mode being responsible for the lion’s share of energy consumption in such devices. Most modern microcontrollers (MCUs) address the issue of idle power consumption by providing two kinds of sleep modes, namely, shallow sleep (fast wake-up and state retention) and deep sleep (longer time to wake-up and does not retain state). Both have lower power consumption than the active mode. An MCU enters and exits shallow sleep mode with little overhead but keeps on-chip memory elements powered-on in order to retain state and, hence, consumes significant power in the process. On the other hand, deep sleep consumes nearly zero power but requires the MCU to checkpoint the current system state to the on-chip non-volatile memory (typically flash). Hence, the overall energy cost of entering and exiting deep sleep is significant (energy intensive flash writes/erases), which deters MCUs from entering deep sleep for short idle periods. An ideal sleep mode should consume as low power as deep sleep and yet retain state with the low overhead of transitioning in and out of shallow sleep. Hypnos [8] is a step in this direction and architects a new sleep mode for MCUs that combines the state retention of shallow sleep modes with the extremely low power of deep sleep modes. It achieves this by scaling the MCU’s supply voltage to just above the SRAM’s data retention voltage when the MCU is in sleep mode.

Another challenge faced by SAF, IGMS, and TPC devices that are powered by ambient energy sources is the unreliability in power supply. The uncertainty in available ambient energy means that the device could suffer a sudden loss of power. Therefore, executing long-running applications is challenging in these devices due to frequent system reboots. Recent research has addressed the issue of enabling long-running computations in such transiently powered computers (TPCs) [60] by checkpointing system state before an imminent power loss (similar to entering deep sleep). Mementos [61] is one such solution that checkpoints system state to the on-chip flash memory. However, the energy intensive erase/write operations of flash eat into the amount of energy available for performing meaningful execution in each power cycle. Advances in memory technologies have seen the emergence of non-volatile memories (NVMs) such as ferroelectric RAM (FeRAM), magnetoresistive RAM (MRAM), resistive RAM (ReRAM), etc., that combine the flexibility and endurance of SRAM with the non-volatility of flash, all at a very low power consumption. Quickrecall [62, 63] and Hibernus [64] use microcontrollers integrated with FeRAM, similar to Ref. [65], to enable efficient checkpointing in TPCs. Quickrecall utilizes FeRAM as a unified memory, thus enabling in-situ checkpointing. Quickrecall also checkpoints on demand and, unlike Mementos, does not need to do any checkpointing-related activity periodically. Hibernus uses FeRAM as the on-chip non-volatile memory instead of flash. Although FeRAM is superior to flash, it is slower than SRAM due to inherent device limitations. However, SRAM is volatile and, hence, data present in SRAM needs to be check-pointed on imminent power loss. Therefore, a trade-off exists between checkpointing and the energy consumed for program execution. Ref. [66] aims to dynamically map a program’s section to FeRAM and SRAM such that the overall energy consumption is minimized. The research works described above adopt a joint hardware-software approach to checkpointing wherein software (along with hardware) performs the checkpoint operation. Recent interest in state retention for energy harvesting applications has resulted in the emergence of non-volatile processors (NVPs) [67], which are hardware-only solutions. NVPs integrate volatile memory elements with emerging NVMs that enable them to take a snapshot of the system state on power loss and restore it on subsequent power availability. The authors in Ref. [68] utilize an NVP to architect a TPC that contains no energy storage or converter. However, by design, NVPs bypass normal boot procedures on power restoration and continue executing instructions where they left off previously. For energy harvesting IoT devices that have little notion of time, continuing to perform a task (such as communication, data sampling, etc.) on power restoration is not always functionally correct, as the check-pointed state might be stale, and hence, invalid upon wake-up. Related research also explores different architectural choices for NVPs [69]. However, the impact that different architectures have on full-system energy consumption remains to be seen in these devices. Finally, recent work proposes programming models to address checkpointing-related consistency issues [70] in energy harvesting TPCs utilizing NVPs.

Energy-efficient operation is also critical for battery-powered wearable devices, especially since they are increasingly being used to execute computationally intensive applications. Many of these applications belong to the domains of recognition, data mining, vision, multimedia, and digital processing, which are inherently error-resilient in nature. Emerging design paradigms such as approximate computing [71, 72], which can exploit this intrinsic error resilience are promising solutions to address the issue of energy-efficiency in these devices.
4. Composable Architectures for Real-Time Control over Wireless Networks

The last decade has witnessed a massive proliferation of the wireless communication systems. Improvements in the performance and cost of wireless technologies have enabled the use of wireless sensor networks as an effective and low-cost means of (open-loop) monitoring in cyber-physical systems (CPS). This has allowed for (almost real-time) data delivery to data centers and system operators without the need for excessive wiring, thereby yielding gains in efficiency and flexibility.

Despite the tremendous promise, thus far embedded wireless networks have largely focused only on open-loop monitoring. Why is this important? Consider, for example, industrial automation and process control systems. Modern industrial systems require fast and effective adaptation to fluctuating market conditions and product diversification. This high level of adaptability can be achieved through the use of Reconfigurable Manufacturing Systems (RMS) featuring: (1) modular components that are easily integrated, scalable, and convertible in terms of functionality, and (2) dynamic control architectures that are distributed, fully modular, and self-configurable, making it a prime candidate for deployment of wireless control [36].

Accordingly, Industrial IoT systems are emerging as a practical means to monitor and operate automation systems with lower setup/maintenance costs; they also introduce a set of logical benefits by facilitating design of wireless setup/maintenance costs; they also introduce a set of logical benefits means to monitor and operate automation systems with lower efficiency and flexibility.

In this section, we present a review of a recently introduced control system design approach for reconfigurable machine tools based on modularized and decentralized Computer Numerical Control (CNC) [36]; this approach exploits a fully decentralized motion control architecture realized through a wireless network of individual axis controllers. In addition, we provide an overview of two programming abstractions we introduced in [38][39], where control functionalities (i.e., control task executions) are assigned to a group of nodes as a single component in order to deal with the inherent unreliability of wireless communication and system components, as well as support compositional control design.

We have committed to the use of Time-Triggered Architectures (TTAs) where communication and computation are scheduled at particular instances of time (i.e., time slots) [40]. This is well aligned with the existing industrial trends, such as the recent wireless standards for industrial automation (WirelessHART and ISA 100.11a). TTAs also simplify modeling of the closed-loop systems, since the network-induced delay is known in advance. Furthermore, TTAs enable natural integration of communication, computation and physical dynamics; closed-loop systems based on TTAs can be modeled as switched control systems [41], allowing for the use of existing techniques for switched-system analysis. However, in this case the system performance depends on communication and computation schedules, which have to be carefully designed and interleaved on a node-by-node basis.

4.1 Plug-n-Play Numerical Control for Reconfigurable Manufacturing Systems

In [36], we described some of the design challenges for PnP automation systems and Reconfigurable Machine Tools, where new system functionalities, such as adding new axes in existing CNC units, can be introduced without significant reconfiguration efforts and downtime costs. Although the focus of that work was on CNC control units, the proposed, fully decentralized motion control architecture, realized through a network of individual axis control modules, can be simply extended to support development of other RMS components. With the proposed design, reconfiguration of motion control systems is carried out by only presenting the controller on each axis with information about machine configuration and the type of axis (e.g., X, Y, or Z) that the controller is running, effectively enabling modularity, reconfigurability, and interoperability of the machine control system.

We implemented the decentralized architecture on low-cost ARM Cortex-M3 based boards [42] on top of the nano-RK real-time operating system (RTOS) [43], and showed that low-power wireless communication can be used for coordination of axis controllers by evaluating system performance on machining of a standard test piece defined in ISO 10791-7 [44].

4.2 Embedded Virtual Machine

To deal with unreliability of wireless links and nodes, as well as support run-time system expansion, in [37][38] we introduce the Embedded Virtual Machine (EVM), a powerful and flexible programming abstraction, where the tasks are assigned to a group of network nodes (referred to as a virtual component), instead of mapping them statically to a specific node at design-time. Our approach is to decouple the functionality (i.e., tasks) from the inherently unreliable physical substrate (i.e., nodes) as well as allow tasks to migrate and adapt to changes in the system (including the network). With the EVM, at every time instance, each control task is mapped to a node within the virtual component, and the appropriate routing is used to deliver information to and from the node. However, with changes in the network/operating conditions, change in the current design of control system that use multi-hop industrial networks, and in which centralized controllers with static routes are specified at design-time.

In this section, we present a review of a recently introduced control system design approach for reconfigurable machine tools based on modularized and decentralized Computer Numerical Control (CNC) [36]; this approach exploits a fully decentralized motion control architecture realized through a wireless network of individual axis controllers. In addition, we provide an overview of two programming abstractions we introduced in [38][39], where control functionalities (i.e., control task executions) are assigned to a group of nodes as a single component in order to deal with the inherent unreliability of wireless communication and system components, as well as support compositional control design.

We have committed to the use of Time-Triggered Architectures (TTAs) where communication and computation are scheduled at particular instances of time (i.e., time slots) [40]. This is well aligned with the existing industrial trends, such as the recent wireless standards for industrial automation (WirelessHART and ISA 100.11a). TTAs also simplify modeling of the closed-loop systems, since the network-induced delay is known in advance. Furthermore, TTAs enable natural integration of communication, computation and physical dynamics; closed-loop systems based on TTAs can be modeled as switched control systems [41], allowing for the use of existing techniques for switched-system analysis. However, in this case the system performance depends on communication and computation schedules, which have to be carefully designed and interleaved on a node-by-node basis.

4.1 Plug-n-Play Numerical Control for Reconfigurable Manufacturing Systems

In [36], we described some of the design challenges for PnP automation systems and Reconfigurable Machine Tools, where new system functionalities, such as adding new axes in existing CNC units, can be introduced without significant reconfiguration efforts and downtime costs. Although the focus of that work was on CNC control units, the proposed, fully decentralized motion control architecture, realized through a network of individual axis control modules, can be simply extended to support development of other RMS components. With the proposed design, reconfiguration of motion control systems is carried out by only presenting the controller on each axis with information about machine configuration and the type of axis (e.g., X, Y, or Z) that the controller is running, effectively enabling modularity, reconfigurability, and interoperability of the machine control system.

We implemented the decentralized architecture on low-cost ARM Cortex-M3 based boards [42] on top of the nano-RK real-time operating system (RTOS) [43], and showed that low-power wireless communication can be used for coordination of axis controllers by evaluating system performance on machining of a standard test piece defined in ISO 10791-7 [44].

4.2 Embedded Virtual Machine

To deal with unreliability of wireless links and nodes, as well as support run-time system expansion, in [37][38] we introduce the Embedded Virtual Machine (EVM), a powerful and flexible programming abstraction, where the tasks are assigned to a group of network nodes (referred to as a virtual component), instead of mapping them statically to a specific node at design-time. Our approach is to decouple the functionality (i.e., tasks) from the inherently unreliable physical substrate (i.e., nodes) as well as allow tasks to migrate and adapt to changes in the system (including the network). With the EVM, at every time instance, each control task is mapped to a node within the virtual component, and the appropriate routing is used to deliver information to and from the node. However, with changes in the network/operating conditions,
the node/routing assignment can change to adapt to the new
operating environment.

To maintain control and timing properties for each control task, the
EVM is capable of migrating control tasks to the most competent set
of physical controllers, while ensuring schedulability of control task
and the corresponding communication flows. In addition, the EVM
architecture, which is based on a FORTH-like interpreter running on
top of an RTOS, allows for runtime extension of the system
capabilities, along with the development of an automated design
flow illustrated in Fig. 2: from Simulink to platform-independent
domain specific languages (DSL), and subsequently, to platform-
dependent code generation. The three-layered design process
enables control engineers to design wireless control tasks with
centralized in-network processing, in a manner that is both largely
platform/protocol independent and extensible to different control
domains (e.g., process control, discrete, aviation, medical).

The use of EVM-based wireless networked control systems was
illustrated on several case studies in industrial automation (on a
FischerTechnik factory module with 22 sensors and actuators) and
process control (e.g., control of heavy oil fractionator) [38].

4.3 Wireless Control Networks

In [39], we introduced the Wireless Control Networks (WCN)
arborcne for decentralized networked control. With the WCN, instead of assigning the computation of the control law to a
particular node in the network, each node in the network acts as
a simple local filter, taking into account only the states of its
neighbors and the sensors in its vicinity. We showed that this simple
scheme causes the entire network itself to act as a dynamical
controller with sparsity constraints imposed by the underlying
network topology. In addition to being suitable for implementation
even on resource constrained nodes, this very simple scheme has
several important benefits. Since at every communication frame,
each node only depends on the transmissions of its immediate
neighbors in order to update the state of its filter, the WCN control
scheme effectively decouples data routing as well as scheduling of
communication and computation from the control design problem;
this results in a significantly simplified scheduling procedure. Even
more importantly, the WCN scheme is compositional in nature since
an existing design can be easily extended to control any new
subsystems added to the plant at runtime, without any effects on the
performance of the previously deployed controllers.

Unlike standard procedures for synthesis of networked controllers,
which are usually focused on minimization of the negative impact
network induced delays could have on the system, the WCN
synthesis procedure explicitly takes into account the plant’s
 dynamics and the network topology to produce the weights for each
of the local filters executed on network nodes (as shown in Fig. 5).
Depending of the desired performance metrics, we derived methods
for design of robust [39], optimal [45], and fault-tolerant controllers
[45] for any network topology. In addition, we designed methods to
program a WCN such that it behaves as an existing controller (i.e.,
to ‘embed’ an existing controller into a WCN) [46]. We also derived
a network synthesis procedure that ensures that a stabilizing WCN
exists [47][48] as well as that the WCN can detect when a subset of
its nodes are compromised [49]. Finally, the use of the WCN in
industrial process control was illustrated in several case studies,
including control of a distillation column over a multi-hop wireless
network [45].

Figure 4. EVM Design Flow [38].

5. Energy-Efficient Manycore Architectures
for IoT application

Since its inception in 1999 [1], the IoT paradigm has evolved from a
simplified vision of connecting things into a complex world-wide
heterogeneous network-of-networks (NoNs) sustaining the
observation, sensing and interaction of humans with the
environment and allowing the control / actuation of various physical
and cyber knobs for enabling smarter transportation / power /
energy / utilities infrastructures. This has contributed to the
emergence of the CPS paradigm that endows the system with real-
time control capabilities. Model Predictive control (MPC) plays a
fundamental role in CPS; for each control interval, the MPC
assesses the overall system evolution by performing state
estimation, anticipates where the system will go and determines the
best sequence of informed control decisions that will bring the
system to desired state (dynamic optimization) [2][3]. While
developments on MPC algorithms for either linear hybrid processes
[2] or nonlinear dynamics [4][5] are well underway, there is also an
urgent need for the design of efficient computational platforms that
can be embedded within this NoNs architecture to provide
intelligence within IoT. To provide real-time control decisions,
these computational platforms need to overcome memory
bandwidth limitation, power inefficiency of large supercomputers
and better exploit the parallelization capabilities dictated by the
characteristics of the application domain and the structural features
of MPC algorithms. Thus, we advocate for exploiting emerging
massive manycore chips for efficiently solving large-scale MPC.
The most important component of these huge manycore chips is the
overall communication backbone, which is predominantly
implemented through a Network-On-Chip (NoC).

5.1 Distributed Computing for IoT

To enable a CPS for large-scale systems, we envision a cognitive
hierarchical architecture that collects information about the sensed
physical processes and its architectural states (via monitored cyber
processes) for building dynamical models and determines real-time
optimal decisions by solving MPC problems. Providing this real-
time analysis, accurate decision-making and efficient closed-loop
control elicit numerous requirements on the computational
components: (i) Monitoring and mining increasing number of
physical and cyber processes to develop accurate and efficient
mathematical and computational models of the system dynamics;
(ii) Meeting the real-time constraints of closed-loop control requires
the exploitation of a high degree of parallelism and an efficient on-
chip communication infrastructure for manycore platforms.
Moreover, it is already shown that the distributed computing
infrastructure preferable for operating IoT based services over a
centralized computing infrastructure [6]. Distributed computing

Figure 5. WCN Synthesis Procedure.
architectures achieve shorter service latency, have lesser operating costs, and involve lower energy consumptions [6].

Considering all the above-mentioned facts, we advocate for distributing the control decision across space and time, instead of solving the MPC in a centralized fashion in a supercomputing center. We propose to endow the CPS with multiple computational units for solving MPC problems efficiently via network-on-chip (NoC) enabled massive manycore chips.

5.2 Network on Chip Architectures

To support effective data analytics, novel architectures such as new NoC platforms are needed to efficiently exploit massive scales of fine-grain parallelism inherent in large-scale IoT applications. In such platforms, both the computing cores and the inter-core communication architecture can be carefully designed and optimized to meet the computation and communication requirements of the considered IoT applications.

For NoC-based manycore platforms, the intra-chip communication infrastructure plays a crucial role. Efficient on-chip communication architectures strive to enable low latency data exchange and high throughput while using the least possible power and resources. Conventional NoCs use multi-hop, packet switched communication. The limitations and design challenges associated with existing NoC architectures are discussed in [27]. To overcome these limitations, a wide variety of novel NoC architectures have been proposed. In the following sub-section we briefly discuss these architectures.

5.2.1 Emerging NoC Interconnection Architectures

Recently NoC architectures employing high radix routers have been proposed [15][16]. In these high radix networks, multiple computing cores are connected to a single NoC router to enable a low average hop-count among the communicating nodes. However, designing power and delay efficient routers incorporating large number of router ports is a challenging task. Reconfigurable NoC architectures incorporating wires with clockless repeaters have been proposed in [17]. These NoC designs, referred as SMART architectures, restrict the operating frequency of interconnect to 1-2 GHz, involve high control overheads and require more complex router design than the traditional NoC routers. The concept of express virtual channels is introduced in. By using virtual express lanes to connect distant cores in the network, it is possible to avoid the router overhead at intermediate nodes, and thereby improve NoC performance in terms of power, latency and throughput.

Small-world graphs (SWGs) have a very short average path length, defined as the number of hops between any pair of nodes. The average shortest path length of SWGs is bounded by a polynomial in \( \log(N) \), where \( N \) is the number of nodes, making them particularly interesting for efficient communication with minimal resources[19]. NoCs incorporating SWGs connectivity can perform significantly better than mesh-like networks [14][20][21][22], yet they require far fewer resources than a fully connected system.

Despite significant performance gains, in the above-discussed schemes, the long-range links are designed with conventional wires. It is already demonstrated that the wireless links are more energy efficient than their wireline counterparts for long-range communications [22]. Hence, the performance improvements by using long-range wireless links will be more than that using wired links. The viability of on chip wireless communication has been already demonstrated through prototype developments [23][24][25][26]. However, in order to harvest its full potential more research is required to address various challenges in several areas including system architecture, circuit design, device fabrication and CAD tool development [27].

A comprehensive survey regarding various wireless NoC architectures is presented in [27], which shows the possibility of creating novel architectures by inserting on-chip wireless links. Design of a small world NoC with long-range wireless links is presented in [28], and distinct advantages of this approach in application-specific designs are shown in [29]. Design of a small-world network enabled WiNoC architecture with millimeter-wave wireless links has been demonstrated in [22].

5.2.2 Proposed NoC Architecture

Our work takes up a very fundamental problem encountered in the synthesis of CPS in the context of IoT (i.e., efficient computation of MPC algorithms) – and leverages the strengths of a small-world NoC-based manycore with on-chip wireless links to achieve the multifaceted objectives of high distributed computing throughput, high communication bandwidth among computing nodes, and energy-efficiency.

Fig. 6 presents an illustration of mapping the MPC task graph to various cores in an NoC [30], where C1 to C8 represent the processing cores. In Fig. 6, without the data recombination links, the task graph could have a tree-like connectivity. However, considering the data recombination links, the task graph no longer remains a tree. Since the terminal nodes in the original task graph could be mapped to disparate computing cores, we would need long-range links to quickly exchange data. This provides the motivation for exploring a small-world connectivity.

To measure the small-worldness of a graph we use the following metric:

\[
S = \frac{C}{C_{rand}}\frac{L}{L_{rand}}
\]

where, \( L \) and \( C \) respectively are the average shortest path and clustering coefficient of a topology. \( L_{rand} \) and \( C_{rand} \) are the same quantities of a randomly constructed Erdos-Renyi graph with the same number of nodes and edges. For a small-world network the clustering parameter is much larger than that of a random network while the average path length is similar. This makes the parameter \( S \) larger than 1. It has been shown in [31] that many real networks...
have small-world characteristics if the quantity $S$ is larger than 1. Fig. 7 shows the $S$ parameter of the graph for various prediction horizons (PH). As evident in this figure, the value of $S$ for each PH is greater than 1. Hence, we can claim that the task graph corresponding to each PH indeed exhibits small-world connectivity.

As explained through Fig. 6, the nodes (computing cores) and edges (communication links) of a manycore platform that is specifically designed for executing parallel MPC algorithm constitute a small-world graph. Hence, the small-world NoC enabled manycore architecture is the most suitable platform for the computation of the parallel MPC algorithm. Thus, our goal is to use the “small-world” approach to build a highly efficient NoC using wired and wireless links. Since the long wired interconnects are costly in terms of power and latency, we propose to use wireless channels for the implementation of the long-range links present in the small-world architecture.

We address a very general version of the MPC problem – and present an architecture optimized for computation workloads and traffic for different quality-of-result (QoR) scenarios (viz., different prediction horizons representing different levels of accuracy and granularity). Thus, our design can build efficient computational platforms that are able to determine the control strategies for large-scale systems that are closely monitored without the need for conventional long distance transmission and processing within supercomputers. In fact, we demonstrate increased efficiency of our architecture when subject to higher QoR requirements. This, we believe, has immense potential in the successful realization of CPS in the IoT era. More precisely, we consider the parallel formulation of nonlinear model predictive control (NMPC), analyze its computational and communication workload characteristics and exploit these patterns for designing highly efficient NoC-based manycore platform. To the best of our knowledge, this is the first attempt at designing an efficient NoC architecture targeted for solving large-scale NMPC problems.

6. CONCLUSIONS

This paper presents a comprehensive tutorial regarding various design challenges that need to be addressed to make the IoT a reality. By integrating mathematical modeling, embedded system, control and microarchitecture content, it will be possible to build an energy efficient, trustworthy and integrable IoT system.

7. ACKNOWLEDGMENTS

PB acknowledges the support by USA National Science Foundation (NSF) under Grant 1331610 and 1453860. PB would also like to thank to Yuankun Xue for valuable discussion and development of the ideas summarized in Section 2. PP acknowledges support from USA NSF grants CCF-1514269, CCF-1162202 and Army Research Office grant W911NF-12-1-0373. MP work was partially supported by the NSF CNS 1505701 grant and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy.

8. REFERENCES


