Computing for Control and Control for Computing

Xinkai Zhang
Department of Electrical and Computer Engineering
University of Nebraska-Lincoln
Lincoln, USA
xinkai.zhang@huskers.unl.edu

Justin Bradley
Department of Computer Science and Engineering
University of Nebraska-Lincoln
Lincoln, USA
jbradley@cse.unl.edu

***Abstract*—Computing can be thought of as a service provided to a system to yield actionable tasks enacted by physical hardware. But rarely is control thought to be in the service of enhancing computation. Consideration of that perspective is what motivates co-regulation, our framework for holistic cyber-physical control of autonomous vehicles. In this paper we elaborate on how co-regulation will enable the next generation of autonomous vehicles precisely because it considers computation as an enabler and consumer of autonomous behavior. We report on the latest advances in this space showing how co-regulation exceeds results in event-triggered, self-triggered, and fixed-rate control strategies yielding more robustness and adaptivity to changing and uncertain conditions – a requirement for next-gen autonomous vehicles. We then describe a co-regulated decision making algorithm based on Markov Decision Processes showing how full consideration of computational resource allocation can increase decision-making capabilities in uncertain environments.**

***Index Terms*—co-regulation, time-varying sampling, resource-aware control, planning, markov decision processes**

I. Introduction

In computationally-controlled vehicles, computation has long been thought of as a service, providing control inputs, decisions, plans, processing images, and otherwise serving the needs of locomoting the vehicle. Designers of planning, navigation, control, and perception algorithms make designs assuming computational quality of service requirements will consistently be met. The guaranteed deadlines serve as important assumptions in performance guarantees in those designs [1]. The larger goals being served by the vehicle are typically prescribed by users somewhere in the loop.

However, vehicles, robotics, and other Cyber-Physical Systems (CPS) are increasingly imbued with the ability to perceive, decide, and with varying levels of autonomy, utilize their physical mechanisms to augment goals, make appropriate decisions, and advance their mission. This suggests that the physical mechanism, its control, planning, navigation, and perception algorithms can also be thought of as a service. Thinking of autonomy as a service mirrors the human-based autonomic system that regulates body functions in response to stimuli and other largely unconscious phenomena [2]. In serving the unconscious goals of the body the human autonomic system adjusts its performance including breathing, cardiac regulation, vasoconstriction, and reflexes [3]. These performance adjustments reallocate blood and other resources where they are needed.

This work was supported in part by USDA-NIFA 2017-67021-25924.

If robotics, autonomous vehicles, and other CPS are to increase their capabilities, and abstract to higher functioning they will need similar structures capable of adjusting performance and reallocating resources as needed. It is with this in mind that we imagine control and planning as a service for computation, adjusting performance through reallocating resources as needed.

In this work we present two examples of how this could work. In the first example we describe a novel concept we have developed and matured called "co-regulation" [4], [5]. Co-regulation augments a traditional control model with a computational model allowing for a co-designed controller to adjust control performance and computational resources simultaneously, and with $O(1)$ complexity. Co-regulation has been impactful in the CPS, control, and real-time communities being adopted for both research [6]–[9] and deployment [10]–[14]. We also have applied this to many systems [4], [5], [15], multi-agent control systems [16], [17], and most recently have hand-designed, co-regulated controllers for a multicopter Unmanned Aircraft Systems (UAS) [18], with provable performance guarantees [19].

In the second example we describe preliminary work on a co-regulated decision-making algorithm based on Markov Decision Processes (MDP) allowing "recursively optimal" decisions to be made as resources are available. To accomplish this we introduce a new concept called a "variable resolution horizon" wherein a state space is rediscretized where valuable information is most likely to be obtained improving decision-making with limited computational resources.

II. Co-regulated Control

Most computer-controlled systems are distributed systems consisting of multiple nodes being executed by a microprocessor with a real-time operating system [20]. The operating system typically uses multiprogramming to multiplex the execution of the various tasks. The CPU time and the communication bandwidth can hence be viewed as shared resources for which the tasks compete [20].

To create the next generation of autonomous vehicles capable of adapting to a wide-range of conditions, conduct varied missions, and remain safe we must design systems that can dynamically adjust behavior and resources to match. This will require new, holistic models and analysis that link together characteristics of computing, control, and communication and provide provable performance guarantees. To enable this we introduce a novel, resource-aware control strategy, co-regulation, which is a coupled strategy wherein a traditional controller's desired performance (e.g., transient control response) is further adjusted by regulating or reallocating a linked resource (e.g., task execution rate) through feedback depending on computational requirements, system states, and surrounding environments. We show how this strategy relates to fixed-rate, event-triggered, and self-triggered sampling strategies. A comparative test of different sampling strategies on an inverted pendulum control problem is presented to analyze the control performances and resource utilization of each strategy.

A. *Overview of Different Sampling Strategies*

For control task, allocating computational resources takes the form of different sampling strategies resulting in the sampled-data control class of systems [21]. Fixed-resource/periodic design (ideally) guarantees a certain level of system performance and robustness – but comes at a steep cost of inflexibility and wasted resources. The fixed allocated resource are calculated offline and are developed for the estimated worst-case noise, obstacle, and maneuver scenarios [22]. For example, controllers are designed to meet margins of stability, transient response, and steady-state requirements, then implemented assuming a fixed rate [23]. If the holistic system and environment exceed the margins, sensitivity, or timing limits, performance and safety are no longer guaranteed [20]. This means, at best, reduced system efficiency or perhaps mission failure, but at worst, means instability or system failure in response to unanticipated circumstances. To achieve good system performances, mission-critical tasks should be executed at high rates to meet the desired precision. As a result, computing resources are wasted when the system states are far from worst-case situations.

Motivation from inefficient allocation of resources led to aperiodic sampling strategies, exemplified by event-triggered control where sensing and actuation is performed when needed. In event-triggered control a triggering condition is continuously monitored and when violated, a sampling and control cycle is executed [24]. These strategies can greatly conserve computational resources but suffer from the disadvantages of all event-triggered systems – that lack of new information cannot be distinguished from failure in detecting/communicating the event [25]. This limits robustness of this control strategy to changing environmental and network conditions and also results in difficulty developing a mathematical foundation for this class of controllers [26]. The research in self-triggered control provide a new type of sampling strategy, time-varying periodic sampling. This reaps design benefits of periodic strategies, but because sampling period changes it conserves computational resources much like aperiodic control strategies. In self-triggered control the next update time is precomputed during the control update based on predictions using previously received data and knowledge of system dynamics [24].

B. *Co-regulated Control Strategy*

Cyber-physical co-regulation is a new hybrid time-varying periodic sampling strategy that can dynamically vary the sampling period at runtime depending on computing demands and system state feedback [4]. The sampling rate and control inputs were simultaneously changed to adjust overall system performance. Figure 1 shows the co-regulation method developed for quadrotor UAS control in our previous work [18], where output from a computational model representing sampling rate is fed to the physical controller (wide-dashed line) which adjusts physical system performance accordingly. Simultaneously, output from the physical plant is fed to the computational controller (narrow-dashed line) which adjusts sampling rate in response to physical performance. The computational controller increases the sampling rate when physical state error becomes higher, and decrease the sampling rate when error drops.

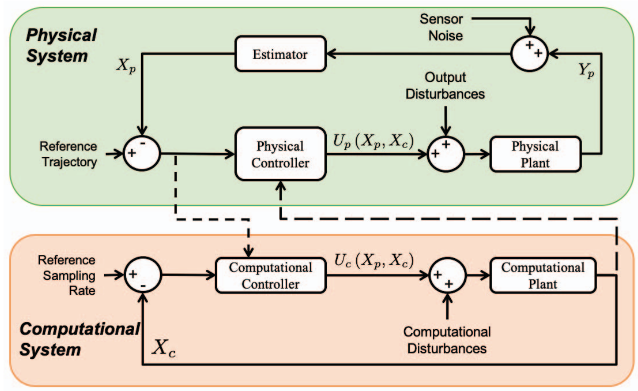


Fig. 1: Co-regulation Block Diagram [18]

The co-regulated system can be described by a stacked state-space system model, where we augment a traditional state space control model with a model of the computational control task

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \dot{x}_c &= u_c \end{aligned} \tag{1}$$

where subscripts c represents "computational". In this case, x_c denotes the control task execution rate of the physical system, which is regulated by the computational control input u_c.

Since the sampling rate dynamically changed at discrete intervals, the system matrices $\mathbf{\Phi}_p$ and $\mathbf{\Gamma}_p$ become functions of the time step k as they must be recalculated as the sampling rate varies. The resulting discrete-time-varying system model with respect to time step k is then,

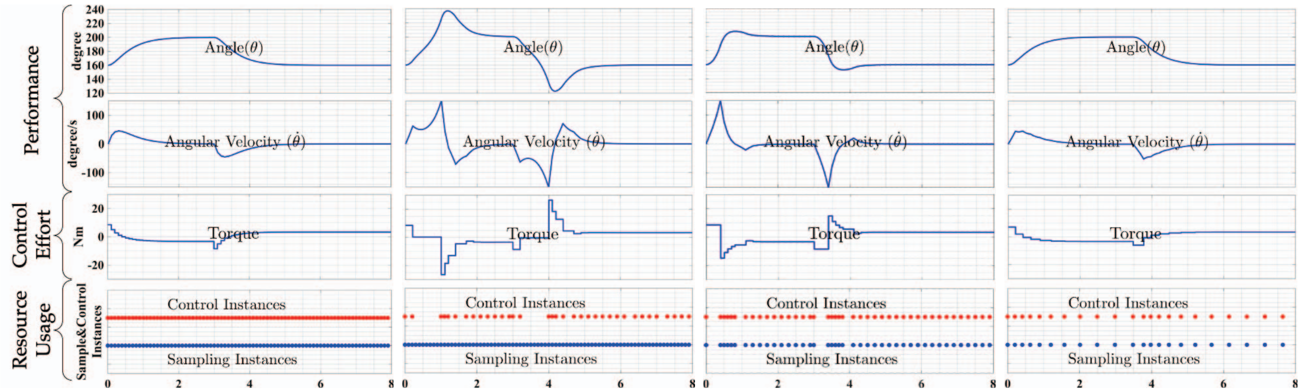
$$x[k+1] = \mathbf{\Phi}[k]x[k] + \mathbf{\Gamma}[k]u[k] \tag{2}$$

and the control input is

$$u[k] = -K[k]x[k] \tag{3}$$

To design the controller we employ a series of discrete optimal control gain matrices (DLQR) [21] for a sequence of sampling rates. The actual controller used is then "gain scheduled" corresponding to the current sampling rate. We call this a "Gain-Scheduled DLQR" control design [18]. Stability of the resulting co-regulated system was shown in [19].

Also, co-regulation needs to know when to take the next sample. We frame this as a feedback computational controller



(a) Fixed-periodic DLQR (b) Event-triggered control (c) Self-triggered control (d) Co-regulation

Fig. 2: Performance Comparison of fixed-periodic, event-triggered, self-triggered control, and co-regulation.

to calculate the coupled control input u_c, which adjusts the sampling rate, in real time, as the dynamics of the system change. In previous work [18] we presented a computational system control law as

$$u_c[k] = K_{cp}\left(x[k] - x_{ref}[k]\right) - K_c\left(x_c[k] - x_{c,ref}[k]\right). \quad (4)$$

The coupling gain, K_{cp}, is used to increase the sampling rate of the system in response to physical state error. The gain, K_c, drives x_c toward the desired reference sampling rate $x_{c,ref}$. To find appropriate values for K_{cp} and K_c, we employ an optimization scheme presented in [18]. Hence at current sampling instance k, the discrete-time computational system model can be denoted as

$$x_c[k+1] = x_c[k] + \frac{1}{x_c[k]} u_c[k]. \quad (5)$$

Hence the next sampling instance time can be calculated by Equations (4) and (5) based on the current state of the plant.

C. Comparative Test of Different sampling strategies

We conduct the comparison test of different sampling strategies by building a nonlinear simulation for an inverted pendulum system. The inverted pendulum is a simple and unstable system that requires control for stabilization, thus it has been widely used as a benchmark in control related research. The states of the system $x = [\theta, \dot{\theta}]^T$ are the angle and angular velocity of the pendulum, the control input $u = \tau$ is the applied torque. The nonlinear system model can be described by

$$\ddot{\theta} = -\frac{g}{l\sin\theta} + \frac{\tau}{ml^2}, \quad (6)$$

where $g = 9.8m/s^2$ is the gravity acceleration, $m = 1kg$ is the point mass, and $l = 1m$ is the length of the massless rigid rod. The inverted pendulum model can be linearized as

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ 9.8 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u. \quad (7)$$

We design fixed-periodic, event-triggered, self-triggered, and co-regulation controllers for this inverted pendulum model to compare and analyze the computational and physical control performances.

To analyze the influences from computing and timing, the controllers are all designed based on a unified optimal DLQR control algorithm. The control parameters were manually tuned for performance as $Q = 100 * \mathbb{I}_{2\times 2}$ and $R = 10$ since they can provide fast state converging. For fixed-periodic control, the sampling rate is set to $10\,\text{Hz}$ as it could provide a sufficient control performance for this inverted pendulum system. For event-triggered control and self-triggered control, the internal sampling interval T_d is set to $10\,\text{Hz}$. The event-triggered control algorithm is adapted from [27] to our inverted pendulum system. This algorithm updates the control signal once a the plant state deviates more than a certain threshold from a desired value. The self-triggered control algorithm is leveraged from [27]. This algorithm decides the next control updates at the previous ones, thus relaxing the need for continuous monitoring of the measurement error.

We conduct the comparison test in Matlab and evaluate the results based on the nonlinear inverted pendulum system response. The total time for each simulation test is set to $8\,\text{s}$. The initial states of the inverted pendulum is set as $x_{init} = [160, 0]^T$. The state reference is set to $x_{ref} = [200, 0]^T$ for time $0\,\text{s} - 3\,\text{s}$, and $x_{ref} = [160, 0]^T$ for time $3\,\text{s} - 8\,\text{s}$.

Figure 2 illustrate a favorable comparison of system performances from different sampling and control strategies. In general, fixed-periodic controller can provide the "gold standard" of control performance with a high computational resource utilization. Thus a closer performance to the fixed-periodic DLQR denotes better control. The sampling & control instances denotes the resource utilization for each strategy, where sparser instances indicates better computational efficiency. When compared with fixed-periodic DLQR controller (Figure 2a), the event-triggered control (Figure 2b) can save computing resources by reducing the number of control instances, but physical control performance is significantly degraded. Settling time of the event-triggered controller is 30% longer than fixed-periodic DLQR controller, along with a 87.5% higher overshoot. The self-triggered control (Figure 2c) can save more computing resources by reducing the number of both sampling instances and control instances. Performance of self-triggered control is better than event-triggered control. Settling time is approximately the same as fixed-periodic controller, but the overshoot of the self-triggered control is 20% higher.

The co-regulated controller (Figure 2d) provides significant improvements in system performances. The co-regulated system can achieve nearly identical physical control performance as the fixed-periodic control with higher sampling rates, which preserves advantages when compared with event-triggered and self-triggered controllers. Also, co-regulated system results in the least number of sampling and control instances among all four strategies, denotes the highest computational efficiency.

III. Co-regulated Planning

In deployed autonomous vehicles control, planning, perception, reasoning, learning, user interaction, and many other algorithms are required to create the kind of nextgen vehicles envisioned [28]. Co-regulation, while a demonstrated technology, only addresses low-level motion control of an autonomous vehicle. Planning for autonomous vehicles takes many forms including optimal control, graph search, automata, and monte carlo methods, each having various strengths and weaknesses [29], [30]. However, a common, shared weakness is the typically large computational resources required to find optimal solutions. Suboptimal, but complete, planners exist, such as rapidly exploring random trees and their variants, and play a critical role particularly in modern UAS autonomy [31]–[33]. The difficulty in creating feasible paths at runtime means the vast majority of commercial autopilots for both fixed-wing and rotorcraft UAS rely on the user to select waypoints which the UAS will then follow using a trajectory generation strategy (e.g., Dubins paths [34]).

Here, we introduce preliminary work on a co-regulated planner based on Markov Decision Processes (MDPs) capable of finding *recursively optimal* solutions to decision-making and planning problems given available computing resources. This, combined with co-regulated controllers will culminate in control and planning algorithms that adjust performance in response to uncertainty in environmental conditions, reallocating resources where they are most needed.

A. Markov Decision Processes

Similar to co-regulated control we need two things to build a co-regulated planner: 1) a performance-adjustable planning algorithm, and 2) a corresponding cyber controller. To make performance-adjustable planning algorithms that can be co-regulated alongside resources we seek modeling paradigms that can be solved iteratively, over time, similar to anytime algorithms [35]–[38]. This is especially appropriate for autonomous vehicles where new goals, hazards, or conditions can appear suddenly and the ability to adapt to them is imperative.

Markov Decision Processes (MDPs) are a widely explored, discrete stochastic control framework for solving decision-making and optimal control problems [29]. MDPs are described as a set of states (S), actions (A), transitions (P), and rewards (R). For each $s \in S$ and $a \in A$ there is a transition probability from one state to another represented by $P(s'|s,a)$ where s' is the resultant state. For each state, action pair a reward matrix, $R(s'|s,a)$, is defined which gives value to each transition. The solution to the MDP is a policy $\pi : state \longrightarrow action$ that prescribes an action to select in each step which maximizes the infinite-horizon reward.

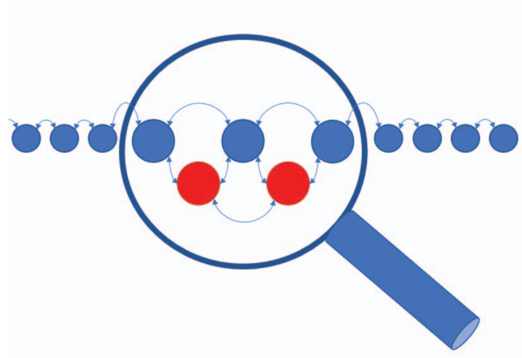


Fig. 3: Rediscretization of MDP state space

MDPs are traditionally solved offline resulting in an optimal policy which can be put into a lookup table and used for online decision making. However, their usefulness is limited by exponential state space growth sometimes rendering the MDP unsolvable by current computing resources [39] let alone at runtime. As a result, researchers have developed tools to reduce the state space and therefore computing resources necessary to solve for the optimal policy [40], [41]. MDP factorizations [42], [43], abstract MDPs [44], and hierarchical MDPs [45] all seek to reduce the state space making the MDP solvable with fewer computational resources – but still not iteratively solvable and hence unable to give back computational resources.

B. Variable Resolution Horizon MDPs

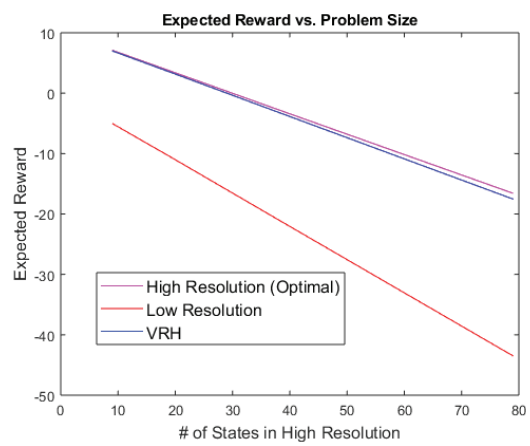


Fig. 4: Performance of VRH MDPs

Like many planning algorithms, the quality of the resulting plan correlates closely with model fidelity and available computing resources. We directly address this through a strategy wherein the model is regularly *rediscretized* according to informational proximity and utility [46], thereby shrinking and expanding the model to maintain higher fidelity as needed, while a receding horizon keeps the MDP solvable with available onboard computing resources (Figure 3). The variable resolution horizon (VRH) MDP is solved iteratively and available computational resources determine the length of the horizon and courseness of the rediscretization, thereby determining solution quality. The result is a performance-adjustable MDP planner providing "recursively optimal" [44] solutions. Figure 4 shows a preliminary exploration of this idea demonstrating that the VRH MDP proposed has only a 5% reduction in expected reward compared to a fully modeled, "flat optimal" MDP [44], but is computationally much simpler.

C. Co-regulated MDPs

With a performance-adjustable planner a corresponding cyber controller to reallocate resources based on performance is needed to complete a "co-regulated planner" design. In co-regulated control, the cyber controller allocates up to a maximum control sampling rate to the controller. Similarly, a cyber controller for a co-regulated planner needs to know how far out the planner must plan. Many vehicle missions or objectives can be broken into phases with varying intensity that tax different parts of the system. In steady flight, for example, 3-4 waypoints is a sufficient planning horizon for transport. But other phases of a mission may require intense planning based on more precise movements, or consider more environmental factors. This, in turn should modify the length of the planning horizon. We are currently working to address this need through the use of new metrics that measure the combined urgency of mission, environment, and upcoming mission phases are applying to the planning algorithm. As an example, high planner urgency would be composed of upcoming mission tasks and their intensity, control performance (good control performance implies less need to replan), and environmental factors. Allocation of computing resources are then computed by the corresponding cyber controller in response to this planner urgency. The result is an autonomous system that knows when to plan, how far in advance to plan, and the resolution of the plans needed to achieve mission objectives.

IV. Discussion

Humans possess the ability to self-assess, adjusting behavior through measuring performance and reallocating resources such as focus, time, and energy. A part of this ability stems from the autonomic system that can adjust performance and reallocate bodily resources rapidly in response to stimuli. Autonomous systems can measure performance to modify behavior but cannot adjust resources to meet new scenarios, goals, and environmental conditions. The proposed co-regulated control framework and co-regulated planner provide the tools, analytical framework, and algorithms for the scientific community to build on to create increasingly intelligent autonomous vehicles.

Co-regulated control provides a new resource-aware design for control in a highly integrated fashion. The primary benefits of the co-regulated control strategy in computing resource saving and physical control performance has been discussed based on the comparison test results. By coupling the computational and physical system controller, the co-regulated control strategy can provide a higher degree of co-designability, as well as a highly robust framework that can provide performance guarantees for different application scenarios. Similarly, co-regulated planning allows complex planning and decision-making problems to be solved iteratively when mission objectives demand it, and in response to available resources.

Our framework will ultimately provide a platform for deployment of "anytime," or similar, algorithms. Anytime algorithms are algorithms that trade computation time for quality of results [35]. When little computation is available, anytime algorithms can produce suboptimal results that may be "good enough." Anytime algorithms have the potential to work well in addressing the fundamental autonomy issues this work addresses. However, research in anytime algorithms generally focuses on the increasing quality of solutions when given more resources, but do not address the corresponding problem of reallocating the freed resources [47], [48]. Our framework provides these reallocation capabilities.

Ideally, an autonomous agent would be capable of reallocating resources where they are needed, adjusting performance of subsystems and sub-processes as needed. While we have focused our attention on low-level controllers with this capability many other aspects of a complex system must be addressed. Guidance [49], navigation [50], perception [51], run-time monitoring [52], learning [53], data collection and processing [54], and communication [17] all should be capable of adjusting performance in response to available resources.

Multi-agent, intelligent systems will be part of future scientific, agricultural, and military operations. Our work will enable hetero-/homo-geneous teams of agents to collect more data, at the right time, over longer durations, and handle changing network topologies and environmental conditions thereby increasing robustness. In the rapidly growing UAS segment, perception algorithms have large computational requirements and their performance can drastically impact control performance and cause failure if dynamic conditions change and resources are not reallocated to match. Our work will help meet these demands as well as targeted and persistent intelligence, surveillance, and reconaissance, flexibility and adaptability of agents to new environments.

References

[1] G. F. Franklin, M. L. Workman, and D. Powell, *Digital Control of Dynamic Systems*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1998.

[2] P. A. Low, *Primer on the Autonomic Nervous System*. Academic Press, 2011.

[3] G. Gabella, *Structure of the Autonomic Nervous System*. Springer Science & Business Media, 2012.

[4] J. M. Bradley and E. M. Atkins, "Coupled cyber–physical system modeling and coregulation of a cubesat," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 443–456, 2015.

[5] J. Bradley and E. Atkins, "Toward Continuous State-Space Regulation of Coupled Cyber-Physical Systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 60–74, Jan. 2012.

[6] S. Balasubramaniyan, S. Srinivasan, F. Buonopane, B. Subathra, J. Vain, and S. Ramaswamy, "Design and verification of cyber-physical systems using truetime, evolutionary optimization and uppaal," *Microprocessors and microsystems*, vol. 42, pp. 37–48, 2016.

[7] A. Nayak, R. Reyes Levalle, S. Lee, and S. Y. Nof, "Resource sharing in cyber-physical systems: modelling framework and case studies," *International Journal of Production Research*, vol. 54, no. 23, pp. 6969–6983, 2016.

[8] S. Wen and G. Guo, "Control and resource allocation of cyber-physical systems," *IET Control Theory & Applications*, vol. 10, no. 16, pp. 2038–2048, 2016.
[9] M. O. Elsedfy, W. A. Murtada, E. Farouk, and M. Gad-Allah, "Efficient design for satellite mission-aware multilevel queue scheduler," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 15, no. 1, 2017.
[10] B. T. Coloe and E. M. Atkins, "Cyber-physical optimization of small unmanned aircraft systems with thermal consideration," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 1373–1380.
[11] M. Eagon, Z. Tu, F. Fei, D. Xu, and X. Deng, "Sensitivity-based dynamic control frequency scheduling of quadcopter mavs," in *Autonomous Systems: Sensors, Processing, and Security for Vehicles and Infrastructure 2019*, vol. 11009. International Society for Optics and Photonics, 2019, p. 110090A.
[12] W. Liu, Q. Gong, H. Han, Z. Wang, and L. Wang, "Reliability modeling and evaluation of active cyber physical distribution system," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 7096–7108, 2018.
[13] C. Lv, X. Hu, A. Sangiovanni-Vincentelli, Y. Li, C. M. Martinez, and D. Cao, "Driving-style-based codesign optimization of an automated electric vehicle: A cyber-physical system approach," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 4, pp. 2965–2975, 2018.
[14] Y. Wang, D. Liu, and Q.-S. Li, "A hybrid system based cps model and control of loads in active distribution network," in *2016 IEEE International Conference on Power System Technology (POWERCON)*. IEEE, 2016, pp. 1–8.
[15] J. M. Bradley and E. M. Atkins, "Computational-physical state co-regulation in cyber-physical systems," in *ACM/IEEE Conference on Cyber-Physical Systems*. Chicago, IL: ACM/IEEE, Apr. 2011.
[16] C. Fernando, C. Detweiler, and J. Bradley, "Co-regulating communication for asynchronous information consensus," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 6994–7001.
[17] ——, "Co-regulated consensus of cyber-physical resources in multi-agent unmanned aircraft systems," *Electronics*, vol. 8, no. 5, p. 569, 2019.
[18] X. Zhang, S. Doebbeling, and J. Bradley, "Co-regulation of computational and physical effectors in a quadrotor unmanned aircraft system," in *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*. IEEE Press, 2018, pp. 119–129.
[19] X. Zhang and J. Bradley, "Stability analysis for a class of resource-aware, co-regulated systems," in *2019 IEEE Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 193–200.
[20] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzen, "How does control timing affect performance? analysis and simulation of timing using jitterbug and truetime," *IEEE control systems magazine*, vol. 23, no. 3, pp. 16–30, 2003.
[21] K. J. Åström and B. Wittenmark, *Computer-controlled systems: theory and design*. Courier Corporation, 2013.
[22] P. T. Kabamba and S. Hara, "Worst-case analysis and design of sampled-data control systems," *IEEE Transactions on Automatic Control*, vol. 38, no. 9, pp. 1337–1358, 1993.
[23] R. Isermann, *Digital control systems*. Springer Science & Business Media, 2013.
[24] W. Heemels, K. H. Johansson, and P. Tabuada, "An introduction to event-triggered and self-triggered control," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE, 2012, pp. 3270–3285.
[25] H. Kopetz, *Real-time systems: design principles for distributed embedded applications*. Springer Science \& Business Media, 2011.
[26] K. J. Åström and B. M. Bernhardsson, "Comparison of riemann and lebesgue sampling for first order stochastic systems," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 2. IEEE, 2002, p. 2011–2016.
[27] A. Eqtami, D. V. Dimarogonas, and K. J. Kyriakopoulos, "Event-triggered control for discrete-time systems," in *Proceedings of the 2010 american control conference*. IEEE, 2010, pp. 4719–4724.
[28] R. R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: the next computing revolution," in *Proceedings of the 47th Design Automation Conference*. ACM, 2010, pp. 731–736.
[29] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. Prentice Hall, 2010.
[30] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous uav guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1-4, p. 65–100, 2010.
[31] M. Kothari and I. Postlethwaite, "A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 2, pp. 231–253, 2013.
[32] T. Cui, Y. Li, and M. Z. Zhang, "Path planning of UAV based on the improved rapidly-exploring random tree algorithm [J]," *Electronic Design Engineering*, vol. 21, no. 12, pp. 15–19, 2013.
[33] Q. Sun, M. Li, T. Wang, and C. Zhao, "UAV path planning based on improved rapidly-exploring random tree," in *2018 Chinese Control And Decision Conference (CCDC)*. IEEE, 2018, pp. 6420–6424.
[34] I. Lugo-Cárdenas, G. Flores, S. Salazar, and R. Lozano, "Dubins path generation for a fixed wing UAV," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2014, pp. 339–346.
[35] S. Zilberstein, "Using anytime algorithms in intelligent systems," *AI magazine*, vol. 17, no. 3, pp. 73–73, 1996.
[36] P. B. Sujit and R. Beard, "Multiple UAV path planning using anytime algorithms," in *2009 American Control Conference*. IEEE, 2009, pp. 2978–2983.
[37] N. Vlassis, R. Elhorst, and J. R. Kok, "Anytime algorithms for multiagent decision making using coordination graphs," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 1. IEEE, 2004, pp. 953–957.
[38] R. Mangharam and A. A. Saba, "Anytime algorithms for GPU architectures," in *2011 IEEE 32nd Real-Time Systems Symposium*. IEEE, 2011, pp. 47–56.
[39] S. Lüdtke, M. Schröder, F. Krüger, S. Bader, and T. Kirste, "State-Space Abstractions for Probabilistic Inference: A Systematic Review," *Journal of Artificial Intelligence Research*, vol. 63, pp. 789–848, 2018.
[40] N. Meuleau, M. Hauskrecht, K.-E. Kim, L. Peshkin, L. P. Kaelbling, T. L. Dean, and C. Boutilier, "Solving very large weakly coupled Markov decision processes," in *AAAI/IAAI*, 1998, pp. 165–172.
[41] P. Poupart, *Exploiting Structure to Efficiently Solve Large Scale Partially Observable Markov Decision Processes*. Citeseer, 2005.
[42] T. Dean, R. Givan, and S. Leach, "Model reduction techniques for computing approximately optimal solutions for Markov decision processes," in *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1997, pp. 124–131.
[43] A. Kolobov, Mausam, and D. S. Weld, "Discovering hidden structure in factored MDPs," *Artificial Intelligence*, vol. 189, pp. 19–47, 2012.
[44] N. Gopalan, M. L. Littman, J. MacGlashan, S. Squire, S. Tellex, J. Winder, and L. L. Wong, "Planning with abstract Markov decision processes," in *Twenty-Seventh International Conference on Automated Planning and Scheduling*, 2017.
[45] S. Feyzabadi and S. Carpin, "Risk-aware path planning using hirerachical constrained markov decision processes," in *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2014, pp. 297–303.
[46] R. Munos and A. Moore, "Variable resolution discretization in optimal control," *Machine learning*, vol. 49, no. 2-3, pp. 291–323, 2002.
[47] I. Zliobaite, A. Bifet, M. Gaber, B. Gabrys, J. Gama, L. Minku, and K. Musial, "Next challenges for adaptive learning systems," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 1, pp. 48–55, Dec. 2012.
[48] D. Ferguson and A. Stentz, "Anytime, Dynamic Planning in High-dimensional Search Spaces," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 1310–1315.
[49] L. Pérez, Í. Rodríguez, N. Rodríguez, R. Usamentiaga, and D. F. García, "Robot guidance using machine vision techniques in industrial environments: A comparative review," *Sensors*, vol. 16, no. 3, p. 335, 2016.
[50] A. Ravankar, A. A. Ravankar, Y. Kobayashi, Y. Hoshino, and C.-C. Peng, "Path smoothing techniques in robot navigation: State-of-the-art, current and future challenges," *Sensors*, vol. 18, no. 9, p. 3170, 2018.
[51] C. Premebida, R. Ambrus, and Z.-C. Marton, "Intelligent robotic perception systems," in *Applications of Mobile Robots*. IntechOpen, 2018.
[52] A. Francalanza, L. Aceto, A. Achilleos, D. P. Attard, I. Cassar, D. Della Monica, and A. Ingólfsdóttir, "A foundation for runtime monitoring," in *International Conference on Runtime Verification*. Springer, 2017, pp. 8–29.
[53] C. Yang, C. Chen, W. He, R. Cui, and Z. Li, "Robot learning system based on adaptive neural control and dynamic movement primitives," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 3, pp. 777–787, 2018.
[54] L. Qingqing, F. Yuhong, J. P. Queralta, T. N. Gia, H. Tenhunen, Z. Zou, and T. Westerlund, "Edge computing for mobile robots: multi-robot feature-based lidar odometry with fpgas," in *2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*. IEEE, 2019, pp. 1–2.