2011 IEEE/ACM Second International Conference on Cyber-Physical Systems

Computational-Physical State Co-Regulation in Cyber-Physical Systems

Justin M. Bradley
Aerospace Dept.
University of Michigan
Ann Arbor, MI
justyn@umich.edu

Ella M. Atkins
Aerospace Dept.
University of Michigan
Ann Arbor, MI
ematkins@umich.edu

***Abstract*—From the perspective of physical system feedback control, the cyber or computer system's role has been to sample and compute control inputs sufficiently fast to maintain acceptable reference command tracking and disturbance rejection in the physical system. This strategy has been successful given the relatively low computational overhead for most control laws compared to computational resource availability. However, in many emerging applications this requirement may be insufficient, not because the computer is incapable of high-speed computations but instead because either more complex computations are required or because processor or network speed must be minimized to conserve energy. We propose the augmentation of traditional physical state models with a computational model to enable a cyber-physical system to co-regulate physical and computational actuation. Ultimately, our goal is to balance resources of the cyber system with quality of control of the physical system to provide a more energy-conscious CPS. As a first step, we propose a continuous-time representation of computational state and derive a continuous "dynamics" model approximation.**

Next, we propose the addition of a computational state into the closed-loop control law for the physical system states. Finally, we augment the derived cyber model with a second-order oscillator and demonstrate control via a LQR controller. In our simulation results, computational state and loop execution rate and oscillator "force" are regulated closed-loop at each control cycle based both on physical and computational state reference commands and errors. Results show that both physical and cyber state can be successfully regulated with the expected degradation in tracking performance as reference computational state (control loop rate) is slowed to values near the stability threshold.

***Index Terms*—Cyber Physical Systems; Real-time Systems; CPS Foundations**

I. Introduction

Cyber-physical systems (CPS) require the ability to manage both their computational and physical resources. In the context of the feedback control system, this means the CPS must achieve required tracking accuracy, disturbance rejection, and robustness levels through synergistic regulation of its physical effectors (e.g., propulsive, steering, switches) and computational effectors (e.g., processing and communication resources). Typically such analyses have been specific to one of these effector classes. Control systems engineers focus on regulating physical actuation but have developed techniques to account for the effects of limited computational resources. Conversely, real-time systems experts have focused on regulating computational resources but are still able to include physical control system performance metrics in computational resource scheduling.

Historically, from the perspective of feedback control systems, the energy required to actuate physical effectors has dominated energy requirements of the cyber system. However, in an era increasingly concerned with energy consumption, form factor, and extended endurance, the physical system must be made more aware of the cyber system and vice versa, a synergy consistent with awareness in biologic entities. With increasing demands placed on the cyber system for distributed data processing, communication, and decision-making, it has become imperative for any resource scheduler to be aware of if, and when, it can scale back resources devoted to reasoning about the physical system and still maintain good quality of control. In terms of performance of the physical system, an increase in resources allotted to the computation of control inputs will result in better performance of the physical system. This can occur either by scheduling additional CPU time for the control task, or by increasing the CPU frequency.

We are pursuing a representation and corresponding theory to unify these disparate notions of "effector regulation" into a common framework. In this paper we propose an incremental advancement toward physical-computational state co-regulation. We first represent computational state in a continuous-time state formulation and derive an approximate continuous model. We then add a control variable representing a single cyber-state to the physical state closed-loop control law. We then combine our derived cyber model with a 2^{nd}-order oscillator and demonstrate enhanced performance given a LQR control law under varying control loop execution frequency. We apply this closed-loop control law in the simulation of a weakly-coupled oscillator-computational state system to demonstrate performance as both physical and computational states are regulated.

In Section II we discuss background research in delayed and digital systems, forming a solid theoretical underpinning for sampled system analysis and representation. We begin with background on models that account for the time delay associated with a digital controller. Recent advances in CPS are then discussed, including related work in feedback scheduling and networked control systems, and we describe why this work

978-0-7695-4361-1/11 $26.00 © 2011 IEEE
DOI 10.1109/ICCPS.2011.27

could be a nontrivial advance within the CPS community. In Section III we review a simple spring-mass-damper system then illustrate the problem of digital control of dynamic systems. We then describe our model of computational state as well as an approximate continuous cyber model. We then propose a closed-loop control law that incorporates a single cyber-state and demonstrate how better quality of control can be achieved by utilizing cyber-state information. In Section IV we combine the physical and cyber models and discuss closed-loop controller implementation details in the simulated results in Section V. Results compare ideal (continuous control) responses with results from the simulated closed-loop CPS, illustrating how regulation of the computational state impacts the response of both the physical and cyber states over time.

II. Background

The approaches for understanding and solving problems in the area of CPS span a wide array of techniques from advancements in modeling physical systems, to advanced scheduling algorithms, to abstract mathematical formulations of hybrid automata. Ultimately, the goal is to appropriately balance the performance of the physical system and the performance of the cyber system. In the first part of this section, we discuss approaches for modeling delay in the physical system and why they do not fully achieve this goal. We then touch on some of the recent advances stemming from a cyber approach to the problem.

A. Modeling Delay

The primary effect of the cyber system on the physical system is the delay associated with a CPS. As a result, an investigation of how to model delay, as part of the state of the system, is important to our goals. Historically, three key research tracks have sought to extend rigorous mathematical formulations of physical systems to accommodate the effects of these delays. These include time-delay systems, digital control, and hybrid systems. Pertinent results from each area of study are highlighted below.

1) Time-Delay Systems: Time-delay systems research has played a prominent role in the definition, control, and stability of systems with delay. The primary difficulty in the development of appropriate tools for modeling these systems is a result of the infinite-dimensional nature of the problems. Hence, traditional dynamics (using ordinary differential equations) and by extension traditional continuous control are inadequate. Functional Differential Equations (FDE), with accompanying analysis, however, have provided a rich framework for investigation of such infinite dimensional systems.

The primary result of delay in a physical system is destabilization. Therefore, research into if, and when, a system becomes unstable has played a key role in this field. While some physical systems are, in fact, $\mathcal{S}_\infty$ stable (*delay-independent asymptotically stable*), most physical systems of interest are $\mathcal{S}_\tau$ stable (*delay-dependent asymptotically stable*). In $\mathcal{S}_\tau$ stability, we are interested in the τ^* (i.e. delay) that results in instability of the system, while values of $\tau < \tau^*$ are stable. Lyapunov stability, and more specifically Lyapunov-Krasovskii and Lyapunov-Razumikhin stability have motivated much of the stability analysis in this field. If we think of a FDE as an evolution in a Euclidean space, the application of Lyapunov's second method becomes more clear—namely, Lyapunov-Krasovskii stability tells us that the derivative of the candidate Lyapunov functional, $\dot{V}$ must be negative along all the system's trajectories. As in traditional nonlinear control theory, Lyapunov's second method is often surprisingly difficult to demonstrate. Lyapunov-Razumikhin stability relaxes the Lyapunov-Krasovskii stability theorem and seeks stability on a subset of trajectories defined by the system evolution an the interval $[t-\tau, t]$ [1]–[3].

There are at least two main obstacles in utilizing time-delay system theory in a CPS system to provide a unified framework. First, our purpose in modeling delay as a part of the CPS is to allow us to choose the optimal delay under changing conditions. While time-delay system analysis can help us analyze the range of stable delays up to τ^*, it has relatively few tools for handling time-varying delays and appropriately choosing them amidst control objectives. Second, while the delay is part of the system model, it does not function as one of the control variables. This means we still cannot utilize the rich theory and practical tools from the control community in our design of an energy-conscious CPS.

2) Digital Control: Time-delay systems analysis primarily considers a continuous delay term. In a CPS the delay induces a zero-order hold effect on the physical system. This effect is better suited to a purely discrete mathematical model than a FDE model [4]. Digital control provides this discrete mathematical framework, as well as familiar control techniques couched in a "digital" domain to design, simulate, and model a system.

Two traditional techniques arise from this area of study. The first is direct digital design. Assuming a fixed sampling rate, the former provides tools to derive a digital model of the system from which design, analysis, and simulation, can be achieved. Utilizing the z-transform, the left half of the s-plane is folded into the unit circle and we can conclude asymptotic stability if the system's eigenvalues (poles) reside within the unit circle. Z-domain analysis, including root-locus, Nyquist stability criterion, etc. are equally valid in the digital domain. State space equations (though now using difference equations rather than differential equations) can be formed, and compensator design, LQ optimal controllers, Kalman filters, etc. retain their familiar form and use.

The second method in digital control design is emulation of the controller/compensator. In this method all design and analysis is done in the continuous domain and the assumption is made that the cyber system sampling rate and control calculation is sufficiently fast to adequately control the system. A transformation using a selected sampling rate is applied to the controller to adapt it to the digital domain [5].

An important consideration in digital control is the selection of the correct sampling rate. It is clear from an energy usage standpoint that lower sampling rates require less energy.

However, they also contribute to the deterioration of system performance. The theoretical lower bound on sampling rate is the familiar Nyquist rate, or $\omega_s/\omega_b > 2$ where ω_s is the sampling rate, and ω_b is the bandwidth of the system. It is, however, well understood that a real physical system will perform poorly, won't have a smooth response, and will be highly sensitive to parameter variations at slow sampling rates. There is relatively little theoretical basis for correctly choosing the sampling rate, and primarily "rules of thumb" have been the norm. Some use simple trial and error techniques. Some have suggested a practical choice of the inverse of the largest real pole (or real part of complex pole). Others suggest the safe choice of $\omega_s/\omega_b > 20$ should suffice for most systems [6], [7].

No matter what sampling rate is chosen, a limiting factor in these methods is that the sampling rate is fixed. Similar to time-delay system analysis, digital control does not tell us how to optimally choose (under changing conditions) a delay, nor does it provide us with tools to treat the delay as a control parameter.

Both the time-delay systems and digital control areas approach the problem of a CPS system from the perspective of the physical system, but they do not address the need for the cyber system to regulate itself *in relation to* the physical system. That is, there is not a clear mechanism by which the cyber system can measure the stability or accuracy of the physical system as a function of its current control algorithm including all delays.

3) Hybrid Systems: Finite state representations have been employed in a variety of physics-based control systems, ranging from timed automata formulations for embedded system verification [8] to formal hybrid system models for dynamic system specification and control [9], [10]. Hybrid systems are capable of capturing discrete and continuous dynamics in a single framework and have been applied to a variety of applications. They provide the ability to model discontinuities through "jumps" between system states during which system state can undergo an instantaneous change in value without capturing this change within any particular state.

Formally, a hybrid system, H, is defined by the tuple $H = \{Q, \Sigma, Inv, J, Init\}$, where set Q is the discrete state set, Σ is the collection of dynamical subsystems associated with states Q, set Inv represents invariants that must be true to remain in a particular state, mapping J represents state transition behaviors, and $Init$ represents the initial conditions, discrete and continuous. For this work, we transcribe computational state to a differentiable representation using a simple hybrid systems formalism. We anticipate hybrid systems models will continue to be of use as we further refine and integrate our computational and physical models.

B. Recent Advances

A number of results emerging from the growing CPS community have also had a large impact on our understanding of coupled CPS. Anytime control, feedback scheduling, and networked control systems are particularly relevant to our work, covering a spectrum of topics related to the dynamic optimal control of the holistic CPS. Contributions in each area are highlighted below.

1) Anytime Control: Anytime control is an attempt to improve control accuracy as CPU time becomes available. These techniques are usually broken down into two improvement strategies: model reduction and performance reduction. In model reduction, the physical system is reduced by partial fraction expansion, modal reduction, or by weakly observable or controllable states. In this way, we can prioritize which control inputs, or how much control input should be calculated given the available resources. In performance reduction the performance of the system is prioritized according to some performance index and the corresponding controls to achieve the performance indices are computed as resources become available [11].

The seminal work by Bhattacharya et al. [12] adapted anytime algorithm techniques to controller design using model reduction and a smooth switching algorithm. Most recently this type of control has been extended to utilize an optimal LQG controller to meet performance criteria when the resources are time-varying and not known *a priori* [13]. In this formulation an unconstrained and constrained formulation are developed and the latter is shown to be an adaptation of Receding Horizon Control.

2) Feedback Scheduling: In contrast to anytime control, wherein a control algorithm is designed to offer increased control with increasing CPU time, feedback scheduling has become popular as a way of adjusting cyber resources based on the needs of the cyber system, including the control algorithm [14]. It is an attempt to adapt traditional control theory to the cyber system in order to regulate the CPS as a whole.

The feedback scheduler determines the appropriate management of resources and ideally allocates CPU utilization to the control task as it needs it. In this scheme models are needed that relate the sampling rate with the control performance [15]. Much work has been done by Cervin et al. in [16] to create a sound framework for feedback scheduling of control systems.

Such algorithms are often computationally intense. A model that directly incorporates the cost of control performance as it relates to cyber system resources would provide an excellent tool for feedback scheduling algorithms which can then utilize such information in choosing appropriate scheduling routines [17].

3) Networked Control Systems (NCS): In a *Networked Control System (NCS)*, feedback control loops are closed across a real-time network [18], [19]. Network communication is required to close a feedback loop whenever the sensor(s), actuator(s), and/or software are not co-located at the same physical processing unit. Typical NCS are faced with three primary problems: delay introduced due to limited bandwidth and competing control tasks, lack of synchronization between data integrated into each controller, and packet loss that may cause data to be unavailable for one or more control cycles.

Researchers have focused on a variety of issues associated with NCS, with much work focused on maintaining controller stability. Techniques from feedback scheduling (cited above) can be applied to assess the impact of delay on performance. Control scheduling techniques specifically addressing NCS issues have also been formulated. For example, [20] formally analyzes stability of a control system in the presence of delays and packet dropout, with a simulation also illustrating the effectiveness of clock synchronization compensation.

Our work is complementary to efforts in feedback scheduling and NCS. Research in feedback scheduling and NCS both offer a foundation for formally analyzing controller stability and in fact optimizing a real-time schedule over a set of controllers. This research, however, has focused on computing stability constraints through *offline* analyses, then capturing these contraints in a real-time processor and/or network scheduler. We instead focus on a more tightly-coupled regulation process, where the rate of control loop execution is in fact regulated at each control cycle based on errors in physical state as well as an ideal "reference" control loop rate that might be computed *a priori* using methods from NCS and feedback scheduling.

To this end, we directly augment our physical system state vector with a new "control loop rate" state. As described below, while the reference value for our new "cyber" (control loop rate) state may be determined from offline stability and performance analyses, the actual value of this state is regulated at each control loop cycle. We introduce a controller rate "knob" as a new forcing term augmenting physical force vector, $\mathbf{u}$, to provide this new regulation capability. The most straightforward application of real-time control loop rate regulation is perhaps for implementations on a processor with variable-speed clock. By varying clock rate, the controller can explicitly trade power consumed by the processor with the rate at which the controller itself executes. Our control scheme provides a direct clock speed regulation mechanism. While our scheme could also be applied to NCS, additional work beyond the scope of this paper would be required to ensure our commanded control loop "rates" were actually respected across all network elements.

III. Problem Formulation and Solution

In this section we describe progress toward unifying cyber and physical models into a single continuous-time framework that supports co-regulation of both models. A straightforward way of accomplishing this is to augment the physical system state model with one or more "computational" states representing the cyber system. This allows us to analyze the system using traditional feedback control system analysis and design techniques from the control systems community. Moreover, an optimal control formulation can be obtained that would balance constraints on both the cyber and physical portions of the system.

Thinking about the cyber system from the perspective of a physical one provides insight into how we might adapt it to a continuous model. For example, what is the "position" of the cyber system? This could be instruction count within a scheduled task, or perhaps progression in an anytime control algorithm. The "velocity" state could be as simple as a sampling rate, or it could correspond with the rate at which we are currently making calculations that specify or contribute toward an anytime or receding horizon control law.

In the following subsections we describe our initial work in this area with application to a simple spring-mass-damper oscillator system to illustrate how the regulated system is modeled and behaves.

A. Spring-Mass-Damper System

We consider a simple spring-mass-damper system as our initial application. A damped oscillator system may be represented as

$$\Sigma_p : \left\{ \begin{bmatrix} \dot{x}_{p1} \\ \dot{x}_{p2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{c}{m} \end{bmatrix} \begin{bmatrix} x_{p1} \\ x_{p2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_p \right. , \qquad (1)$$

where x_{p1} is the position and x_{p2} is the velocity of the physical system. For this system, throughout this paper, we have chosen $k = 39.4784$, $m = 1$, and $c = 1.2566$. The eigenvalues for the system are

$$\begin{aligned} \lambda_1 &= -0.6283 + 6.2517j \\ \lambda_2 &= -0.6283 - 6.2517j. \end{aligned}$$

Note that since all the eigenvalues are in the Left-Half Plane (LHP) the system is stable. We can also deduce that the system has a fairly slow response (observing that $\mathfrak{Re}(\lambda_i)$ is fairly close to 0), and the system will have moderate to large oscillations (observing that $\mathfrak{Im}(\lambda_i)$ is significantly larger than $\mathfrak{Re}(\lambda_i)$). Because the system is stable, our simulations in this paper will consist of the system response to the initial conditions

$$\mathbf{x}_p(0) = \mathbf{x}_{p0} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

The plot of the open loop response is shown in Figure 1.

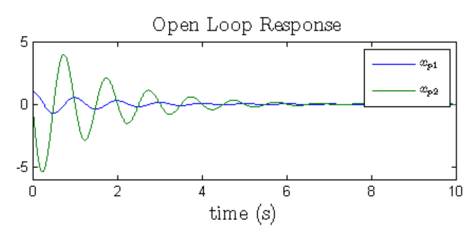

Fig. 1. Open Loop Response

1) Closed-loop Continuous Controller: Even an inherently stable system can be augmented with a feedback controller that provides a more desirable response (e.g., faster response time and/or less overshoot). For simplicity, and because we are only driving the system to its equilibrium in the simple simulations presented in this paper (i.e. $x_{p1} = 0$, $x_{p2} = 0$) we design a proportional-derivative feedback controller. Let

$$u_p = -k_{p1}x_{p1} - k_{p2}x_{p2}$$

be the control, where $k_{p1} = 3.5$, $k_{p2} = 2$ are gains chosen for a good response. Simulating the system we obtain the plot in Figure 2. Note that the response has improved significantly relative to open-loop.

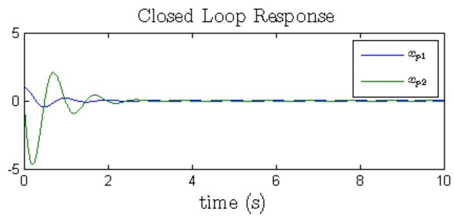

Fig. 2. Closed-loop Response

2) Closed-loop Digital Controller: The underlying assumption for the above closed-loop response time history (simulated in MATLAB) is that the controller samples at an infinitely fast rate. Of course in a real CPS it does not, nor does it need to. An appropriate sampling rate is typically chosen such that requirements for smoothness, stability, and tracking are met. It is often recommended that the sampling rate, ω_s, be chosen such that $\omega_s/\omega_b > 20$, where ω_b is the desired bandwidth [5]. If using digital control theory we transform our previous control u_p into a digital equivalent using $\omega_s = 62.83$ (which corresponds with a sampling period $T_s = 0.1$). We obtain

$$u_p\left(k\right) = -k_{p1}x_{p1}\left(k\right) - k_{p2}x_{p2}\left(k\right), \tag{2}$$

where $k_{p1} = -0.5537$, $k_{p2} = 1.7897$. The response is shown in Figure 3.

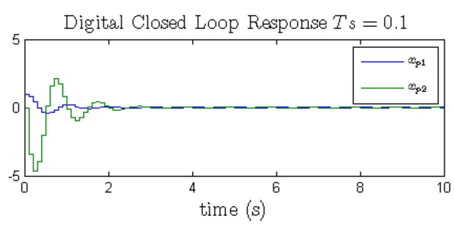

Fig. 3. Closed-loop Response of Digital System with $T_s = 0.1$

If a constant gain is held as sampling rate decreases, general response and stability decreases. Digital control theory allows us to analyze and design the controller using a fixed sampling rate T_s.

B. Cyber System Model

We now turn to the development of a cyber model that can be integrated into the physical system model. This prompts us to consider the cyber system from the perspective of a physical one. We desire a state, x_c, that represents the "position" of the cyber system, and whose derivative, $\dot{x}_c$, is the frequency. A natural choice for x_c is to think of instructions within the control task being executed as a linear function of time from the beginning of the task to the end. In reality, instructions are executed as a function of discrete time. We assume clock frequency sufficiently fast that such effects are negligible.

Definition 1: Let $x_{c,max}$ be the total number of instructions within the control task and let f be the frequency of executing instructions. Then the period of computing a control input u, is

$$T_c = \frac{x_{c,max}}{f}.$$

To simplify, we only consider the cyber system model Σ_c defined on the period T_c. Then

$$x_c = \{x_c = ft|0 \leq t \leq T_c\}.$$

This, then, clearly implies that $\dot{x}_c = f$ which is, indeed, a natural velocity of the cyber system. Our cyber model is then

$$\Sigma_c : \{\dot{x}_c = \left[0\right] x_c + \left[1\right] f.$$

In Figure 4 we show a plot of x_c as a function of time as it would appear in a cyber system. In this scenario we leave $f = 1$ GHz fixed in time. We now show x_c as function of

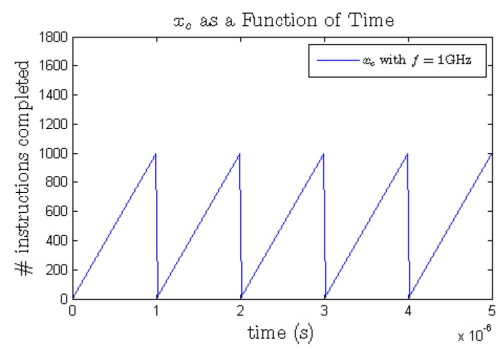

Fig. 4. x_c as a Function of Time

time with varying f. This is shown in Figure 5.

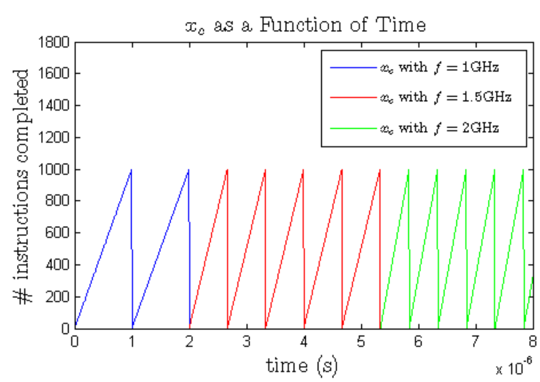

Fig. 5. x_c as a Function of Time with Varying f

Clearly this system is not realizable in continuous time. At each period, T_c, however, it is closely approximated by a double integrator. Let x_{c1} be the "position" of the cyber system, and let x_{c2} be the "velocity" or frequency. Then we can represent the cyber system as

$$\Sigma_c : \left\{ \begin{bmatrix} \dot{x}_{c1} \\ \dot{x}_{c2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_{c1} \\ x_{c2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_c \right. \tag{3}$$

1) Quality of Control of the Physical System: To effect how quickly a control input, u, is computed, a real-time scheduler can adjust the control task to receive a larger (or smaller) fraction of available CPU time. In our continuous CPS model, this corresponds to adjusting the frequency, f, of the control loop, and with our model this rate can be updated as frequently as physical force is regulated, once per control loop cycle. In this manner, our controller directly influences the quality of control of the physical and computational states through closed-loop regulation.

In this section, we examine how the quality of control of the physical system is affected by changing the frequency of the cyber system. As an example, it is instructive to observe the effects of changing the sampling rate in a digital system. In Figure 6 we show the response of the spring-mass system for several sampling rates. The sampling rate, in part, determines the bandwidth of the system, and hence to which disturbances, noise, and reference commands the system can respond. Robustness of the specific control law aside, the faster the sampling rate, the more robust the system will be (in general). As sampling rate decreases the system suffers from increasing overshoot, slower rise times, and otherwise degrades in performance.

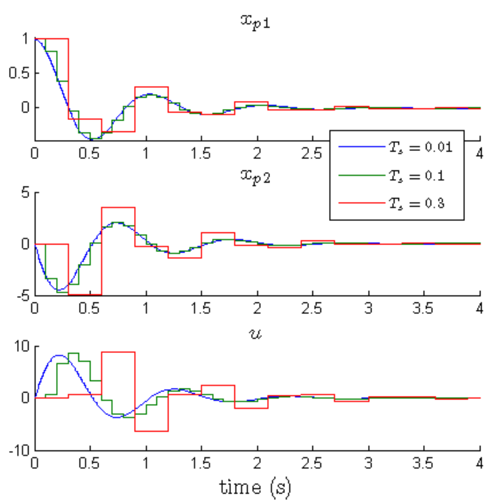

Fig. 6. Effects of Varying Sampling Rate on the Response of the Spring-mass System

To measure the quality of control, ρ, of the digital system we propose the average error of the response to some initial conditions over a range of sampling rates. Let $n_i =$ the number of samples for the simulation of the system with period $T_{s,i}$. Also define $\mathbf{e}_{p,i} = \mathbf{x}_{p,r} - \mathbf{x}_p$, where $\mathbf{x}_{p,r}$ is the reference input. Then

$$\rho_p\left(T_{s,i}\right) = \left(\frac{1}{n_i}\sum_{k=1}^{n_i}\left\|\mathbf{e}_{p,i}\right\|^2\right)^{-1}$$

where ρ_p is the quality of control of the system. For our system above, using the gains $k_{p1} = -0.5537$, $k_{p2} = 1.7897$, we vary T_s from 0.001s to 0.4s. The results are in Figure 7.

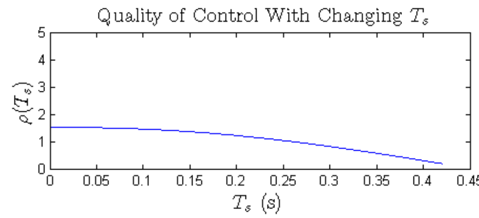

Fig. 7. Quality of Control with Changing Frequency

We note that this curve appears to coincide with results from other researchers in this area [21].

C. Adding a Cyber Control to the Physical System

Above we have described the addition of a computational state to the continuous-time state vector traditionally used for physical system control. In this section, we introduce a new "virtual" control effector to enable the physical system to depend upon the cyber-state. In this simulation, the physical system is controlled via a closed-loop control law that includes a dependence upon the cyber-state, however, the cyber-state is controlled open loop. Computational frequency is a natural control choice for the cyber system. Let γ be the commanded sampling frequency of the cyber system, and let γ_r be the desired reference for the sampling frequency. When we designed the digital controller in Subsection III-A2 we selected our sampling frequency as $\gamma = 1/T_s = 10$. Therefore, from the perspective of the physical system, we would ideally choose $\gamma_r = 10$. Let $\mathbf{e}_p = \mathbf{x}_p - \mathbf{x}_r$ be the error between desired and actual physical states. We now write the control law as

$$u = \mathbf{K}_p\mathbf{e}_p k_\gamma\sqrt{\frac{\gamma}{\gamma_r}}, \tag{4}$$

where $\mathbf{K}_p = \begin{bmatrix}-0.5337 & 1.7897\end{bmatrix}$ from above and $k_\gamma = 1.73$ is a tuning gain on the γ-term. Using the same mechanism as in Subsection III-B1, we simulate the system over the same range of frequencies. Figure 8 shows the results.

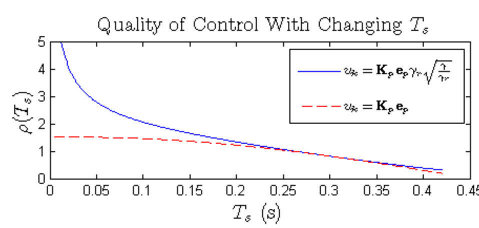

Fig. 8. Quality of Control with Changing Frequency

In this simulation the controller in (4) surpasses the quality of the controller from (2). The largest improvement in quality comes at the extremes of the frequency range, but in intermediate frequencies the benefit is still apparent. The above result illustrates how regulation of the control loop rate can in fact improve performance of the physical system. However, we note that the feedback law proposed above is nonlinear. Below we propose an augmented model to which linear analysis and optimal control can be applied thus providing the coupling and closed-loop regulation of both the cyber and physical states.

IV. Coupled CPS Model

In this section we introduce a linear CPS system and feedback control law, and show how it, together with a hybrid automaton, can properly represent the system of interest. We also describe a few details of our implementation in Matlab.

A. Continuous Dynamics

We can augment the physical model Σ_p in (1) with Σ_c in (3) to obtain a new continuous model representing the dynamics of both the cyber and physical portions of the system.

$$\begin{bmatrix} \dot{x}_{p1} \\ \dot{x}_{p2} \\ \dot{x}_{c1} \\ \dot{x}_{c2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m} & -\frac{c}{m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{p1} \\ x_{p2} \\ x_{c1} \\ x_{c2} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_p \\ u_c \end{bmatrix}$$

where u_p and u_c are the control inputs for the physical and cyber systems respectively. This system is unstable as seen from Σ_c already in Jordan form. However, it is controllable, and therefore stabilizable. In this model, the x_{c1} state representing the cyber "position" is not particularly meaningful outside of the dynamics equations. The physical system is affected by x_{c2} through the sample-hold mechanism inherent in a CPS. As a result, the coupling between the two systems is critical for proper co-regulation. We describe this coupling in part in the implementation Subsection IV-C, and in part in the LQR controller formulation in Subsection V-A.

B. Hybrid System Model

The system $\mathbf{\Sigma} = [\Sigma_p \Sigma_c]$ can be modeled as a hybrid automaton as shown in Figure 9.

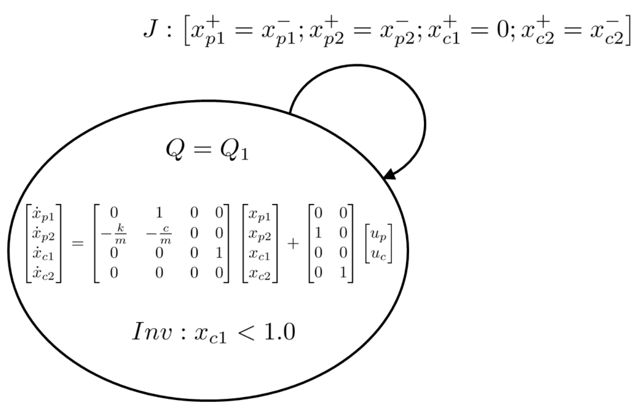

Fig. 9. Hybrid System Model

During each cycle, input vector $\mathbf{u} = [u_p, u_c]^T$ is constant since the modeled computations are required before a new update is available. At the end of each cycle, computational state "jumps" to zero, and input vector $\mathbf{u}$ is updated to its new value. The model shown here is preliminary but provides insight into how cyber and physical states can be integrated into a single continuous state-space representation that allows both cyber and physical states to be regulated in the same framework.

C. Implementation Details

The hybrid automaton in Figure 9 can be implemented to provide a simulation of the CPS. In our implementation, at each cycle, we use a 4^{th}-order Runge-Kutta variable time step ordinary differential equation solver. As x_{c2} (control loop frequency) changes according to the CPS system dynamics and desired reference frequency the length of a cycle, and hence range of integration, varies and appropriately captures the changing dynamics of the cyber system. The zero-order hold nature of the CPS is captured by holding the input constant until the next cycle. Additionally, a cyber system cannot have negative state values, although this would be allowed by our continuous state model. To address this, we set $x_{c1} = 0$ in our simulation as it currently has no bearing on the CPS due to the zero-order hold, and we artificially restrict x_{c2} to not fall below some reasonable threshold. Although this lower limit on x_{c2} introduces a nonlinearity into the system, it is analogous to saturation in a linear system and can be analyzed in that way. In our following simulation we limit $x_{c2,min} = 3.33$ as that is the frequency at which the spring-mass system response approaches instability.

To summarize we present the following pseudocode:

while $t < t_{max}$ **do**
  u=-K*$\mathbf{x}_{prev}$
  tspan=[t_{prev}, $t_{prev} + x_{c2}$]
  [t,x]=ode45(@CPSmodel,t_{span},$\mathbf{x}_{prev}$)
  t_{all}=[t_{all};t]
  $\mathbf{x}_{all}$=[$\mathbf{x}_{all}$;x]
end while

V. Simulations

In this section we present results from a coupled LQR controller design for our linear CPS to demonstrate the ability to apply traditional control techniques to the coupled system model.

A. LQR Formulation

We designed a traditional infinite horizon LQR controller which minimizes

$$J = \int \left(\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u} \right) dt$$

We add an integrator state to the system and the LQR gain is solved for the augmented system

$$\begin{bmatrix} \dot{\mathbf{e}} \\ \dot{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{C} \\ 0 & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{x} \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{B} \end{bmatrix} \begin{bmatrix} u_p \\ u_c \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k}{m} & -\frac{c}{m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Case	$T_{s,0}$	$T_{s,r}$	$x_{c2,0}$	$x_{c2,r}$
1	0.005s	0.0125s	200Hz	80Hz
2	0.3s	0.05s	3.33Hz	20Hz
3	0.02s	0.05s	50Hz	20Hz
4	0.025s	0.05s 0.1s @t=4.7s	40Hz	20Hz 10Hz @t=4.7s

TABLE I
TEST CASES

$$\mathbf{e} = \begin{bmatrix} x_{p1} - x_{p1,r} \\ x_{c2} - x_{c2,r} \end{bmatrix}$$

where $x_{p1,r}$ and $x_{c2,r}$ are the reference inputs for the physical and cyber system respectively. Additionally, while creating this controller, we introduced a small amount of coupling between the cyber and physical systems by forming

$$\mathbf{Q} = \left[\begin{array}{cc|c} q_{1,1} & q_{1,2} & \\ q_{2,1} & q_{2,2} & \\ \hline & & \mathbf{0} \end{array}\right]$$

In choosing appropriate $\mathbf{Q}$ and $\mathbf{R}$ matrices we faced a tradeoff. The double-integrator representing the cyber system can have fast rise time as would be expected in a cyber system. That is, if we desire a different frequency we can set it immediately (ignoring transmission and context switching/scheduling delays). This corresponds with a large $\mathbf{Q}$ and small $\mathbf{R}$. However, the double integrator will overshoot the reference frequency, a condition that worsens as proportional gains are increased. If we choose small $\mathbf{Q}$ and large $\mathbf{R}$ we can make a smooth transition with little to no overshoot. This of course is undesirable if the cyber system must rapidly respond to stabilize the physical system. Our chosen values of $\mathbf{Q}$ and $\mathbf{R}$ in the simulations shown below place higher importance on response time since we artificially limit the lower bound on x_{c2}.

B. Results

We present plots and analysis of four test cases given in Table I. $T_{s,0}$ is the initial sampling rate of the system and corresponds to $x_{c2,0}$. Similarly, $T_{s,r}$ is the reference sampling rate and corresponds to $x_{c2,r}$. In all cases we drive the physical system to zero. Each of the following test cases presents a comparison of the same LQR controller, one simulated continuously, and one simulated as a CPS. Note that in the CPS system the control loop rate state is used in real-time by the software to control the simulator's control output update rate as described above.

In the first case the initial frequency, $x_{c2,0}$ is over twice as high as $x_{c2,r}$. However, both values are sufficiently fast with respect to the destabilization frequency that the system suffers no significant loss of quality of control due to the co-regulation of physical force and control loop rate. Indeed, as seen in Figure 10 the physical x_{p1}, x_{p2} and cyber state x_{c2} very closely align.

In the next case, shown in Figure 11, the initial frequency is at the lower stability threshold $x_{c2,min}$ value, a condition that might occur in processing overload or critical energy

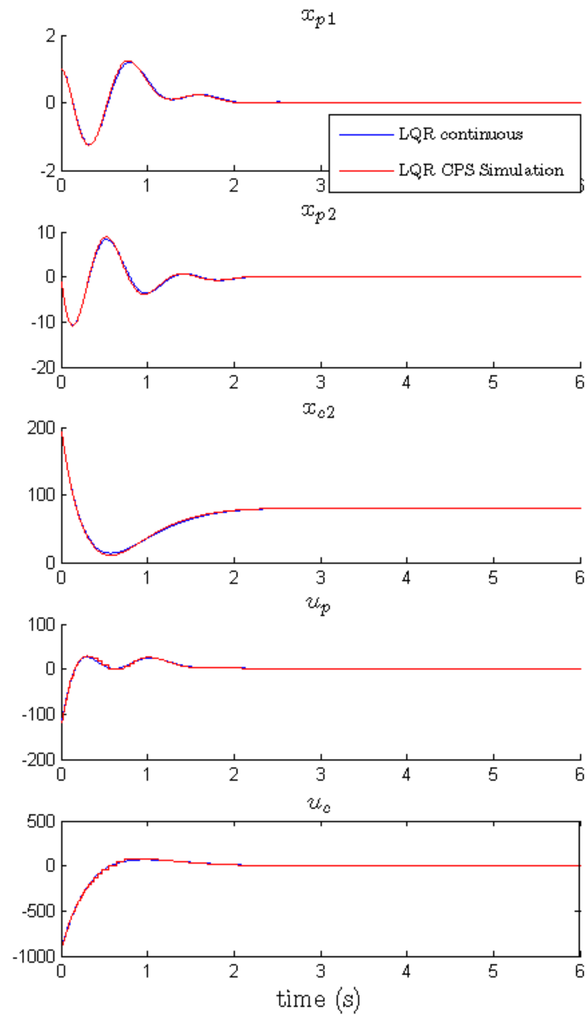

Fig. 10. Simulation for Case 1. $x_{c2,0} = 200$Hz, $x_{c2,r} = 80$Hz

conservation situations. To accurately respond, the CPS will push the cyber system faster to achieve higher quality of control of the physical system. Early in the above simulation x_{c2} is slow and the system does not have good quality of control resulting in the high transients. However, by $t = 2$s the system has sped up sufficiently and can provide high quality of control to drive the physical states to zero as designed. Additionally, x_{c2} approaches its nominal value $x_{c2,r} = 20$Hz once the system responds to this faster reference command. A closer look at the x_{c2} plot reveals that x_{c2} actually takes two cycles (at 0.3 s each) to respond. Due to the slow sampling rate, the feedback controller integral error term needs both cycles to obtain a control input u_c that pushes x_{c2} higher.

In case three we approach $x_{c2,r} = 20$Hz from a higher frequency. The results, shown in Figure 12, show how the cyber system, x_{c2}, overshoots the target, approaching $x_{c2,min}$ which starts to destabilize the system, due to our LQR preference for fast response time with sacrifice in overshoot. Within a cycle the controller drives x_{c2} higher, stabilizing the system, and

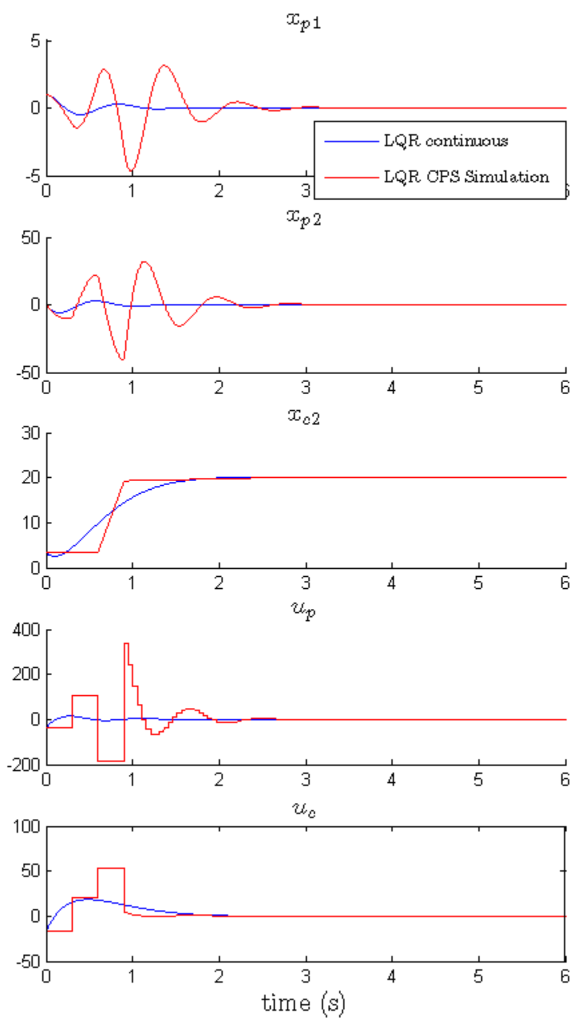

Fig. 11. Simulation for Case 2. $x_{c2,0} = 3.33\text{Hz}$, $x_{c2,r} = 20\text{Hz}$

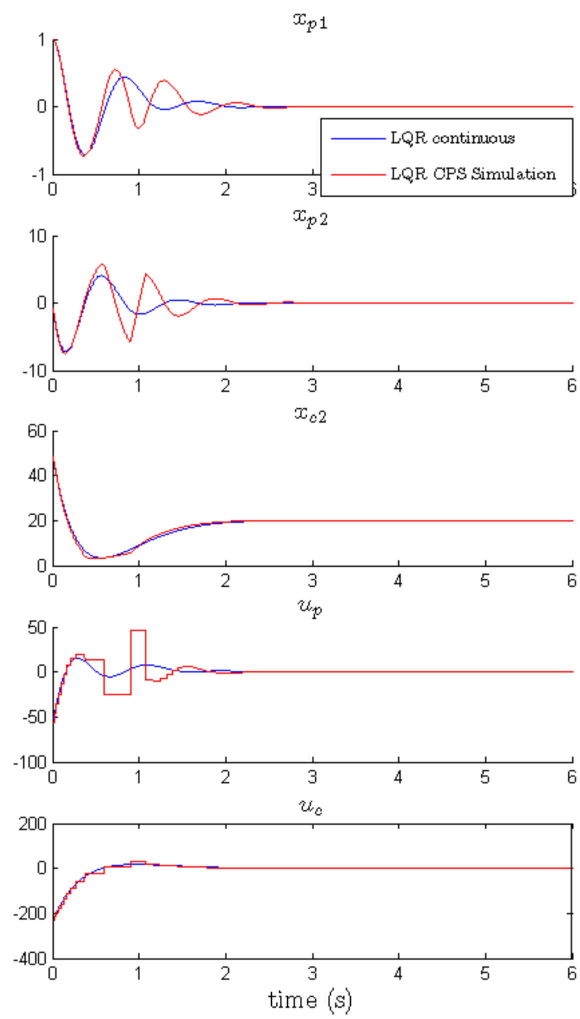

Fig. 12. Simulation for Case 3. $x_{c2,0} = 50\text{Hz}$, $x_{c2,r} = 20\text{Hz}$

ultimately driving the physical states to zero, their reference values.

In the final case we illustrate the impact of a single step disturbance (in physical velocity) followed by a changing control loop rate reference command. First, at time $t = 4.5$s we inject a nontrivial velocity impulse into the physical system. This requires that the controller regulate the physical states back to zero over time. Then, closely following at $t = 4.7$s we command the cyber system to a lower frequency $x_{c2,r} = 10\text{Hz}$ so that the system has to deal with the disturbance while attempting to move to a lower frequency. The plot is shown in Figure 13. While the CPS system is not as fast to respond as the continuous controller due to its low update frequency, it is still effective in regulating the disturbance. Because x_{c2} is not near the threshold for instability, the CPS is able to respond with sufficient speed. Note that x_{c2} in the CPS simulation almost exactly tracks the continuous controller. This example of disturbance-induced co-regulation is promising and consistent with our expectations that quality of control will be high so long as we do not approach the destabilization control loop rate.

VI. Conclusions

We have presented a novel CPS representation in which physical and computational systems are represented as a single continuous multi-variable linear system. This representation enables co-regulation of physical and computational state to optimally balance computational load with physical system stability and disturbance rejection at each control loop cycle. A simple 2^{nd}-order oscillator system is used to illustrate a "physical" model, while a single "control loop rate" computational state and virtual control "knob" are used for the "cyber" model. We have developed a simulation of our CPS and have shown that our proposed model represents the expected behavior of the corresponding CPS.

Much work remains before the proposed model will be of practical use as a complement to existing feedback scheduling and NCS strategies. First, although our CPS is "coupled" in

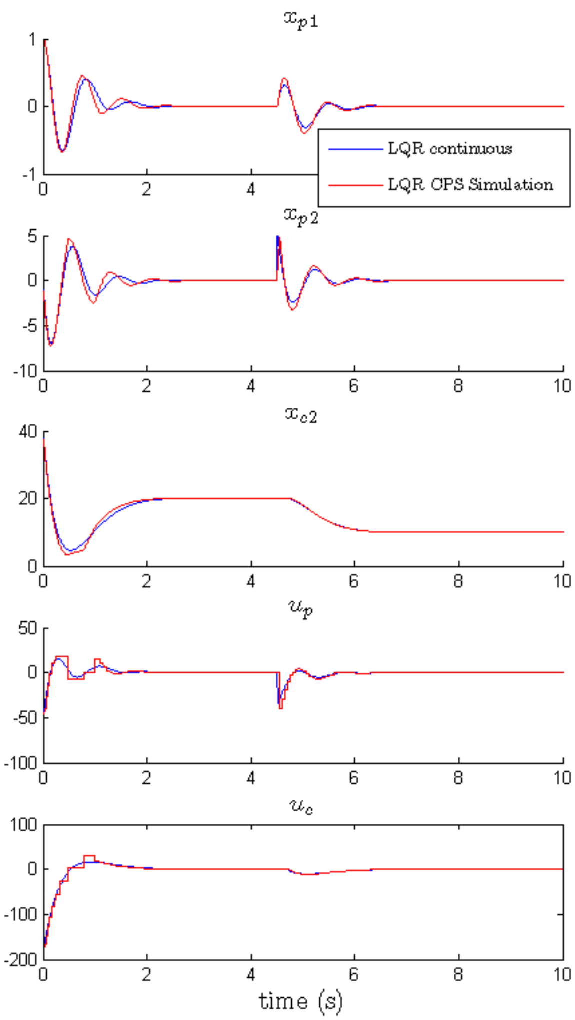

Fig. 13. Simulation for Case 4. $x_{c2,0} = 40$Hz, $x_{c2,r} = 20$Hz then $x_{c2,r} = 10$Hz at $t = 4.7$s while responding to disturbance.

that physical state errors influence control loop rates, and control loop rate influences applied physical force, this coupling is very weak in our current model, appearing only through low-gain feedback in our LQR formulation. Exploration of that coupling is needed to understand more precisely how the effects of sampling can be effectively utilized by the proposed continuous model. Further, LQR is only one of several possibilities for a feedback controller. Other options need to be explored. Finally, this work needs to be extended to more complex as well as marginally-stable or unstable physical systems.

Our long-term goal is to adapt these methods to challenges in air transportation and unmanned aircraft surveillance applications. The proposed next-generation airspace architecture encompassing air transportation and unmanned operations will rely on decentralized decision-making. Decentralization in turn implies increased demand for onboard computation. Particularly for surveillance operations in which high-bandwidth sensor data factors into the mission, computational and communication requirements will place ever-higher demands on resources. The co-regulation of cyber and physical controls will therefore be critical to overall system optimization.

Acknowledgment

The authors would like to thank Kang Shin and other cyber-physical systems project collaborators for their for their support. This work was funded in part under NSF grant CNS-0930813.

References

[1] L. Dugard and E. Verriest, *Stability and Control of Time-Delay Systems*, L. Dugard, Ed. Springer, 1998.

[2] K. Gu, V. L. Kharitonov, and J. Chen, *Stability of Time-Delay Systems*, W. S. Levine, Ed. Birkhauser, 2003.

[3] Y. Xia, M. Fu, and P. Shi, *Analysis and Synthesis of Dynamical Systems with Time-delays*. Springer Verlag, 2009.

[4] G. Stépán, *Retarded Dynamical Systems: Stability and Characteristic Functions*. Burnt Mill, Harlow, Essex, England : New York: Longman Scientific & Technical ; Wiley, 1989. [Online]. Available: http://mirlyn.lib.umich.edu/Record/001823389

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

[6] M. Gopal, *Digital Control Engineering*. New Age International, 1988.

[7] B. C. Kuo, *Digital Control Systems*. Oxford University Press, 1992.

[8] R. Alur, T. Henzinger, and P. Ho, "Automatic symbolic verification of embedded systems," *IEEE Transactions on Software Engineering*, vol. 22, no. 3, pp. 181–201, 2002.

[9] C. Tomlin, J. Lygeros, and S. Sastry, "Synthesizing controllers for nonlinear hybrid systems," *Hybrid Systems: Computation and Control*, pp. 360–373, 1998.

[10] A. Nerode and W. Kohn, "Models for Hybrid Systems: Automata, Topologies, Controllability, Observability," *Hybrid systems*, pp. 317–356, 1993.

[11] D. Fontanelli, L. Greco, and A. Bicchi, "Anytime Control Algorithms for Embedded Real-time Systems," *Hybrid Systems: computation and control*, pp. 158–171, 2008.

[12] R. Bhattacharya and G. Balas, "Anytime control algorithm: Model reduction approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 5, 2004.

[13] V. Gupta, "On an anytime algorithm for control," in *Proceedings of the 48th IEEE Conference on Decision and Control*. IEEE, 2010, pp. 6218–6223.

[14] L. Sha, T. Abdelzaher, K. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A. Mok, "Real-time Scheduling Theory: A Historical Perspective," *Real-time systems*, vol. 28, no. 2, pp. 101–155, 2004.

[15] K. Arzén, A. Cervin, J. Eker, and L. Sha, "An introduction to control and scheduling co-design," in *Proceedings of the 39th IEEE Conference on Decision and Control*, vol. 5. IEEE, 2002, pp. 4865–4870.

[16] A. Cervin, J. Eker, B. Bernhardsson, and K. Årzén, "Feedback–feedforward scheduling of control tasks," *Real-Time Systems*, vol. 23, no. 1, pp. 25–53, 2002.

[17] J. Eker, P. Hagander, and K. Årzén, "A feedback scheduler for real-time control tasks," *Control Engineering Practice*, vol. 8, no. 12, pp. 1369–1378, 2000.

[18] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[19] M. Branicky, S. Phillips, and W. Zhang, "Scheduling and Feedback Co-Design for Networked Control Systems," *IEEE Transactions on Automatic Control*, vol. 2, pp. 1211–1217, 2002.

[20] W. Zhang, M. Branicky, and S. Phillips, "Stability of Networked Control Systems," *IEEE control systems magazine*, vol. 21, no. 1, pp. 84–99, 2001.

[21] E. Atkins and R. Sanner, "QOS Tradeoffs for Guidance, Navigation, and Control," in *2002 IEEE Aerospace Conference Proceedings.*, vol. 7, 2002, pp. 7–3333.