Dynamic Resolution in Distributed Cyber-Physical System Simulation

Dylan Pfeifer
The University of Texas at Austin
2501 Speedway, Stop C0803
Austin, TX 78712, USA
011-512-232-4297
dpcfeifer@ieee.org

Andreas Gerstlauer
The University of Texas at Austin
2501 Speedway, Stop C0803
Austin, TX 78712, USA
011-512-232-8294
gerstl@ece.utexas.edu

Jonathan Valvano
The University of Texas at Austin
2501 Speedway, Stop C0803
Austin, TX 78712, USA
011-512-471-5141
valvano@mail.utexas.edu

ABSTRACT
Cyber-physical systems challenge distributed simulation techniques for reasons of the heterogeneous tools used to model system components at different levels of abstraction, each with potentially different notions of time. The SimConnect and SimTalk distributed cyber-physical system simulation tools meet the synchronization challenge of distributed simulation, but also offer dynamic resolution among coordinated simulators for tradeoffs in simulation speed versus accuracy. This paper discusses the dynamic resolution capabilities of SimConnect and SimTalk, and evaluates the tools in distributed simulation of a closed-loop motor control system. Results show selectable tradeoffs in speedup versus accuracy over non-dynamic coordination.

Categories and Subject Descriptors
I.6.8 [**Modeling and Simulation**]: Types of Simulation – *Combined, Distributed, Parallel.*

General Terms
Experimentation

Keywords
Kahn Process Networks, dynamic distributed hybrid co-simulation, heterogeneous co-simulation, co-simulation backplanes, cyber-physical system simulation, DC motor PID/PWM simulation

1. INTRODUCTION
Cyber-physical systems (CPS) are engineered systems that integrate computation and physical processes [1]. They inherit the field of real-time embedded systems and challenge modern electronic design automation [2] as computation elements continue to proliferate in quantity, decrease in area and power, and increase in system-on-chip (SOC) complexity. Heterogeneous by definition, cyber-physical systems are a challenge to simulate at the system level because a design may include hardware, software, mechanical, or even biological components [1]. While individual simulators may specialize in modeling some of these components, no single simulator yet performs superlatively in modeling all of them, especially as the complexity of components continues to grow in the diverse range of CPS [3][4][5]. For example, while a Spice 3.0-based simulator solution [6] may excel in modeling cyber-physical system analog electronics, it may not excel in modeling a microcontroller at an instruction-set accurate level of abstraction such as the TExaS [7] Freescale 9S12 simulator does. Therefore, cyber-physical system simulation can benefit from coordinating multiple different simulators, each specializing in an engineering domain required by the system.

Heterogeneous simulator coordination brings a range of challenges, principally the synchronization and causality challenge of independent simulators running with local time and independent state. Solutions to this challenge are given in the field of parallel and distributed simulation (PADS) [8][9][10][11]. Once the coordination challenge is overcome, a remaining challenge is to reduce the simulation time required, since some simulators can increase in simulation time exponentially as model complexity increases [12].

One means of PADS simulated time reduction is through dynamic time resolution, a service offered by the SimConnect and SimTalk distributed cyber-physical system tools [13][14]. In applying dynamic resolution to the distributed simulation of a closed-loop motor control system, we find speed up offered with the tools with a configurable tradeoff in speed versus accuracy.

2. RELATED WORK
SimConnect and SimTalk present a backplane based [15][16][17] solution to the distributed cyber-physical system challenges. The architecture of the tools, how they relate to previous backplane techniques, and how they perform among both multiple homogenous and heterogeneous simulators is covered in [13][14]. One benefit of the tools is the reduction of the backplane control structure to the properties of a Kahn Process Network (KPN) [18], simplifying implementation and enforcing simulation causality by limiting the tokens of the backplane KPN to a type defined as "interpolated events" (IEs). Interpolated events, covered in [14], provide compliance to the *local causality constraint* [8] in distributed simulation, but also enable dynamic time management during the simulation.

Previous results with SimConnect and SimTalk achieved distributed simulation speed up by increasing spatial distribution, increasing parallelism in the simulation model [12], or relaxing IE resolutions *statically* configured. In results of this paper, however, new speed-ups are obtained by *dynamic* resolution of the IE duration as the simulation proceeds.

SIGSIM-PADS'13, May 19-22, 2013, Montréal, Québec, Canada.

Infrastructure and support is added to the SimConnect/SimTalk tools achieve this capability. With dynamic resolution, time segments during the simulation can be "zoomed in" in event resolution or "zoomed out" at demand. A higher resolution means more synchronization traffic, for increased simulation time. Lower resolution means less synchronization traffic, so simulators achieve a faster local speed.

The potential speedup benefit of dynamic time management in distributed simulation is discussed in [19], with application specific solutions offered in [20][21][22]. However, for cyber-physical systems, the High Level Architecture (HLA)-based [23] techniques discussed in [19] require HLA support among cyber-physical system simulators, a development yet to be evaluated for the challenges discussed [14]. The SimConnect/SimTalk based solution to dynamic time management, by contrast, is straightforward through the interpolated event data type. Implementation of the service added less than 500 lines of C code to the tools from their code size given in [14].

In regard to interoperating with real-time distributed simulation systems (a feature supported by HLA [23] and former DIS-like solutions [25]), while SimConnect/SimTalk can support real-time interoperability and scaled wall-clock time execution, on the other hand, the domain challenge of cyber-physical system simulation state complexity currently prohibits simulating at speeds anywhere close to real-time or wall-clock time. That is, the micro-event focus of CPS engineering design, focusing on events and signals in the simulation space with kilohertz, megahertz or gigahertz frequencies results in simulation times that must run at "as fast as possible" speeds. Even simulating as fast as possible on current workstation-grade machines, the execution time of seconds of simulation time in the systems hosted by SimConnect/SimTalk still can take minutes of wall-clock time due to the computation time of connected CPS simulators (see Figure 16). Therefore, interoperability with real-time distributed simulations or systems with "humans in the loop" [25] is not yet a feature demanded of SimConnect/SimTalk CPS simulations. When the simulation can execute simulation time faster than wall-clock time, real-time wall-clock scaling can be enforced by the SimConnect backplane. This is a difference in domain focus required by CPS engineering design, even though it is still a PADS application. Further discussion of this domain challenge of CPS in regard to event focus is given in [14].

3. DYNAMIC RESOLUTION WITH INTERPOLATED EVENTS

The interpolated event type is defined in [13]. IEs are 3-tuple set elements (v, t_m, t_n) from the product set $V \times T \times T$, where $\{ V \}$ is a set of values, and $\{ T \}$ is a set of tags. This extends the value/tag "(v, t)" definition of an event covered in [24]. For a given interpolated event (v, t_m, t_n), the value v is defined to be constant on the interval [t_m, t_n) specified in the IE, such that the tag set $\{ T \}$ is ordered. $\{ T \}$ is conventionally the real number set $\boldsymbol{R}^1$ in timed, event driven simulations, representing the simulation time stamp when an event occurs. For an interpolated event (v, t_m, t_n), the range [t_m, t_n) assigns a "stable" time to the signal value v for producers and consumers.

If a simulator consumes an interpolated event (v, t_m, t_n), it may assume the value v is constant on the tag range [t_m, t_n), and not need to input the value again until expiration time t_n. So, an interpolated event encapsulates both communication (the signal value) and synchronization (the start and end time). In terms of PADS synchronization, the expiration t_n value can be seen as a look ahead value [9] and lower bound on the t_m value on all future IEs posted on a FIFO. It can also be considered the *lower-bound time stamp* (LBTS) value [19] for input signal IEs in the HLA RTI conservative synchronization terminology [25]. An IE (v, t_m, t_n) can also be considered as the union of event (v, t_m) and the set of all Chandy/Misra/Bryant NULL messages [9][10] { (t_i , NULL) } such that $t_m < t_i < t_n$. Discussions of deadlock, synchrony, and blocking with IEs are given in [14].

The SimConnect backplane routes interpolated events from producer to consumers in a publish/subscribe client/server architecture. Signal producers and consumers are independently running simulators. Simulators connect to the SimConnect server through TCP/IP sockets and publish or subscribe interpolated events as they proceed in simulation. Mapped to nodes in a Kahn Process Network, simulators consume IEs, run, and produce IEs until the expiration tag of the last consumed IE. At this time point simulators sample their FIFOs again for a new IE. If their input FIFOs are empty, simulators block, enforcing the *local causality constraint* [8] because each simulator cannot advance in time beyond the expiration tags of IEs on its input FIFOs. Simulators may be distributed across any compute/network topology with TCP/IP services, and they share no run-time variables or state with each other. They only connect to the backplane through the SimTalk protocol and produce IEs to the backplane, or consume IEs from it, agnostic to other simulators in the distribution.

Selection of services to implement the networking and interoperability requirements of this hierarchy are flexible, but TCP/IP and BSD sockets were chosen for simulators offering an OS-level programming interface for familiarity and facility. The Message Passing Interface (MPI), for example, could be used to transport SimTalk IE messages, and the backplane could store IEs in a relational database such as MySQL, but implementation with socket programming and minimal library dependencies in the SimConnect backplane was desired. More examples and details of SimConnect and SimTalk backplane/simulator based configurations are given in [12][13][14].

Dynamic resolution can be achieved with interpolated events through two means. First, a signal producer has complete write authority over a posted t_n value, the expiration time of an IE (v, t_m, t_n). Therefore, a signal producer may vary this value anytime during the simulation based on internal knowledge of the signal's change frequency or some other application criteria. Changing the IE t_n value will change the time the consuming simulator of the IE next queries the backplane, there by changing the time of next synchronization. As t_n increases beyond t_m, the IE duration increases, and therefore the event resolution relaxes (the time between synchronizations on the IE signal increases). As t_n approaches, but yet is still greater than t_m, the IE duration decreases, so the event resolution increases (more synchronization events). This

is similar to dynamically varying an HLA-based LBTS values [19], but with implementation simplicity.

A second method to achieve dynamic resolution with IEs is for the backplane, simulation operator, or another simulator to *command* a signal producer to change its $(t_n - t_m)$ duration for future IEs. In this way an agent may externally vary the IE duration of a signal producer, there by even throttling the rate of its incoming signals. External IE resolution change requests are registered to the backplane in the form of (*signal_name, time, resolution*) 3-tuples stored in the backplane. These can be entered at any time during the simulation through the SimTalk protocol to the backplane by any connected simulation participant.

For a backplane resolution change tuple (*signal_name, time, resolution*), the tuple specifies that for the signal specified by string *signal_name*, all IEs posted by the signal producer for which the IE t_n value is greater than or equal the tuple *time* value must have a duration value of $(t_n - t_m) = resolution$. If a series of resolution tuples is entered for a signal, such as the series (*signal_name,* t_0, r_0), (*signal_name,* t_1, r_1), (*signal_name,* t_2, r_2), (*signal_name,* t_3, r_3), such that $t_0 < t_1 < t_2 < t_3$, then for all IEs (v, t_m, t_n) of *signal_name* with $t_0 < t_n < t_1$, the duration of the IE, $(t_n - t_m)$ shall be value r_0. For all IEs (v, t_m, t_n) on *signal_name* such that $t_1 < t_n < t_2$, the duration of the IE, $(t_n - t_m)$ shall be value r_1, for all $t_2 < t_n < t_3$, the resolution shall be r_2, and so on. Therefore, a string of resolution tuples may be registered in the backplane for arbitrary IE resolution adjustment.

Because a signal producer has no idea what resolution tuples have been registered in the backplane, the backplane notifies the signal producer of a resolution change when its broadcasted local time (indicated an IE posted t_n value) reaches a time greater than or equal to resolution tuple time entry. The backplane then sends notification of a resolution change through a SimTalk "res" request message in response to the posted IE, a SimTalk "res [*signal_name*] [*value*]" message. Receipt of this message by any agent in the backplane indicates that the agent shall adjust the resolution $(t_n - t_m)$ of future IEs it is posting on *signal_name* to the value given in the [*value*] field of the message until further notice from the backplane. The simulator proceeds to post IEs at the new resolution value until its local time reaches another resolution tuple point, indicated by an IE t_n value. SimTalk "res" messages only occur at points of IE resolution change, so "res" messages are infrequent relative to IE "get" and "set" messages [13].

If a simulator itself posts a "res" message *to* the backplane, versus receiving one, it logs a new resolution tuple. In this way simulators may post resolution change requests to signal producers (or any broadcasted signal in the simulation), as well as adjust resolution on their own outgoing signals.

4. EXPERIMENT

4.1 Background and Purpose

In [14], a means was needed to evaluate the accuracy of a SimConnect/SimTalk hosted distributed simulation against a truth condition. The microcontroller-based, software PID/PWM controlled DC motor was chosen as the system to simulate because it contained sufficient model diversity (a microcontroller model, Spice electrical model, and Matlab/Simulink electro-mechanical model) to demonstrate SimConnect and SimTalk facility with heterogeneous simulation while also being a system with known behavior. The Matlab/Simulink-only simulation of the PID/PWM control system served as a single-simulator truth condition to evaluate the accuracy of the more realistic 3-simulator based simulation. However, the studies of [14] used static IE resolution. For this experiment, dynamic resolution is employed. The same system (DC motor control system) is chosen as the simulation target to have a means to evaluate accuracy. The 2-simulator representation of the system (TExaS and Matlab/Simulink) was selected for simplicity, and could be evaluated for accuracy against the statically controlled 2-simulator based system in [14].

To support dynamic resolution messaging, source code was added to the SimConnect backplane to implement resolution tuple storage and new SimTalk "res" message passing. Support for SimTalk "res" messages was also added to each SimTalk software plugin for the TExaS [7] and Matlab/Simulink simulators [26].

The PID/PWM microcontroller-based DC motor controller system in [14] was re-simulated in the 2-simulator case but with dynamic resolution. The setup of the model is repeated for illustration. For a control condition, the system is first modeled in a 1-simulator case with Matlab/Simulink.

4.2 1-Simulator setup

The DC motor is modeled in Simulink block-diagram form with summing, integrating, and gain blocks in Figure 1.

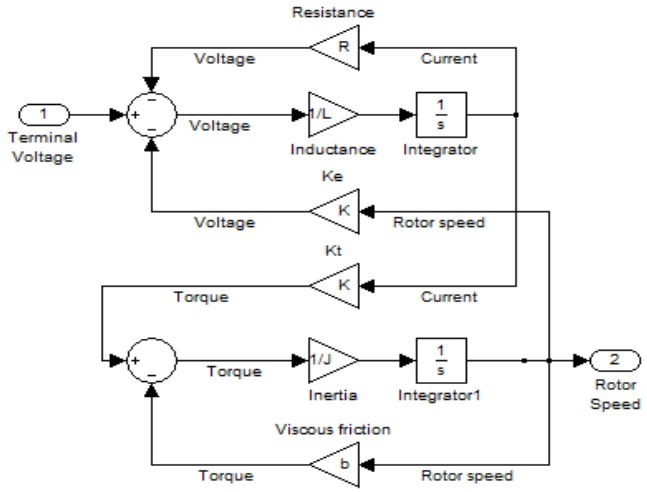


Figure 1. Simulink DC motor electro-mechanical model

The model is parameterized for the simulation as follows:

R	motor terminal electrical resistance	1.0 Ω
L	motor terminal inductance	0.001 H
K_t	torque constant	0.1 Nm/A
K_e	electrical constant	0.1 Nm/A
b	rotor viscous friction coefficient	0.001 Nm · s
J	rotor moment of inertia	0.01 kg · $(m/s)^2$

Figure 2. DC motor model parameters

The model is driven with the Simulink PID continuous controller block in Figure 3, configured to a set point of half-speed 23.44 radians/s, 5 Volt output ceiling, with K_p, K_i, and K_d coefficients of 8, 2, and 1 respectively, with clamping anti-integrator windup. The closed-loop transient response is plotted in Figure 6.

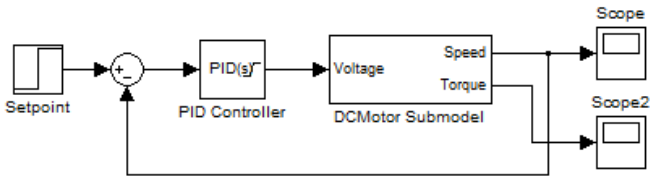


Figure 3. Simulink continuous PID controller

4.3 2-Simulator setup

For distributed modeling and increased realism, the controller is refined to a software-based PID difference-equation algorithm with PWM actuators hosted on the 9S12 microcontroller, simulated with TExaS at the cycle-estimating, instruction-set architecture (ISA) level. The SimConnect/SimTalk signal structure is given in Figure 4.

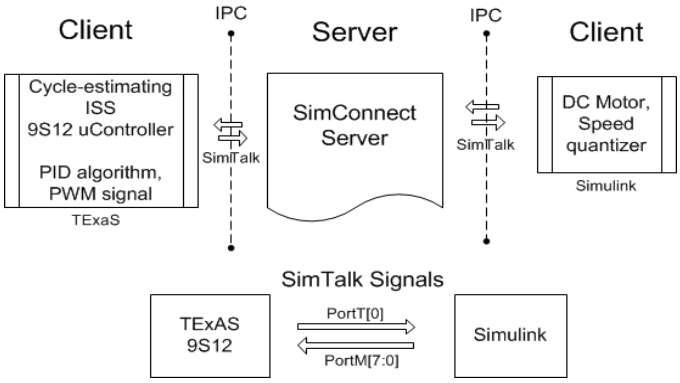


Figure 4. 2-Simulator configuration

The software-based PID algorithm in 9S12 assembly is adapted from [7]. Conversion of a continuous-time frequency-domain specified controller to a digital controller is covered in [27]. This refinement uses a free-running 1 kHz sampling rate PID main loop of 63 9S12 assembly instructions, and a total code length of 158 instructions. The algorithm also incorporates anti-integrator windup and output limit checking. The refined Simulink model is given in Figure 5. The "socket_input" and "socket_output" S-Functions register SimTalk signals PortT[0] and PortM[7:0] for exchange with the SimConnect server. The PortT[0] digital signal is the PWM wave generated by TExaS. The 0/1 signal is amplified to 5 Volts for application to the motor terminals. The PWM wave and voltage in this configuration is modeled as ideal (zero rise/fall time).

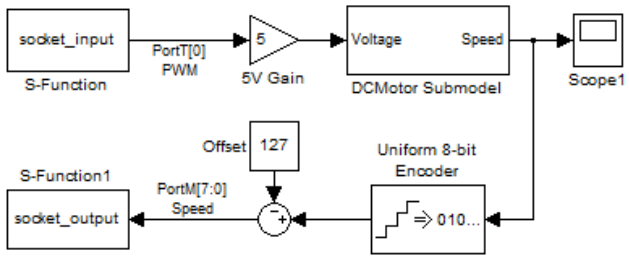


Figure 5. Simulink DC motor model with SimTalk I/O interface

5. RESULTS

The 2-simulator model is conducted first at a static 100 µs IE ($t_n - t_m$) resolution on signals PortT[0] and PortM[7:0]. The transient response is plotted against the Simulink-only classical continuous and cases in Figure 6 to verify the functionality of the distributed modeling of the digital PID/PWM microcontroller-based simulation versus the Simulink continuous-time 1-simulator controller.

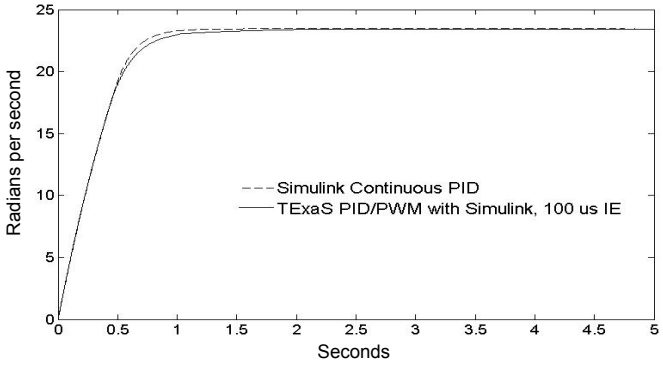


Figure 6. Model output speed versus time with Simulink-only and 2-simulator PID control model cases

In Figure 6, there is some departure from the continuous model because the applied terminal signal is a PWM wave from the 9S12 microcontroller in a software PID loop. However, the control profile shows set point agreement. The static 100 µs 2-simulator Case A is used as a baseline for checking dynamic resolution experiments, with simulation times given in Figure 16. In Case B, the static simulation localized to one machine to demonstrate the effect of network latency as a distribution cost. In Case C, the IE resolution is dynamically changed early in the simulation. The resolution begins at 100 µs IE resolution to set initial conditions, and is relaxed to 10 ms at simulation time 1 ms.

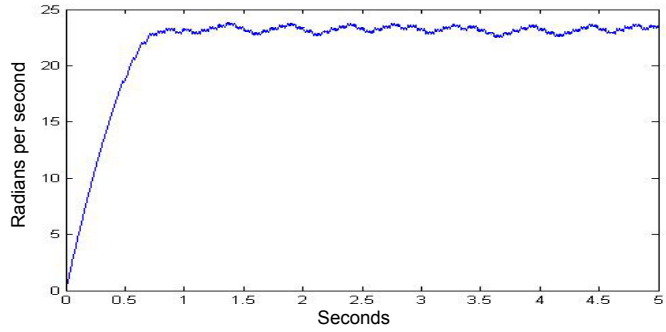


Figure 7. Case C model output speed versus time

Figure 7 shows set point approach, but steady state instability as Simulink and TExaS only receive signal updates every 10 ms (the relaxed IE duration). However, the simulation time decreases significantly, as shown in Figure 16.

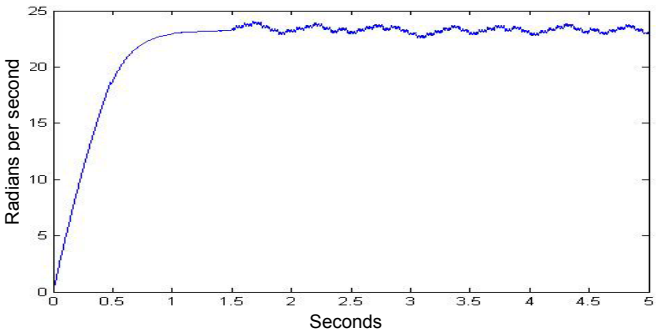


Figure 8. Case D model output speed versus time

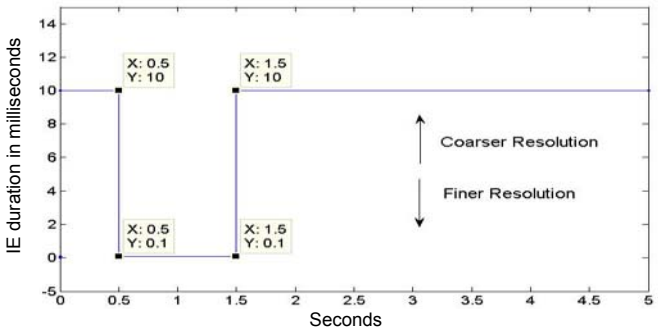


Figure 9. Case D dynamic IE duration change

In Case D, coarse resolution (10 ms IE) is used in the motor run-up phase (0 to 0.5) seconds, then finer resolution (100 µs IE) during the set point approach phase (0.5 to 1.5 seconds), then coarse resolution again in the steady state phase (1.5 to 5 seconds). The control profile matches the static high-resolution case up to 1.5 seconds, where the controller is unstable again in the error due to relaxed resolution (10 ms IE signals). The simulation time is still decreased (Figure 16).

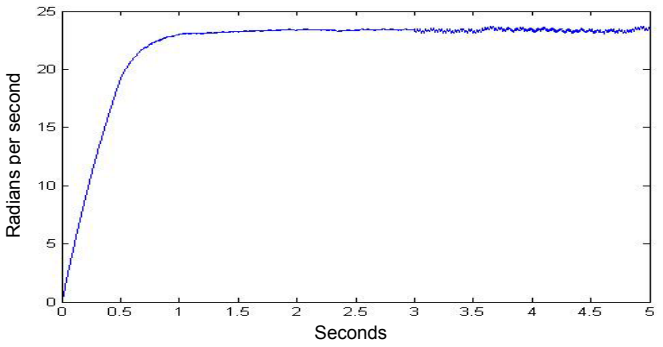


Figure 10. Case E model output speed versus time

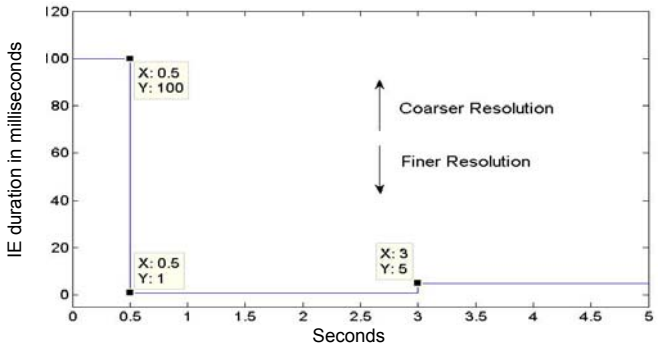


Figure 11. Case E dynamic IE duration change

In Case E, coarse resolution IE (100 ms) is used to 0.5 seconds in the simulation, then finer resolution (1 ms IE) as the set point approaches, and then 5 ms IE resolution in the steady state after 3 seconds. The instability around the set point is reduced over the Case D 10 ms IE resolution after 3 seconds, but variance persists due to controller only getting samples once in every five PID loops (5 ms IE with a 1 kHz PID loop).

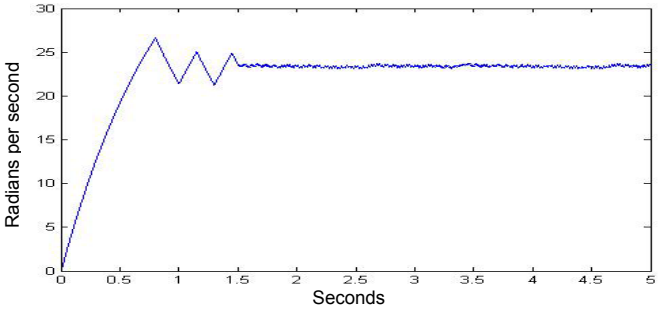


Figure 12. Case F model output speed versus time

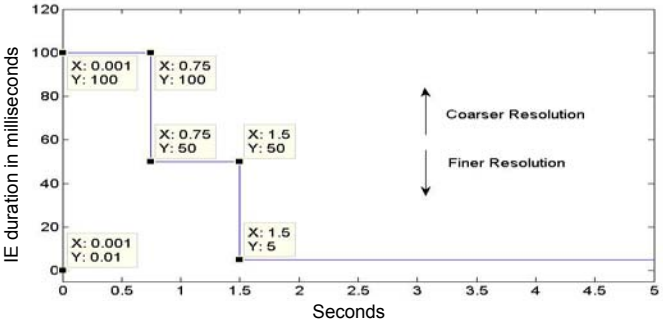


Figure 13. Case F dynamic IE duration change

In Case F, we suppose coarse resolution might apply during the run-up and set point approach phases. Coarse resolution IE (100 ms) from 0 to 0.75 seconds is applied, medium resolution (50 ms IE) from 0.75 to 1.5 seconds, and then 5 ms IE resolution after 1.5 seconds. Figure 12 shows that although the control oscillates coarsely under low resolution, the oscillation decreases significantly when the resolution increases around controller steady state set point.

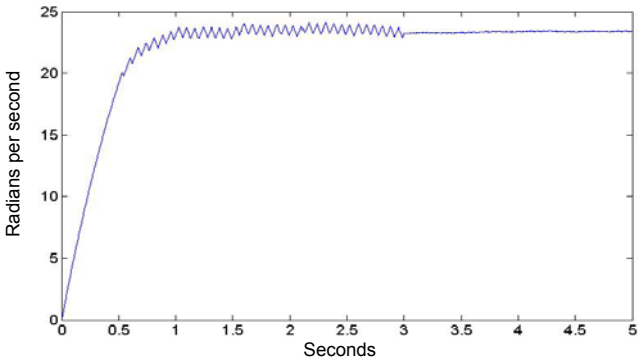


Figure 14. Case G model output speed versus time

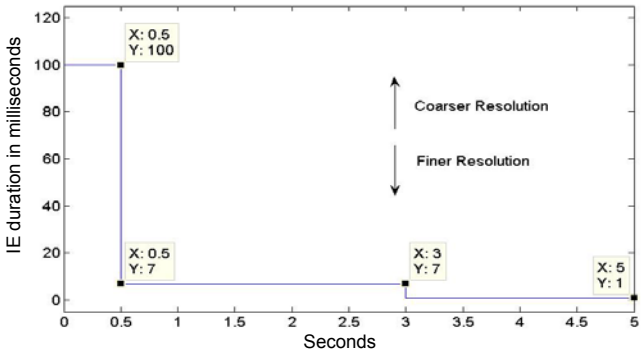
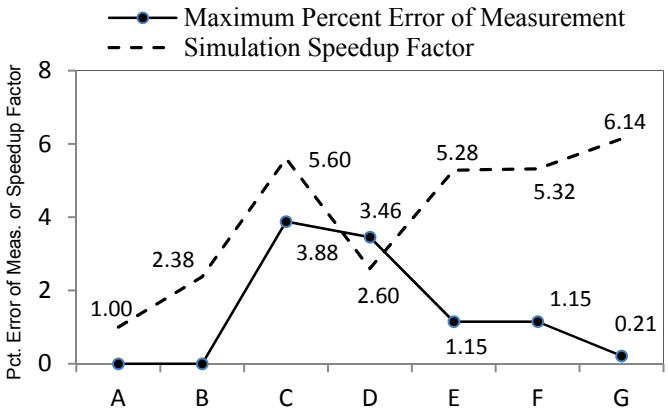


Figure 15. Case G dynamic IE duration change

In Case G, coarse resolution IE (100 ms) is used from 0 to 0.5 seconds, finer resolution IE (7 ms) approaching the set point, and then 1 ms IE resolution after 3 seconds. The steady state oscillation decreases significantly as the PID controller receives IE updates ever 1 ms. Noise still exists due to 1 ms IE sampling of the PortT[0] PWM wave that has a higher duty cycle resolution than 1 ms. However, Case F shows that we can *arbitrarily* bring down the error term around the set point by increasing the resolution, while using coarse resolution in the run-up and approach phases.

In Figure 16, simulation runtime counters and the speed up multiplier of each experiment (A through G) is plotted against the static resolution simulation time of Case A. The maximum percent error of measurement around the set point value after 3 seconds for each experiment is also plotted, with Cases A and B considered the truth condition. Figure 16 shows that percent error of around the set point can be reduced arbitrarily while decreasing simulation time by dynamic resolution. In the best trial, Case G, there is a maximum 0.21 percent error of measurement around the set point but with a 6.14 times faster simulation time than the static resolution Case A.

In regard to IE token communication cost, Figure 16 shows that distribution (2 machines versus 1 machine in Cases A and B) affects simulation time, since IEs must travel over a local area network. However, even with this cost, dynamic resolution still decreases simulation time in Cases C through G over the static resolution cases. The SimConnect server in all cases but B ran on GNU/Linux 2.6.16 kernel Intel Xeon 2.93 GHz machine. The TExaS and Matlab/Simulink simulators ran on a Windows 8 OS Intel Core i5 2.50 GHz laptop computer.



A	B	C	D	E	F	G	H	I
A. S,T,I	100 µs	3	4000	20	11.8	5.0	13:35	2
B. S,T,I	100 µs	3	4000	20	11.8	5.0	5:51	1
C. S,T,I	(dyn)	3	4.8	20	11.8	5.0	2:29	2
D. S,T,I	(dyn)	3	800	20	11.8	5.0	5:21	2
E. S, T,I	(dyn)	3	24	20	11.8	5.0	2:38	2
F. S,T,I	(dyn)	3	6.5	20	11.8	5.0	2:37	2
G. S, T,I	(dyn)	3	19.7	20	11.8	5.0	2:16	2

Column Legend

A Experiment case and distributed simulators : Simulink(S), TExaS(T), SimConnect server(I)
B Static IE resolution, or (dyn) if dynamic resolution
C Number of instantiated SimTalk connectors
D Number of SimTalk messages (times 10^3)
E Number of 9S12 cycles (times 10^6)
F Number of 9S12 instructions (times 10^6, rounded 10^5)
G Simulated Time (seconds)
H Simulation Execution Time (minutes: seconds)
I Number of host machines

Figure 16. Simulation times, configurations and traffic

In regard to scaling, generally, SimConnect can service connections from simulators hosted anywhere on the Internet. As the number of connected, coordinated simulators increases, the latency of servicing SimTalk messages in the SimConnect backplane increases, since it must maintain and service more socket connections. SimConnect presently services socket states in a serial, cyclic loop. This latency does not include SimTalk message delivery latencies, which depend on the network. The scaling profiles of simulation time versus number of connected simulators is under study, but in the experiments of [13], which coordinated up to 128 independent Ngspice simulators, the latency of simulation was dominated by the execute time of individual simulators as they ran in each time slice specified in an IE. In [12], the message servicing and delivery cost of the simulation latency was still less than one percent of the total latency from one IE consumption to the next measured in a consuming simulator. Therefore, scaling is presently a function of the compute time of individual simulators. We have not yet found an inflection point where the backplane message servicing latency

dominates the simulation, but from the results of [13] we anticipate it will require many hundreds of simulators. This estimate does not include pathological cases where a network fault or distribution abnormality introduces an asymmetrical message delivery delay.

5.1 Interpretation of Results

Figure 16 emphasizes the simulation speed up potential offered by dynamic resolution, essentially relaxing resolution during periods of infrequent inter-simulator signaling, and increasing resolution during frequent signaling. While speed up with dynamic resolution control is not unique to this study, the benefit of the SimConnect/SimTalk approach is that dynamic resolution is straightforward to implement using the KPN/IE dataflow dynamics that SimConnect/SimTalk strictly adheres to, as long as a blocking-API can be applied to connected simulators through a SimTalk connector.

A limitation, however, is that the interpolated event (IE) data token incurs similar limitations and accuracy tradeoffs of an ideal sample-and-hold based analog-to-digital converter. That is, the IE is a sample-and-hold, piecewise-constant based representation of the underlying signal. Primarily this results in delay of signal change information that is at minimum the duration of the IE. The cost of delay inaccuracies are seen in [12]. The cost of quantization error in control systems with sample-and-hold signal converters is covered in [27]. Therefore, whether IEs are varied statically with repeated simulations, or varied dynamically in one simulation per this study, there is an unavoidable, inversely proportional speed-versus-accuracy tradeoff with the KPN/IE approach. The effect of this may be negligible for some systems, particularly if the IE resolution is on the order of the minimum clock period of the simulated system, or it may be completely determinable based on bandwidth and gain parameters for control systems [27]. However, the approach currently does not have a means to guarantee a metric of accuracy that is independent of the system being simulated. That is, the accuracy effects of IE resolution are highly tied to the coupling interface of the system under simulation where the signal couple is represented by an IE. Fortunately, analog-to-digital converters are a common coupling point in CPSs, lending to IE representation. However, if a clocked digital gate or storage element is represented by an IE, the IE duration can introduce a delay of the circuit change state, or completely mask the change state. So, speed-versus-accuracy tradeoffs are presently system-by-system determined.

Another limitation is that the system requires connectable simulators to offer an OS-level programming interface for the SimTalk connector. Fortunately, preeminent simulators in CPS simulation offer this.

6. SUMMARY AND CONCLUSIONS

Two independent simulators, TExaS and Matlab/Simulink, were coordinated with the SimConnect/SimTalk tools to model a closed loop, software-based PID/PWM control system with Freescale 9S12 microcontroller ISA and 2nd order DC motor model realism. Dynamic resolution demonstrated up to 6.14 times speed up over the non-dynamic case, with configurable tradeoffs in speed up versus accuracy. Dynamic resolution implementation in the SimConnect/SimTalk tools was straight forward with the interpolated event (IE) data type and SimTalk "res" messages. In the space of simulated software-based PID control, these results show that more coarse resolution is tolerable while the PID error value is high, while a higher resolution is required as the controller settles around the set point. As an application specific result, this can speed up PID/PWM-based simulations in cyber-physical systems by relaxing resolution when a new controller set point is ordered, then increasing resolution through steady state of the simulated controller.

For future work, adaptive control of the simulation resolution is being developed for the SimConnect backplane. Because the backplane services all SimTalk messages in the system, it can monitor any signal or groups of signals consting of streams of IEs. Any numerical metric can be applied to these, such as finite differences applied to the $\{ v \}$ values of IEs evaluated at their $\{ t_m \}$ time points, for numerical differentiation of the IE signal. Similarly the IEs can be integrated at their $\{ v, t_m \}$ values using standard numerical integration techniques. The backplane can calculate combinations of metrics on IE signals and adjust the simulation resolution (through *res* messages described here) based on preset thresholds. The most simple adaptive control would be to relax resolution as an IE signal stream stabilizes (measured by a such as a numerical derivative applied the IE stream), and to increase resolution dynamically as the signal changes more frequently. Alternatively, resolution can change when the backplane moniters an event on a specific signal, such as an interrupt pin in the system represented as SimTalk singal stream, causing the backplane to employ high resolution during the interrupt service routine in the simulated software. There are many possibilities for adaptive control in the SimConnect backplane, and work is underway on these features.

7. REFERENCES

[1] Lee, E.A., "Cyber-Physical Systems: Design Challenges," *The University of California at Berkeley Center for Hybrid and Embedded Software Systems,* Technical Report No.UCB/EECS-2008-8, Jan. 2008.

[2] Sangiovanni-Vincentelli, A., "Quo Vadis, SLD? Reasoning about the Trends and Challenges of System Level Design," *Proceedings of the IEEE*, vol.95, no.3, pp.467-506, March 2007.

[3] National Science Foundation (NSF), "Cyber-Pysical Systems," http://www.nsf.gov/pubs/2012/nsf12520/nsf12520.htm, 2006.

[4] Klesh, A.T., Cutler, J.W., and E.M. Atkins, "Cyber-Physical Challenges for Space Systems," *The 2012 IEEE/ACM Third International Conference Cyber-Physical Systems (ICCPS),* pp. 45-52, 17-19 April 2012.

[5] Rajkumar, R., Lee, I., Sha, L., and J. Stankovic, "Cyber-Physical Systems: The Next Computing Revolution," *The 2010 ACM/IEEE 47th Design Automation Conference (DAC),* pp.731-736, 13-18 June 2010

[6] Nenzi, P., and V. Holger, "Ngspice Users Manual." V. 22, Sep 2010, ngspice.sourceforge.net.

[7] Valvano, J., Embedded Microcomputer Systems: Real Time Interfacing, 3rd ed. Stamford, CT: Cengage Learning, 2011.

[8] Fujimoto, R.M., "Parallel and Distributed Simulation," *Proceedings of the 1995 Winter Simulation Conference*, pp. 118-125, 3-6 Dec 1995.

[9] Fujimoto. R.M. "Parallel Discrete Event Simulation," *Proceedings of the 1989 Winter Simulation Conference*, pp. 19-28, 4-6 Dec 1989.

[10] Chandy, K.M., and J. Misra, "Distributed Simulation: A Case Study in Design and Verification of Distributed Programs," *IEEE Transactions on Software Engineering,* vol. SE-5, no. 5, Sep 1979.

[11] Jefferson, D.R., "Virtual Time," *ACM Transactions on Programming Languages and Systems*, vol. 7, no. 3, 1985.

[12] Pfeifer, D. and A. Gerstlauer, "Expression-level Parallelism for Distributed Spice Circuit Simulation," *The 15th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications, DS-RT 2011.* 4-7 September, 2011.

[13] Pfeifer, D. and J. Valvano, "Kahn Process Networks Applied to Distributed Heterogeneous HW/SW Cosimulation," *The 2011 Electronic System Level Synthesis Conference, ECSI*. 5-6 June 2011.

[14] Pfeifer, D., J. Valvano, and A. Gerstlauer. "SimConnect and SimTalk for Distributed Cyber-Physical System Simulation." *Simulation: Transactions of the Society for Modeling and Simulation International.* OnlineFirst, DOI: 10.1177/0037549712472755. 5 March 2013.

[15] Schmerler, S., Tanurhan, Y., and K.D. Muller-Glaser, "A Backplane Approach for Cosimulation in High-level System Specification Environments," *Proceedings of the European Design Automation Conference, EURO-DAC '95 with EURO-VHDL*, pp. 262-267, 18-22 Sep 1995.

[16] Atef, D., Salem, A., and H. Baraka, "An Architecture of Distributed Cosimulation Backplane," *The 42nd Midwest Symposium on Circuits and Systems*, vol. 2, pp.855-858, 1999.

[17] Sung, W., and S. Ha, "A Hardware Software Cosimulation Backplane with Automatic Interface Generation," *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC '98,* pp.177-182, 10-13 Feb 1998.

[18] Kahn, G., "The Semantics of a Simple Language for Parallel Programming," *Information Processing*, pp. 471-475, Stockholm, Sweden, Aug 1974.

[19] Dou Zhiwu; Li Yanfeng; , "Dynamic Time Management Algorithms Research in Simulation System HLA-Based," *Computer Science and Engineering, 2009. WCSE '09. Second International Workshop on* , vol.1, no., pp.580-583, 28-30 Oct. 2009

[20] Lungeanu, D.; Shi, C.-J.R.; , "Distributed simulation of VLSI systems via lookahead-free self-adaptive optimistic and conservative synchronization," *Computer-Aided Design, 1999. Digest of Technical Papers. 1999 IEEE/ACM International Conference on* , vol., no., pp.500-504, 1999

[21] Lee, C.; Coe, E.; Clark, J.M.; Stepanek, J.; Raghavendra, C.; Bhatia, S.; Puri, R.; , "Scalable time management algorithms using active networks for distributed simulation," *DARPA Active NEtworks Conference and Exposition, 2002. Proceedings* , vol., no., pp. 366- 378, 2002

[22] Moo-Kyoung Chung; Chong-Min Kyung; , "Improving Lookahead in Parallel Multiprocessor Simulation Using Dynamic Execution Path Prediction," *Principles of Advanced and Distributed Simulation, 2006. PADS 2006. 20th Workshop on* , vol., no., pp.11-18, 2006

[23] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)-- Framework and Rules," *IEEE Std 1516-2010*, vol., no., pp.1-38, Aug. 18 2010.

[24] Lee, E.A., and A. Sangiovanni-Vincentelli, "Comparing Models of Computation," *The IEEE/ACM International Conference on Computer-Aided Design Digest of Technical Paper,s ICCAD-96*, pp.234-241, 10-14 Nov 1996.

[25] Fujimoto, R.M. and R.M Weatherly, "Time Management in the DoD High Level Architecture," *The Proceedings of the 1996 10th Workshop on Parallel and Distributed Simulation*, pp. 60-67, 1996.

[26] The MathWorks Corp. www.mathworks.com. 2012.

[27] Franklin, G., J.D. Powell, and A. Emami-Naeini. Feedback Control of Dynamic Systems, 4th ed. Upper Saddle River, NJ: Prentice Hall, 2002.