AFRL-OSR-VA-TR-2013-0422

MODELING OF EMBEDDED HUMAN SYSTEMS

Jonathan Sprinkle

University of Arizona

JULY 2013
Final Report

DISTRIBUTION A: Approved for public release.

AIR FORCE RESEARCH LABORATORY
AF OFFICE OF SCIENTIFIC RESEARCH (AFOSR)
ARLINGTON, VIRGINIA 22203
AIR FORCE MATERIEL COMMAND

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 23-07-2013	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 6/1/2009-4/30/2012

4. TITLE AND SUBTITLE Modeling of Embedded Human Systems	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER FA9550-091-0519
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Jonathan Sprinkle	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Arizona 1230 E Speedway Blvd, Bldg #104 Tucson, AZ 85721-0104	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFOSR
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-OSR-VA-TR-2013-0422

12. DISTRIBUTION / AVAILABILITY STATEMENT

Publicly Releasable

DISTRIBUTION A: APPROVED FOR PUBLIC RELEASE

13. SUPPLEMENTARY NOTES

n/a

14. ABSTRACT

This final report provides the results of the Modeling of Embedded Human Systems (MEHS) project, Award #FA9550-091-0519. The goal of the work was a new approach to modeling embedded human systems; that is, systems where components are both human and computational. These systems are unique since the differences in time for a computational controller, as opposed to a human-centric controller, will require special consideration for system-wide feasibility and stability. Our approach focused on several research areas for modeling the interaction and control of such systems for robust behavior, determining the key abstractions for defining families of such systems, and developing a methodology for simulating, exercising, and potentially verifying these models based on the known qualities of the human, computational, and physical components. Our approach leveraged results from control of distributed vehicle systems, system stability and controllability, modeling, controller synthesis, and real-time systems.

15. SUBJECT TERMS

Control systems; software; modeling, domain-specific modeling; reachability; human-in-the-loop; embedded human

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Jonathan Sprinkle
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)* 510 914 1460

Final Report: Modeling of Embedded Human Systems

Principal Investigator: Jonathan Sprinkle
Institution: University of Arizona

Program Manager: Dr. Robert Bonneau AFOSR/RSL
Directorate: Mathematics, Information and Life Sciences (RSL)
Thrust: Systems and Software

Abstract

This final report provides the results of the Modeling of Embedded Human Systems (MEHS) project, Award #FA9550-091-0519. The goal of the work was a new approach to modeling embedded human systems; that is, systems where components are both human and computational. These systems are unique since the differences in time for a computational controller, as opposed to a human-centric controller, will require special consideration for system-wide feasibility and stability. Our approach focused on several research areas for modeling the interaction and control of such systems for robust behavior, determining the key abstractions for defining families of such systems, and developing a methodology for simulating, exercising, and potentially verifying these models based on the known qualities of the human, computational, and physical components. Our approach leveraged results from control of distributed vehicle systems, system stability and controllability, modeling, controller synthesis, and real-time systems.

Modeling of Embedded Human Systems: Statement of Objectives

We addressed the problem of modeling human-in-the-loop (i.e., embedded human) systems, specifically of the differences between automatic control and/or decision points (typically performed by computational components), and human-centric control and/or decision points (typically performed by human components). Unless such systems are properly modeled, heterogenous systems may not be flexible enough to substitute human components for computational components over the lifetime of a single deployment, or even the lifetime of the system, due to constraints and latencies in the communication and human/computational components.

Previous work focused on fundamental abstractions for mixed-component systems, but with the following constraints: the system was made up of a single vehicles under control, and only two vehicles (including the vehicle under control) under consideration. However, those approaches permitted freedom of expression for the modes of the system under control, and the results focused on synthesis of proof-carrying models that demonstrated the safety and robustness of system models, even under disturbance.

The final results can be correlated into the below core areas:

- **Develop abstractions of the embedded human system** as a fully computational stochastic system with bounded latencies and bounded probabilistic execution time; we used these abstractions to analyze the properties of such a system, and examine or develop methods to compose the behavior of various computational components, to relieve issues of latency; we then extend these systems to determine whether the introduction of an embedded human decision component obeys the same models of composition.
- **Develop a mathematical framework for integrating embedded human systems**, in which the kinematic behavior and saturation bounds of a vehicle under control can be used to synthesize a system-level simulation that takes into account the computational or embedded human component's decision points.
- **Extend integrative modeling techniques** to synthesize a series of initial conditions to exercise the system; we then use the discrete models of the mission controller, combined with the continuous models of the vehicle controller, to search for corner cases that may reveal weaknesses in the overall design.

These three major research tasks leverage existing work by others, as well as the proposer. The broad areas of existing research are control, modeling, and real-time systems. We next discuss particular research results give us foundations upon which we can build new abstractions, frameworks, and models that enable scalability and confidence of these distributed embedded human systems.

Modeling of Embedded Human Systems: Project Narrative

1 Motivation

The distributed battlefield presents a tremendous set of new capabilities, with the usual condition that new problems are introduced. The management, monitoring, and control of these new capabilities by a central authority requires significant information processing by the computational back-end, as well as humans in the decision loop. These human operators in the loop are so-called *embedded humans*, meaning that the human operator is a piece of the system as much as an embedded computer. In manned systems, embedded humans flying on board may need to make decisions based on information from local or distributed computational systems, as well as work with unmanned systems. In in the case of unmanned systems, an embedded human must make decisions and issue commands without the benefit of situational awareness. For unmanned systems especially, an embedded human is acting on data which is old (due to latency); their decision, once given, will take time to arrive at the system; and, the information needed to make that decision may be supplemented to give the same assurances of decisions made by an on-board human.

Given this new model of many sources of information requiring decision or behavior specification from human operators, the information and control choices available to an operator or monitor at any time (whether on board or not) is comparable to the choices available to an aircraft pilot. There are several characteristics of these distributed battlefield systems:

- timing of decisions is important, as communication and decision latency is on the order of seconds;
- complexity of the global state requires multiple perspectives of multiple interacting systems;
- despite significant differences in unmanned systems platforms, the decision points by humans in the loop may be similar;
- even when controlling unmanned systems, humans are in the loop to perform mission-level decision-making; and
- dependable human reaction requires training with the system; this includes system and protocol training, as well as training with any software user interface.

These characteristics give rise to the following observations:

- the latency of a human's decision, coupled with the potential for human error, is indistinguishable from a poorly designed algorithm, or poorly scheduled/high-latency system;
- system operation may require rapid decisions to be made, and latency (further) shrinks this decision window;
- cascading latencies may relegate certain designs to be unusable with a human-in-the-loop;
- system complexity, or ethical protocol, may require a human in the loop even for "simple" decisions;
- different decisions may utilize different subsets of the full state of the system; and
- the control inputs required by an embedded human may vary based on whether the human is making decisions for local or distributed systems.

These observations provide the context in which we propose this research: namely, that the system operation relies on human interaction, and that human interaction always carries with it the risk of latency of response, or incorrect use of the user interface. These problems plagued embedded systems designs, where instead of a human operator, the issue at hand was a buggy component, or a communications/scheduling model that did not scale with new components. New advancements in theoretical computer science have solved many of these problems in scaleability, but these also depend on a decidable worst-case execution time for each component in the system for static scheduling. Such metrics may be available for autonomous controllers, but decision points involving an embedded human will require additional models or degraded modes of operation to ensure high-confidence in the system.

2 Background

Our approach leveraged results from control of distributed vehicle systems, computational methods in control, modeling, and real-time embedded systems. These are included again in the final report to provide context for our results.

2.1 Distributed Vehicle Systems

Control of distributed vehicle systems is a rich research area in the domain of control systems. In particular, decentralized algorithms are useful for teams of homogeneous and heterogeneous systems in a mixed-initiative environment. Previous work in formation flight (as a decentralized controller) is given in [1], which also includes flight information regarding an actual test between a human and autonomous pilot. The proposed work does not center specifically on decentralized control, so this work is somewhat tangential: however, it relates the issue of communication between a human pilot and an autonomous craft, specifically on the behavior in degraded communication. Such is the effort described in [2], where communications were severed by software, and a reliable behavior of the UAV resulted that was predictable by the leader (human pilot). This work also includes several interesting cooperative modeling languages, among them the Computation and Control Language (CCL) [3], which provides a method to specify the aspects of control software using high-level directives.

Given that many distributed vehicle systems utilize both a discrete and continuous model to describe their behaviors and algorithms, the use of a hybrid system formalism can provide a concise representation of such models. From [4], a general hybrid system automaton H is defined as follows:

$$H = (Q, X, \Sigma, V, Init, f, Inv, R) \tag{1}$$

where Q is the set of discrete states of the automaton, X is the set of continuous state variables, Σ is the set of discrete input variables, V is the set of continuous input variables, $Init$ is the set of initial states of the system, f is a vector field representing the evolution of the continuous state variables, Inv is the set of states and inputs for which continuous evolution is allowed, and R is a reset relation defining the discrete transitions in the hybrid automaton.

2.2 Computational Methods in Control

Existing research in computational methods for system stability and controllability concentrates on the existing design of a controller, and knowledge of undesired (or desired) target states (in the past or future).

Many physical systems implicitly contain undesired states, and in mode-switching software transitions can be prevented based on known physical state. Computational foundations have been developed to understand safe transitions between control modes of a single vehicle [5], as well as controlled safe behavior of multiple vehicles [6]. Further, Tomlin and Sastry’s ISAT study [7] for DARPA in 2005[1] concretized the notion of an embedded human, who is a necessary component of the system.

The proposed work integrates notions of reachability of both physical (continuous) and discrete systems. In order to build upon this work, a brief summary of reachability calculations is given. The below approach examines the reachability of continuous systems, not that of hybrid systems, which has a discrete reachability formulation (see [8] for more information) in addition to piecewise continuous formulations.

Reachability Calculations The finite horizon backwards *reach set* $\mathcal{G}(t)$ is the set of states from which trajectories arise that lead to some target set $\mathcal{G}_0$ after exactly a specified time t. If the system’s dynamics include inputs, those inputs can be chosen to drive the trajectories toward or away from the target set; the interpretation of these inputs as either optimal controls or worst case disturbances depends on the circumstance. In Fig. 1, three snapshots of potential system states are shown in t_i, t_j, t_k.

Forward reachability is the determination of all potential future states given initial conditions and disturbances. If $t_i < t_k$, then Fig. 1 represents forward reachability of initial conditions at time t_i. Thus, points 1, 2, and 4 are not reachable, and the system may be valued at point 3 within $t_k - t_i$ seconds. *Backward*

[1] This study was performed under Dr. John Bay in IXO.

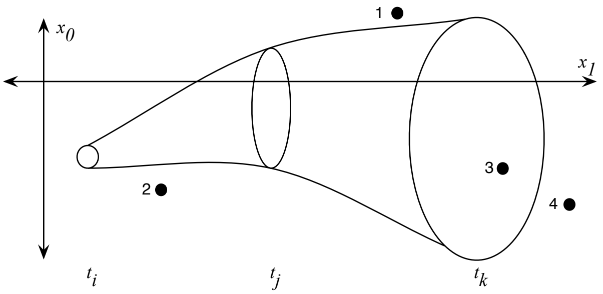

Figure 1: Abstraction of reachability. The ellipses denoted by $t_{\{i,j,k\}}$ show the potential values in $x_0 \times x_1$ of the system at those times.

reachability is the determination of all initial conditions that could result in some final condition. If $t_k > t_k$, then Fig. 1 represents a backward reachability, of target conditions at time t_i. Thus, points 1, 2, and 4 cannot reach anywhere in the final condition area within $t_k - t_i$ seconds. Depending on the formulation used for reachable sets, they can represent target desired, or undesired states (safety, or unsafety).

The target and reach sets are represented by an implicit surface function $\psi_{\text{surf}}(x,t)$, $\mathcal{G}_0 = \{x \in \mathbb{X} \mid \psi_{\text{surf}}(x,0) \leq 0\}, \mathcal{G}(t) = \{x \in \mathbb{X} \mid \psi_{\text{surf}}(x,t) \leq 0\},$, where $\mathbb{X}$ is some well-behaved configuration space. Construction of the implicit surface function $\psi_{\text{surf}}(x,0)$ for $\mathcal{G}_0$ is generally straightforward: there are simple functions for common shapes such as circles, spheres, cylinders, halfspaces and prisms, and the set operations of union, intersection and complement become the pointwise functional operations minimum, maximum and negation.

Calculation of the backwards reach set then reduces to solving an initial value HJ PDE. Let the system dynamics be given by $\dot{x} = f(x,u)$, where f is bounded and Lipschitz continuous in x. Choose the input signal u to drive trajectories toward $\mathcal{G}_0$ and assume that it is measurable and bounded at each point of time $u(t) \in \mathcal{U}$, where $\mathcal{U}$ is compact. Then the implicit surface function $\psi_{\text{surf}}(x,t)$ for the backwards reach set $\mathcal{G}(t)$ is the solution to the initial value HJ PDE

$$\frac{\partial \psi_{\text{surf}}}{\partial t} - H\left(x, \frac{\partial \psi_{\text{surf}}}{\partial x}\right) = 0, \tag{2}$$

where $H(x,p) = \min_{u \in \mathcal{U}} p^T f(x,u)$ and $\psi_{\text{surf}}(x,0)$ defined as above. For more details, see [9].

Analytic solution of this nonlinear PDE is not usually possible; in fact, it often does not have a classical differentiable solution at all. Fortunately, a well defined weak solution exists, and numerical methods have been designed to approximate it. Mitchell's Toolbox of Level Set Methods [10] is a collection of such algorithms that runs in Matlab.

2.3 Modeling and Metamodeling

Formalisms to address complexities in software and systems can be developed to abstract the necessary aspects of the system into *models*. These models are computational (as in Simulink models of transfer functions), as well as structural (as in class diagrams of the software architecture), and several aspects may be required to cover the spectrum of abstractions required for a system.

The research discipline of *model-integrated computing* emphasizes the explicit specification of a model of that system's behavior, structure, and/or analysis, and attaches that model to some *semantic domain*, where it can be analyzed, executed, or used to generate software or other models. This style of modeling allows for families of systems (domains) to be formalized by a *metamodel*, and new domains to be created by composing metamodels. Metamodeling has recently been applicable to describing complex systems as a family, thus permitting instances to be generated based on policies described generally to the family of systems. Work by Vangheluwe et al. [11] addresses the use of models for designing a reactive system. Mosterman, Sztipanovits and others [12] have shown the relevance of metamodeling in control systems design, particularly abstractions for such models. Fig. 2 shows the methodology of MIC, to properly abstract a system into a model-integrated program synthesis (MIPS) environment, which can be interpreted to synthesize application software, or analysis software, in order to locate design weaknesses.

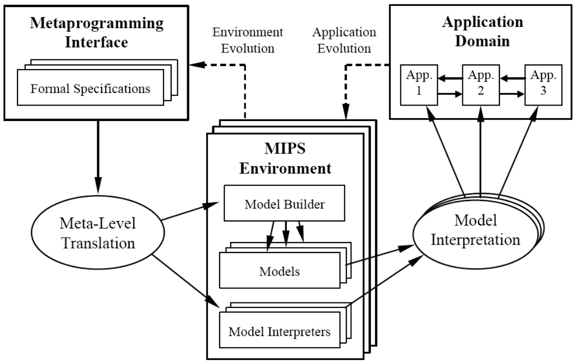

Figure 2: Model-Integrated Computing provides an infrastructure for synthesizing applications based on models of their structure and behavior. Additional interpretations of the same model, provide analysis points for system verification, etc.

For our proposed research, we will be focusing on (a) the key abstractions for an embedded human system, and (b) what kinds of analysis can be done on such models. As such, we plan to utilize the MIC infrastructure as a lightweight method to develop our algorithms, and their proofs of concept.

2.4 Real-time Embedded Systems

For systems where computation time is a large portion of the system's overall runtime, small variances in computation time can lead to corner cases of the systems state space where behavior diverges from expected values. The algorithm for a particular system component may not converge for certain areas of the statespace, and a value may not be returned as quickly (or at all).

Solutions to problems in such *ad hoc* scheduling and performance of real-time embedded systems include the introduction of logical execution time (LET) semantics, through Henzinger's Giotto [13]. Such solutions involve setting a logical execution time that is sufficiently longer than the worst-case-execution-time (WCET) of a computational component [14].

Additionally, work by Lee et al. to understand models of computation of component-based systems [15, 16] provides a common description for certain classes of systems, and limitations which those classes have. Of particular relevance to this research are the rendezvous domain, and the process networks domain. In the rendezvous domain (also called synchronous message passing), a process p_1 sends a message to process p_2, and does not progress until p_2 is ready to receive the message. The process networks domain is slightly different, in that processes do not block execution while communicating, but rather communicate by FIFO buffers. If the timing of each process is deterministic, then the collection of processes executes deterministically; if the timing of processes is *not* well-defined, or is nondeterministic, then nondeterministic behavior can emerge.

3 Results

Objective In order to address the problem, our main approach developed new models that take into account the different requirements and expectations of computational or human components in a system. This included formalisms to account for specification of various latencies of communication, computation, decision, etc., as well as formalisms for hybrid behavior of various systems given contradictory commands, latent responses, unsafe commands, etc.

Key papers core to this mission are:

- J. Sprinkle and D. Chu, "Modeling languages applied to decision controllers for embedded human systems," in *Seventh IEEE International Conference and Workshops on Engineering of Autonomic and Autonomous Systems (EAsE 2010).*
- J. Ding, J. Sprinkle, C. Tomlin, S. S. Sastry, and J. L. Paunicka, "Reachability calculations for

vehicle safety during manned/unmanned vehicle interaction," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 138–152, 2012.

- D. Chu. "Design Validation of Multi-mode Systems." Ph.D. Dissertation, *University of Arizona*, (tentative), 2013. Included as an attachment.

This project followed a research agenda to address this problem, by providing a path of research objectives that culminates in metrics of success to evaluate the research progress. The specific objectives are below.

- **Developed abstractions of the embedded human system** as a fully computational stochastic system with bounded latencies and bounded probabilistic execution time; the specific abstraction we developed is described in the attached papers, and focuses the system into a bipartite graph, where transition between modes is alternately enabled by human action, and computer decision.
- **Developed a mathematical framework for integrating embedded human systems**, in which the kinematic behavior and saturation bounds of a vehicle under control can be used to synthesize a system-level simulation that takes into account the computational or embedded human component's decision points. These results are codified in the attached papers through the in-depth investigation of the approach to several problems in the area of mixed mode manned/unmanned systems.
- **Extended integrative modeling techniques** to synthesize a series of initial conditions to exercise the system; we then used the discrete models of the mission controller, combined with the continuous models of the vehicle controller, to search for corner cases that may reveal weaknesses in the overall design.

3.1 Abstractions of an Embedded Human System

In "Modeling languages applied to decision controllers for embedded human systems" we described in detail the abstractions (both fundamental models, and actual domain-specific models) for embedded human system interaction. These abstractions have a few key benefits:

- Scalability: systems can be composed from one another, and connected at well-defined points;
- Decoupled: since some modes are computer controlled and others are human controlled, we have decoupled the verification problem into two-mode transitions; multi-mode verification is possible by verifying that each mode settles to an invariant region of the state space;
- Readability: models are easier to understand, and can be analyzed visually as well as with algorithms.

3.2 A Mathematical Framework for Integrating Embedded Human Systems

In "Reachability calculations for vehicle safety during manned/unmanned vehicle interaction" we demonstrated our mathematical framework for verification and integration of the embedded human system. Those results validated the previous result in the following ways:

- Invariant sets: we demonstrated the ability to verify that invariant regions of the state space could be determined;
- Transition logic: we demonstrated the ability to backwards-propagate the transition between modes, by ensuring that reach sets satisfied the conditions of transition;
- Recovery: we demonstrated the ability to recover from modal confusion, by selecting from reach sets that are known to be safe over a short horizon

3.3 Extending Integrative Modeling Techniques

We demonstrate in Chu's attached thesis draft, the power of our modeling techniques to generalize the process of synthesizing, executing, and simulating the mathematical and modeling frameworks we developed. Key results in Chu's thesis include:

- The use of the modeling language and its bipartite graph requirements;
- Concurrent calculation of the reachsets across an arbitrary number of nodes;
- Synthesis of simulations from initial conditions, where a computer autonomously executes human commands;
- Synthesis of simulations where a human must send necessary transition commands;
- Display of relevant reachsets during execution.

4 Additional Results

In addition to these core results, this award was also instrumental in the below papers and their results, which either directly or indirectly contributed to the core research of the proposal:

- S. Whitsitt and J. Sprinkle, "Modeling autonomous systems," *AIAA Journal of Aerospace Information Systems*, (in press, accepted in final form), 2013.
- J. M. Eklund, J. Sprinkle, and S. S. Sastry, "Switched and Symmetric Pursuit/Evasion Games With Online Model Predictive Control," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 604–620, 2012.
- J. Sprinkle and B. Eames, "Time-triggered buffers for event-based middleware systems," *Innovations in Systems and Software Engineering*, vol. 7, no. 1, pp. 9–22, 2011.
- D. Chu, J. Sprinkle, R. Randall, and S. Shkarayev, "Simulations and flight experiments of transition maneuvers of a VTOL micro air vehicle," *International Journal of Micro Air Vehicles*, vol. 2, no. 2, pp. 69–89, June 2010, ISSN 1756-8293.
- A. Schuster and J. Sprinkle, "Synthesizing executable simulations from structural models of component-based systems," *Electronic Communications of the European Association of Software Science and Technology (EASST)*, vol. 21, p. 10 pages, 2009, ISSN 1863-2122.
- E. Jones and J. Sprinkle, "autoVHDL: a domain-specific modeling language for the auto-generation of VHDL core wrappers." in *Proceedings of the compilation of the co-located workshops on DSM'11, TMC'11, AGERE!'11, AOOPES'11, NEAT'11, & VMIL'11.* New York, NY, USA. ACM, pages 71-76, 2011. `http://dx.doi.org/10.1145/2095050.2095063`
- J. Sprinkle, "Analysis of a metamodel to estimate complexity of using a domain-specific language," in *Proceedings of the 10th Workshop on Domain-Specific Modeling*, ser. DSM '10. New York, NY, USA: ACM, 2010, pp. 13:1–13:6. `http://dx.doi.org/10.1145/2060329.2060359`
- D. Chu, J. Sprinkle, R. Randall, and S. Shkarayev, "Automatic control of VTOL micro air vehicle during transition maneuver," in *AIAA Guidance, Navigation and Control Conference.* AIAA, August 2009, p. 16 pages, AIAA-2009-5875.

In addition, the below results are pending acceptance or revision:

- Jonathan Sprinkle. "Metamodel-Based Metrics for Complexity of Using a DSML." in preparation for submission to *Software and Systems Modeling*, (revising per reviewer comments) (2013)
- Sean Whitsitt, Diyang Chu, and Jonathan Sprinkle "Closing the Loop: New Modeling Techniques for Verifying Constraints on Cyber-Physical Systems." in preparation for submission to *Software and Systems Modeling*, (revising per reviewer comments) (2013)
- Diyang Chu and Jonathan Sprinkle. "Parallel Computation of Reachable Sets for Multi-mode Controllers." in preparation for submission to *AIAA J. Guidance, Control, and Dynamics*, (revising per reviewer comments) (2013)

5 Other Information

5.1 Personnel

The following personnel were part of the project:

- Jonathan Sprinkle (PI)
- Diyang Chu (PhD Student, 2013 (scheduled))
- Kun Zhang (PhD Student, 2014 (scheduled))
- Maribel Hudson (MS, 2011)
- Ivan Lizarraga (BS, 2010)
- Sean Whitsitt (PhD Student, 2014 (scheduled))

5.2 Awards

- In 2011, Sprinkle was elevated to IEEE Senior Member
- In 2009, Hudson was named "Outstanding Senior" for the Arizona College of Engineering
- In 2009, Sprinkle was awarded UA's College of Engineering Faculty Support Grant, made possible by the generosity of Ed and Joan Biggers to the University of Arizona College of Engineering

References cited

[1] T. Keviczky, F. Borrelli, and G. J. Balas, "Decentralized receding horizon control for large scale dynamically decoupled systems," *Automatica*, vol. 42, pp. 2105–2115, Dec. 2006.

[2] S. Waydo, J. Hauser, R. Bailey, E. Klavins, and R. Murray, "Uav as a reliable wingman: A flight demonstration," *Control Systems Technology, IEEE Transactions on*, vol. 15, pp. 680–688, July 2007.

[3] E. Klavins, "A language for modeling and programming cooperative control systems," *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, vol. 4, pp. 3403–3410 Vol.4, 26-May 1, 2004.

[4] C. Tomlin, J. Lygeros, and S. Shankar Sastry, "A game theoretic approach to controller design for hybrid systems," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 949–970, Jul 2000.

[5] A. M. Bayen, I. M. Mitchell, M. M. K. Oishi, and C. J. Tomlin, "Aircraft autolander safety analysis through optimal control-based reach set computation," *AIAA Journal of Guidance, Control Dynamics*, vol. 30, pp. 68–77, January-February 2007.

[6] M. Oishi, I. Mitchell, A. Bayen, and C. Tomlin, "Invariance-preserving abstractions of hybrid systems: Application to user interface design," *Control Systems Technology, IEEE Transactions on*, vol. 16, no. 2, pp. 229–244, March 2008.

[7] C. J. Tomlin and S. S. Sastry, "Embedded humans," tech. rep., DARPA ISAT Study, 2005.

[8] R. Alur, T. Henzinger, G. Lafferriere, and G. Pappas, "Discrete abstractions of hybrid systems," *Proceedings of the IEEE*, vol. 88, pp. 971–984, Jul 2000.

[9] I. Mitchell, A. Bayen, and C. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *Automatic Control, IEEE Transactions on*, vol. 50, pp. 947–957, July 2005.

[10] I. M. Mitchell, "The flexible, extensible and efficient toolbox of level set methods," *Journal of Scientific Computing*, December 2007.

[11] T. H. Feng, M. Zia, and H. Vangheluwe, "Multi-formalism modelling and model transformation for the design of reactive systems," in *Summer Computer Simulation Conference (SCSC)*, pp. 505–512, Society for Computer Simulation International (SCS), July 2007.

[12] P. Mosterman, J. Sztipanovits, and S. Engell, "Computer-automated multiparadigm modeling in control systems technology," *Control Systems Technology, IEEE Transactions on*, vol. 12, no. 2, pp. 223–234, March 2004.

[13] T. Henzinger, B. Horowitz, and C. Kirsch, "Giotto: A time-triggered language for embedded programming," *Proceedings of the IEEE*, vol. 91, pp. 84–99, 2003.

[14] S. Matic and T. Henzinger, "Trading end-to-end latency for composability," *Real-Time Systems Symposium, 2005. RTSS 2005. 26th IEEE International*, pp. 12 pp.–, Dec. 2005.

[15] J. Liu and E. Lee, "Timed multitasking for real-time embedded software," *Control Systems Magazine, IEEE*, vol. 23, no. 1, pp. 65–75, Feb 2003.

[16] J. Liu and E. A. Lee, "A component-based approach to modeling and simulating mixed-signal and hybrid systems," *ACM Trans. Model. Comput. Simul.*, vol. 12, no. 4, pp. 343–368, 2002.