From Cold Start to Critical: Formal Synthesis of Hybrid Controllers

PI: Dane A. Sabo
dane.sabo@pitt.edu

Advisor: Dr. Daniel G. Cole
dgcole@pitt.edu

Sunday 14th September, 2025

1 Goals and Outcomes

The goal of this research is to develop a unified framework combining temporal logic synthesis with continuous-time verification methods to create autonomous hybrid control systems with complete correctness guarantees. Hybrid control systems have great potential for autonomous control applications because they can switch between different control laws based on discrete triggers in the system's operating range. This approach allows autonomous controllers to use several tractable control laws optimized for different regions in the state space, rather than relying on a single controller across the entire operating range. But, the discrete transitions between control laws in hybrid controllers present significant challenges in proving stability and liveness properties for the complete system. While tools from control theory can establish properties for individual control modes, these guarantees do not generalize when mode switching is introduced. Conversely, significant advances in formal methods have enabled automatic synthesis of discrete controllers from temporal logic specifications—tools like Strix can generate provably correct switching logic for complex logical requirements. However, these synthesis approaches assume instantaneous mode transitions and operate purely in discrete state spaces. In hybrid systems, transitions occur along continuous trajectories governed by differential equations, creating a fundamental verification gap that neither purely discrete synthesis nor traditional control theory can address alone.

This research addresses a fundamental challenge in hybrid controller synthesis and verification by unifying discrete system synthesis with continuous system analysis. We will leverage formal methods to create controllers that are correct-by-construction, enabling guarantees about the complete system's behavior. To demonstrate this approach, we will develop an autonomous controller for nuclear power plant start-up procedures. Nuclear power represents an excellent test case because the continuous reactor dynamics are well-studied, while the discrete mode switching requirements are explicitly defined in regulatory procedures and operating guidelines. Current nuclear reactor control *is* already a hybrid system. For example, during reactor startup, operators must transition from initial cold conditions through controlled heating phases to predetermined power levels. Each phase employs different automated controllers: temperature ramp controllers during heatup, reactivity controllers approaching criticality, and load-following controllers during operation. The decision of when to switch between these controllers currently relies on human operators interpreting written procedures. Our approach would formalize such transition conditions and synthesize the switching logic automatically.

The capability to create high-assurance hybrid control systems has significant potential to reduce labor costs in operating critical systems by removing human operators from control loops. Nuclear power stands to benefit substantially from increased controller autonomy, as operations and maintenance represent the largest expense for current reactor designs. While emerging technologies such as microreactors and small modular reactors will reduce maintenance costs through factory-manufactured replacement components, they face increased per-megawatt operating costs if required to maintain traditional staffing levels. However, if increased autonomy can be safely introduced, these economic challenges can be addressed while maintaining safety standards.

If this research is successful, we will achieve the following outcomes:

1. **Formalize mode switching requirements as logical specifications that can be synthesized into discrete controller implementations.** The discrete transitions between continuous controller modes are often explicitly defined in operating procedures and regulatory requirements for critical systems. These natural language requirements will be translated

into temporal logic specifications, which will then be synthesized into provably correct discrete controllers for continuous mode switching.

2. **Develop and verify formal characterizations of hybrid mode dynamics and safety conditions.** We will establish mathematical frameworks distinguishing transitory modes with reachability requirements to target states from stabilizing modes with invariant maintenance properties. For linear dynamics, classical control theory will establish stability and performance within each mode. For nonlinear systems, reachability analysis will verify that transitory modes drive the system toward intended transitions while maintaining safety constraints, and that stabilizing modes preserve their designated operating regions. This unified approach will enable provable conditions for safe state space traversal and transition timing.
3. **Prove that hybrid system implementations achieve safety and performance specifications across operational mode sequences.** By synthesizing discrete controller transitions from logical specifications using correct-by-construction methods and verifying that continuous components perform appropriately between discrete transitions, we can establish mathematical guarantees that the hybrid system maintains safety constraints and meets performance requirements during autonomous operational sequences such as reactor startup procedures, where multiple control modes must be coordinated to achieve higher-level operational objectives.

2 State of the art and limits of current practice

This research will improve our ability to build high assurance autonomous hybrid control systems by connecting tools from different areas of the scientific literature. We will combine control and dynamics theory of hybrid systems with the discrete systems analysis found in the computer science space. These two fields are disparate, but both approach the problem of high assurance hybrid systems. First, we will discuss the control theory side. Hybrid systems from a differential equation perspective will be considered, and comparisons to the current state of the art of nuclear power control will be discussed. Then, we will discuss approaches for discrete and hybrid systems from the logical and computer science perspective. These gap between these two fields is the crux of the intellectual merit of this research.

2.1 Control Theory and Hybrid Systems

Hybrid systems have two components to their behavior. They have continuous dynamics called 'flow', and have discrete dynamics called 'jumps'. Hybrid systems can often be described as a set of differential and difference equations. An hybrid system can be defined as follows:

$$\dot{x}(t) = f(x(t), q(t), u(t)) \tag{1}$$

$$q(k+1) = \nu(x(k), q(k), u(k)) \tag{2}$$

In this description there are two functions: $f(\cdot)$ and $\nu(\cdot)$. $f(\cdot)$ defines the continuous dynamics, while $\nu(\cdot)$ defines the discrete dynamics. These functions take three inputs: the continuous states x, the discrete state q, and an optional control input u. $f_i(\cdot)$ here is a nonlinear function, where q changes the mode of the continuous dynamics, but is otherwise dependent on the current continuous states and control input. $\nu_i(\cdot)$ can be much more generally created. The only restriction on $\nu_i(\cdot)$ is that it must not create an infinite number of jumps in a finite amount of time. $\nu_i(\cdot)$ can depend a control input u, or can be *autonomous*, where the discrete mode is defined by the system states x and q. While an autonomous system may not have a control input, that does not mean the system is not controlled, as the continuous dynamics $f_i(\cdot)$ can be constructed such that the system is controlled by the continuous and discrete states x and q.

The hybrid systems we are most interested in here are *continuous* autonomous hybrid systems. These systems are hybrid systems whose continuous states x do not change during a jump, and do not rely on an external control input. This research is primarily aimed at control of physical systems whose continuous states x are physically required to be continuous across jumps. For example, a nuclear reactor control rod system may jump from a warm-up ramp control mode to a load following controller, but the continuous states of the reactor such as temperature and rod position can not instantaneously change.

Continuous hybrid systems are often constructed by piecing together control systems for different regions in the state space. This way of building a hybrid control system is intuitive–controllers are built for local regions with local objectives. The problem arises when analysis of these local controllers is generalized to the entire, global, hybrid system. Even in the most simple case of linear time invariant controller modes, global guarantees of stability can not be made using linear control theory. Instead, approaches using stitched networks of Lyapunov functions have been introduced. These Lyapunov functions are difficult to find, but can for some linear switched hybrid systems provide a way to verification.

Another way of verifying stability of hybrid control systems includes performing reachability analysis. Reachability is a formal methods tool that uses abstractifications of a system to determine output ranges of dynamic systems for a given input. Reachability is not limited to linear systems, and can be performed on nonlinear systems as well. Reachability for hybrid systems suffers from two main deficiencies. First, the analysis can take a very long time to compute. Second, reachability uses approximations. Hybrid systems that have trajectories of significantly long times can cause problems when using reachability because either the computation time makes the analysis infeasible, or the compounding approximation errors makes the analysis meaningless.

2.2 Formal Methods and Reactive Synthesis

Hybrid systems often are realized as cyber-physical systems. Cyber-physical systems have a computational component responsible for control of a physical process based on live sensors and actuators. Verification of these systems has drawn two different communities together. We have previously discussed the engineering perspective coming from the physical control theory background, but in this section we discuss work coming from the cyber oriented persuasion.

Many cyber systems lend themselves to a finite number of possible states. For these systems, a map of the possible states and the transition between them can be drawn and analyzed. This is called a *finite state machine*. Finite state machines can be easily verified, as all possible states can be exhaustively checked to meet requirements in a process called model checking. Model checking has been used extensively in high-assurance digital systems and to ensure software correctness for critical systems.

In order to check correctness, requirements for these automata must be defined. Typically this is done using a formal language such as temporal logic. Temporal logic allows system behaviors to be defined with temporal relations and includes four operators: next (X), eventually (F), globally (G), and until (U). With these operators, controller specifications can be created. In a nuclear context, one might use temporal logic to define safety behaviors of a reactor core control system. Let's consider an example:

Suppose we are trying to write safety requirements about SCRAM behavior. In plain English, one might say "If a high temperature alarm is triggered, the control rods will immediately be inserted and unable to be withdrawn unless reset by an operator." In a linear temporal logic specification, this would be written as follows:

$$\mathscr{G}(HighTemp \rightarrow \mathscr{X}(RodsInserted \wedge (\neg RodsWithdrawn \; \mathscr{U} \; OperatorReset))) \tag{3}$$

A significant body of work has been done around the translation of English requirements into temporal logic specifications. The Formal Requirements Elicitation Tool (FRET) developed by the NASA Ames Research Center to help bridge the gap between natural language and mathematical specification. Using FRET, one is able to take a bulk number of near-English requirements in a language called FRETish, and translate them automatically into linear temporal logic specifications. From this point, it can be examined whether or not the set of requirements define a realizable system, or if there exists conflicts between different specifications.

We have previously discussed that a set of specifications can be checked as to whether or not the constitute a realizable system. If a system is realizable, there are a significant number of tools that can synthesize reactive control systems from the set of logical specifications. Reactive systems are those that take an input, and produce a reaction (an output). They depend on the current

system state and input to produce the next state. Competitions such as the Reactive Synthesis Competition (SYNTCOMP) have existed for over a decade where different groups try to produce the best reactive synthesis algorithm. These systems are tested against a series of benchmarks to examine the number, quality, and resources consumed to produce realizations of reactive systems from logical specifications.

LIMITATION: while reactive synthesis exists, and we have an extensive amount of documentation on nuclear power regulation and operating procedures exists, we have not tried to combine the two together.

Finally, formal methods has contributed the hybrid automata and differential dynamic logic to try and solve the hybrid system verification problem. Hybrid automata are an expansion of finite automata. Hybrid automata define each node as being a control mode, similar to how finite state automata define each node as a single state. For hybrid automata, the node represents the *discrete state*. Meanwhile, the transitions between states indicate the transitions between continuous modes. These transitions represent the executions of $v(\cdot)$ that change the discrete state q. Hybrid automata introduce a way to graphically represent the transitions between continuous dynamic modes.

Differential dynamic logic (dL), on the other hand, is an expansion of linear temporal logic to include support for real numbers and differential equation solving. dL introduces two new operators focused on including dynamic behaviors. The first is the box modality $[\alpha]\phi$, which states that for all possible executions of the hybrid system α, ϕ holds. The second is the diamond modality $\langle\alpha\rangle\phi$, which states that for the hybrid system α, there is a trajectory where ϕ holds. With these two additional modalities, hybrid systems can be reasoned about directly while including the continuous dynamics. That does not mean that working with dL is easy however, as the effort to perform verification is encumbered by the knowledge requirement of differential equations, logical specifications, and then finally, sequent calculus to actually try and prove things written in dL. dL is expressive enough to capture any hybrid system behavior, but the effort to actually prove requirement adherence is challenging, even with automated proof assistant tools.

In the next section, we will discuss how this research will address these limitations and provide a path forward for building high assurance hybrid control systems.

References