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

Friday 19th September, 2025

High-Assurance Hybrid Controller Synthesis from Logical Specifications

PI: Dane A. Sabo, dane.sabo@pitt.edu
Advisor: Daniel G. Cole, dgcole@pitt.edu

Goal: The goal of this research is to use mathematical statements of requirements and automated tools to construct hybrid controllers that are provably free from design defects.

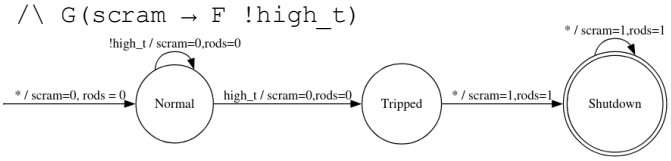
Outcomes: If this research is successful, we should be able to do the following:

1. Formalize design requirements and operational conditions as temporal logic specifications
2. Determine whether a controller implementation can be realized from a set of specifications
3. For unrealizable systems, identify where specification refinement is necessary
4. For realizable systems, synthesize the formal specification into a controller implementation

Controller Synthesis

- During normal operation, if a high temperature alarm is triggered, the reactor will immediately shutdown by inserting the control rods
- Once shutdown, the reactor will be unable to restart

```
G(high_t → X scram)
/\ G(scram → G rods)
/\ G(scram → (!scram U high_t))
/\ G(scram → F !high_t)
```



This research converts high-level requirements into logical specifications and uses automated tools to synthesize controller implementations.

Approach: Nuclear reactors are well studied systems of which we identify critical requirements during design. These requirements will be formalized as logical specifications. The core challenge of this research will be **automatically synthesizing** these specifications into a controller implementation that is **provably free of defects**. This process will include establishing whether a set of requirements are realizable. Realizable controllers in this context are derived from a set of specifications that provide sufficient detail to create an actual controller implementation. If a controller is not realizable, specification refinement guidance will be provided. Freedom from defects will be ensured by the use of automated formal methods tools.

Impact: Operations and maintenance are the largest costs in nuclear power. Conventional reactors are custom-built and require large maintenance teams, but relatively few operators. Small modular reactors (SMRs) and microreactors (MRs) invert this model: factory-made modules reduce construction and maintenance costs, but operator wages will become a major expense. If staffing needs stay the same, SMRs and MRs face stiff economic headwinds compared to other sources of power generation. By developing autonomous controllers with strong safety and performance guarantees, we can reduce operator burden, lower staffing requirements, and improve the economic viability of SMRs and MRs.

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 aims to advance high-assurance autonomous hybrid control systems by bridging disparate approaches from control theory and computer science. While both fields tackle hybrid system verification, they approach the problem from fundamentally different perspectives. Control theory emphasizes performance and stability, while computer science focuses on correctness through formal verification. This gap represents both the primary challenge and the key opportunity for intellectual contribution.

2.1 Control Theory and Hybrid Systems

Hybrid systems combine continuous dynamics ('flows') with discrete transitions ('jumps'). Following the standard formulation, a hybrid system can be expressed as:

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

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

Here, $f(\cdot)$ defines the continuous dynamics while $\nu(\cdot)$ governs discrete transitions. The continuous states x, discrete state q, and control input u interact to produce hybrid behavior. The discrete state q defines the current continuous dynamics mode. The only constraint on $\nu(\cdot)$ is avoiding Zeno behavior—infinite jumps in finite time. Our focus centers on continuous autonomous hybrid systems, where continuous states remain unchanged during jumps and no external control input is required. Physical systems naturally exhibit this property—a nuclear reactor switching from warm-up to load-following control cannot instantaneously change its temperature or rod position.

An intuitive approach to building hybrid controllers is to stitch together multiple simple controllers for different state space regions. This approach creates significant verification challenges. Even with linear time-invariant modes, global stability cannot be guaranteed using linear control theory alone. Instead, researchers have developed Lyapunov-based approaches, though finding appropriate Lyapunov functions remains challenging. Stability conditions for switched linear systems can provide necessary and sufficient conditions using multiple Lyapunov functions [1], but these methods require restrictive assumptions. Individual Lyapunov functions must be monotonically nonincreasing at every switching time, which proves impractical for many real systems. Common Lyapunov functions face even stricter existence conditions, often impossible for systems with fundamentally different dynamics across modes [2, 3].

LIMITATION: Lyapunov-based methods of ensuring stability are prohibitively challenging to apply to hybrid control systems. Finding Lyapunov functions to prove stability in the sense of Lyapunov is not practical for working with hybrid systems. Additional requirements on the Lyapunov functions makes finding appropriate functions extremely difficult.

Reachability analysis offers an alternative verification approach by computing system output ranges for given inputs. Unlike Lyapunov methods, reachability extends naturally to nonlinear systems. Hamilton-Jacobi frameworks established the mathematical foundation for computing reachable sets in continuous and hybrid systems, enabling formal verification of safety-critical applications [4].

LIMITATION: Reachability analysis is not scalable to large hybrid systems. Reachability analysis faces two critical limitations. First, computational complexity grows exponentially with state dimension—current methods remain limited to 6-8 dimensional systems despite recent algorithmic advances. Second, approximation errors compound over long time horizons, potentially

rendering analysis meaningless for extended trajectories. Zonotope-based methods suffer accuracy degradation when propagating across mode boundaries, while ellipsoidal methods produce conservative over-approximations that become increasingly pessimistic with each mode transition.

Recent work on using control barrier functions has improved hybrid system verification efforts. Neural network based control barrier function approximations achieve 10-100X speedup over classical Hamilton-Jacobi methods while maintaining safety guarantees for 7-dimensional autonomous racing systems [5]. This breakthrough demonstrates that deep neural networks can approximate Hamilton-Jacobi partial differential equations and enables application to previously intractable high-dimensional systems, but future work must address the explainability problem for neural networks to ensure control barrier functions are valid.

2.2 Formal Methods and Reactive Synthesis

Correctness requirements are specified using temporal logic, which captures system behaviors through temporal relations. Linear Temporal Logic (LTL) provides four fundamental operators: next (X), eventually (F), globally (G), and until (U). Consider a nuclear reactor SCRAM requirement:

Natural language: "If a high temperature alarm triggers, control rods must immediately insert and remain inserted until operator reset."

This plain language requirement can be translated into a rigorous logical specification.

LTL specification:

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

Once requirements are translated into these logical specifications, they can be checked using computational tools. Cyber-physical systems naturally exhibit discrete behavior amenable to formal analysis. Systems with finite states can be modeled as finite state machines, where all possible states and transitions are explicitly enumerable as logical specifications. This enables exhaustive verification through model checking—a technique extensively employed in high-assurance digital systems and safety-critical software. This mathematical framework has been extended to hybrid automata [6], which established the standard model combining discrete control graphs with continuous dynamics. Hybrid automata bridge program-analysis techniques to hybrid systems with infinite state spaces through symbolic model-checking based on reachability analysis [7] but are not scalable for the same limitations mentioned earlier with other reachability techniques.

NASA's Formal Requirements Elicitation Tool (FRET) bridges natural language and mathematical specifications through FRETish—a structured English-like language automatically translatable to temporal logic [8]. FRET enables hierarchical requirement organization, realizability checking for specification conflicts, integration with verification tools like CoCoSim, and runtime monitoring through their Copilot tool.

From realizable specifications, reactive synthesis tools automatically generate controllers. The Reactive Synthesis Competition (SYNTCOMP) has driven algorithmic improvements for over a decade, with tools like Strix dominating recent competitions through efficient parity game solving [9,10]. Strix is able to translate linear temporal logic specifications into deterministic automata automatically while maximizing generated automata quality.

LIMITATION: Unexplored application of temporal logic requirements in nuclear con-

trol. Despite extensive nuclear power documentation and mature reactive synthesis tools, little work has combined these for nuclear control applications. Nuclear procedures are written in structured natural language that maps well to temporal logic, yet the synthesis community has not engaged with this domain. This represents a significant unexplored opportunity where formal methods could provide immediate practical impact.

Hybrid automata extend finite automata by representing discrete states as control modes with continuous dynamics. Transitions between nodes indicate discrete state changes through $v(\cdot)$ execution. This provides intuitive graphical representation of mode switching logic. Differential dynamic logic (dL) expands this idea and offers the most complete logical foundation for hybrid verification. dL introduced two key modalities [11, 12]:

- Box modality $[\alpha]\phi$: for all executions of hybrid system α, property ϕ holds
- Diamond modality $\langle\alpha\rangle\phi$: there exists an execution of α where ϕ holds

These modalities enable direct reasoning about hybrid systems including continuous dynamics. The KeYmaera X theorem prover implements dL through axiomatic tactical proving, successfully verifying collision avoidance in train and aircraft control [13].

LIMITATION: There is a high expertise barrier for dL verification, and scalability remains an issue. While dL is expressive enough for any hybrid behavior, verification requires expertise in differential equations, logical specifications, and sequent calculus. The proof effort remains challenging even with automated assistance. Users must understand both the mathematical intricacies of their system and the logical framework, then guide the prover through complex proof steps. This expertise barrier prevents wider adoption despite the framework's theoretical completeness.

The state of the art reveals a field in transition. Traditional boundaries between control theory and formal methods are dissolving through learning-based approaches and compositional techniques. While fundamental challenges remain—particularly scalability and the theory-practice gap—the convergence of approaches promises to enable verification and synthesis for systems of unprecedented complexity. The next section describes how this research will contribute to bridging these gaps through unified synthesis frameworks applied to nuclear reactor control.

3 Research Approach

This research will overcome the limits of current practice to build high assurance hybrid control systems for critical infrastructure. To do this, we must accomplish three main thrusts:

1. Translate operating procedures and requirements into temporal logic formulae
2. Create the discrete half of a hybrid controller using reactive synthesis
3. Develop continuous controllers to operate between modes, and verify their correctness using reachability

In the following sections I will discuss how these thrusts will be accomplished.

3.1 $(Procedures \wedge FRET) \rightarrow TemporalSpecifications$

The motivating factor behind this work is the fact that commercial nuclear power operations have remained manually controlled by human operators, despite advances in control systems sophistication. The frustrating part of this is that the procedures performed by human operators are highly prescriptive. Human operators in nuclear power plants may not be entirely necessary with the technology we have today.

Written procedures and requirements in nuclear power are descriptive enough we may be able to translate them into logical formulae with little effort. If we can accomplish this, we can automate existing procedures without inducing reengineering. The easiest way to accomplish this task will be through the use of automated translational tools. Tools like FRET can help accomplish this task.

FRET uses a specialized requriements languaged called FRETish to restrict requirements to be written in easy to understand components, but without leaving room for ambiguity. FRET does this by forcing requirements to contain six possible parts:

1. Scope: *What modes does this requirement hold?*
2. Condition: *Scope + more specificity*
3. Component: *What does this requirment affect?*
4. Shall
5. Timing: *When does the response happen?*
6. Response: *What should happen?*

FRET provides functionality to check the *realizability* of a system. Realizability checks whether or not the written requirements are complete by examining the six components that make up requirements. Complete requirements are those that do not conflict with one another, and do not leave any behavior as undefined. Systems that are not realizable from the procedure definitions and design requirements are problematic beyond realizing autonomy. These systems have behavior in their systems that is a physical equivalent of a software bug. With FRET, we can catch these errors while building an autonomous controller. The second type of error including undefined behaviors are those that may be left up to human judgement during control. This is not desirable for high assurance systems, as however trained humans can be, are still prone to errors. Addressing these vulnerabilities in FRET while building an autonomous controller will deliver a controller free of these vulnerabilities.

FRET also provides the capability to export the requirements created in a temporal logic format. This capability allows us to leave FRET and move onto the next step of our approach, where we will synthesize the discrete mode switching behavior.

3.2 $(TemporalLogic \wedge ReactiveSynthesis) \rightarrow DiscreteAutomata$

In this section of our approach we describe how the discrete component of the hybrid system will be created. The formal specifications created in FRET will be used with reactive synthesis tools to generate the mode switching components. These components effectively make up the human component of reactor operation by automating the decision points typically specified in written procedures. By removing the human component, we eliminate the possibility of human error and advance hybrid system autonomy.

Reactive synthesis is an active field in computer science that focuses on the synthesis of discrete controllers created from temporal logic specifications. Reactive defines that the system responds to inputs to create outputs. These systems are finite in size, where each node represents a unique set of discrete states q.

References

[1] José C Geromel and Patrizio Colaneri. Stability and stabilization of continuous-time switched linear systems. *SIAM Journal on Control and Optimization*, 45(5):1915–1930, 2006.

[2] Michael S Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

[3] Daniel Liberzon. *Switching in systems and control*. Birkhäuser Boston, 2003.

[4] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on Automatic Control*, 50(7):947–957, 2005.

[5] Shuo Yang, Yiwei Chen, Xiang Yin, and Rahul Mangharam. Learning local control barrier functions for hybrid systems. *arXiv preprint arXiv:2401.14907*, 2024.

[6] Rajeev Alur, Costas Courcoubetis, Thomas A Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, pages 209–229. Springer, 1993.

[7] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.

[8] Dimitra Giannakopoulou, Anastasia Mavridou, Julian Rhein, Thomas Pressburger, Johann Schumann, and Nija Shi. Capturing and analyzing requirements with fret. Technical Report NASA/TM-20220007610, NASA Ames Research Center, 2022.

[9] Philipp J Meyer and Michael Luttenberger. Strix: Explicit reactive synthesis strikes back! In *International Conference on Computer Aided Verification*, pages 578–586. Springer, 2018.

[10] Swen Jacobs, Roderick Bloem, Romain Brenguier, et al. The 4th reactive synthesis competition (syntcomp 2017): Benchmarks, participants & results. In *6th Workshop on Synthesis*, volume 260 of *EPTCS*, 2017.

[11] André Platzer. Differential dynamic logic for hybrid systems. *Journal of Automated Reasoning*, 41(2):143–189, 2008.

[12] André Platzer. A complete uniform substitution calculus for differential dynamic logic. *Journal of Automated Reasoning*, 59(2):219–265, 2017.

[13] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. Keymaera x: An axiomatic tactical theorem prover for hybrid systems. In *International Conference on Automated Deduction*, pages 527–538. Springer, 2015.