From Cold Start to Critical:
Formal Synthesis of Autonomous Hybrid Controllers

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

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

Track: PhD Mechanical Engineering

Monday 23rd February, 2026

The goal of this research is to develop a methodology for creating autonomous control systems with event-driven control laws that have guarantees of safe and correct behavior.

Nuclear power relies on extensively trained operators who follow detailed written procedures to manage reactor control. Based on these procedures and operators' interpretation of plant conditions, operators make critical decisions about when to switch between control objectives. But, reliance on human operators has created an economic challenge for next-generation nuclear power plants. Small modular reactors face significantly higher per-megawatt staffing costs than conventional plants. Autonomous control systems are needed that can safely manage complex operational sequences with the same assurance as human-operated systems, but without constant supervision.

To address this need, we will combine formal methods from computer science with control theory to build hybrid control systems that are correct by construction. Hybrid systems use discrete logic to switch between continuous control modes, similar to how operators change control strategies. Existing formal methods generate provably correct switching logic but cannot handle continuous dynamics during transitions, while traditional control theory verifies continuous behavior but lacks tools for proving discrete switching correctness. We will bridge this gap through a three-stage methodology. First, we will translate written operating procedures into temporal logic specifications using NASA's Formal Requirements Elicitation Tool (FRET), which structures requirements into scope, condition, component, timing, and response elements. This structured approach enables realizability checking to identify conflicts and ambiguities in procedures before implementation. Second, we will synthesize discrete mode switching logic using reactive synthesis to generate deterministic automata that are provably correct by construction. Third, we will develop continuous controllers for each discrete mode using standard control theory and reachability analysis. We will classify continuous modes based on their transition objectives, and then employ assume-guarantee contracts and barrier certificates to prove that mode transitions occur safely and as defined by the deterministic automata. This compositional approach enables local verification of continuous modes without requiring global trajectory analysis across the entire hybrid system. We will demonstrate this on an Emerson Ovation control system. This approach will demonstrate autonomous control can be used for complex nuclear power operations while maintaining safety guarantees.

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

1. *Synthesize written procedures into verified control logic.* We will develop a methodology for converting written operating procedures into formal specifications. These specifications will be synthesized into discrete control logic using reactive synthesis tools. Control engineers will be able to generate mode-switching controllers from regulatory procedures with little formal methods expertise, reducing barriers to high-assurance control systems.
2. *Verify continuous control behavior across mode transitions.* We will develop methods using reachability analysis to ensure continuous control modes satisfy discrete transition requirements. Engineers will be able to design continuous controllers using standard practices while ensuring system correctness and proving mode transitions occur safely at the right times.
3. *Demonstrate autonomous reactor startup control with safety guarantees.* We will implement this methodology on a small modular reactor simulation using industry-standard control hardware. Control engineers will be able to implement high-assurance autonomous controls on industrial platforms they already use, enabling users to achieve autonomy without retraining costs or developing new equipment.

Contents

1 Goals and Outcomes

The goal of this research is to develop a methodology for creating autonomous hybrid control systems with mathematical guarantees of safe and correct behavior.

Nuclear power plants require the highest levels of control system reliability, where failures can result in significant economic losses, service interruptions, or radiological release. Currently, nuclear plant operations rely on extensively trained human operators who follow detailed written procedures and strict regulatory requirements to manage reactor control. These operators make critical decisions about when to switch between different control modes based on their interpretation of plant conditions and procedural guidance. This reliance on human operators prevents autonomous control capabilities and creates a fundamental economic challenge for next-generation reactor designs. Small modular reactors, in particular, face per-megawatt staffing costs far exceeding those of conventional plants and threaten their economic viability.

What is needed is a method to create autonomous control systems that safely manage complex operational sequences with the same assurance as human-operated systems, but without constant human supervision. To address this need, we will combine formal methods with control theory to build hybrid control systems that are correct by construction. Hybrid systems use discrete logic to switch between continuous control modes, mirroring how operators change control strategies. Existing formal methods can generate provably correct switching logic from written requirements, but they cannot handle the continuous dynamics that occur during transitions between modes. Meanwhile, traditional control theory can verify continuous behavior but lacks tools for proving correctness of discrete switching decisions. By synthesizing discrete mode transitions directly from written operating procedures and verifying continuous behavior between transitions, we can create hybrid control systems with end-to-end correctness guarantees. If existing procedures can be formalized into logical specifications and continuous dynamics verified against transition requirements, then autonomous controllers can be built that are provably free from design defects. This approach will enable autonomous control in nuclear power plants while maintaining the high safety standards required by the industry.

This work is conducted within the University of Pittsburgh Cyber Energy Center, which provides access to industry collaboration and Emerson control hardware, ensuring that developed solutions align with practical implementation requirements.

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

1. **Translate written procedures into verified control logic.** We will develop a methodology for converting existing written operating procedures into formal specifications that can be automatically synthesized into discrete control logic. This process will use structured intermediate representations to bridge natural language procedures and mathematical logic. Control system engineers will generate verified mode-switching controllers directly from regulatory procedures without formal methods expertise, lowering the barrier to high-assurance control systems.
2. **Verify continuous control behavior across mode transitions.** We will establish methods for analyzing continuous control modes to ensure they satisfy discrete transition requirements. Using classical control theory for linear systems and reachability analysis for nonlinear dynamics, we will verify that each continuous mode safely reaches its intended transitions. Engineers will design continuous controllers using standard practices while iterating

to ensure broader system correctness, proving that mode transitions occur safely and at the correct times.

3. **Demonstrate autonomous reactor startup control with safety guarantees.** We will apply this methodology to develop an autonomous controller for nuclear reactor startup procedures, implementing it on a small modular reactor simulation using industry-standard control hardware. This demonstration will prove correctness across multiple coordinated control modes from cold shutdown through criticality to power operation. We will demonstrate that autonomous hybrid control can be realized in the nuclear industry with current equipment, establishing a path toward reduced operator staffing while maintaining safety.

The innovation in this work is unifying discrete synthesis with continuous verification to enable end-to-end correctness guarantees for hybrid systems. If successful, control engineers will create autonomous controllers from existing procedures with mathematical proof of correct behavior. High-assurance autonomous control will become practical for safety-critical applications. This capability is essential for the economic viability of next-generation nuclear power. Small modular reactors offer a promising solution to growing energy demands, but their success depends on reducing per-megawatt operating costs through increased autonomy. This research will provide the tools to achieve that autonomy while maintaining the exceptional safety record the nuclear industry requires.

2 State of the Art and Limits of Current Practice

The principal aim of this research is to create autonomous reactor control systems that are tractably safe. To understand what is being automated, we must first understand how nuclear reactors are operated today. This section examines reactor operators and the operating procedures we aim to leverage, then investigates limitations of human-based operation, and concludes with current formal methods approaches to reactor control systems.

2.1 Current Reactor Procedures and Operation

Nuclear plant procedures exist in a hierarchy: normal operating procedures for routine operations, abnormal operating procedures for off-normal conditions, Emergency Operating Procedures (EOPs) for design-basis accidents, Severe Accident Management Guidelines (SAMGs) for beyond-design-basis events, and Extensive Damage Mitigation Guidelines (EDMGs) for catastrophic damage scenarios. These procedures must comply with 10 CFR 50.34(b)(6)(ii) and are developed using guidance from NUREG-0900 [1, 2], but their development relies fundamentally on expert judgment and simulator validation rather than formal verification. Procedures undergo technical evaluation, simulator validation testing, and biennial review as part of operator requalification under 10 CFR 55.59 [3]. Despite this rigor, procedures fundamentally lack formal verification of key safety properties. No mathematical proof exists that procedures cover all possible plant states, that required actions can be completed within available timeframes, or that transitions between procedure sets maintain safety invariants.

LIMITATION: *Procedures lack formal verification of correctness and completeness.* Current procedure development relies on expert judgment and simulator validation. No mathematical proof exists that procedures cover all possible plant states, that required actions can be completed within available timeframes, or that transitions between procedure sets maintain safety invariants. Paper-based procedures cannot ensure correct application, and even computer-based procedure systems lack the formal guarantees that automated reasoning could provide.

Nuclear plants operate with multiple control modes: automatic control, where the reactor control system maintains target parameters through continuous reactivity adjustment; manual control, where operators directly manipulate the reactor; and various intermediate modes. In typical pressurized water reactor operation, the reactor control system automatically maintains a floating average temperature and compensates for power demand changes through reactivity feedback loops alone. Safety systems, by contrast, operate with implemented automation. Reactor Protection Systems trip automatically on safety signals with millisecond response times, and engineered safety features actuate automatically on accident signals without operator action required.

The division between automated and human-controlled functions reveals the fundamental challenge of hybrid control. Highly automated systems handle reactor protection—automatic trips on safety parameters, emergency core cooling actuation, containment isolation, and basic process control [4, 5]. Human operators, however, retain control of strategic decision-making: power level changes, startup/shutdown sequences, mode transitions, and procedure implementation.

2.2 Human Factors in Nuclear Accidents

Current-generation nuclear power plants employ over 3,600 active NRC-licensed reactor operators in the United States [6]. These operators divide into Reactor Operators (ROs), who manipulate reactor controls, and Senior Reactor Operators (SROs), who direct plant operations and serve as shift supervisors [7]. Staffing typically requires at least two ROs and one SRO for current-generation units [8]. Becoming a reactor operator requires several years of training.

The persistent role of human error in nuclear safety incidents—despite decades of improvements in training and procedures—provides the most compelling motivation for formal automated control with mathematical safety guarantees. Operators hold legal authority under 10 CFR Part 55 to make critical decisions, including departing from normal regulations during emergencies. The Three Mile Island (TMI) accident demonstrated how a combination of personnel error, design deficiencies, and component failures led to partial meltdown when operators misread confusing and contradictory readings and shut off the emergency water system [9]. The President's Commission on TMI identified a fundamental ambiguity: placing responsibility for safe power plant operations on the licensee without formal verification that operators can fulfill this responsibility does not guarantee safety. This tension between operational flexibility and safety assurance remains unresolved: the person responsible for reactor safety is often the root cause of failures.

Multiple independent analyses converge on a striking statistic: 70–80% of nuclear power plant events are attributed to human error, versus approximately 20% to equipment failures [10]. More significantly, the root cause of all severe accidents at nuclear power plants—Three Mile Island, Chernobyl, and Fukushima Daiichi—has been identified as poor safety management and safety culture: primarily human factors [11]. A detailed analysis of 190 events at Chinese nuclear power plants from 2007–2020 [12] found that 53% of events involved active errors, while 92% were associated with latent errors—organizational and systemic weaknesses that create conditions for failure.

LIMITATION: *Human factors impose fundamental reliability limits that cannot be overcome through training alone.* The persistent human error contribution despite four decades of improvements demonstrates that these limitations are fundamental rather than a remediable part of human-driven control.

2.3 HARDENS and Formal Methods

The High Assurance Rigorous Digital Engineering for Nuclear Safety (HARDENS) project represents the most advanced application of formal methods to nuclear reactor control systems to date [13].

HARDENS aimed to address a fundamental dilemma: existing U.S. nuclear control rooms rely on analog technologies from the 1950s–60s. This technology is obsolete compared to modern control systems and incurs significant risk and cost. The NRC contracted Galois, a formal methods firm, to demonstrate that Model-Based Systems Engineering and formal methods could design, verify, and implement a complex protection system meeting regulatory criteria at a fraction of typical cost. The project delivered a Reactor Trip System (RTS) implementation with full traceability from NRC Request for Proposals and IEEE standards through formal architecture specifications to verified software.

HARDENS employed formal methods tools and techniques across the verification hierarchy. High-level specifications used Lando, SysMLv2, and FRET (NASA Formal Requirements Elicitation Tool) to capture stakeholder requirements, domain engineering, certification requirements, and safety requirements. Requirements were analyzed for consistency, completeness, and realizability using SAT and SMT solvers. Executable formal models used Cryptol to create a behavioral model of the entire RTS, including all subsystems, components, and limited digital twin models of sensors, actuators, and compute infrastructure. Automatic code synthesis generated verifiable C implementations and SystemVerilog hardware implementations directly from Cryptol models—

eliminating the traditional gap between specification and implementation where errors commonly arise.

Despite its accomplishments, HARDENS has a fundamental limitation directly relevant to hybrid control synthesis: the project addressed only discrete digital control logic without modeling or verifying continuous reactor dynamics. The Reactor Trip System specification and verification covered discrete state transitions (trip/no-trip decisions), digital sensor input processing through discrete logic, and discrete actuation outputs (reactor trip commands). The project did not address continuous dynamics of nuclear reactor physics. Real reactor safety depends on the interaction between continuous processes—temperature, pressure, neutron flux—evolving in response to discrete control decisions. HARDENS verified the discrete controller in isolation but not the closed-loop hybrid system behavior.

LIMITATION: *HARDENS addressed discrete control logic without continuous dynamics or hybrid system verification.* Verifying discrete control logic alone provides no guarantee that the closed-loop system exhibits desired continuous behavior such as stability, convergence to setpoints, or maintained safety margins.

HARDENS produced a demonstrator system at Technology Readiness Level 2–3 (analytical proof of concept with laboratory breadboard validation) rather than a deployment-ready system validated through extended operational testing. The NRC Final Report explicitly notes [13] that all material is considered in development, not a finalized product, and that "The demonstration of its technical soundness was to be at a level consistent with satisfaction of the current regulatory criteria, although with no explicit demonstration of how regulatory requirements are met." The project did not include deployment in actual nuclear facilities, testing with real reactor systems under operational conditions, side-by-side validation with operational analog RTS systems, systematic failure mode testing (radiation effects, electromagnetic interference, temperature extremes), NRC licensing review, or human factors validation with licensed operators in realistic control room scenarios.

LIMITATION: *HARDENS achieved TRL 2–3 without experimental validation.* While formal verification provides mathematical correctness guarantees for the implemented discrete logic, the gap between formal verification and actual system deployment involves myriad practical considerations: integration with legacy systems, long-term reliability under harsh environments, human-system interaction in realistic operational contexts, and regulatory acceptance of formal methods as primary assurance evidence.

3 Research Approach

Autonomous control systems are fundamentally different from automatic control systems. The difference between these systems is the level at which they operate. Automatic control systems are purely operational systems,

To build a high-assurance hybrid autonomous control system (HAHACS), a mathematical description of the system must be established. This work will make use of automata theory while including logical statements and control theory. The nomenclature and lexicon between these fields is far from homogenous, and the reviewer of this proposal is not expected to be an expert in all fields simultaneously. To present the research ideas as clearly as possible in this section, the following syntax is explained.

A hybrid system is a dynamical system that has both continuous and discrete states. The specific type of system discussed in this proposal are continuous autonomous hybrid systems. This means that these systems a) do not have external input [1] and b) continuous states do not change instantaneously when discrete states change. For our systems of interest, the continuous states are physical, and are always Lipschitz continuous. This nomenclature is heavily borrowed from [**?**], but is redefined here for convenience:

$$H = (\mathscr{Q}, \mathscr{X}, \mathbf{f}, Init, \mathscr{G}, \mathscr{R}, Inv) \tag{1}$$

where:

- $\mathscr{Q}$: is the discrete states of the system
- $\mathscr{X}$: is the continuous states of the system
- $\mathbf{f} : \mathscr{Q} \times \mathbb{R} \to \mathbb{R}$, where $\mathbf{f}_i$ is a vector field that defines the continuous dynamics for each q_i
- *Init*: the initial states of q and x
- G: guard conditions that define when discrete state transitions occur
- $\delta : \mathscr{Q} \times G \to \mathscr{Q}$, are the discrete state transition functions
- R: Reset maps that define state 'jumps'
- *Inv*: Safety invariants on the continuous dynamics

The creation of a HAHACS essentially boils down to the creation of such a tuple where there are proof artifacts that the intended behavior of the control system are satisfied by the actual implementation of the control systems. But to create such a HAHACS, we must first completely describe its behavior.

3.1 System Requirement and Specifications

Temporal logic is a powerful set of semantics to build systems that can have complex but deterministic behavior.

Human control of nuclear power can be divided into three different scopes: strategic, operational, and tactical. Strategic control is the high-level and long term decision making for the plant. This level has objectives that are complex and economic in scale, such as managing labor needs and supply chains to optimize sheduled maintenence and downtime. The time scale on this level

[1] This is not strictly true in our case because we allow strategic inputs. For example, a remote powerplant may receive a start-up or shutdown command from a different location, but only this binary high level input is a strategic input.

Strategic, operational, tactical placeholder

Figure 1: Breakdown of control scope

of control is long, often over months or years. The lowest level of control is the tactical level. This is the individual control of pumps, turbines, and chemistry of the plant. This level of control has already been somewhat automated today in nuclear power, and is generally considered ’automatic control’ when autonomous. These controls are almost always continuous systems, and have a direct impact on the physical state of the plant. Tactical control objectives are things like maintaining a pressurizer level, maintaining a certain core temperature, or adjusting reactivity with a chemical shim. The level of control linking these two levels, then, is the operational control scope. Operational control is the primary responsibility of human operators today. Operational control takes the current strategic objective, and implements tactical control objectives to drive the plant towards strategic goals. In this way, it is the bridge between high and low level goals. A strategic goal may be to perform refueling at a certain time, while the tactical level of the plant currently is focused on mainting a certain core temperature. The operational level is what issues the shutdown procedure of the plant, using several smaller tactical goals along the way to achieve this objective. Thus, the combination of the operational and tactical level of the plant fundamentally forms a hybrid controller. The tactical level is the continuous evolution of the plant according to the control input and control law, while the operational level is a discrete state evolution which determines the tactical control law to reach different operational states.

This operational control level is the main reason for the requirement of human opeartors in nuclear control today. The hybrid nature of this control system makes it difficult to prove that a controller will perform according to the strategic requirements, as the infrastructure to build hybrid systems today dooes not exist. Humans have been used for this layer because the general intelleigence of humans has be relied upon as a safe way to manage the hybrid nature of this system. But, these operators are using prescriptive operating manuals to perform their control with strict procedures on what control to implement at a given time. These procedures are the key to the operational control scope.

The method of constructing a HAHACS in this proposal leverages two key points of the way this control scope is done today: first, the operational scope control is effectively discrete control. Second, the rules of implementing this control are described a priori to their implementation in operating procedures. We can make great use of these facts by formalizing the rules for transitioning between discrete states. To do this, we will use temporal logic to formalize discrete switching behavior.

Temporal logic is a rich syntax that allows for the definition of logical calculations including time related bounds. For this reason, we can make statements relating discrete control modes to one another. Using temporal logic, we can effectively describe all of the requirements of a

HAHACS. The guard conditions G are easily defined by determining boundary conditions between discrete states and defining their behavior, while continuous mode invariants can be defined using temporal logic statements as well. These form the basis of any proofs about a HAHACS, and are the fundamental 'truth' statements about what the behavior of the system is designed to be.

To build these temporal logic statements, an intermediary tool called FRET is planned to be used. FRET stands for Formal Requirements Elicitation Tool, and was designed by NASA to build high assurance timed systems. FRET is an intermediarly language between temporal logic and natural language that allows for rigid definitions of temporal behvarior while using a logic-novice friendly syntax. This benefit is crucial for the feasibility of this methodology for industry, as minimizing the barrier to formal methods is a critical component of their scucess. By reducing the expert knowledge required to use these tools, their adoption with current workforce becomes easier.

A key feature of FRET is the ability to start with logically imprecise statements and consecutively refine them into a well-posited specification. We can use this to our advantage by directly dumping in operating procedures and design requirements into FRET in natural language, and iteratively refining them into the specifications for a HAHACS. This has two distinct but important benefits. First, it allows us to draw a direct link from the design documentation to the digital system implementation. Second, it clearly demonstrates where the natural language documents are insufficient. These procedures may still be used by human operators, so any wiggle room for interpretation is a weakness that must be addressed.

Once system requirements we defined as temporal logic specifications we will use the specifications to build the discrete control system. To do this, reactive synthesis tools will be utilized. Reactive synthesis is a field in computer science that deals with the automated creation of reactive programs from temporal logic specifications. A reactive program is one that for a given state takes an input, and produces an output. Our systems, such as the discrete portion of the controller, fit exactly this this mold. The current discrete state, and status of guard conditions are the input to the system, while the output is the next discrete state. The output of a reactive synthesis algorithim is a discrete automata.

Reactive synthesis' main advantage is the fact that at no point in the production of a discrete automata of the program is human engineering required. The resultant automata is correct by construction. This method of construction eliminates the possibility of human error outright at the implementation state. Instead, the effort on the human designer is directed at the specification of the system behavior itself.

CONTINUOUS CONTROLLER TRANSITIONS VS DISCRETE GUARD CONDITIONS

3.2 Continuous Controllers

The synthesis of the discrete operational controller is only half of an autonomous controller. These control systems are hybrid, with both discrete and continuous components. In this section, we will talk about the continuous control modes that are the transitions between discrete modes, how they may be synthesized, and how we plan to verify them.

The operational control scope defines go/no-go decisions that themselves are deciding what kind of continuous control to implement. To this end, the entry or exit of a discrete state triggers are themselves the guard conditions G that define the barriers of the continuous controller. These continuous controllers all share a large state space, but each individual continuous control mode operates within it's own partition defined by the discrete state q_i and guard conditions G. This partitioning of the continuous state space amongst several discrete vector fields controlled by the given q_i has traditionally been a difficult problem for validation and verification of systems properties. Typically, the discontinuity of the vector fields at discrete state interfaces make things like reachability analysis computationally expensive, and analytic solutions become intractable.

We circumnavigate these issues by designing our hybrid system from the bottom up with this verification in mind. Each continuous control mode has an input and output set clearly defined by our discrete transitions *a priori*. Consider that we define the continuous state space as X. Whenever we create guard functions from our design requirements for a given system, we are effectively creating subsets $X_{entry,i}$ and $X_{exit,i}$ for each discrete mode q_i. These subsets define when the state transitions occur between discrete modes, but more importantly when building continuous control modes, they become control objectives.

4 Metrics for Success

This research will be measured by advancement through Technology Readiness Levels, progressing from fundamental concepts to validated prototype demonstration. This work begins at TRL 2–3 and aims to reach TRL 5, where system components operate successfully in a relevant laboratory environment. This section explains why TRL advancement provides the most appropriate success metric and defines the specific criteria required to achieve TRL 5.

Technology Readiness Levels provide the ideal success metric because they explicitly measure the gap between academic proof-of-concept and practical deployment—precisely what this work aims to bridge. Academic metrics like papers published or theorems proved cannot capture practical feasibility. Empirical metrics like simulation accuracy or computational speed cannot demonstrate theoretical rigor. TRLs measure both dimensions simultaneously. Advancing from TRL 3 to TRL 5 requires maintaining theoretical rigor while progressively demonstrating practical feasibility. Formal verification must remain valid as the system moves from individual components to integrated hardware testing.

The nuclear industry requires extremely high assurance before deploying new control technologies. Demonstrating theoretical correctness alone is insufficient for adoption; conversely, showing empirical performance without formal guarantees fails to meet regulatory requirements. TRLs capture this dual requirement naturally. Each level represents both increased practical maturity and sustained theoretical validity. Furthermore, TRL assessment forces explicit identification of remaining barriers to deployment. The nuclear industry already uses TRLs for technology assessment, making this metric directly relevant to potential adopters. Reaching TRL 5 provides a clear answer to industry questions about feasibility and maturity that academic publications alone cannot.

Moving from current state to target requires achieving three intermediate levels, each representing a distinct validation milestone:

TRL 3 ***Critical Function and Proof of Concept*** For this research, TRL 3 means demonstrating that each component of the methodology works in isolation. Startup procedures must be translated into temporal logic specifications that pass realizability analysis. A discrete automaton must be synthesized with interpretable structure. At least one continuous controller must be designed with reachability analysis proving transition requirements are satisfied. Independent review must confirm that specifications match intended procedural behavior. This proves the fundamental approach on a simplified startup sequence.

TRL 4 ***Laboratory Testing of Integrated Components*** For this research, TRL 4 means demonstrating a complete integrated hybrid controller in simulation. All startup procedures must be formalized with a synthesized automaton covering all operational modes. Continuous controllers must exist for all discrete modes. Verification must be complete for all mode transitions using reachability analysis, barrier certificates, and assume-guarantee contracts. The integrated controller must execute complete startup sequences in software simulation with zero safety violations across multiple consecutive runs. This proves that formal correctness guarantees can be maintained throughout system integration.

TRL 5 ***Laboratory Testing in Relevant Environment*** For this research, TRL 5 means demonstrating the verified controller on industrial control hardware through hardware-in-the-loop testing. The discrete automaton must be implemented on the Emerson Ovation control system and verified to match synthesized specifications exactly. Continuous controllers must execute at required rates. The ARCADE interface must establish stable real-time communication between the Emerson Ovation hardware and SmAHTR simulation. Complete autonomous startup sequences must execute via hardware-in-the-loop across the full operational envelope. The controller must handle off-nominal scenarios to validate that expulsory modes function correctly. For example, simulated sensor failures must trigger appropriate fault detection and mode transitions, and loss-of-cooling scenarios must activate SCRAM procedures as specified. Graded responses to minor disturbances are outside this work's scope. Formal verification results must remain valid, with discrete behavior matching specifications and continuous trajectories remaining within verified bounds. This proves that the methodology produces verified controllers implementable on industrial hardware.

Progress will be assessed quarterly through collection of specific data comparing actual results against TRL advancement criteria. Specification development status indicates progress toward TRL 3. Synthesis results and verification coverage indicate progress toward TRL 4. Simulation performance metrics and hardware integration milestones indicate progress toward TRL 5. The research plan will be revised only when new data invalidates fundamental assumptions. This research succeeds if it achieves TRL 5 by demonstrating a complete autonomous hybrid controller with formal correctness guarantees operating on industrial control hardware through hardware-in-the-loop testing in a relevant laboratory environment. This establishes both theoretical validity and practical feasibility, proving that the methodology produces verified controllers and that implementation is achievable with current technology.

5 Risks and Contingencies

This research relies on several critical assumptions that, if invalidated, would require scope adjustment or methodological revision. The primary risks to successful completion fall into four categories: computational tractability of synthesis and verification, complexity of the discrete-continuous interface, completeness of procedure formalization, and hardware-in-the-loop integration challenges. Each risk has associated indicators for early detection and contingency plans that preserve research value even if core assumptions prove false. The staged project structure ensures that partial success yields publishable results and clear identification of remaining barriers to deployment.

5.1 Computational Tractability of Synthesis

The first major assumption is that formalized startup procedures will yield automata small enough for efficient synthesis and verification. Reactive synthesis scales exponentially with specification complexity, creating risk that temporal logic specifications derived from complete startup procedures may produce automata with thousands of states. Such large automata would require synthesis times exceeding days or weeks, preventing demonstration of the complete methodology within project timelines. Reachability analysis for continuous modes with high-dimensional state spaces may similarly prove computationally intractable. Either barrier would constitute a fundamental obstacle to achieving the research objectives.

Several indicators would provide early warning of computational tractability problems. Synthesis times exceeding 24 hours for simplified procedure subsets would suggest complete procedures are intractable. Generated automata containing more than 1,000 discrete states would indicate the discrete state space is too large for efficient verification. Specifications flagged as unrealizable by FRET or Strix would reveal fundamental conflicts in the formalized procedures. Reachability analysis failing to converge within reasonable time bounds would show that continuous mode verification cannot be completed with available computational resources.

The contingency plan for computational intractability is to reduce scope to a minimal viable startup sequence. This reduced sequence would cover only cold shutdown to criticality to low-power hold, omitting power ascension and other operational phases. The subset would still demonstrate the complete methodology while reducing computational burden. The research contribution would remain valid even with reduced scope, proving that formal hybrid control synthesis is achievable for safety-critical nuclear applications. The limitation to simplified operational sequences would be explicitly documented as a constraint rather than a failure.

5.2 Discrete-Continuous Interface Formalization

The second critical assumption concerns the mapping between boolean guard conditions in temporal logic and continuous state boundaries required for mode transitions. This interface represents the fundamental challenge of hybrid systems: relating discrete switching logic to continuous dynamics. Temporal logic operates on boolean predicates, while continuous control requires reasoning about differential equations and reachable sets. Guard conditions requiring complex nonlinear predicates may resist boolean abstraction, making synthesis intractable. Continuous safety regions that cannot be expressed as conjunctions of verifiable constraints would similarly create insurmountable verification challenges. The risk extends beyond static interface definition to dynamic behavior across transitions: barrier certificates may fail to exist for proposed transitions, or continuous modes may be unable to guarantee convergence to discrete transition boundaries.

Early indicators of interface formalization problems would appear during both synthesis and verification phases. Guard conditions requiring complex nonlinear predicates that resist boolean abstraction would suggest fundamental misalignment between discrete specifications and continuous realities. Continuous safety regions that cannot be expressed as conjunctions of half-spaces or polynomial inequalities would indicate the interface between discrete guards and continuous invariants is too complex. Failure to construct barrier certificates proving safety across mode transitions would reveal that continuous dynamics cannot be formally related to discrete switching logic. Reachability analysis showing that continuous modes cannot reach intended transition boundaries from all possible initial conditions would demonstrate the synthesized discrete controller is incompatible with achievable continuous behavior.

The primary contingency for interface complexity is restricting continuous modes to operate within polytopic invariants. Polytopes are state regions defined as intersections of linear half-spaces, which map directly to boolean predicates through linear inequality checks. This restriction ensures tractable synthesis while maintaining theoretical rigor, though at the cost of limiting expressiveness compared to arbitrary nonlinear regions. The discrete-continuous interface remains well-defined and verifiable with polytopic restrictions, providing a clear fallback position that preserves the core methodology. Conservative over-approximations offer an alternative approach: a nonlinear safe region can be inner-approximated by a polytope, sacrificing operational flexibility to maintain formal guarantees. The three-mode classification already structures the problem to minimize complex transitions, with critical safety properties concentrated in expulsory modes that can receive additional design attention.

Mitigation strategies focus on designing continuous controllers with discrete transitions as primary objectives from the outset. Rather than designing continuous control laws independently and verifying transitions post-hoc, the approach uses transition requirements as design constraints. Control barrier functions provide a systematic method to synthesize controllers that guarantee forward invariance of safe sets and convergence to transition boundaries. This design-for-verification approach reduces the likelihood that interface complexity becomes insurmountable. Focusing verification effort on expulsory modes—where safety is most critical—allows more complex analysis to be applied selectively rather than uniformly across all modes, concentrating computational resources where they matter most for safety assurance.

5.3 Procedure Formalization Completeness

The third assumption is that existing startup procedures contain sufficient detail and clarity for translation into temporal logic specifications. Nuclear operating procedures, while extensively detailed, were written for human operators who bring contextual understanding and adaptive reasoning to their interpretation. Procedures may contain implicit knowledge, ambiguous directives, or references to operator judgment that resist formalization in current specification languages. Underspecified timing constraints, ambiguous condition definitions, or gaps in operational coverage would cause synthesis to fail or produce incorrect automata. The risk is not merely that formalization is difficult, but that current procedures fundamentally lack the precision required for autonomous control, revealing a gap between human-oriented documentation and machine-executable specifications.

Several indicators would reveal formalization completeness problems early in the project. FRET realizability checks failing due to underspecified behaviors or conflicting requirements would indicate procedures do not form a complete specification. Multiple valid interpretations of procedural steps with no clear resolution would demonstrate procedure language is insufficiently precise for automated synthesis. Procedures referencing “operator judgment,” “as appropriate,” or similar discretionary language for critical decisions would explicitly identify points where human reasoning cannot be directly formalized. Domain experts unable to provide crisp answers to specification questions about edge cases would suggest the procedures themselves do not fully define system behavior, relying instead on operator training and experience to fill gaps.

The contingency plan treats inadequate specification as itself a research contribution rather than a project failure. Documenting specific ambiguities encountered would create a taxonomy of formalization barriers: timing underspecification, missing preconditions, discretionary actions, and undefined failure modes. Each category would be analyzed to understand why current procedure-writing practices produce these gaps and what specification languages would need to address them. Proposed extensions to FRETish or similar specification languages would demonstrate how to bridge the gap between current procedures and the precision needed for autonomous control. The research output would shift from “here is a complete autonomous controller” to “here is what formal autonomous control requires that current procedures do not provide, and here are language extensions to bridge that gap.” This contribution remains valuable to both the nuclear industry and formal methods community, establishing clear requirements for next-generation procedure development and autonomous control specification languages.

Early-stage procedure analysis with domain experts provides the primary mitigation strategy. Collaboration through the University of Pittsburgh Cyber Energy Center enables identification and resolution of ambiguities before synthesis attempts, rather than discovering them during failed synthesis runs. Iterative refinement with reactor operators and control engineers can clarify procedural intent before formalization begins, reducing the risk of discovering insurmountable specification gaps late in the project. Comparison with procedures from multiple reactor designs—pressurized water reactors, boiling water reactors, and advanced designs—may reveal common patterns and standard ambiguities amenable to systematic resolution. This cross-design analysis would strengthen the generalizability of any proposed specification language extensions, ensuring they address industry-wide practices rather than specific quirks.

6 Broader Impacts

Nuclear power presents both a compelling application domain and an urgent economic challenge. Recent interest in powering artificial intelligence infrastructure has renewed focus on small modular reactors (SMRs), particularly for hyperscale datacenters requiring hundreds of megawatts of continuous power. Deploying SMRs at datacenter sites would minimize transmission losses and eliminate emissions from hydrocarbon-based alternatives. However, nuclear power economics at this scale demand careful attention to operating costs.

According to the U.S. Energy Information Administration's Annual Energy Outlook 2022, advanced nuclear power entering service in 2027 is projected to cost \$88.24 per megawatt-hour [14]. Datacenter electricity demand is projected to reach 1,050 terawatt-hours annually by 2030 [15]. If this demand were supplied by nuclear power, the total annual cost of power generation would exceed \$92 billion. Within this figure, operations and maintenance represents a substantial component. The EIA estimates that fixed O&M costs alone account for \$16.15 per megawatt-hour, with additional variable O&M costs embedded in fuel and operating expenses [14]. Combined, O&M-related costs represent approximately 23–30% of the total levelized cost of electricity, translating to \$21–28 billion annually for projected datacenter demand.

This research directly addresses the multi-billion-dollar O&M cost challenge through high-assurance autonomous control. Current nuclear operations require full control room staffing for each reactor, whether large conventional units or small modular designs. These staffing requirements drive the high O&M costs that make nuclear power economically challenging, particularly for smaller reactor designs where the same staffing overhead must be spread across lower power output. Synthesizing provably correct hybrid controllers from formal specifications can automate routine operational sequences that currently require constant human oversight. This enables a fundamental shift from direct operator control to supervisory monitoring, where operators oversee multiple autonomous reactors rather than manually controlling individual units.

The correct-by-construction methodology is critical for this transition. Traditional automation approaches cannot provide sufficient safety guarantees for nuclear applications, where regulatory requirements and public safety concerns demand the highest levels of assurance. Formally verifying both the discrete mode-switching logic and the continuous control behavior, this research will produce controllers with mathematical proofs of correctness. These guarantees enable automation to safely handle routine operations—startup sequences, power level changes, and normal operational transitions—that currently require human operators to follow written procedures. Operators will remain in supervisory roles to handle off-normal conditions and provide authorization for major operational changes, but the routine cognitive burden of procedure execution shifts to provably correct automated systems that are much cheaper to operate.

SMRs represent an ideal deployment target for this technology. Nuclear Regulatory Commission certification requires extensive documentation of control procedures, operational requirements, and safety analyses written in structured natural language. As described in our approach, these regulatory documents can be translated into temporal logic specifications using tools like FRET, then synthesized into discrete switching logic using reactive synthesis tools, and finally verified using reachability analysis and barrier certificates for the continuous control modes. The infrastructure of requirements and specifications already exists as part of the licensing process, creating a direct pathway from existing regulatory documentation to formally verified autonomous controllers.

Beyond reducing operating costs for new reactors, this research will establish a generalizable framework for autonomous control of safety-critical systems. The methodology of translating operational procedures into formal specifications, synthesizing discrete switching logic, and verifying continuous mode behavior applies to any hybrid system with documented operational requirements. Potential applications include chemical process control, aerospace systems, and autonomous transportation, where similar economic and safety considerations favor increased autonomy with provable correctness guarantees. Demonstrating this approach in nuclear power—one of the most regulated and safety-critical domains—will establish both the technical feasibility and regulatory pathway for broader adoption across critical infrastructure.

7 Schedule, Milestones, and Deliverables

This research will be conducted over six trimesters (24 months) of full-time effort following the proposal defense in Spring 2026. The work progresses sequentially through three main research thrusts before culminating in integrated demonstration and validation.

The first semester (Spring 2026) focuses on Thrust 1, translating startup procedures into formal temporal logic specifications using FRET. This establishes the foundation for automated synthesis by converting natural language procedures into machine-readable requirements. The second semester (Summer 2026) addresses Thrust 2, using Strix to synthesize the discrete automaton that defines mode-switching behavior. With the discrete structure established, the third semester (Fall 2026) develops the continuous controllers for each operational mode through Thrust 3, employing reachability analysis and barrier certificates to verify that each mode satisfies its transition requirements. Integration and validation occupy the remaining three semesters.

Figure 2 shows the complete project schedule including research thrusts, major milestones, and planned publications.

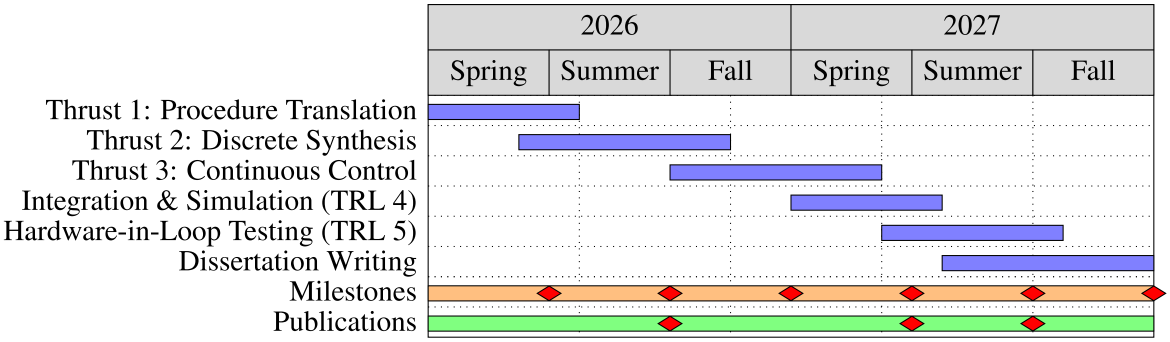

Figure 2: Project schedule showing major research thrusts, milestones (orange row), and publications (green row). Red diamonds indicate completion points. Overlapping bars indicate parallel work where appropriate.

7.1 Milestones and Deliverables

Six major milestones mark critical validation points throughout the research. M1 (Month 4) confirms that startup procedures have been successfully translated to temporal logic using FRET with realizability analysis demonstrating consistent and complete specifications. M2 (Month 8) validates computational tractability by demonstrating that Strix can synthesize a complete discrete automaton from the formalized specifications. This milestone delivers a conference paper submission to NPIC&HMIT documenting the procedure-to-specification translation methodology. M3 (Month 12) achieves TRL 3 by proving that continuous controllers can be designed and verified to satisfy discrete transition requirements. This milestone delivers an internal technical report demonstrating component-level verification. M4 (Month 16) achieves TRL 4 through integrated simulation demonstrating that component-level correctness composes to system-level correctness. This milestone delivers a journal paper submission to IEEE Transactions on Automatic Control presenting the complete hybrid synthesis methodology. M5 (Month 20) achieves TRL 5 by demonstrating practical implementability on industrial hardware. This milestone delivers a conference paper submission to NPIC&HMIT or CDC documenting hardware implementation and experimental validation. M6 (Month 24) completes the dissertation documenting the entire methodology, experimental results, and research contributions.

References

[1] U.S. Nuclear Regulatory Commission, "Guidelines for the preparation of emergency operating procedures," Tech. Rep. NUREG-0899, U.S. Nuclear Regulatory Commission, 1982.

[2] U.S. Nuclear Regulatory Commission, "10 CFR Part 50.34." Code of Federal Regulations.

[3] U.S. Nuclear Regulatory Commission, "10 CFR Part 55.59." Code of Federal Regulations.

[4] "Westinghouse RPS System Description," tech. rep., Westinghouse Electric Corporation.

[5] C. D. Gentillon, D. Marksberry, D. Rasmuson, M. B. Calley, S. A. Eide, and T. Wierman, "Westinghouse reactor protection system unavailability, 1984-1995." Number: INEEL/CON-99-00374 Publisher: Idaho National Engineering and Environmental Laboratory.

[6] U.S. Nuclear Regulatory Commission, "Operator Licensing." `https://www.nrc.gov/reactors/operator-licensing`.

[7] U.S. Nuclear Regulatory Commission, "Part 55—Operators' Licenses." `https://www.nrc.gov/reading-rm/doc-collections/cfr/part055/full-text`.

[8] U.S. Nuclear Regulatory Commission, "§ 50.54 Conditions of Licenses." `https://www.nrc.gov/reading-rm/doc-collections/cfr/part050/part050-0054`.

[9] J. G. Kemeny *et al.*, "Report of the president's commission on the accident at three mile island," tech. rep., President's Commission on the Accident at Three Mile Island, October 1979.

[10] World Nuclear Association, "Safety of nuclear power reactors." `https://www.world-nuclear.org/information-library/safety-and-security/safety-of-plants/safety-of-nuclear-power-reactors.aspx`, 2020.

[11] L. Högberg, "Root causes and impacts of severe accidents at large nuclear power plants," vol. 42, no. 3, pp. 267–284.

[12] M. Zhang, L. Dai, W. Chen, and E. Pang, "Analysis of human errors in nuclear power plant event reports," vol. 57, no. 10, p. 103687.

[13] J. Kiniry, A. Bakst, S. Hansen, M. Podhradsky, and A. Bivin, "High assurance rigorous digital engineering for nuclear safety (hardens) final technical report," Tech. Rep. TLR-RES-RES/DE-2024-005, Galois, Inc. / U.S. Nuclear Regulatory Commission, 2024. NRC Contract 31310021C0014.

[14] U.S. Energy Information Administration, "Levelized costs of new generation resources in the annual energy outlook 2022," report, U.S. Energy Information Administration, March 2022. See Table 1b, page 9.

[15] Environmental and Energy Study Institute, “Data center energy needs are upending power grids and threatening the climate.” Web article, 2024. Accessed: 2025-09-29.