energies

MDPI

Review

Towards Trustworthy Autonomy in Advanced Nuclear Systems: A Review of Advanced Control, Digital Twins, and Generative AI

Yang Liu

Department of Nuclear Engineering, Texas A&M University, College Station, TX 77843, USA; y-liu@tamu.edu

Abstract

Advanced nuclear systems are expected to operate across a wider range of missions, plant configurations, staffing models, and external couplings than the existing light-water reactor fleet. The operating practices of that fleet were developed for large single units at steady baseload and cannot be assumed to transfer unchanged to flexible, multi-module, remotely supervised, or multi-product plants. This review examines how advanced control, digital twins, and generative artificial intelligence (GenAI) can extend those practices toward trustworthy autonomy. Model-based and predictive control, learning-enabled control, control-oriented digital twins, supervisory operation, and GenAI grounded in validated engineering tools are synthesized within a single framework organized by level of automation. The framework considers plant layer, degree of autonomy, and evidence maturity from conceptual proposals to operating-plant experience. The review finds that higher autonomy is most credible within a layered architecture. Independent protection and qualified regulatory control remain the safety foundation that every advanced layer must defer to. Predictive and supervisory control coordinate constrain plant-wide operation, and reinforcement learning is best confined to offline-trained supervisory decisions screened by verified filters. Digital twins supply state estimation, prediction, and controller-development infrastructure whose role narrows from action to advice as uncertainty grows. GenAI is best introduced as a grounded interface that retrieves controlled information, invokes validated tools, and prepares recommendations for human approval. The principal gap is integrated demonstration, and a staged roadmap in which the level of automation rises only with accumulated evidence is offered from the author's perspective.

Keywords: advanced reactor control; digital twin; generative artificial intelligence; large language model; human-supervised autonomy

Check for updates

Academic Editor: Vincenzo Paolo Loschiavo

Received: 7 September 2026
Revised: 9 October 2026
Accepted: 22 October 2026
Published: 2 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Data centers that train and serve artificial intelligence (AI) have added a large new electrical load that nuclear energy is well placed to meet. The need for firm, carbon-free electricity is one reason advanced nuclear energy systems are expected to do more than hold a single electrical setpoint. Flexible grid support, industrial process heat, hydrogen production, thermal storage, multi-module stations, and remotely supervised microreactors introduce coupled objectives and disturbances that a single-setpoint design never had to reconcile. These missions do not remove the need for proven protection and regulatory control; they add plant-wide decisions above those functions. The existing light-water reactor (LWR) fleet shows how well a control approach can perform when its assumptions hold.

Advanced nuclear systems change several of those assumptions at once. The relationship between nuclear energy and AI also runs in the other direction: advanced control, digital twins, and generative AI can help design and operate the plants that will meet that demand. How far such tools can be trusted with plant operation is the question examined here. The argument of this review is that trustworthy autonomy should be layered, bounded, and reversible: each advanced method should receive only the degree of autonomy that its models, interfaces, and validation evidence can support. The term follows the trustworthy-AI literature, which asks for robustness, safety, security, transparency, accountability, and human oversight [1]. Safety-critical industries also require evidence that learned components behave as intended [2], and a nuclear study has turned this requirement into deterministic acceptance gates [3].

1.1. Missions, Plant Classes, and Modernization

Large-LWR operating practice is the natural starting point because it is proven. It combines independent protection, qualified regulatory control loops, extensive procedures, trained licensed crews, and mature maintenance programs, all refined over decades of operation [4,5]. This combination was optimized under identifiable assumptions: a large unit size, a unit-centric control room, substantial onsite staffing, well-characterized transients, and a stable baseload mission. Within those assumptions, single-loop proportional–integral–derivative (PID) regulation with cascade structures, gain scheduling, and interlocks remains sufficient for most reactor loops. Difficulties appear when missions, staffing models, or plant configurations move outside the assumptions, as flexible and multi-product operation requires.

Advanced reactors differ from this baseline in their physics before they differ in their missions. Integral pressurized-water small modular reactors (SMRs) compress the primary circuit, which tightens coupling among the core, pressurizer, and steam generators. Sodium-cooled fast reactors combine strong temperature feedback with large pool thermal inertia, so slow thermal states dominate the response to power maneuvers. High-temperature gas-cooled reactors respond through massive graphite cores that buffer transients but slow the response to control actions. Molten-salt and fluoride-salt-cooled designs add coolant transport delays, temperature-dependent salt properties, and in some concepts moving fuel. Heat-pipe and other microreactor concepts concentrate reactivity control in rotating drums, which leaves few independent actuators and makes trajectory planning more consequential [6]. Classical, sliding-mode, and nonlinear predictive controllers have tracked ramps near ten percent of full power per minute on a drum-actuated microreactor model but only in simplified-model simulation with matched design and evaluation models [7].

The differences in physics translate into a different control task. Passive safety widens the tolerable envelope but changes the available recovery actions and the boundary between normal control and protection. High temperatures, chemically active coolants, and compact vessels restrict instrumentation placement, and first-of-a-kind uncertainty weakens the models on which advanced controllers depend. Limited actuator redundancy and strong cross-system coupling then raise the value of robust state estimation, constrained control, and a pre-defined safe response when a controller cannot continue. Beyond physics, advanced reactors can be designed from the outset for flexible, multi-module, remote, or multi-product service, which adds mission requirements on top of the new dynamics.

Integrated energy systems illustrate these differences. Analyses of integrated nuclear and renewable systems now treat grid support, process heat, hydrogen production, and thermal storage as coordinated products rather than independent add-ons [8,9]. Each product adds controlled outputs, manipulated variables such as storage charge flows and bypass valves, constraints on delivered temperatures and ramp rates, and economic objectives that compete with tracking accuracy. The shared challenge is coordinated management of

reactor power, thermal inventories, component stress, product demand, and equipment health. Faster power maneuvering alone does not solve it. Simulation studies of a fluoride-salt-cooled reactor coupled to molten-salt storage, with a supervisory governor enforcing secondary-side temperature limits and a decision layer switching among charging, discharging, and load-following modes, show the supervisory and mission layers such coordination requires [10,11].

Moving from this foundation to advanced reactors therefore requires changes in control and operating practice, and those changes should build on what the LWR fleet has already validated. Qualified control and protection systems embody decades of validation effort, and operating crews hold tacit plant knowledge that procedures and data records capture only in part, so the credible path is augmentation around the qualified core. Near-term candidates include improved load-following control and state estimation (Section 2), advisory digital twins that add state awareness without taking on any control function (Section 4), and language-based tools for procedure retrieval at the operator interface (Section 6). Each step generates the operating data needed to justify the next.

1.2. Coordinated Operation and Bounded Autonomy

Multi-module and remote deployment move the unit of control from one reactor to a network. Modules can share a control room, turbine, heat-rejection system, or industrial customer. Thus, module dispatch, staggered startups, and common-cause disturbances become control problems, and controllers tuned in isolation can interact through shared equipment. Plant-level testing of the HTR-PM, a high-temperature gas-cooled reactor plant in which two reactor modules feed a shared turbine, shows that coordinated operation can be built into a commercial design, while larger module counts and richer shared systems remain primarily a simulation problem [12]. Remote microreactors face the same coordination needs with far fewer people on site. Preliminary economic evaluation indicates that remote monitoring and centralized offsite control, supported by digital twins, are likely necessary for viable deployment, so intermittent communication and delayed external support become normal operating conditions [13].

Figure 1 organizes these requirements as layers distinguished by what each decides and how directly it acts on the plant. The main point is that the higher a layer sits, the less directly it acts on the plant, and that independent protection stays outside every advanced layer. Independent protection takes autonomous protective action and stays isolated from every higher layer. Component and regulatory control holds process variables through qualified feedback loops that issue direct actuator commands. Plant-wide advanced control and optimization generates constrained setpoints and constraint margins for those loops. Supervisory and mission coordination selects operating modes, dispatches modules, and allocates shared resources, with consequential changes gated by approval. Operator-facing decision support, increasingly including generative artificial intelligence (GenAI), informs and recommends but does not act. A digital twin and plant-knowledge band runs alongside all five layers, supplying state estimates, predictions, and validation evidence, but it acquires no control role merely by being connected to the plant. The left panel of the figure lists the deployment classes, each of which weights these layers differently: a conventional plant concentrates activity in the lower layers, whereas a remote microreactor or an integrated energy system leans on supervisory coordination and decision support. An earlier framework for multimodular reactors combined control, diagnosis, and decision-making for nearly autonomous operation [14], and the layers used here add GenAI decision support.

Risk depends on the level of automation as much as on layer position. A given system may display information, recommend an action, generate a constrained setpoint, select among validated modes, or issue direct actuator commands. Collapsing these roles into

a single label such as "AI control" obscures which of them is meant. Human–automation research distinguishes levels of automation from information acquisition through action implementation and warns that higher autonomy degrades operator awareness and complicates recovery when automation reaches its limits [15,16]. Bounded autonomy keeps one layer's errors from propagating to another. A poor market forecast may reduce revenue but should not relax a fuel-temperature limit. A learning policy or GenAI agent may propose a mode or setpoint, but a deterministic governor, a constraint checker, and an operator approval gate can still reject it. The architecture is heterogeneous by design: different methods are used where their assumptions hold, and each higher layer can hand control back to a simpler verified layer.

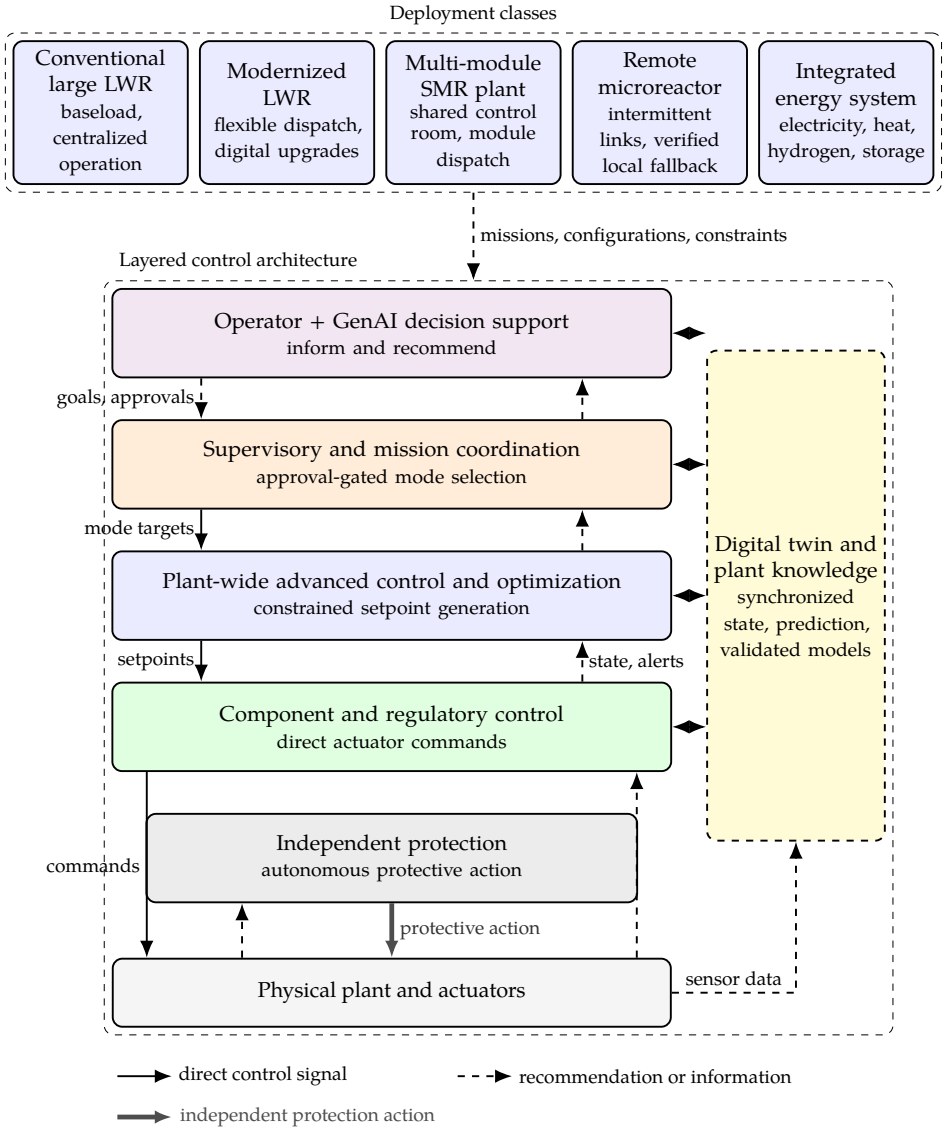


Figure 1. Deployment contexts and the bounded control layers used to organize the review.

1.3. Scope and Organization

The review covers technologies that can change plant state directly or materially shape a control decision: model-based and predictive control, learning-enabled control, digital twins, supervisory operation, and GenAI connected to engineering tools. Monitoring, diagnosis, prognostics, cybersecurity, and human factors are treated only where they change a control decision. Typical cases are a diagnosis that triggers a mode change, an estimate that feeds a controller, and an operator who must approve or override an automated action.

Earlier reviews synthesized reactor modeling and control [4,5], digital twin concepts with uncertainty for nuclear power applications [17], and digital twins for nuclear condition monitoring [18]. The present article connects control methods, digital twins, and GenAI within one framework centered on levels of automation.

The purpose of this review is to determine what level of automation the current evidence supports for each technology in advanced nuclear systems, and what demonstrations would justify raising it. Four questions guide the synthesis of each technology: (1) which plant layer fits the method, (2) how the method represents hard constraints and uncertainty, (3) what information, computation, and human approval must precede action, and (4) what demonstration would justify raising its level of automation. For the last question, each study is placed on an evidence maturity scale. Its lowest levels are a conceptual proposal, simplified-model simulation, and high-fidelity simulation. Above them come software-in-the-loop (SIL) and hardware-in-the-loop (HIL) testing, a representative physical testbed, demonstration at a nuclear research facility, and finally deployment at an operating plant.

The review focuses on nuclear applications published from 2015 to 2026. The search began in the author's own research area, where the evidence and its gaps were already known in detail. It was then widened to independent and international work through Google Scholar and citation chaining. Each search paired a plant term, such as nuclear, small modular reactor, microreactor, or a named reactor type, with a technology term covering predictive, supervisory, or autonomous control, reinforcement learning, digital twins, uncertainty quantification, or large-language-model agents. Among numerous studies, the review cites those that reach the highest evidence stage for a method and plant class, introduce a distinct approach, or mark the latest step in a line of work. Methods from control, machine learning, human factors, and cybersecurity are cited through textbooks, high-impact papers, and surveys.

The remainder of the article follows the layers of Figure 1 from the bottom up. Section 2 reviews model-based and constrained control, the layer closest to the qualified loops. Section 3 examines offline learning, safe reinforcement learning, and runtime assurance as candidate proposers above that layer. Section 4 turns to the digital twins on which both depend. Section 5 integrates these technologies into multi-module, integrated-energy, and remote operating strategies under human supervision and gathers the cybersecurity and human–automation risks they share. Section 6 evaluates grounded GenAI at the operator and engineering interface, Section 7 offers a staged roadmap from the author's perspective, and Section 8 provides summary remarks.

2. Model-Based and Advanced Control Strategies

The layered architecture of Section 1 assigns its two lowest layers, component regulation and plant-wide constrained coordination, to model-based control. Each model-based family is judged below by whether it preserves constraints and predictable behavior across the intended operating envelope, what it requires of its plant model, and what degree of autonomy the nuclear evidence justifies. Tracking error on a single nominal model says little about the first of these properties, yet the higher layers of Sections 3 and 5 rely on it.

2.1. Conventional Control Foundations and Model-Based Extensions

Operating reactors regulate power, temperature, pressure, and level almost entirely with single-loop PID control, cascade structures, feedforward compensation, gain scheduling, sequence logic, and interlocks [5,19]. These methods persist because operators and reviewers understand their behavior, their computational burden is negligible, and their verification is comparatively tractable. State feedback with observers, linear-quadratic regulation, and reference governors extend this foundation with an auditable structure [4].

Conventional loops also define the architecture into which any advanced controller must fit; a supervisory algorithm that ignores actuator saturation, anti-windup logic, or the bandwidth of the loops beneath it can fail however well it performs alone.

The limits of classical control appear where constraints, strong multivariable coupling, and wide operating envelopes interact, as in deep load following, startup and shutdown, and multi-module coordination. In these regimes, scheduled single-loop controllers must be tuned and interlocked case by case. Robust, adaptive, and nonlinear methods each address one source of difficulty. Adaptive control follows changing operating points, and H_∞ and μ synthesis trade nominal performance for guaranteed stability over a bounded uncertainty set [20]. Sliding-mode control rejects matched disturbances, and feedback linearization cancels strong nonlinearity when an accurate structural model exists [21,22]. Nuclear demonstrations remain concentrated in simplified-model simulation. On a drum-controlled 22 MWth microreactor model, a super-twisting sliding-mode controller cut the integrated tracking error of ramp following by roughly an order of magnitude relative to a tuned PID. The gain came at about nine times the control effort, and chattering persisted. Because the evaluation and design models coincided, robustness to mismatch was never tested [7]. A formal robustness claim built on an untested uncertainty description is weaker than a simpler controller evaluated against representative mismatch. Adaptive laws can learn changing or unknown dynamics online while keeping the loop stable, as in an adaptive MPC scheme that controls a heat-pipe microreactor model through a single heat-pipe failure [23] and in impedance learning for a human-guided robot in contact with an unknown environment [24]. Adaptive designs raise a separate concern: online adaptation is hard to audit under configuration control and should be bounded, monitored, and reversible.

Neighboring industries show which parts of this toolbox transfer. Process industries provide the deepest operating record for model predictive control (MPC), state estimation, and optimization above regulatory loops [25,26]. Aerospace contributes verified reconfiguration, analytical redundancy, and dissimilar backup control laws for flight-critical systems [27,28]. Power systems contribute hierarchical dispatch under communication constraints and adversarial conditions [29]. The algorithms transfer more readily than the commissioning assumptions. Process plants tolerated years of step-testing and occasional constraint violations, and an off-specification product is a recoverable loss; reactivity control allows none of this, since perturbation experiments on an operating core are restricted, and some violations are irreversible in licensing terms even when physically benign.

2.2. Model Predictive and Differentiable Predictive Control for Constrained Operation

Of the model-based families, MPC receives the most attention because it builds constraints into the control law. At each update it predicts the plant response over a finite horizon, optimizes a sequence of actions subject to state and actuator limits, applies the first action, and re-solves at the next sample [26,30,31]. This suits advanced reactors, where slow thermal inventories, xenon dynamics, storage states, and product demands must be weighed together. The variants map onto different needs. Linear MPC has a mature theory and fast solvers, and nonlinear MPC handles reactivity and thermal-hydraulic nonlinearity. Economic MPC optimizes plant value directly [32], and robust and stochastic MPC methods carry disturbance sets and chance constraints into the guarantee [33,34]. Explicit MPC precomputes a piecewise control law for verification [35,36], and distributed MPC coordinates local problems across modules.

In compact form, for the discrete-time prediction model

$$x_{i+1|k} = f\left(x_{i|k}, u_{i|k}, d_{i|k}\right), \qquad y_{i|k} = h\left(x_{i|k}\right), \tag{1}$$

the controller at sampling instant k solves

$$\begin{aligned} \mathbf{u}_k^\star = \arg\min_{\mathbf{u}_k} \; & \sum_{i=0}^{N-1} \ell\big(x_{i|k}, u_{i|k}, r_{i|k}\big) + V_f\big(x_{N|k}\big) \\ \text{s.t.} \quad & x_{0|k} = \hat{x}_k, \\ & x_{i+1|k} = f\big(x_{i|k}, u_{i|k}, d_{i|k}\big), \\ & x_{i|k} \in \mathcal{X}, \quad u_{i|k} \in \mathcal{U}, \quad \Delta u_{i|k} \in \Delta\mathcal{U}, \\ & x_{N|k} \in \mathcal{X}_f, \qquad i = 0, \dots, N-1. \end{aligned} \tag{2}$$

Here, $\hat{x}_k$ is the current state estimate, $r_{i|k}$ and $d_{i|k}$ are reference and disturbance forecasts, N is the prediction horizon, $\mathcal{X}$ and $\mathcal{U}$ are admissible state and input sets, and V_f and $\mathcal{X}_f$ are the terminal cost and terminal set. A common tracking cost is $\ell = \|y - r\|_Q^2 + \|u - u^{\text{ref}}\|_R^2 + \|\Delta u\|_S^2$. Only the first move, $u_k = u_{0|k}^\star$, is applied before the problem is solved again.

Nuclear applications lag industrial practice: receding-horizon power control was demonstrated on pressurized-water reactor (PWR) models in 2004 [37], nonlinear MPC built on feedback linearization has since been applied to full PWR plant models [22], and an MPC enabling anticipatory control strategies has been developed and assessed for a heat-pipe system relevant to microreactors [38]. On a simplified model of a sodium fast reactor's primary system, MPC built on Laguerre functions rejected reactivity and inlet-temperature disturbances faster than digital PI control, and it was embedded in a field-programmable gate array for hardware-in-the-loop tests [39]. In the microreactor comparison above, nonlinear MPC with a ten-step horizon and an extended Kalman filter tracked best but used by far the most control effort, again with robustness untested [7]. The larger benefit of MPC is explicit constraint handling, which no error integral captures.

The prediction model now differentiates nuclear MPC studies more than the optimization formulation does. Physics-based models are traceable but often exceed real-time budgets; reduced-order models and identified surrogates such as Koopman-operator predictors recover speed at the cost of approximation error [40,41]. Neural ordinary differential equation (ODE) models add continuous-time nonlinear prediction: on a three-loop thermal-fluid testbed, a neural ODE trained on twin-generated and experimental transients forecast four actuator channels in about 70 ms with 34% lower error than experiment-only training and supplied empirical 95th-percentile error bounds as constraint-tightening margins [42].

Differentiable predictive control (DPC) keeps the finite-horizon objective of MPC but moves the optimization offline into the training of an explicit policy [43]. With a differentiable predictor f_ϕ, a policy π_θ, scenario features $z_i^{(m)} = (\hat{x}_i^{(m)}, r_i^{(m)}, d_i^{(m)})$, and constraint-violation penalties p_x and p_u, a basic training problem is

$$\begin{aligned} \theta^\star = \arg\min_{\theta} \; & \frac{1}{M} \sum_{m=1}^{M} \sum_{i=0}^{N-1} \Big[\ell\Big(\hat{x}_i^{(m)}, u_i^{(m)}, r_i^{(m)}\Big) + p_x\Big(\hat{x}_i^{(m)}\Big) + p_u\Big(u_i^{(m)}\Big) \Big] \\ \text{s.t.} \quad & u_i^{(m)} = \pi_\theta\Big(z_i^{(m)}\Big), \\ & \hat{x}_{i+1}^{(m)} = f_\phi\Big(\hat{x}_i^{(m)}, u_i^{(m)}, d_i^{(m)}\Big), \qquad i = 0, \dots, N-1. \end{aligned} \tag{3}$$

Automatic differentiation propagates the gradient through the N-step rollout over sampled initial conditions, references, and disturbances. Online evaluation is then the explicit map $u_k = \pi_{\theta^\star}(z_k)$, with no nonlinear program at run time. Penalty training does not guarantee hard constraints, however, so such policies still need independent validation. For consequential functions they also need projection, command governance, or a verified safety filter.

Whichever variant is chosen, deployment requires bounded solver timing, a defined response to infeasibility, a model error that tightens constraints, and reversion to a verified regulatory mode. These requirements favor supervisory MPC and command governors above existing loops, the pattern the next subsection examines.

2.3. Supervisory and Fault-Tolerant Coordination

Supervisory, hierarchical, fault-tolerant, and reconfigurable control all address how the plant should change strategy when goals, equipment availability, or plant conditions change. Their shared response is a coordination layer above the local loops that manages mode transitions, setpoint generation, control allocation among redundant actuators, constraint management, and verified degraded modes [27,28]. Reference and command governors are the most attractive entry point because they leave the regulator intact and modify only commands that would violate a predicted constraint.

The most complete nuclear demonstrations in the reviewed literature come from one line of work on fluoride-salt-cooled reactors. A reference governor over three PID loops on a System Analysis Module (SAM) model [44,45] of a 320 MWth pebble-bed reactor accepted, trimmed, or rejected power setpoint requests so that underactuated secondary-side temperatures stayed within limits during a 40% power reduction at 5% per minute, halting the ramp at about 347 s before a violation, and it kept enforcing constraints under Gaussian sensor noise [10]. The follow-on project extended the same reactor to an integrated energy system with two-tank molten-salt storage of up to 440 MWh and combined mode-dependent PID control under a multivariable command governor, physics-based diagnostics with Markov component models, and a decision layer that converts projected failure probabilities into mode transitions and maintenance policies [11,46]. In its end-to-end scenario, heat-exchanger fouling was detected at about 100 s and injected stuck-valve faults triggered automated transitions back to a load-following mode, with operator override retained. These demonstrations remain in high-fidelity simulation, and their authors list static surrogate matrices, constant assumed failure rates, and the absence of physical validation as open items.

Fault tolerance adds a chain of detection, isolation, state reconstruction, decision, reconfiguration, and recovery to this coordination layer [27]. Passive designs tolerate a predefined fault set through fixed margins and are simpler to verify but conservative; active designs recover more capability after diagnosis but inherit the failure modes of the diagnostic chain. The chain also defines how health information enters control. A confirmed diagnosis can tighten limits, reweight an objective, substitute an analytically redundant sensor, or trigger a conservative mode, as the command-governor architecture does with evolving failure probabilities; it should not relax a safety constraint without independently validated uncertainty treatment.

Supervisory structures also provide the interface through which learning and generative methods can enter the plant. A learning policy can propose a setpoint, a digital twin can forecast the response, and a governor can accept, trim, or reject the proposal before local control acts. This decomposition lets new decision methods be evaluated without direct actuator access, and Sections 3 and 6 build on it.

2.4. Comparative Synthesis

The method map in Figure 2 positions the families by dependence on identified versus physics-based models and by explicit constraint enforcement. The map shows that validation strength runs opposite to constraint handling. PID and gain scheduling, which handle constraints least explicitly, have operating-plant evidence, while most predictive families have been tested only in simulation. Each node records the strongest nuclear validation setting, so the map is a qualitative guide to role and evidence rather than a maturity score. Table 1 summarizes the same families by role, principal strength, and the condition under which that strength holds.

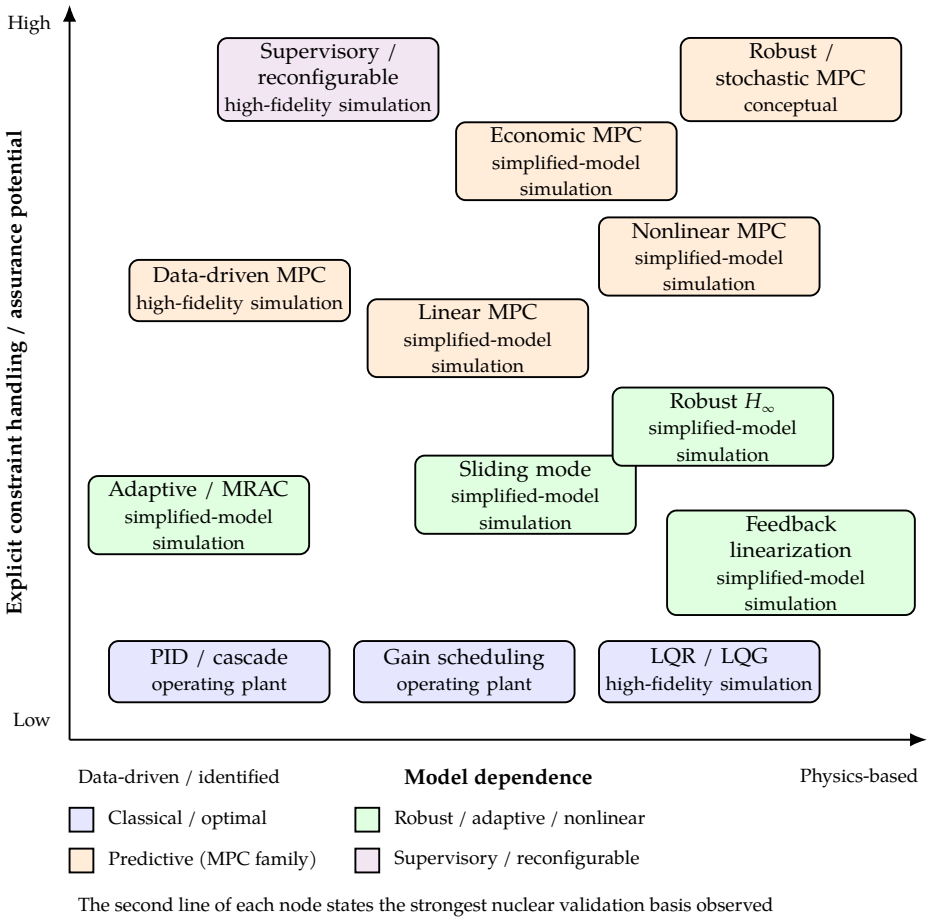


Figure 2. Qualitative landscape of model-based control families, positioned by model dependence and explicit constraint handling, with the strongest nuclear validation setting shown in each node.

Table 1. Qualitative comparison of model-based control families for advanced nuclear operation.

Method Family	Best-Suited Role	Principal Strength and Condition	Representative Nuclear Work
Conventional and state-space control	Fast component and regulatory loops	Transparent, low-computation behavior; wide-range operation requires scheduling and interlocks	Reviews and plant-control foundations [4,5]
Robust and nonlinear control	Disturbance rejection and wide-envelope regulation	Useful when uncertainty bounds or nonlinear structure are physically defensible	PWR and microreactor studies [7,21]
Model predictive control	Constrained multivariable coordination	Direct treatment of limits and competing objectives; depends on a credible real-time model	Reactor maneuvering and integrated systems [11,22,37]
Reference and command governors	Bounded setpoint supervision	Preserves qualified local loops while screening requests	Advanced-reactor supervisory control [10]
Fault-tolerant and reconfigurable control	Degraded-mode operation	Coordinates diagnosis, reallocation, and safe-mode reversion; degree of autonomy must reflect diagnostic confidence	Integrated supervisory demonstrations [11,46]
Learned predictive models	Fast forecasting within MPC or supervision	Computationally efficient, but constraint margins must include surrogate error	Testbed-linked neural models [42]

The comparison rests on two distinctions. The first separates nominal tracking from safety-relevant capability. Sliding-mode and predictive controllers cut tracking error by roughly an order of magnitude in matched microreactor comparisons that test no model mismatch [7]; the supervisory studies improve constraint management and fault response directly, the capability flexible operation requires [10,11]. The second concerns validation settings. Classical control has operating-plant evidence, and plant-scale testing of coordinated multi-modular operation at the HTR-PM shows that multi-module concepts can reach that level when the plant is designed for them [12]. Among the studies reviewed here, evidence for optimization-based controllers is concentrated in simulation, the neural ODE forecaster is the only element evaluated on a physical testbed [42], and none has entered sustained operation at a nuclear facility.

Conventional loops remain the permanent lower layer, and robust and nonlinear methods are useful where a physically meaningful uncertainty or structure can be stated. MPC is the strongest candidate for constrained plant-wide coordination. Supervisory and fault-tolerant structures supply the mode logic and safe reversion that higher-level automation needs, and learned predictors reduce computation within a declared operating domain and monitored prediction error. The near-term research priority is therefore integrated evaluation of the complete control stack, with estimator error, actuator saturation, communication delay, infeasibility, model drift, and takeover by the backup controller exercised on the same scenarios. That systems view carries directly into the learning-enabled methods of Section 3, which require the supervisory interface described above before they can be deployed.

3. Learning-Enabled Control and Reinforcement Learning

Section 2 ended with a supervisory layer that can accept, trim, or reject proposals from methods it does not itself verify. Reinforcement learning (RL) is the most prominent candidate proposer because it handles sequential decisions that resist compact optimization. Its best-known successes, however, come from domains where exploration is inexpensive and failure is recoverable, and neither condition holds in nuclear operation. The evidence reviewed below places learning at the slower supervisory layers, trained offline on simulators or digital twins, with verified constraints kept outside the learned policy.

3.1. Appropriate Decisions and Data Pathways

The Markov decision process (MDP) provides the common language [47]. The state collects core power, precursor concentrations, coolant temperatures and flows, xenon inventory, and component health. The observation is only the subset visible through instrumentation, so most reactor problems are partially observed. Actions range from continuous actuator commands to discrete supervisory choices, and horizons span seconds to months. Because limits on fuel temperature, thermal stress, and reactivity insertion cannot be reduced to average reward, the constrained MDP represents them as explicit constraints [48].

Formally, an MDP is $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $P(s' \mid s, a)$ is the transition model, $R(s, a)$ is the one-step reward, and the policy $\pi_\theta(a \mid s)$ maps a state to an action distribution. A constrained policy-optimization problem can be written as

$$\begin{aligned} \theta^\star \in \arg\max_\theta \quad & J_R(\theta) = \mathbb{E}_{\pi_\theta, P}\left[\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t)\right] \\ \text{s.t.} \quad & J_{C_j}(\theta) = \mathbb{E}_{\pi_\theta, P}\left[\sum_{t=0}^{T-1} \gamma_c^t C_j(s_t, a_t)\right] \le d_j, \quad j = 1, \ldots, p. \end{aligned} \tag{4}$$

The reward can represent production value minus tracking penalties, and each C_j a quantity to be limited, such as temperature-limit proximity. For a fixed policy, the value function satisfies the Bellman relation

$$V^{\pi}(s) = \mathbb{E}_{\substack{a\sim\pi(\cdot|s)\\ s'\sim P(\cdot|s,a)}}[R(s,a) + \gamma V^{\pi}(s')]. \tag{5}$$

Actor–critic algorithms approximate V^{π} or Q^{π} with a critic and move π_{θ} toward higher estimated return. The expectation constraints in Equation (4) bound the average discounted cost under the sampled distribution and leave individual unsafe actions possible, the gap the deployment-time filters of the next subsection close.

The learning paradigms differ mainly in the plant interaction they require, which is the scarcest and most safety-relevant resource. Model-free actor–critic methods handle continuous actions but need extensive interaction [49]. Offline RL, imitation learning, and model-based RL replace plant interaction with historical records, expert demonstrations, and simulation [50,51]. Each source has a characteristic limitation: plant records cluster near steady state and sanctioned maneuvers, so coverage of the transients a controller most needs is poor; demonstrations carry tacit knowledge of sequencing and margins but also inconsistency and habit; and simulator trajectories inherit model bias. Conservative offline algorithms suppress optimistic extrapolation beyond the dataset but cannot create coverage the data lack [52,53].

Learning must also justify itself against the simpler methods of Section 2. Conventional and predictive control already handle fast single-loop regulation well, as the drum-controlled microreactor comparisons showed [7]. The credible openings are multivariable coupling, long-horizon supervisory decisions, and objectives such as mode selection or health-aware derating. Recent nuclear studies address these targets: task analysis of power-increase procedures has structured a proximal policy optimization action space for an SMR [54], RL has optimized discrete control sequences for advanced reactors [55], deep RL has performed multistep criticality search and power shaping on microreactor models [6], and a deep deterministic policy gradient agent has coupled multivariable control of a small PWR [56]. At the plant-wide level, an autonomous emergency operation system that uses AI techniques has been designed and prototyped to perform emergency control functions with reduced operator intervention [57].

3.2. Safety Mechanisms and Runtime Assurance

Safe RL methods are best organized by where their safety mechanism acts. Penalty terms, constrained policy optimization, and chance constraints discourage unsafe behavior statistically under the training distribution [58]. Action shielding checks each proposed action against a formal specification [59], and control barrier functions, Lyapunov filters, and predictive safety filters project or reject an action unless a safe continuation remains available [60,61]. Runtime assurance completes the set with a monitor and a verified fallback controller that takes over when the policy leaves the analyzed envelope.

These mechanisms divide into two classes with different defensibility. Training-time constraints shape what the policy learns, but their guarantee depends on the training distribution. Shields, filters, and runtime assurance instead prevent unsafe actions by construction, at the price of a validated model or certified safe set. A credible nuclear architecture therefore needs the preventive class for consequential functions, and its filter model and safe set become safety-significant artifacts that must be validated as such.

Runtime assurance follows the Simplex pattern [62]. An advanced controller, here a learned policy, runs beside a verified baseline controller, and a decision module switches to the baseline whenever the advanced output cannot be shown to be safe. For supervisory setpoints, the decision module can be a reference governor, itself a deterministic safety

filter [63]. Let r_{k-1} be the setpoint in force and r_k^{RL} the setpoint proposed by the policy at step k. The governor applies

$$\begin{aligned} r_k &= r_{k-1} + \kappa_k\left(r_k^{\mathrm{RL}} - r_{k-1}\right), \\ \kappa_k &= \max\left\{\kappa \in [0,1] \,:\, \left(\hat{x}_k,\ r_{k-1} + \kappa\left(r_k^{\mathrm{RL}} - r_{k-1}\right)\right) \in O\right\}, \end{aligned} \tag{6}$$

where $\hat{x}_k$ is the state estimate and O is the set of state and constant-setpoint pairs for which the predicted closed-loop response satisfies every constraint. The filter passes the proposal when $\kappa_k = 1$, trims it when $0 < \kappa_k < 1$, and holds the current setpoint when $\kappa_k = 0$. Because O is invariant under a constant setpoint, holding r_{k-1} always remains admissible, so the filter has a safe option whatever the policy proposes. When the state estimate leaves the region in which O was computed, or the policy input leaves its training envelope, a monitor returns the setpoint to the baseline. For direct actuator commands, a control barrier function filter plays the same role [60]. The fluoride-salt-cooled reactor governor of Section 2.3 has this structure [10], so a learned proposal can replace the requested setpoint without changing the filter. The guarantee holds only for the model used to compute O, a data-driven linear surrogate in that study, so O must be tightened by the surrogate error bound (Section 4.3).

Published nuclear work demonstrates useful components without end-to-end deployment. A constraint-aware RL algorithm has been studied for supervisory power-plant control [64], and deep RL has been applied to boiling-water-reactor maneuvering [65]. Most of the nuclear RL papers reviewed here report average tracking performance and omit violation counts, rare-disturbance response, and sensor-fault behavior. The drum-controlled microreactor study of Tunkle et al. [66] is a partial exception, with a matched PID baseline and explicit fault cases. Its proximal policy optimization agents roughly halved short-ramp tracking error relative to the tuned PID, kept lower error under measurement noise with at least 10% less control effort, and, in the multi-agent variant, compensated for a randomly disabled drum; yet PID remained more accurate over a 20,000 s xenon-dominated transient. In that study, RL's advantage lay on short anticipatory horizons and its weakness on slow feedback dynamics absent from training. Self-healing control on a generic full-scope PWR simulator is a second exception: an autoencoder and an RL agent returned the plant to normal operation after false data were injected into steam generator level signals [67]. All of these studies remain simulation based; their main contribution is to identify the decision structures and failure cases that a safety filter and validation program must address.

3.3. Simulation Transfer and Deployment Lifecycle

Model-based RL shifts interaction from the plant to a system code, reduced-order model, or digital twin [51]. Short rollouts limit the compounding of model error, the central failure mode of learned world models [68]; stability-aware variants restrict policy updates to a region the model can certify [69]; and ensembles, parameter randomization, and injected noise reduce overfitting to one simulator. None of these removes the sim-to-real problem, because a policy exploits simulator errors in neutronic feedback, thermal-hydraulic closures, or sensor modeling as if they were physics. Validation therefore must be layered: independent simulators, out-of-distribution scenarios, SIL and HIL tests, and shadow-mode comparison against the incumbent controller form a sequence of increasingly plant-like evidence.

A testbed-linked surrogate study shows this discipline in practice. A graph neural ODE surrogate was pretrained on SAM transients of a three-loop thermal-fluid facility and fine-tuned on 30 measured sequences from the facility itself [70]. Fine-tuning recovered a heat-exchanger flow exponent near 0.87, close to the value of 0.8 used in SAM, which suggests that transferable physical structure was learned; the model also sustained forecasts for five permanently uninstrumented volumes, but its ensemble uncertainty coverage collapsed

outside the training power envelope. The same applies to RL: train with accelerated models, but qualify policies against independent physics and hardware, and declare and monitor the envelope within which the learned model is credible.

A practical deployment pattern follows, shown in Figure 3. The key point is that a learned policy gains a larger role only after independent testing, and every update returns through the same offline gate. The learned system first operates as an offline design aid, then as a shadow policy, then as an approval-gated recommender; the design-aid and recommender roles appear credible in the near term. Bounded automatic action for low-consequence functions is a plausible mid-term step, considered only after the policy is enclosed by a deterministic filter, monitored for domain departure, and paired with a verified backup controller, while end-to-end learned actuator control across a broad plant envelope remains a research objective. The deployed policy stays frozen, and retraining on new plant data re-enters the pipeline under approval, because silent online learning invalidates the evidence assembled before deployment. Every step presupposes a simulator or twin credible enough to train against and independent enough to test against, the subject of Section 4.

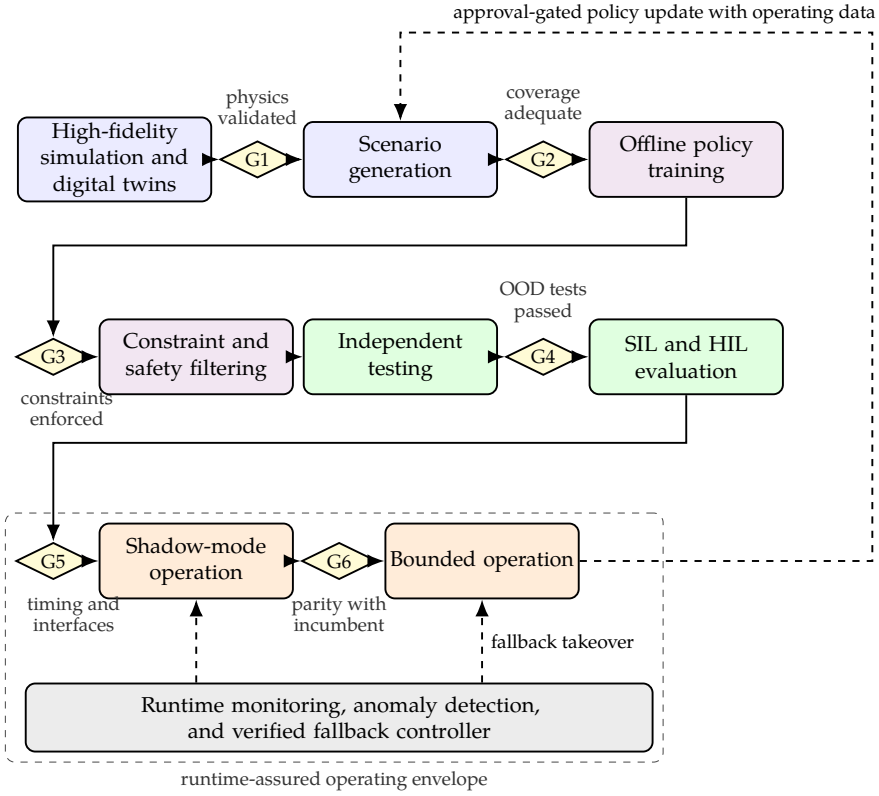


Figure 3. Offline development, independent testing, runtime assurance, and approval-gated updates for learning-enabled control.

4. Digital Twins for Control Design and Online Operation

The two preceding sections ended at the same dependency. Predictive control needs a plant model that runs faster than the decision it informs, learned control needs an environment to train in and an independent one to test against, and both need estimates of states that sensors cannot reach. Digital twins are the infrastructure intended to meet these needs. Their control value lies in three functions: estimating unmeasured states, predicting the effect of candidate actions, and providing a common environment for controller design and testing. Twin outputs deserve a control role only in proportion to demonstrated synchronization, quantified uncertainty, and validated operating domain, and the definitions and evidence below support that position.

4.1. Definition, Synchronization, and Observability

A digital twin is best defined by the functions it performs for a physical asset; a simulation model alone is not sufficient [71,72]. Manufacturing research separates a digital model with no automatic plant connection, a digital shadow with one-way data flow, and a digital twin with bidirectional exchange [73]. Control needs finer levels above the shadow level: a synchronized twin estimates the plant state from live measurements, a predictive twin also forecasts responses to disturbances and candidate actions, and a prescriptive twin recommends or supports actions, so its errors are consequential. The term is used more loosely than these definitions permit. A model becomes a twin only through a defined data connection to its physical counterpart [74], yet many nuclear studies attach the label to offline simulation models [18]; the National Academies identify data assimilation, uncertainty quantification (UQ), and verification and validation as what separates twins from models [75]. An unsynchronized model provides none of the state awareness that online decision-making assumes.

Synchronization turns a model into a twin through sensor reconciliation, state estimation, parameter updating, and configuration management. Ensemble Kalman methods remain the standard method for nonlinear state estimation [76], moving-horizon estimation handles constraints explicitly, and Bayesian parameter updating tracks slow drift such as fouling. In nuclear thermal-hydraulics, an ensemble Kalman filter coupled non-intrusively to a system code performed comparably to an extended Kalman filter on separate-effect and integral-effect tests [77]. Machine learning calibration has also corrected a fast, low-accuracy plant twin online against measured data from a steam generator tube rupture experiment in the PKL III facility [78]. Hybrid machine learning schemes exist for reactor-operation twins [79], and probabilistic graphical models treat sensing, state, and decision as one Bayesian network [80]. Virtual sensing extends estimation to uninstrumented locations: a calibrated SAM twin of a three-loop thermal-fluid facility exposes 247 unmeasured parameters as virtual sensors [81], and a graph neural ODE surrogate reconstructs five permanently uninstrumented volumes from twelve thermocouples [70]. Both are tied to a physical facility, but observability remains a hard limit: a state not constrained by measurements is inferred through the model and inherits its discrepancy.

Drift should be treated as normal. Growing innovation sequences, persistent residual bias, and collapsing interval coverage are practical indicators. The graded response is recalibration, demotion to an advisory role, or removal from service. Plant configuration, model version, calibration data, and parameter updates must stay synchronized too, or an accurate model can still represent the wrong plant state [75].

4.2. Physics-Based, Learned, Hybrid, and Whole-System Models

The physics core of most nuclear twins is a best-estimate system thermal-hydraulics code such as SAM, whose strengths are physically consistent prediction within the validated domain and whose weaknesses are execution speeds near real time, non-differentiable solvers, and expert recalibration [45]. Projection-based reduced-order models recover speed while remaining tied to the original equations but degrade under strong nonlinearity and wide parameter variation [40]. Learned models occupy the complementary corner: sparse regression can recover interpretable governing equations [82], neural operators learn solution maps across whole parameter families [83,84], and graph neural networks add a relational bias that mirrors plant topology [85]. Learned models win on speed and differentiability; physics codes win on auditability and degrade more predictably outside the training data, though use beyond the validated domain still requires uncertainty treatment. No single class combines these strengths.

Hybrid twins combine these strengths most directly. Learned closures embedded in SAM and coarse-mesh turbulence models preserve system-level conservation while correcting the weakest empirical components [86,87], and physics-informed reduced-order and point-kinetics models apply the same logic at the reactor-physics level [88,89]. Hybridization reduces data requirements because the learned component spans only the residual left by the physics. The hazard is compensating error: a flexible closure can match observations for the wrong physical reason, so the learned correction, its training domain, and its uncertainty should be made explicit.

Compensating error has a theoretical root in non-identifiability. When model parameters and a discrepancy term are calibrated against the same data, many combinations fit equally well, so close agreement does not show that each part is correct [90]. In a hybrid twin the learned closure acts as the discrepancy term and can absorb errors in other closures, boundary conditions, or sensor calibration. The combined model then fails when the plant moves to a regime where those hidden errors change, which is where control depends most on the twin. Four checks expose it. The closure is validated on separate-effect data before integral tests [91]. Learned physical parameters are compared with established values, as for the flow exponent of Section 3.3. Held-out quantities, such as unused sensors or conserved totals, are predicted. The closure is disabled to see whether the error reappears elsewhere. Passing these checks reduces the risk without removing it.

Whole-system twins extend these ideas to plant networks, with components as typed nodes and interactions as typed edges. A heterogeneous graph twin for EBR-II and a generic fluoride-salt-cooled reactor infers the full system state, including unmeasurable peak fuel temperature, from a few sensor nodes; it reproduces SAM transients with about 3% average error and carries ensemble uncertainty from fifteen models, but it has been trained and tested only against high-fidelity simulation, and steady-state offsets near 10 K remain [92]. The graph neural ODE extension adds continuous-time dynamics, physics-guided message passing, and fine-tuning on facility data [70]; both build on earlier work in the same research line that integrated learned closures into system codes and established a validated SAM baseline with quantified uncertainty for the FFTF loss-of-flow benchmark [86,87,93]. Graph structure suits multi-module plants, yet transfer across module counts and designs is untested. Neural operators offer a further route to fast field prediction and virtual sensing [94,95]. For every class, control relevance depends less on speed than on whether prediction error and domain departure are visible to the controller, and that visibility is the subject of the next subsection.

4.3. Uncertainty Quantification and Automation Gating

A control-oriented twin needs a calibrated predictive distribution rather than a point estimate alone, because constraint tightening and automation gating rely on it. A review of uncertainty quantification and software risk analysis for digital twins in nearly autonomous management and control systems reaches the same conclusion: twin credibility must be quantified before twin outputs are allowed to shape control decisions [96]. Forward UQ propagates known input and parameter uncertainty through the model; inverse UQ infers parameter distributions, and ideally a model-discrepancy term, from experimental data and is the harder task in nuclear applications and the one with greater effect on control. The modular Bayesian formulation with Gaussian-process emulators replaced expert-judgment uncertainty ranges for thermal-hydraulics code parameters with distributions learned from transient data, first in theory and then in its application to the TRACE code [97,98]; a later survey documents persistent obstacles of parameter identifiability, confounding between parameter uncertainty and model discrepancy, and scarce validation experiments [99].

Integrated Bayesian assessment and deep Gaussian-process emulation extend the framework to expensive nuclear models [100,101].

Modular Bayesian analysis has also been applied at the closure level, where model form is weakest. Applications range from boiling heat transfer and wall boiling to bubbly flow with machine learning closures [102–106]. Bayesian-optimized deep ensembles gave learned surrogates better uncertainty estimates for thermal stratification and critical heat flux [107,108]. In each case a model discrepancy should reduce confidence before it becomes a constraint violation.

Automation gating follows directly. A twin inside a validated domain with calibrated residuals may initialize an optimizer or provide bounded forecast margins; as uncertainty grows, the dependent function should step down from automatic action to recommendation to information. In the graph neural ODE study, nominal 90% intervals covered 91% of observed-node errors inside the training envelope and only 52% outside it [70]. A learned model's interval cannot be assumed valid everywhere, so the controller must monitor model credibility as well as the plant state.

Constraint tightening turns this principle into a control law. Let y be a constrained output with limit $y_{\max}$, and let $\hat{y}_{i|k}$ be the twin prediction i steps ahead. The predictive controller of Equation (2) then enforces

$$\hat{y}_{i|k} \leq y_{\max} - \delta_{i,k}, \qquad i = 1, \ldots, N, \tag{7}$$

where the margin $\delta_{i,k}$ is an upper quantile of the twin's i-step prediction error, estimated on validation data and updated online from the residuals. When the observed coverage of the quantile falls below its nominal level over a moving window, $\delta_{i,k}$ is enlarged. When no admissible action satisfies Equation (7), the dependent function steps down to advice, and the regulatory loops hold the last verified setpoint. Tube-based and stochastic MPC provide the formal basis for such margins [33,34]. On the three-loop facility, the neural ODE forecaster reports empirical 95th- and 99th-percentile error bounds suited to the margin [42]. The coverage drop reported above shows why the margin must be monitored. Closed-loop tightening has not yet been demonstrated.

4.4. Prediction and Controller-Development Infrastructure

Prediction serves control only when it arrives before the decision it informs, and reported latencies now fall well inside typical supervisory update periods. The heterogeneous-graph whole-system twin processes five time steps per second; the graph neural ODE surrogate completes a 300 s rollout in 2.9 s; the latent neural ODE forecasts four actuator channels one second ahead in about 70 ms; and a recurrent accelerator of the calibrated SAM testbed twin reaches 119 ms median inference with a full 300 s forecast in 1.45 s [42,70,81,92]. Such margins leave room for computation, uncertainty propagation, constraint checking, and communication within realistic supervisory periods. The coupling pattern then sets where twin error enters: the state estimate that initializes MPC, forecast trajectories and constraint margins for look-ahead supervision, or candidate-action evaluations for learned policies.

Beyond online service, the same twin supports the controller lifecycle, from requirements analysis and scenario generation through SIL and HIL testing to commissioning and post-event reconstruction [109,110]. A common twin links design assumptions to operating evidence, but a controller validated only against its own design model inherits every shared blind spot. The three-loop platform shows both sides: it couples a physical facility, a calibrated SAM twin with 1.2 °C mean temperature error and flow deviations under 2.8%, and a recurrent accelerator, while programmable logic controller interlocks remain the deterministic safety layer [81]. Twin-generated transients beyond experimental safety limits improved the neural ODE actuator forecaster by 34% relative to experiment-only

training [42]. Synthetic scenarios are nonetheless a modeled product whose plausibility, coverage, and inherited bias need assessment before they train a controller, and training and acceptance models must stay separate.

Rare events need deliberate generation because the transients that matter most for control, such as pump trips, loss of heat sink, and actuator failures, are the ones a plant or test facility seldom provides. System codes can produce them through fault injection, through dynamic event trees that branch at each stochastic event and follow every branch in time [111], and through sampling that concentrates runs where failures occur. Such data improve accuracy when a model learns plant structure from abundant code transients before a small measured set corrects it, as in the graph neural ODE pretrained on SAM transients and fine-tuned on 30 measured sequences [70]. They improve robustness because training reaches states that cannot be run safely on hardware. The limit is inherited model-form error, so a model trained on synthetic transients must be accepted against independent data.

Figure 4 makes this lifecycle explicit: development stays offline until an independent validation gate is passed, synchronized operation is monitored for drift, and detected drift routes back through a model-update gate to recalibration and renewed validation. In short, a twin earns and keeps a control role only through validation gates, and drift lowers that role until the twin is revalidated. Fleet data can improve future versions, but a deployed twin should not update itself in a way that bypasses the drift-response logic or configuration control.

The lifecycle separates three kinds of model update, each with its own control. State synchronization changes only the estimated plant state. It runs continuously through data assimilation and needs no requalification, although its residuals feed the drift indicators of Section 4.1. Parameter recalibration adjusts a few physically meaningful parameters, such as fouling factors, within ranges validated in advance. It can be automated when each update is logged, checked against held-out measurements, and reversible. Structural change covers new closures, retrained components, and changes in plant topology. It is made offline and passes the independent validation gate of Figure 4 again. Each released version is kept under configuration control with its training data, validation record, and validated domain (Section 7.3). Table 2 summarizes these capabilities, their safeguards, and representative work.

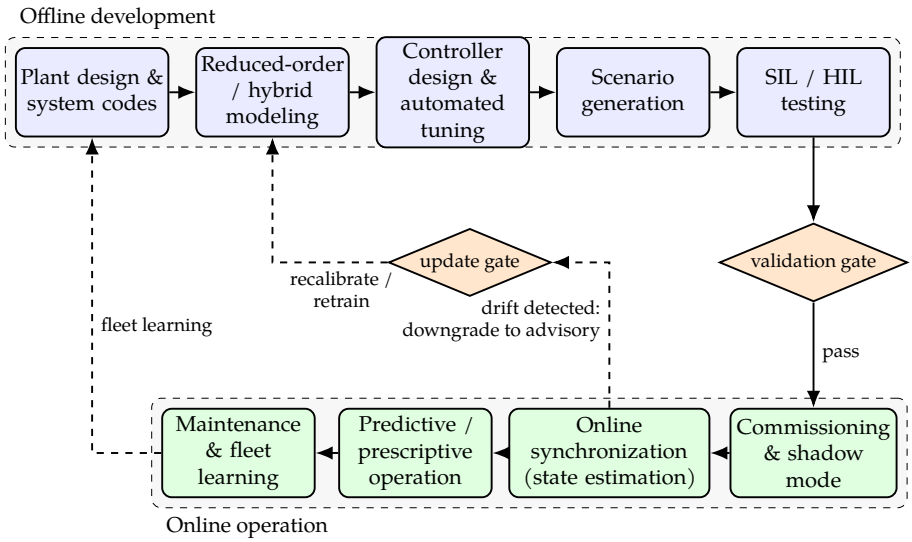


Figure 4. Digital-twin-enabled control lifecycle linking offline model and controller development to independently gated deployment, synchronized operation, drift-triggered fallback, and controlled model updates.

Table 2. Representative control-oriented digital-twin capabilities and their present limitations.

Capability	Contribution to Control	Required Safeguard	Representative Work
Synchronized plant twin	State estimation, virtual sensing, and operator context	Sensor reconciliation, configuration control, and drift monitoring	Thermal-fluid testbed and reactor-operation frameworks [79,81]
Whole-system graph twin	Sparse-sensor inference across connected components	Independent validation beyond training simulations	Heterogeneous graph and graph neural ODE twins [70,92]
Hybrid physics–ML model	Fast prediction with conservation structure	Explicit treatment of learned-closure error	SAM-linked closures and reduced models [86–88]
Uncertainty-aware twin	Constraint tightening and graded automation	Calibrated prediction intervals and discrepancy monitoring	Bayesian UQ and ensemble methods [99,100,107]
Neural-operator virtual sensing	Rapid field reconstruction from limited measurements	Domain detection and validation of reconstructed states	Reactor neural operators [94,95]
Controller-development twin	Scenario generation, tuning, and SIL/HIL testing	Independent acceptance model or facility	Testbed-linked forecasting [42]

4.5. Credibility Limits

The evidence above supports a consistent list of limits. No reviewed nuclear study demonstrates a synchronized, uncertainty-quantified twin exercising sustained closed-loop control at an operating reactor; representative physical testbeds mark the validation frontier. Observability is incomplete, model-form error persists even in hybrid twins, and abnormal-event data are sparse, so the regions where a twin matters most are the least validated. Latency, software and model versioning, and the cybersecurity of connected plant models add constraints that grow with connectivity [18,75]. Cyber risks shared by all layers are gathered in Section 5.4.

A twin intended to influence control becomes credible when seven items are documented: intended use, validated operating domain, quantified prediction and state-estimation error, independent testing, drift detection, defined behavior when the twin is removed from service, and configuration control. Advisory use can tolerate larger error than constraint enforcement, and the governing principle is that the twin's control role should contract automatically when model credibility declines. Section 5 now places twins, the controllers of Section 2, and the learned policies of Section 3 into plant-level operating strategies. There, that principle must hold across modules, missions, and operators.

5. Supervisory Control and Intelligent Operation Strategies

The preceding three sections examined model-based controllers, learning-enabled controllers, and digital twins largely as individual technologies. Their plant-level value depends on how they combine into operating strategies for advanced reactors, multi-module stations, and integrated energy systems. Three questions follow: how decisions are assigned across layers, what level of automation each layer receives, and which safe modes apply when models, communications, or equipment fail. Supervisory control assigns modes and setpoints within plant constraints, so its design should begin with functional allocation and not with a preferred algorithm.

Figure 5 shows the resulting pattern. The central point is that optimization and remote support only propose, while local control and protection keep the plant safe when either is lost. Mission-level optimization proposes schedules, an operator approval gate bounds consequential changes, and the plant supervisor distributes admissible setpoints to local module and balance-of-plant controllers. Remote support is nonessential to safe local regulation, and loss of communication or a constraint violation invokes a verified hold or setback.

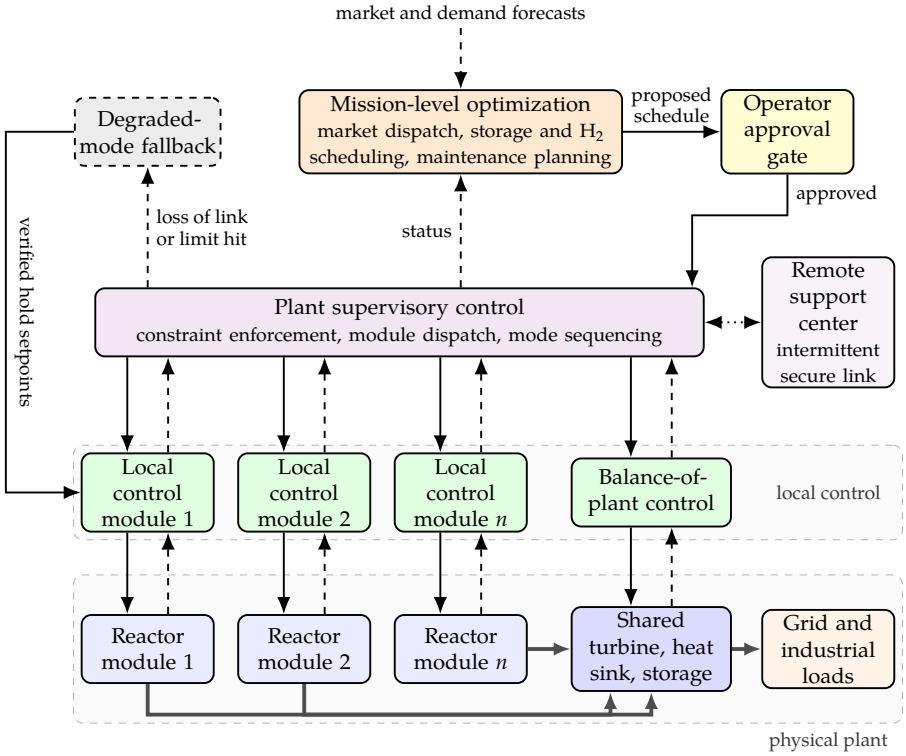


Figure 5. Supervisory architecture for multi-module and remote operation, showing bounded mission-level scheduling, operator approval, local control, secure remote support, and verified degraded-mode fallback.

5.1. Functional Allocation and Transition Management

The level of automation should be assigned separately to each function. Human–automation research grades automation from information display, through recommendation, approval-gated action, and bounded automatic action, to mission-level autonomy [15,112]. The scale measures actual control over plant state, not algorithmic sophistication: an RL policy that only recommends a setpoint sits at a lower level of automation than an interlock that acts automatically. Intermediate automation levels can also degrade situation awareness while reducing workload only modestly [16]. Seven criteria govern the assignment: decision timescale, consequence severity, reversibility, observability, model uncertainty, communication availability, and time available for operator intervention. Fast, reversible, well instrumented, and well modeled functions suit bounded automatic action. Rare, consequential, or poorly observable decisions justify approval gates.

Planned transitions concentrate the complexity to which this allocation applies. Startup requires permissive checks, a controlled approach to criticality, and bounded heat-up trajectories, and shutdown imposes analogous constraints in reverse. Transitions among electricity, heat, storage, and hydrogen modes couple the reactor to turbine and process dynamics far from nominal conditions. Local regulators then track process variables; supervisory logic checks permissives, sequences modes, and trims references before constraints are reached; operators approve high-consequence transitions; and digital twins can preview trajectories while permissive logic remains deterministic. The fluoride-salt-cooled reactor governor studies of Section 2 illustrate this separation, with PID loops regulating the plant, a governor screening requested maneuvers, and a higher layer coordinating storage and fault response [10,11]. Transition automation is most credible when sequencing, constraint enforcement, and direct regulation are distinct functions.

Hybrid-systems and discrete-event theory give this separation a formal basis. An integrated energy system can be modeled as a hybrid automaton [113]. Each discrete mode, such as load following, storage charging or discharging, hydrogen production, or hold, has its own continuous dynamics and an invariant, the operating limits that must hold in that mode. Each transition has a guard, the permissive conditions under which it may start, and a reset that assigns new references. Safety of the transition logic then becomes a reachability question: whether any admissible sequence of mode switches and bounded disturbances can drive the state across a limit. At the discrete level, supervisory control theory synthesizes a supervisor that admits only legal event sequences [114], for example by blocking a mode change while a previous transition is in progress. The reviewed integrated-energy studies switch among such modes [11] but do not report a formal verification of the switching logic, which remains an open step.

Requirements differ between the existing fleet and advanced designs. LWR startup, shutdown, and load-following sequences are procedurally mature and partially automated. Advanced reactors change the constraint set: strong passive feedback and large thermal inertia can simplify power regulation, while drum actuation, coolant freezing margins, and sparse operating experience add limits that LWR practice never exercised. Drum-controlled microreactor comparisons and procedure-structured RL studies show that several algorithms can execute nominal ramps [7,54,66], but robust transition management under mismatch, sensor faults, and abnormal mode changes is unresolved. Automated sequencing is therefore credible where transition models are validated; first-of-a-kind transitions should remain conservative and human-approved until operating experience accumulates.

5.2. Multi-Module, Integrated-Energy, and Remote Operation

Multi-module stations require plant-level dispatch because shared turbines, heat sinks, storage, or product systems couple otherwise independent modules. The supervisor therefore allocates power, staggers transitions, and preserves reserve. Four problems are specific to such stations. A transient in one module disturbs the others through shared equipment, which calls for the coordinated control compared below. Modules at different burnup or equipment health should carry different shares of the load, and unit-specific allocation has reduced control-rod movement [115]. A trip in one module must not spread, so each module keeps its own protection and local control. Fewer operators supervise more modules, and operators have asked for AI help in monitoring less-attended units but not in autonomous diagnosis or safety-critical control [116].

Operating evidence exists only at small module counts: plant-level testing of the two-module HTR-PM supports coordinated operation, while larger dispatch schemes remain conceptual or simulation based and interaction risk grows with module count [12]. Fleet operation extends coordination across sites, and a preliminary economic assessment indicates that centralized offsite control rooms supported by digital twins could serve multiple remote microreactors [13]. Because every nuclear controller or model update must be validated against the site licensing basis, fleet learning should feed an offline qualification pipeline and not direct online adaptation.

Three architectures can coordinate the modules, trading optimality against fault tolerance and scalability [117]. A centralized MPC or scheduler reaches the plant-wide optimum and handles shared-equipment constraints directly. It is also a single point of failure, and it grows and must be revalidated with every change in module configuration. Distributed MPC gives each module its own controller. Non-cooperative controllers settle at a compromise, cooperative ones approach the centralized solution at a higher communication cost, and both degrade gracefully when a module drops out. Hierarchical control places a slow plant-level scheduler above local controllers that track the allocated targets. HTR-PM shows

the state of practice. Its implemented coordinated control law for two modules feeding one turbine was verified in power-step tests at the plant [118]. A multivariable dynamic matrix controller coordinated parallel modules in simulation [119], and coordinated load following was developed for multi-modular cogeneration plants [120]. Centralized or hierarchical schemes remain tractable for two modules, and distributed MPC gains as module counts grow.

Integrated energy systems add the mission-level decisions at the top of the hierarchy. Dynamic models of nuclear and renewable systems producing electricity and hydrogen show that mission value depends on when and where energy is routed [8,9]. Economic MPC, stochastic MPC with forecasting, mixed-integer scheduling, and distributed optimization are established tools [32,34]. They remain appropriate provided mission-level optimization only allocates energy among products and issues bounded targets that lower layers realize under thermal, reactivity, and equipment constraints. A poor forecast then affects revenue without compromising safety. In the integrated-energy prototyping study, an MDP asset-management layer with one-day decision steps produced interpretable repair-versus-defer policies, deferring an intermediate-heat-exchanger fouling repair, for example, when a scheduled shutdown fell within roughly 110 days [11]. The formulations are mature in combined heat-and-power and microgrid practice; the nuclear gaps are validated thermal-cycling cost models and disciplined coupling of equipment health into dispatch.

Remote operation raises each of these supervisory questions in a harder form because continuous onsite staffing can consume a prohibitive share of a microreactor's revenue [13], so the technical program is local autonomy with intermittent, secure remote supervision. Criticality-search and load-following studies automate routine reactivity management at the simulation level [6,66], but protection, local regulation, and setpoint-hold logic must continue without connectivity or remote services. A secure remote-supervision demonstration for a 4.5 MWth lead-cooled fast-reactor simulator shows the minimum infrastructure: a lumped-parameter twin verified against SAM, communication over OPC Unified Architecture (OPC-UA) with certificate-based authentication and encryption, a rate-limiting reference governor on an industrial soft controller, and a digital-twin and GenAI operator interface [121]. The controlled system is a simplified lumped-parameter simulator, and command scope is bounded by the rate-limiting governor. Passive heat removal gives many advanced designs hours of grace time, but radiological consequences and weeks-long repair access raise the cost of a wrong action, so degraded-mode behavior such as reduced-power holds and restart after link loss belongs in the validated design basis. Cyber threats to remote supervision and the transition to local safe operation are treated in Section 5.4.

In all three contexts, coordination may improve mission performance, but each module must retain an independently verified path to stable local operation, and taking that path in time depends on the diagnosis and human supervision addressed next.

5.3. Diagnosis and Fault Response

Monitoring and diagnosis become part of control when they trigger sensor substitution, actuator reallocation, derating, or a mode change, the classical scope of fault-tolerant control [27]. At the simulation level, the integrated-energy architecture couples a physics-based diagnostic code to online risk updates and automated mode transitions after injected fouling and stuck-valve faults [11], a Generation-IV digital-twin framework applies RL to health-aware supervisory control [122], and a nearly autonomous management and control system assembles digital twins for diagnosis, strategy inference, and prognosis to recommend operator actions during a simulated advanced-reactor transient [123]. At the research-facility level, the AGN-201 digital twin supports remote monitoring and au-

tonomous anomaly detection without closing a control loop [124,125]. Detection therefore has stronger validation than control-consequential response.

Diagnostic confidence should determine the response. A single classifier output should not initiate a consequential maneuver without temporal confirmation, independent physics checks, and corroborating measurements, because a false alarm converts a monitoring error into an induced transient. The required evidence scales with the action: sensor substitution based on analytical redundancy can rest on high-fidelity simulation or testbed evidence, whereas letting remaining-useful-life estimates relax limits demands validated UQ and independent confirmation that the reviewed literature has not yet demonstrated.

5.4. Cybersecurity and Human–Automation Risks

Cybersecurity and human–automation interaction affect every layer reviewed above. Each layer adds data paths that can be corrupted through falsified or replayed data, injected commands, lost links, or tampered models [126], and interfaces at which operators can misread automation. The shared mitigation is that each advanced function can be withdrawn cleanly while local control, protection, and the operator keep the plant in a verified mode. Attacks specific to GenAI are quantified in Section 6.4.

Remote command paths widen the attack surface, so the local layer must remain safe under falsified telemetry, replayed commands, or stale data [29,127]. Under the zone-and-conduit model of industrial control security [128], independent protection and local regulation sit in the innermost zone and accept no remote commands. Telemetry leaves the site through a one-way path, and remote commands enter only through one authenticated and encrypted conduit that accepts a few typed requests, such as bounded setpoint changes. The local supervisor screens each remote request with the same governor and permissive logic that screen local requests. A compromised remote center can therefore request only actions that the site would accept from its own operators.

The transition to local safe operation should be deterministic and rehearsed. A heartbeat watchdog in the local supervisor declares loss of link when no authenticated message arrives within a set timeout. The supervisor then rejects remote requests and holds the last validated setpoints, so the transfer is bumpless. If the link is still absent after a second delay, the supervisor ramps the plant to a pre-analyzed reduced-power state. Remote supervision resumes only after re-authentication, a comparison of remote and local plant states, and operator approval. No step depends on the remote center, so the same sequence covers ordinary outages and denial-of-service attacks. The lead-cooled fast-reactor demonstration did not test loss of the link, and end-to-end tests under injected link loss remain open (Section 7.2).

Human supervision remains essential at the boundaries between modes, models, and automation levels, and the shift from manipulation to supervision has well-documented failure modes. Automation removes routine practice while leaving humans the rare and hardest interventions, so manual skill erodes where it is most needed [129]. Operators must maintain mode awareness across automated transitions, sustain vigilance through long low-workload periods, and calibrate trust to demonstrated reliability [130,131]; miscalibration harms performance through misuse under overtrust and disuse under distrust [132]. Aviation, process, and vehicle automation show recurring incidents rooted in hidden mode changes and automation surprise [16].

Two cognitive models explain these failure modes. Situation awareness has three levels: perception of the relevant plant elements, comprehension of their meaning, and projection of their near-term status [133]. Automation that hides its mode or data sources degrades the first two, and an operator who cannot project the next automated action cannot intervene in time. Trust is calibrated when reliance matches the actual capability of the automation, and it rests on observed performance, understanding of the process, and purpose [130].

Interfaces should therefore show what the automation is doing, why, and what it expects next. The testbed assistant cites live measurements and margins to configured limits [81], and the bounded-context monitor shows a status banner computed without the language model [134]. Operator trust was not measured in these studies, and operator trials remain future work (Section 7.2).

Operators must be able to see the active mode, the source and age of data, the limits applied by the supervisor, the action proposed, and the response that follows if approval is withheld or communication is lost. Planned automated actions should be previewed before execution, using the faster-than-real-time twins of Section 4. Degraded-mode behavior must be simple enough to predict under stress. Training should include automation surprise, misleading recommendations, stale twin states, and takeover from degraded modes, so that responsibility matches the level of automation granted. GenAI enters the plant at this interface, as an information and recommendation layer above the supervisory structure, and Section 6 examines what it can add there and under what conditions.

6. Generative-AI-Enabled Nuclear Control and Operation

Section 5 left operators supervising a layered plant in which protection, regulatory control, model-based optimization, and digital twins each hold bounded roles, and it identified assembling context from procedures, trends, alarms, and models as the operator's main burden. GenAI is relevant to nuclear operation primarily at that interface, as a language, knowledge, and tool-orchestration layer. Large language models (LLMs) can translate intent into structured tasks, retrieve dispersed information, and coordinate engineering software, but they do not inherently provide numerical correctness, stability, constraint satisfaction, or hard real-time behavior, so the surrounding architecture must supply those properties.

6.1. Capabilities and Grounded Architecture

The capabilities that matter for control and operation are a narrow subset of what foundation models offer [135]. Instruction tuning lets engineering intent be expressed in plain language [136]. Retrieval-augmented generation (RAG) attaches a searchable memory of version-controlled plant documents that can be updated and cited without retraining [137,138]. Tool-use patterns delegate arithmetic, data access, simulation, and optimization to deterministic functions, and interleaving reasoning with tool calls lets tool observations update subsequent action selection [139,140]. Structured output schemas make requests and results machine-checkable for downstream validation.

For plant use, these mechanisms should compose into a fixed chain: retrieve controlled sources, convert the request into a typed task, invoke only authorized tools, check numerical and logical consistency, and log every source, tool call, and decision for audit. Tool permissions follow least privilege, and no free-text output should reach an actuator. Two nuclear systems realize this chain. A lead-cooled fast reactor agent found that online tool invocation introduced unacceptable latency, so its physics diagnostics ran in parallel and fed the agent a structured state dictionary [121]. A bounded-context monitoring system placed tool governance in the host software, where in-flight and cooldown guards gated diagnostic dispatches without blocking the 1 Hz monitoring loop [134]. In both, the language model consumes verified numbers and proposes actions, while deterministic software decides what it may touch.

The chain exists because language competence and a control role are different things. Token prediction does not guarantee a mass balance, a feasible trajectory, or recognition of an out-of-domain state, and hallucinated technical content remains a documented failure mode [141]. Figure 6 therefore places deterministic computation, constraint checking, human approval, and independent protection between the language layer and the plant.

The essential property is that no path leads from the language model to an actuator without deterministic checks and an approval gate. An operator goal passes through controlled retrieval, schema-checked task generation, and least-privilege tool routing into a digital twin and into an MPC or RL optimizer from Sections 2 and 3. An independent constraint checker screens the candidate, the operator approves where required, and conventional controllers execute the setpoints. Independent protection senses the plant directly and depends on no generative or optimization layer.

The chain also fixes the response to a failed tool call, independent of the language model. A timeout, a schema or execution error, an out-of-range or stale value, and a candidate rejected by the constraint checker all receive the same deterministic response. The host software discards the result, retries a bounded number of times when the failure is transient, and otherwise reports the failed tool to the operator. No partial result is passed on, and the regulatory controllers keep the last verified setpoints through the hold path of Figure 6. A failed call can therefore delay a recommendation but cannot change a setpoint.

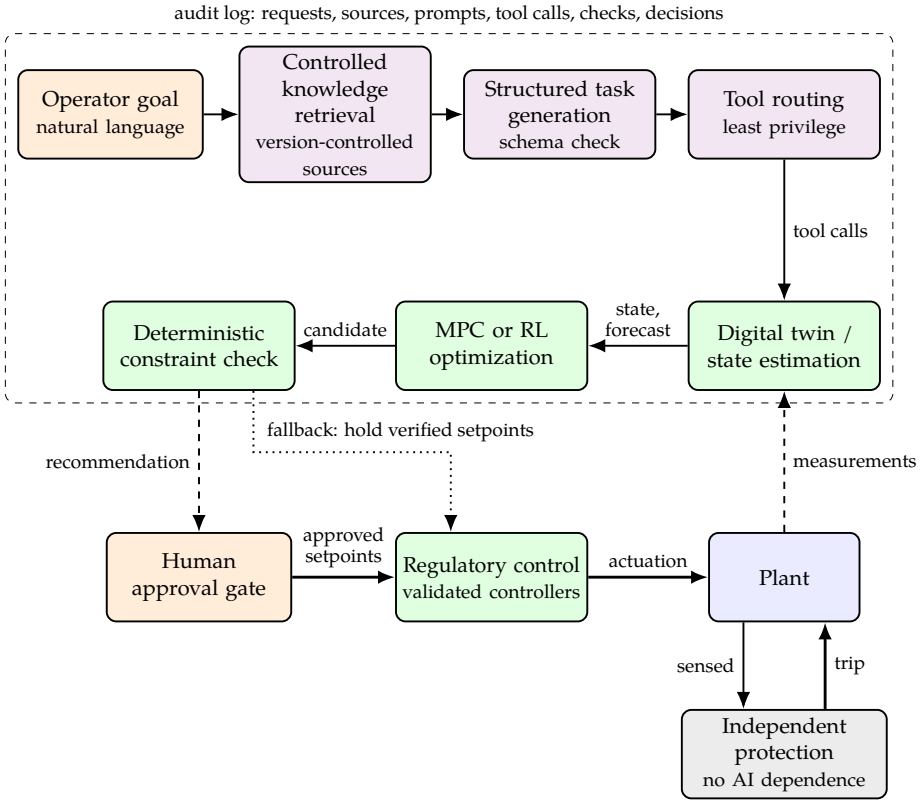


Figure 6. Grounded GenAI architecture in which deterministic tools, constraint checks, human approval, and independent protection bound automated action.

6.2. Operator and Engineering Support

Operator-facing uses share the purpose identified at the end of Section 5: reducing the effort required to assemble context. Figure 7 shows the pattern such assistants follow. The point of the figure is that the assistant reaches the plant only through tools and data, while the human keeps manual command. A user query carrying instructions, a description of the plant or testbed, and a summary of recent data and predictions enters an agentic system. Inside it, the LLM plans a response, a tool-calling agent turns the plan into actions, and an executor issues requests, with a knowledge graph supplying domain structure. Tools reach the digital twin and its control model, and the physical twin streams sensor data back. The resulting state update and output message return to the user, who retains manual command of the physical system. A thermal-fluid testbed assistant gave natural-

language access to live measurements and digital-twin forecasts, correctly correlating 31 simultaneous measurements and citing safety margins against configured limits while retaining an advisory-only role [81]. Human–AI interaction guidelines apply directly, since the assistant must state what it can do and how the operator can correct or dismiss it [142].

Sustained advisory operation raises further systems questions. A bounded-context monitor ran a locally deployed model as a persistent 1 Hz advisory layer for roughly 34 minutes over a replayed heater-runaway transient [134]. All 2224 consumed alert snapshots were classified as anomalies, 99.84% of the numeric tokens in anomaly explanations matched a source value in the same turn's context, and routine turns took a median of 462 ms, while anomaly explanations took a median of 1163 ms and often exceeded the one-second budget. These results quantify instruction-following fidelity under bounded memory. They do not establish independent detection skill, because a deterministic status banner displayed on every turn was part of the model input, and that distinction determines how such systems can be credited in a safety argument.

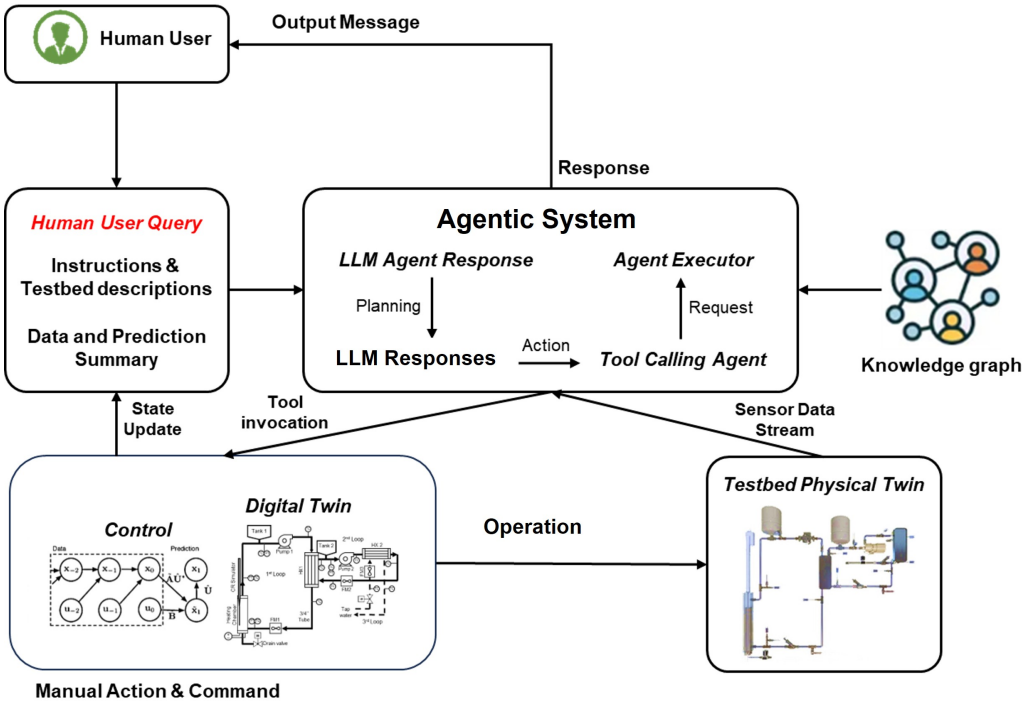


Figure 7. Overview of a GenAI-based agentic system in which a user query is planned and executed through tool calls to the digital twin and physical testbed, with the human retaining manual command.

Engineering automation is a lower-consequence entry point because its outputs are checked before they matter. AutoFLUKA embeds Monte Carlo domain knowledge in retrieval-grounded LLM agents that automate input generation, batch execution, and post-processing for radiation-transport workflows, resolving input and execution errors in under a minute against the days such queries traditionally wait on user forums [143]. A companion study automated data-driven modeling on a critical-heat-flux benchmark of roughly 25,000 experimental points. Agent-built models outperformed the standard lookup table on all blind test cases and matched expert-built Bayesian-optimized deep ensembles, and a supervisor-orchestrated multi-agent pipeline proved more reliable and token efficient than a single self-repairing agent [144]. In both cases conventional software provides the acceptance test, since inputs must parse, simulations must execute, and results must pass physics checks.

Multi-agent systems divide such workflows among domain, data, simulation, and verification roles [145,146]. An empirical taxonomy built from more than 1600 annotated traces across seven frameworks identified fourteen recurring failure modes, clustered into system design flaws, inter-agent misalignment, and weak verification or termination, and found that organizational design often limited multi-agent gains even when model capability was adequate [147]. In a plant, conflicting agent objectives and unclear decision responsibility during abnormal events are credible hazards, so multi-agent autonomy belongs in offline analysis campaigns while operational interfaces remain serialized and approval gated.

6.3. Nuclear Demonstrations

Nuclear-specific demonstrations now span simulation automation, knowledge assistance, digital-twin access, supervisory recommendation, and local monitoring. Most of the demonstrations discussed in this subsection come from the author's laboratory, and they are assessed by the same evidence criteria applied to independent work. Figure 8 shows the physical testbed and the communication, twin, and agent infrastructure on which the control-relevant ones were built. It shows that the language agent works at the remote end beside the digital twin, while PID control and interlocks remain on the local facility controllers. AutoFLUKA and the critical-heat-flux modeling agents cover the engineering workflows [143,144]. RADIANT-LLM proposes a local-first, multimodal, provenance-aware foundation-model environment for nuclear engineering, motivated by the data-control concerns that cloud-hosted models raise for plant information [148]. It targets a deployment constraint that the studies below flag as future work.

Among these demonstrations, AROMA-GPT is the most complete control-relevant example. It connects a retrieval-grounded agent to the 4.5 MWth lead-cooled fast-reactor twin of Section 5 through physics tools, a state estimator, a predictive rod planner, and a cladding-integrity check [121]. The twin's steady state was verified against SAM with deviations below 3%. The agent operates beside a rate-limiting reference governor, and its recommended control signals matched those the governor generated independently. Because the governor runs whether or not the agent responds, a failed tool call would remove advice without changing the control path, although the reported tests did not inject such failures. In a disturbance-rejection case, it recommended rod withdrawal to 86.18%, and the restored power settled at 3.06 MWth against a 3.0 MWth setpoint, a 2.0% offset attributed to its linearized tool model. It also flagged an inlet-temperature drift before the governor's hard constraints activated. A human vetted every recommendation. The reactor model was a simplified lumped-parameter Simulink model, the industrial-controller connection was software in the loop, and the acknowledged limitations were the cloud-hosted frozen model and the absence of repeated-trial statistics.

The strongest validation comes from the physical testbed of Figure 8. The three-loop platform integrates a calibrated SAM twin, a recurrent surrogate, and a context-engineered assistant over OPC-UA. The twin tracks a multi-step power transient with a primary-loop outlet mean absolute error of 0.667 K, the surrogate generates a 300 s forecast in 1.45 s, and the assistant grounds its guidance in both live sensors and predicted states while remaining stateless and never actuating the facility [81]. The bounded-context monitor extends the same facility class to persistent operation with a local model and audited session-state management [134], so the two systems bracket the current design space, one stateless and cloud based, one persistent and local.

Publication status and scope also bound the evidence. AutoFLUKA, the testbed platform, AROMA-GPT, the modeling-agent study, and RADIANT-LLM are peer-reviewed journal articles, whereas the bounded-context monitor is a conference paper. None of these demonstrations involves an operating reactor, and single scenarios and facilities do not yet

support generalization to fleet practice. Because these systems share one laboratory and one facility class, independent replication elsewhere is also still needed. Their strongest contribution is architectural: language models can be placed beside validated computation without becoming the source of safety claims.

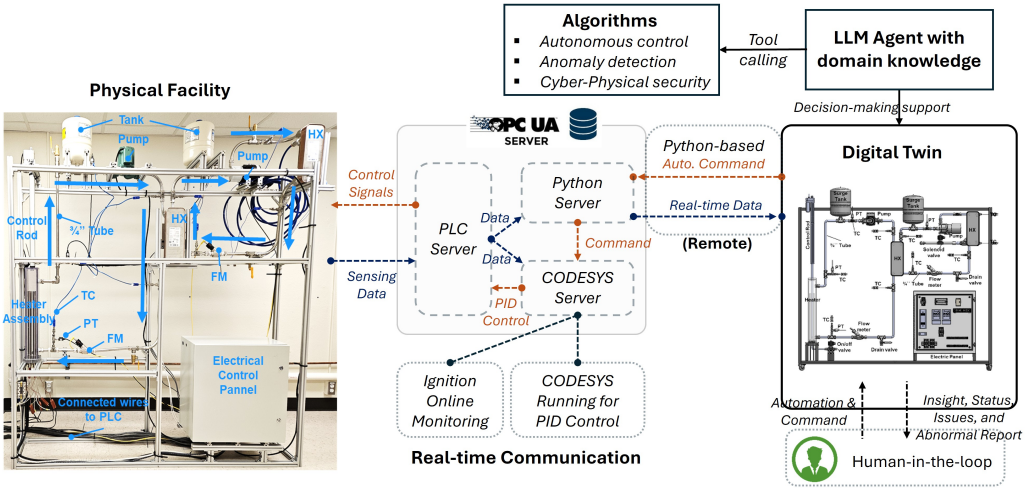


Figure 8. The thermal-fluid testbed developed in the author's laboratory, showing the physical facility, the OPC-UA real-time communication layer, the remote digital twin, and the tool-calling LLM agent with a human in the loop.

Independent groups report related demonstrations, so far in simulation or on text benchmarks. A tool-using LLM agent handled a boron concentration anomaly in a reactor simulator [149], and a compact language model selected reactor control actions validated in closed-loop simulation [150]. Both act on the simulated plant, which the laboratory systems above do not. AtomicGPT, adapted to nuclear text for on-premises use, addresses the data-control concern behind RADIANT-LLM [151]. None of these studies has yet used physical hardware or measured plant data.

6.4. Limits on Automated Action and Evaluation

The reviewed studies support GenAI for language interaction, knowledge synthesis with provenance, tool selection, planning under supervision, and code and input generation. They do not support claims of numerical accuracy, closed-loop stability, constraint satisfaction, hard real-time latency, reproducibility, calibrated confidence, or reliable self-assessment [141,152]. These gaps are documented across model generations and should be assumed for any model not evaluated for the role. On the automation scale of Section 5, near-term roles should remain at the inform and recommend levels: engineering copilots, procedure retrieval, and approval-gated preparation of supervisory objectives. Bounded automatic action, backed by conventional controllers, should be considered only after model output has passed the deterministic schema, tool, constraint, and runtime-monitoring stages above. Direct generative actuator control has no current technical basis for deployment.

Evaluation should match the role and should not rely on generic language benchmarks. A retrieval assistant should be tested for source fidelity and refusal when evidence is missing, a tool-using agent for tool selection and recovery from failed calls, and an operator assistant additionally for latency, workload, and out-of-domain recognition. Holistic multi-metric evaluation [152], cost-controlled agent benchmarks with genuine holdouts [153], and failure-mode taxonomies usable as annotation instruments [147] support this shift, and domain-curated scientific benchmarks are markedly harder for frontier models than generic ones, which argues for nuclear-specific test suites [154].

These requirements translate into measurable criteria. Retrieval accuracy is the share of answers supported by the retrieved controlled sources, and the unsupported-response rate is the share of claims, including numbers, that no source supports [155]. Tool-selection accuracy is the share of requests that call the correct tool with correct arguments. Response time is reported as median and tail latency against the operating deadline, and cybersecurity as the success rate of prompt-injection attacks, including those carried by retrieved content [156]. Nuclear values are still sparse. A nuclear engineering benchmark reported exam accuracy up to 95% [157], and the bounded-context monitor reported 99.84% numeric consistency with median latencies of 462 ms and 1163 ms against a 1 s budget [134]. In a simulated control room, adaptive attacks caused the loss of a critical safety function in 8.7% to 12.1% of sessions [158]. A failed criterion, such as a missing source, an unsupported number, a missed deadline, or a security alarm, withdraws the GenAI function through the deterministic response of Section 6.1, which leaves the regulatory controllers on the last verified setpoints. The cyber risks that GenAI shares with the other layers are gathered, with their mitigations, in Section 5.4.

Documentation should record the intended use, model and retrieval versions, evaluated scenarios, and failure modes so that decisions on the level of automation inherit an explicit evidence trail [159,160]. Fluency can improve access to engineering capability, but it cannot substitute for the validated models and controllers that make a plant action safe. Section 7 asks what would move these technologies from demonstration toward deployment.

7. Research Gaps and Staged Roadmap

Across the five technologies reviewed, algorithmic capability has outpaced deployment evidence. Most control studies in the reviewed literature remain in simulation, few digital twins are validated against the physical systems they represent, and GenAI has reached representative physical testbeds only in isolated cases.

Table 3 supports this assessment with 21 representative entries from 2017 to 2026. In the table, automatic means closed-loop action without approval of each action. Automatic operation at an operating plant appears only for the classical coordinated control of the two HTR-PM modules. Advanced controllers reach hardware-in-the-loop testing at best, in an MPC embedded in a field-programmable gate array (FPGA) for a sodium fast reactor and in real-time tests of a microreactor automated control system. Digital twins reach measured facility data and one research reactor, and none of them closes a control loop there. Outside simulation, GenAI functions stay at the inform and recommend levels, and only two simulator studies let a language agent execute control actions. Robustness evidence is thinner still. Only seven of the 20 implemented entries report tests under faults, upsets, model mismatch, or attacks, and each covers one or a few scenarios, as described with each study in Sections 2–4. The only operating-plant tests, at HTR-PM, included turbine and reactor trips [12]. The other entries report nominal operation, disturbance rejection, or an abnormal transient that is itself the task. Repeated trials that report worst-case constraint violation and recovery time are rare, and Section 7.1 proposes them as benchmark measures.

In the author's view, the most useful next step is a sequence of shared experiments that test models, interfaces, operators, and safe-mode reversion as one decision system. What follows is offered as a perspective on where the gaps lie and how they might be closed.

7.1. Common Benchmarks and Credible Digital Twins

The first gap is that MPC, RL, digital-twin, and GenAI studies are evaluated on plant models and scenarios typically chosen by the teams that designed the methods, so reported performance is difficult to compare or evaluate objectively. Accuracy-only leaderboards in adjacent fields reward needlessly complex systems, and inadequate holdouts produce

gains that do not transfer [153]. Holistic multi-scenario evaluation [152] and domain-expert benchmarks that prove harder than generic tests [154] offer templates the nuclear community could adapt. A nuclear control benchmark would need to fix the plant configuration, uncertainty ranges, hard constraints, and baseline controllers. Its scenarios would cover normal maneuvers, model mismatch, and degraded sensors. Reporting would move beyond average tracking error toward worst-case constraint violation, recovery time, and backup activation frequency, with reproduction by independent groups. A small set of well-instrumented benchmark problems with controlled holdouts would likely be more useful than any generic leaderboard.

Table 3. Representative studies by technology, plant, level of automation exercised, and evidence stage.

Study	Technology	Plant or Facility	Level of Automation	Evidence Stage
Model-based and supervisory control				
Wood et al. [14]	Nearly autonomous control framework	Advanced reactors	Automatic (concept)	Conceptual
Dong et al. [12], Jiang et al. [118]	Coordinated plant control law	HTR-PM, two HTGR modules	Automatic	Operating plant
da S. C. Pereira et al. [39]	Multivariable MPC on an FPGA	SFR primary system	Automatic	HIL (FPGA in the loop)
McConnell et al. [161]	Automated control hardware with PID loops	MARVEL and MAGNET models	Automatic	HIL (real time)
Dave et al. [10], Ponciroli et al. [11]	Reference governor and mode-decision layer	Pebble-bed FHR with salt storage	Automatic	Simulation
Oncken et al. [23]	Adaptive MPC with learned predictor	Heat-pipe microreactor model	Automatic	High-fidelity simulation
Abdulraheem et al. [7]	PID, sliding-mode, and nonlinear MPC	Drum-controlled microreactor	Automatic	Simplified-model simulation
Learning-enabled control				
Tunkle et al. [66]	Single- and multi-agent PPO	Drum-controlled microreactor	Automatic	Simulation
Lee et al. [54]	PPO with task-analysis action space	SMR power increase	Automatic	Simulation
Yoo et al. [67]	Self-healing control with autoencoder and RL	Generic PWR full-scope simulator	Automatic	High-fidelity simulation
Kim et al. [57]	Autonomous emergency operation system	Nuclear power plant	Automatic	Simulation (prototype)
Digital twins				
Song et al. [78]	Online calibration of a fast twin by machine learning	PKL III integral test facility	Inform	Measured test data, replayed
Almukhametov et al. [70]	Graph neural ODE twin	Three-loop facility	Inform	Physical testbed data
Liu et al. [92]	Whole-system graph twin	EBR-II and generic FHR	Inform	High-fidelity simulation
Stewart et al. [124]	Remote-monitoring twin	AGN-201 research reactor	Inform	Research facility
Lin et al. [123]	Nearly autonomous management and control	Advanced reactor	Recommend	Simulation
Generative AI				
Lim et al. [81]	Grounded LLM assistant with SAM twin	Three-loop facility	Inform	Physical testbed
Garimidi et al. [134]	Bounded-context local LLM monitor	Three-loop facility	Inform	Physical testbed data, replayed
Ndum Ndum et al. [121]	LLM agent beside a reference governor	4.5 MWth LFR model	Recommend	SIL
Lee et al. [149]	Tool-using LLM agent	Reactor simulator	Automatic (simulator)	Simulation
Lee et al. [150]	Compact language model selecting control actions	Reactor simulator	Automatic (simulator)	Simulation

The twin itself belongs in the benchmark, because twin credibility becomes a system-level bottleneck once controllers and agents act on its predictions. Continuous residual monitoring, configuration synchronization, and explicit model discrepancy appear to be the minimum before twin outputs may influence constraints [18,75], and the partial-observability forecasters of Section 4 still lack long-duration validation against drifting facilities [70]. Rare-event coverage is the second unresolved twin problem: controllers intended for fault

response cannot be trained or evaluated only on the benign transients an operating plant provides, so rare-event data should come from system codes, generative trajectory models, and physical experiments, and be checked against an independent model or experiment. Instrumented testbeds coupled to system-code twins, such as the SAM-linked thermal-fluid platform, are a practical bridge because faults can be exercised repeatedly [81].

7.2. Safe Integration, Human Supervision, and Cyber-Resilient Deployment

Given shared benchmarks and credible twins, the hardest open questions occur at the interfaces between methods. Natural-language goals must become typed optimization tasks; RL proposals must pass deterministic safety filters of the kind described in Section 3 [60,61]; twin uncertainty must tighten control envelopes online so that a degrading twin lowers the level of automation of the controllers that depend on it; and runtime assurance must arbitrate among intelligent controllers and verified backup controllers. Component-level correctness does not compose into system-level safety: systems-theoretic accident analysis attributes most losses in complex systems to flawed interactions among individually correct components [162], and the multi-agent failure taxonomy of Section 6 locates failures in the same place [147]. The entire chain from sensing through optimization, approval, execution, and recovery should therefore be tested as a unit, with stale data, conflicting objectives, and failed tools injected deliberately.

Human supervision of adaptive automation remains the least quantified element of these architectures. The open questions follow from Section 5: how functions are allocated between crews and automation, how multi-module and remote plants are staffed when onsite crews shrink, and how operator trust stays calibrated and manual skill is retained as routine actions are automated. Because out-of-the-loop performance loss grows with automation level [16], human-in-the-loop experiments should focus on intervention quality more than on nominal task completion, with scenarios that include hidden model drift and plausible but wrong recommendations, and with measures of whether operators recognized the active automation level and intervened in time. The human-factors review structure for control-room modernization offers a framework for such campaigns [163]. Organizational questions belong in the same program: responsibility for AI-generated recommendations, approval of model updates, and change control for prompts and knowledge bases.

Deployment infrastructure needs direct study because most GenAI demonstrations assume cloud inference and unconstrained connectivity. Nuclear implementations will more likely require local inference, network segmentation, signed model versions, and continued operation during disconnection, consistent with operational-technology security practice [164] and nuclear computer-security guidance [165]. Early evidence that bounded-context local agents can sustain continuous monitoring without cloud dependence suggests the constraint is tractable [134], although long-duration reliability under plant workloads remains open. Cyber-physical tests should inject corrupted sensors, poisoned retrieval sources, malicious prompts, and loss of the remote link, drawing on secure-control theory for adversaries who understand plant dynamics [29,127,166]. Success in such tests means that the plant retains a verified local mode and the higher-level function is withdrawn cleanly rather than failing unpredictably.

7.3. Stage-Gated Demonstrations and Qualification

Figure 9 organizes this agenda into three readiness stages defined by the level of automation a function has earned. The main message is that a function moves to a higher level of automation only after it passes the evidence gate between stages. In the near term, functions inform and recommend but do not act: engineering copilots, simulation automation, advisory twins, HIL controller testing, and bounded setpoint recommendations, all of

which can mature on today's evidence base. In the intermediate term, approval-gated and supervised action becomes admissible once independent validation and SIL/HIL demonstration have been reproduced; supervised MPC and safety-filtered learning then move onto representative facilities or research reactors, together with multi-module coordination and human-in-the-loop trials. In the long term, bounded automatic action for selected missions, such as coordinated microreactor or SMR fleets, becomes admissible once physical demonstration, human-in-the-loop evaluation, and long-duration reliability evidence have accumulated. Generative models act through tool calls at every stage, so even in the long term their proposals reach the plant only as typed setpoints filtered by validated physics models, deterministic checks, and hard limits.

Progression between stages occurs only when the corresponding evidence gate is passed. One practical sequence of demonstrations runs: common simulation benchmark; independent reproduction; uncertainty and adversarial testing; SIL; HIL; representative thermal-fluid facility; research-reactor shadow mode; human-in-the-loop evaluation; bounded supervised operation; long-duration reliability trial; and a regulator-facing evidence case. A function that fails a gate need not halt the program; it reverts to the highest automation level its evidence supports, keeping the roadmap incremental and reversible. Acceptance at each gate uses the benchmark measures of Section 7.1, such as worst-case constraint violation and recovery time, and for GenAI functions the criteria of Section 6.4, with numeric thresholds set in each application's safety analysis. HTR-PM coordinated control has completed this path, from verification on a full-scale simulator to tests at the operating plant [12,118,167].

Qualification effort should be graded by the level of automation of the function: a document assistant, a setpoint recommender, and a direct controller should not require identical evidence. For each class, the record would link intended use, provenance, verification results, uncertainty limits, human responsibilities, and post-deployment monitoring into one auditable case, reusing datasheet and model-card conventions [159,160]. For digital twins and learned components, the three update tiers of Section 4.4 map onto the configuration management that nuclear safety-system software already follows [168]. Current IAEA, NIST, and U.S. Nuclear Regulatory Commission guidance frames AI deployment around application risk and human oversight [169–172], but these documents do not yet specify how adaptive or foundation-model-based functions should be qualified for control-relevant roles.

The levels of automation used in this review can be read against two recent U.S. regulatory developments. The NRC AI Strategic Plan describes notional autonomy levels for AI in nuclear activities [172]. At the insight level a machine assists human decisions, at the collaboration level, a human vets and carries out machine recommendations, and at the operation level, a human supervises machine decisions. Regulatory scrutiny is expected to rise with the level. The inform role of this review corresponds to insight, the recommend and approval-gated roles to collaboration, and bounded automatic action to operation. No reviewed architecture proposes full autonomy. The risk-informed, technology-inclusive framework of 10 CFR Part 53 grades the treatment of plant equipment by safety significance derived from risk evaluations [173]. It also requires an explicit allocation of safety functions among human action, automation, and passive or inherent features, and it gives staffing flexibility to plants whose safety functions need no human action. The layered architecture reviewed here fits this approach. When safety rests on independent protection and inherent features, advanced controllers, digital twins, and GenAI tools could be classified as not safety significant. The evidence stages of Section 1.3 also parallel the analyses, tests, prototypes, and operating experience through which Part 53 expects design features to be demonstrated.

Early engagement among researchers, vendors, operators, and regulators on what such packages should contain is, in the author's view, an important near-term step. It would let the demonstrations above generate the right artifacts from the outset.

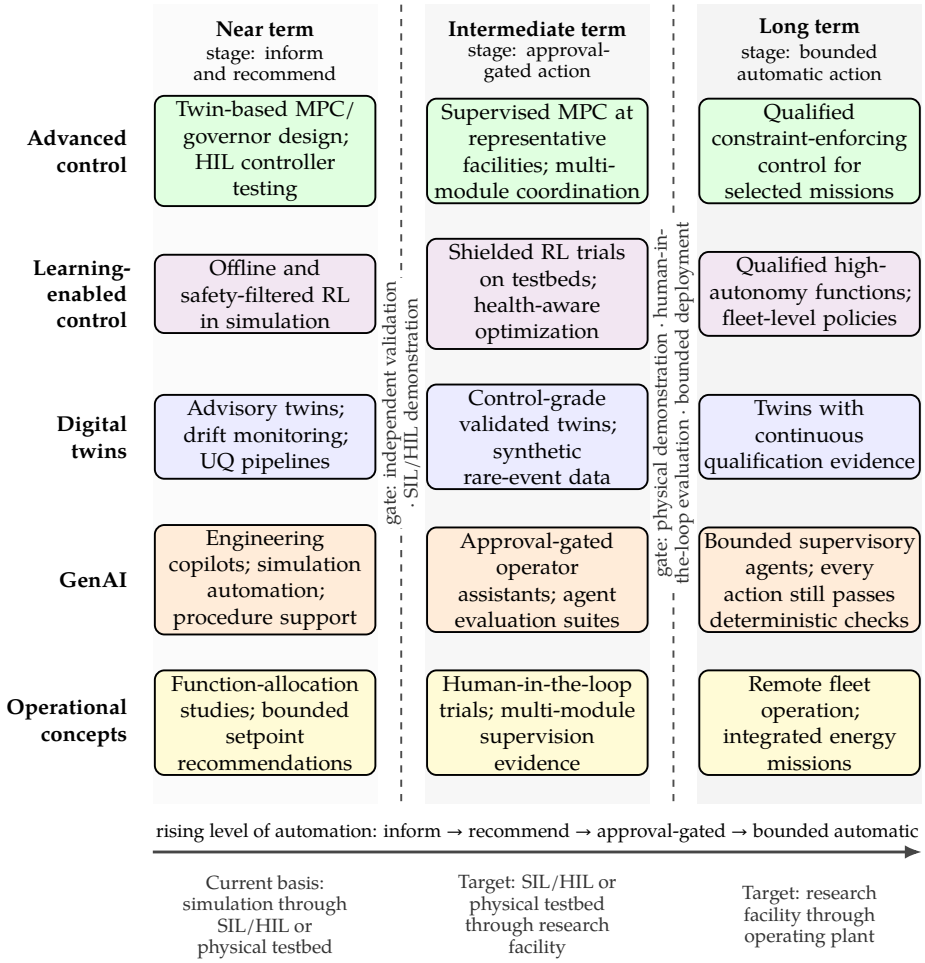


Figure 9. Stage-gated roadmap linking increasing levels of automation to progressively stronger demonstrations.

8. Conclusions

This review set out to determine what level of automation the current evidence supports for advanced control, digital twins, and GenAI in advanced nuclear systems, and what demonstrations would justify raising it. It answered the four questions of Section 1.3 for each technology, with the evidence organized by level of automation and by evidence stage. The practical implication and research directions that follow the findings are the author's perspective.

Three main findings follow from the evidence in Table 3. First, each technology has a distinct and bounded role. MPC is the clearest extension for constrained multivariable operation above the existing loops, and reinforcement learning fits offline-trained supervisory decisions behind verified safety filters. Digital twins supply state estimates and predictions whose credibility must be monitored, and GenAI is most useful as a grounded interface to validated tools. Second, capability has outpaced evidence. Advanced controllers reach hardware-in-the-loop testing at best, automatic operation at an operating plant remains classical, and GenAI stays at the inform and recommend levels outside simulation. Third, robustness under faults, model mismatch, and attacks is seldom tested, and most studies report nominal performance.

The practical implication is that autonomy should be extended one function at a time. Each function should earn its level of automation through a documented evidence record, while independent protection and qualified regulatory control remain unchanged. Advisory tools can be introduced now, because their errors are recoverable. Supervisory functions need independent validation, SIL and HIL testing, and human-in-the-loop trials before they act.

Three research directions would supply the demonstrations that higher levels of automation require. The first is shared benchmarks and representative facilities on which complete control chains are tested under faults, mismatch, and attack. The second is qualification methods for learned and generative components. The third is human-in-the-loop evidence on workload, trust, and intervention. Progress on these fronts, more than larger models, will decide how far autonomy can responsibly extend in advanced nuclear plants.

Funding: This work was supported by the U.S. Department of Energy, Office of Nuclear Energy, Distinguished Early Career Program under contract number DE-NE0009468, and by the Texas A&M Institute of Data Science Thematic Labs Program.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: The author gratefully acknowledges the support of the U.S. Department of Energy Office of Nuclear Energy Distinguished Early Career Program and the Texas A&M Institute of Data Science Thematic Labs Program.

Conflicts of Interest: The author declares no conflicts of interest.

Abbreviations

AI	Artificial intelligence	NIST	National Institute of Standards and Technology
CFR	Code of Federal Regulations	NRC	U.S. Nuclear Regulatory Commission
DPC	Differentiable predictive control	ODE	Ordinary differential equation
FHR	Fluoride-salt-cooled high-temperature reactor	OPC-UA	OPC Unified Architecture
FPGA	Field-programmable gate array	PI	Proportional–integral
GenAI	Generative artificial intelligence	PID	Proportional–integral–derivative
HIL	Hardware-in-the-loop	PPO	Proximal policy optimization
HTGR	High-temperature gas-cooled reactor	PWR	Pressurized-water reactor
HTR-PM	High-temperature gas-cooled reactor pebble-bed module	RAG	Retrieval-augmented generation
IAEA	International Atomic Energy Agency	RL	Reinforcement learning
LFR	Lead-cooled fast reactor	SAM	System Analysis Module
LLM	Large language model	SFR	Sodium-cooled fast reactor
LWR	Light-water reactor	SIL	Software-in-the-loop
MDP	Markov decision process	SMR	Small modular reactor
ML	Machine learning	UQ	Uncertainty quantification
MPC	Model predictive control		

References

1. Kaur, D.; Uslu, S.; Rittichier, K.J.; Durresi, A. Trustworthy Artificial Intelligence: A Review. *ACM Comput. Surv.* **2022**, *55*, 39. [CrossRef]
2. Perez-Cerrolaza, J.; Abella, J.; Borg, M.; Donzella, C.; Cerquides, J.; Cazorla, F.J.; Englund, C.; Tauber, M.; Nikolakopoulos, G.; Flores, J.L. Artificial Intelligence for Safety-Critical Systems in Industrial and Transportation Domains: A Survey. *ACM Comput. Surv.* **2024**, *56*, 176. [CrossRef]
3. Abdelrahman Ibrahim, A.; Lim, H.K. A Deterministic Assurance Framework for Licensable Explainable AI Grid-Interactive Nuclear Control. *Energies* **2025**, *18*, 6268. [CrossRef]
4. Dong, Z.; Cheng, Z.; Zhu, Y.; Huang, X.; Dong, Y.; Zhang, Z. Review on the Recent Progress in Nuclear Plant Dynamical Modeling and Control. *Energies* **2023**, *16*, 1443. [CrossRef]

5. Li, G.; Wang, X.; Liang, B.; Li, X.; Zhang, B.; Zou, Y. Modeling and Control of Nuclear Reactor Cores for Electricity Generation: A Review of Advanced Technologies. *Renew. Sustain. Energy Rev.* **2016**, *60*, 116–128. [CrossRef]
6. Radaideh, M.I.; Tunkle, L.; Price, D.; Abdulraheem, K.; Lin, L.; Elias, M. Multistep Criticality Search and Power Shaping in Nuclear Microreactors with Deep Reinforcement Learning. *Nucl. Sci. Eng.* **2026**, *200*, S309–S321. [CrossRef]
7. Abdulraheem, K.K.; Choi, S.; Shen, Q.; Kochunas, B.; Radaideh, M.I. A Load Following Reactivity Control System for Nuclear Microreactors. *Prog. Nucl. Energy* **2025**, *184*, 105676. [CrossRef]
8. Bragg-Sitton, S.M.; Boardman, R.; Rabiti, C. Reimagining Future Energy Systems: Overview of the US Program to Maximize Energy Utilization via Integrated Nuclear–Renewable Energy Systems. *Int. J. Energy Res.* **2020**, *44*, 8156–8169. [CrossRef]
9. Jacob, R.A.; Zhang, J. Modeling and Control of Nuclear–Renewable Integrated Energy Systems: Dynamic System Model for Green Electricity and Hydrogen Production. *J. Renew. Sustain. Energy* **2023**, *15*, 046302. [CrossRef]
10. Dave, A.J.; Lee, T.; Ponciroli, R.; Vilim, R.B. Design of a Supervisory Control System for Autonomous Operation of Advanced Reactors. *Ann. Nucl. Energy* **2023**, *182*, 109593. [CrossRef]
11. Ponciroli, R.; Moiseytseva, V.; Dave, A.J.; Nguyen, T.N.; Wang, H.; Vilim, R.B.; Kochunas, B.; Cilliers, A. Design and Prototyping of Advanced Control Systems for Advanced Reactors Operating in the Future Electric Grid: Final Report; Technical Report ANL/NSE-24/40; Argonne National Laboratory: Lemont, IL, USA, 2024
12. Dong, Z.; Zhang, Z.; Dong, Y.; Shi, L.; Huang, X.; Zhu, Y.; Jiang, D. Testing the Feasibility of Multi-Modular Design in an HTR-PM Nuclear Plant. *Nat. Commun.* **2025**, *16*, 2778. [CrossRef] [PubMed]
13. Bryan, H.C.; Jesse, K.W.; Miller, C.A.; Browning, J.M. Remote Nuclear Microreactors: A Preliminary Economic Evaluation of Digital Twins and Centralized Offsite Control. *Front. Nucl. Eng.* **2023**, *2*, 1293908. [CrossRef]
14. Wood, R.T.; Upadhyaya, B.R.; Floyd, D.C. An Autonomous Control Framework for Advanced Reactors. *Nucl. Eng. Technol.* **2017**, *49*, 896–904. [CrossRef]
15. Parasuraman, R.; Sheridan, T.B.; Wickens, C.D. A Model for Types and Levels of Human Interaction with Automation. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2000**, *30*, 286–297. [CrossRef] [PubMed]
16. Endsley, M.R. From Here to Autonomy: Lessons Learned from Human–Automation Research. *Hum. Factors* **2017**, *59*, 5–27. [CrossRef] [PubMed]
17. Kochunas, B.; Huan, X. Digital Twin Concepts with Uncertainty for Nuclear Power Applications. *Energies* **2021**, *14*, 4235. [CrossRef]
18. Mendoza, H.; Yadav, V.; Alam, S.B.; Eskins, D.; Iyengar, R. Advances in Digital Twins and AI/ML for Condition Monitoring in Nuclear Applications. *Front. Energy Res.* **2026**, *14*, 1716514. [CrossRef]
19. Astrom, K.J.; Murray, R.M. *Feedback Systems: An Introduction for Scientists and Engineers*; Princeton University Press: Princeton, NJ, USA, 2008.
20. Zhou, K.; Doyle, J.C.; Glover, K. *Robust and Optimal Control*; Prentice Hall: Upper Saddle River, NJ, USA, 1996.
21. Yadav, D.K.; Gupta, A.; Munshi, P. Design of NDI-SMC Based Robust Hybrid Nonlinear Controller for Load Following Operation in Pressurized Water Reactor. *Nucl. Eng. Des.* **2020**, *363*, 110604. [CrossRef]
22. Naimi, A.; Deng, J.; Vajpayee, V.; Becerra, V.M.; Shimjith, S.R.; Arul, A.J. Nonlinear Model Predictive Control Using Feedback Linearization for a Pressurized Water Nuclear Power Plant. *IEEE Access* **2022**, *10*, 16544–16555. [CrossRef]
23. Oncken, J.; Lin, L.; Agarwal, V. Adaptive Model Predictive Control for Heat Pipe–Cooled Microreactors Under Normal and Heat Pipe Failure Conditions. *Nucl. Technol.* **2024**, *210*, 2274–2289. [CrossRef]
24. Xing, X.; Burdet, E.; Si, W.; Yang, C.; Li, Y. Impedance Learning for Human-Guided Robots in Contact with Unknown Environments. *IEEE Trans. Robot.* **2023**, *39*, 3705–3721. [CrossRef]
25. Qin, S.J.; Badgwell, T.A. A Survey of Industrial Model Predictive Control Technology. *Control Eng. Pract.* **2003**, *11*, 733–764. [CrossRef]
26. Rawlings, J.B.; Mayne, D.Q.; Diehl, M.M. *Model Predictive Control: Theory, Computation, and Design*, 2nd ed.; Nob Hill Publishing: Madison, WI, USA, 2017.
27. Blanke, M.; Kinnaert, M.; Lunze, J.; Staroswiecki, M. *Diagnosis and Fault-Tolerant Control*, 3rd ed.; Springer: Berlin, Germany, 2016. [CrossRef]
28. Zhang, Y.; Jiang, J. Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems. *Annu. Rev. Control* **2008**, *32*, 229–252. [CrossRef]
29. Teixeira, A.; Shames, I.; Sandberg, H.; Johansson, K.H. A Secure Control Framework for Resource-Limited Adversaries. *Automatica* **2015**, *51*, 135–148. [CrossRef]
30. Garcia, C.E.; Prett, D.M.; Morari, M. Model Predictive Control: Theory and Practice—A Survey. *Automatica* **1989**, *25*, 335–348. [CrossRef]
31. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O.M. Constrained Model Predictive Control: Stability and Optimality. *Automatica* **2000**, *36*, 789–814. [CrossRef]

32. Ellis, M.; Durand, H.; Christofides, P.D. A Tutorial Review of Economic Model Predictive Control Methods. *J. Process Control* **2014**, *24*, 1156–1178. [CrossRef]
33. Mayne, D.Q.; Seron, M.M.; Rakovic, S.V. Robust Model Predictive Control of Constrained Linear Systems with Bounded Disturbances. *Automatica* **2005**, *41*, 219–224. [CrossRef]
34. Mesbah, A. Stochastic Model Predictive Control: An Overview and Perspectives for Future Research. *IEEE Control Syst. Mag.* **2016**, *36*, 30–44. [CrossRef]
35. Bemporad, A.; Morari, M.; Dua, V.; Pistikopoulos, E.N. The Explicit Linear Quadratic Regulator for Constrained Systems. *Automatica* **2002**, *38*, 3–20. [CrossRef]
36. Borrelli, F.; Bemporad, A.; Morari, M. *Predictive Control for Linear and Hybrid Systems*; Cambridge University Press: Cambridge, UK, 2017. [CrossRef]
37. Na, M.G.; Jung, D.W.; Shin, S.H.; Lee, S.M.; Lee, Y.J.; Jang, J.W.; Lee, K.B. Design of a Nuclear Reactor Controller Using a Model Predictive Control Method. *KSME Int. J.* **2004**, *18*, 2080–2094. [CrossRef]
38. Lin, L.; Oncken, J.; Agarwal, V.; Permann, C.; Gribok, A.; McJunkin, T.; Eggers, S.; Boring, R. Development and Assessment of a Model Predictive Controller Enabling Anticipatory Control Strategies for a Heat-Pipe System. *Prog. Nucl. Energy* **2023**, *156*, 104527. [CrossRef]
39. da S. C. Pereira, L.F.; Batista, E.A.; Pinto, J.O.P.; Upadhyaya, B.R.; Hines, J.W.; Coble, J.B. Model Predictive Control for Sodium Fast Reactors Based on Laguerre Functions and FPGA-in-the-Loop Environment. *Nucl. Eng. Des.* **2022**, *400*, 112041. [CrossRef]
40. Benner, P.; Gugercin, S.; Willcox, K. A Survey of Projection-Based Model Reduction Methods for Parametric Dynamical Systems. *SIAM Rev.* **2015**, *57*, 483–531. [CrossRef]
41. Korda, M.; Mezic, I. Linear Predictors for Nonlinear Dynamical Systems: Koopman Operator Meets Model Predictive Control. *Automatica* **2018**, *93*, 149–160. [CrossRef]
42. Lim, D.; Almukhametov, A.; Young, C.; Liu, Y. Autonomous Control of Thermal-Fluid Dynamics via Digital Twin–Enabled Neural ODE Models for Small-Scale Advanced Reactors. *Nucl. Sci. Eng.* **2026**, *200*, 1–20. [CrossRef]
43. Drgoňa, J.; Kis, K.; Tuor, A.R.; Vrabie, D.L.; Klaučo, M. Differentiable Predictive Control: Deep Learning Alternative to Explicit Model Predictive Control for Unknown Nonlinear Systems. *J. Process Control* **2022**, *116*, 80–92. [CrossRef]
44. Hu, R. A Fully-Implicit High-Order System Thermal-Hydraulics Model for Advanced Non-LWR Safety Analyses. *Ann. Nucl. Energy* **2017**, *101*, 174–181. [CrossRef]
45. Hu, R.; Zou, L.; O'Grady, D.; Mui, T.; Ooi, Z.J.; Hu, G.; Cervi, E.; Yang, G.; Andrs, D.; Lindsay, A.; et al. SAM: A Modern System Code for Advanced Non-LWR Safety Analysis. *Nucl. Technol.* **2025**, *211*, 1883–1902. [CrossRef]
46. Fortier, L.; Cetiner, S. Autonomous Operations for Advanced Reactors Utilizing Supervisory Control. *Trans. Am. Nucl. Soc.* **2024**, *131*, 306–309. [CrossRef]
47. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
48. Altman, E. *Constrained Markov Decision Processes*; Chapman and Hall/CRC: Boca Raton, FL, USA, 1999. [CrossRef]
49. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 1861–1870.
50. Levine, S.; Kumar, A.; Tucker, G.; Fu, J. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *arXiv* **2020**, arXiv:2005.01643. [CrossRef]
51. Moerland, T.M.; Broekens, J.; Plaat, A.; Jonker, C.M. Model-Based Reinforcement Learning: A Survey. *Found. Trends Mach. Learn.* **2023**, *16*, 1–118. [CrossRef]
52. Kumar, A.; Zhou, A.; Tucker, G.; Levine, S. Conservative Q-Learning for Offline Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; Volume 33, pp. 1179–1191.
53. Fujimoto, S.; Gu, S.S. A Minimalist Approach to Offline Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021; Volume 34, pp. 20132–20145.
54. Lee, H.J.; Lee, D.; Kim, J. Automating Power-Increase Operation for Small Modular Reactors Based on Task Analysis with Proximal Policy Optimization. *Nucl. Eng. Technol.* **2026**, *58*, 103767. [CrossRef]
55. Nguyen, K.H.N.; Rivas, A.; Delipei, G.K.; Hou, J. Reinforcement Learning-Based Control Sequence Optimization for Advanced Reactors. *J. Nucl. Eng.* **2024**, *5*, 209–225. [CrossRef]
56. Chen, J.; Xiao, K.; Huang, K.; Yang, Z.; Chu, Q.; Jiang, G. A Multi-Variable Coupled Control Strategy Based on a Deep Deterministic Policy Gradient Reinforcement Learning Algorithm for a Small Pressurized Water Reactor. *Energies* **2025**, *18*, 1517. [CrossRef]
57. Kim, J.; Lee, D.; Yang, J.; Lee, S. Conceptual Design of Autonomous Emergency Operation System for Nuclear Power Plants and Its Prototype. *Nucl. Eng. Technol.* **2020**, *52*, 308–322. [CrossRef]
58. Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained Policy Optimization. In Proceedings of the 34th International Conference on Machine Learning, Sydney, NSW, Australia, 6–11 August 2017; Volume 70, pp. 22–31.

59. Alshiekh, M.; Bloem, R.; Ehlers, R.; Konighofer, B.; Niekum, S.; Topcu, U. Safe Reinforcement Learning via Shielding. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32, pp. 2669–2678. [CrossRef]
60. Ames, A.D.; Xu, X.; Grizzle, J.W.; Tabuada, P. Control Barrier Function Based Quadratic Programs for Safety Critical Systems. *IEEE Trans. Autom. Control* **2017**, *62*, 3861–3876. [CrossRef]
61. Wabersich, K.P.; Zeilinger, M.N. A Predictive Safety Filter for Learning-Based Control of Constrained Nonlinear Dynamical Systems. *Automatica* **2021**, *129*, 109597. [CrossRef]
62. Sha, L. Using Simplicity to Control Complexity. *IEEE Softw.* **2001**, *18*, 20–28. [CrossRef]
63. Garone, E.; Di Cairano, S.; Kolmanovsky, I. Reference and Command Governors for Systems with Constraints: A Survey on Theory and Applications. *Automatica* **2017**, *75*, 306–328. [CrossRef]
64. Sun, Y.; Khairy, S.; Vilim, R.B.; Hu, R.; Dave, A.J. A Safe Reinforcement Learning Algorithm for Supervisory Control of Power Plants. *Knowl.-Based Syst.* **2024**, *301*, 112312. [CrossRef]
65. Chen, X.; Ray, A. Deep Reinforcement Learning Control of a Boiling Water Reactor. *IEEE Trans. Nucl. Sci.* **2022**, *69*, 1820–1832. [CrossRef]
66. Tunkle, L.; Abdulraheem, K.; Lin, L.; Radaideh, M.I. Nuclear Microreactor Transient and Load-Following Control with Deep Reinforcement Learning. *Energy Convers. Manag. X* **2025**, *27*, 101090. [CrossRef]
67. Yoo, S.; Mohler, G.; Zhang, F. Self-Healing Control of Nuclear Power Plants Under False Data Injection Attacks. *Nucl. Sci. Eng.* **2025**, *199*, 162–175. [CrossRef]
68. Janner, M.; Fu, J.; Zhang, M.; Levine, S. When to Trust Your Model: Model-Based Policy Optimization. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
69. Berkenkamp, F.; Turchetta, M.; Schoellig, A.; Krause, A. Safe Model-Based Reinforcement Learning with Stability Guarantees. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
70. Almukhametov, A.; Lim, D.; Hu, R.; Liu, Y. Graph Neural ODE Digital Twins for Control-Oriented Reactor Thermal-Hydraulic Forecasting Under Partial Observability. *IEEE Trans. Nucl. Sci.* **2026**, *73*, 3837–3855. [CrossRef]
71. Grieves, M.; Vickers, J. Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In *Transdisciplinary Perspectives on Complex Systems*; Kahlen, F.J., Flumerfelt, S., Alves, A., Eds.; Springer: Cham, Switzerland, 2017; pp. 85–113. [CrossRef]
72. Tao, F.; Zhang, H.; Liu, A.; Nee, A.Y.C. Digital Twin in Industry: State-of-the-Art. *IEEE Trans. Ind. Inform.* **2019**, *15*, 2405–2415. [CrossRef]
73. Kritzinger, W.; Karner, M.; Traar, G.; Henjes, J.; Sihn, W. Digital Twin in Manufacturing: A Categorical Literature Review and Classification. *IFAC PapersOnLine* **2018**, *51*, 1016–1022. [CrossRef]
74. Wright, L.; Davidson, S. How to Tell the Difference between a Model and a Digital Twin. *Adv. Model. Simul. Eng. Sci.* **2020**, *7*, 13. [CrossRef]
75. National Academies of Sciences, Engineering, and Medicine. *Foundational Research Gaps and Future Directions for Digital Twins*; Technical report; The National Academies Press: Washington, DC, USA, 2024. [CrossRef] [PubMed]
76. Evensen, G. *Data Assimilation: The Ensemble Kalman Filter*, 2nd ed.; Springer: Berlin, Germany, 2009. [CrossRef]
77. Gong, L.; Peng, C.; Huang, Q. Toward a Digital Twin for Nuclear Reactors: Data Assimilation in Thermal-Hydraulics Using EnKF. *Prog. Nucl. Energy* **2026**, *195*, 106305. [CrossRef]
78. Song, H.; Song, M.; Liu, X. Online Autonomous Calibration of Digital Twins Using Machine Learning with Application to Nuclear Power Plants. *Appl. Energy* **2022**, *326*, 119995. [CrossRef]
79. Hong, L.Z.; Gong, H.L.; Ji, H.J.; Lu, J.L.; Li, H.; Li, Q. Optimizing Near-Carbon-Free Nuclear Energy Systems: Advances in Reactor Operation Digital Twin through Hybrid Machine Learning Algorithms for Parameter Identification and State Estimation. *Nucl. Sci. Tech.* **2024**, *35*, 135. [CrossRef]
80. Kapteyn, M.G.; Pretorius, D.; Willcox, K.E. A Probabilistic Graphical Model Foundation for Enabling Predictive Digital Twins at Scale. *Nat. Comput. Sci.* **2021**, *1*, 337–347. [CrossRef] [PubMed]
81. Lim, D.; Liu, Y.; Ndum Ndum, Z.; Young, C.; Hassan, Y. An AI-Driven Thermal-Fluid Testbed for Advanced Small Modular Reactors: Integration of Digital Twin and Large Language Models. *AI Therm. Fluids* **2025**, *1*, 100023. [CrossRef]
82. Brunton, S.L.; Proctor, J.L.; Kutz, J.N. Discovering Governing Equations from Data by Sparse Identification of Nonlinear Dynamical Systems. *Proc. Natl. Acad. Sci. USA* **2016**, *113*, 3932–3937. [CrossRef] [PubMed]
83. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning Nonlinear Operators via DeepONet Based on the Universal Approximation Theorem of Operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229. [CrossRef]
84. Li, Z.; Kovachki, N.; Azizzadenesheli, K.; Liu, B.; Bhattacharya, K.; Stuart, A.; Anandkumar, A. Fourier Neural Operator for Parametric Partial Differential Equations. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 3–7 May 2021.

85. Battaglia, P.W.; Hamrick, J.B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R.; et al. Relational Inductive Biases, Deep Learning, and Graph Networks. *arXiv* **2018**, arXiv:1806.01261. [CrossRef]
86. Liu, Y.; Hu, R.; Zou, L.; Nunez, D. SAM-ML: Integrating Data-Driven Closure with Nuclear System Code SAM for Improved Modeling Capability. *Nucl. Eng. Des.* **2022**, *400*, 112059. [CrossRef]
87. Liu, Y.; Hu, R.; Balaprakash, P.; Kraus, A.; Obabko, A. Data-Driven Modeling of Coarse Mesh Turbulence for Reactor Transient Analysis Using Convolutional Recurrent Neural Networks. *Nucl. Eng. Des.* **2022**, *390*, 111716. [CrossRef]
88. Gong, H.; Cheng, S.; Chen, Z.; Li, Q. Data-Enabled Physics-Informed Machine Learning for Reduced-Order Modeling Digital Twin: Application to Nuclear Reactor Physics. *Nucl. Sci. Eng.* **2022**, *196*, 668–693. [CrossRef]
89. Prantikos, K.; Tsoukalas, L.H.; Heifetz, A. Physics-Informed Neural Network Solution of Point Kinetics Equations for a Nuclear Reactor Digital Twin. *Energies* **2022**, *15*, 7697. [CrossRef]
90. Kennedy, M.C.; O'Hagan, A. Bayesian Calibration of Computer Models. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2001**, *63*, 425–464. [CrossRef]
91. Oberkampf, W.L.; Roy, C.J. *Verification and Validation in Scientific Computing*; Cambridge University Press: Cambridge, UK, 2010. [CrossRef]
92. Liu, Y.; Alsafadi, F.; Mui, T.; O'Grady, D.; Hu, R. Development of Whole System Digital Twins for Advanced Reactors: Leveraging Graph Neural Networks and SAM Simulations. *Nucl. Technol.* **2025**, *211*, 2206–2223. [CrossRef]
93. Liu, Y.; Mui, T.; Xie, Z.; Hu, R. Benchmarking FFTF LOFWOS Test No. 13 Using SAM Code: Baseline Model Development and Uncertainty Quantification. *Ann. Nucl. Energy* **2023**, *192*, 110010. [CrossRef]
94. Kobayashi, K.; Alam, S.B. Deep Neural Operator-Driven Real-Time Inference to Enable Digital Twin Solutions for Nuclear Energy Systems. *Sci. Rep.* **2024**, *14*, 2101. [CrossRef] [PubMed]
95. Hossain, R.; Ahmed, F.; Kobayashi, K.; Koric, S.; Abueidda, D.; Alam, S.B. Virtual Sensing-Enabled Digital Twin Framework for Real-Time Monitoring of Nuclear Systems Leveraging Deep Neural Operators. *npj Mater. Degrad.* **2025**, *9*, 21. [CrossRef]
96. Lin, L.; Bao, H.; Dinh, N. Uncertainty Quantification and Software Risk Analysis for Digital Twins in the Nearly Autonomous Management and Control Systems: A Review. *Ann. Nucl. Energy* **2021**, *160*, 108362. [CrossRef]
97. Wu, X.; Kozlowski, T.; Meidani, H.; Shirvan, K. Inverse Uncertainty Quantification Using the Modular Bayesian Approach Based on Gaussian Process, Part 1: Theory. *Nucl. Eng. Des.* **2018**, *335*, 339–355. [CrossRef]
98. Wu, X.; Kozlowski, T.; Meidani, H.; Shirvan, K. Inverse Uncertainty Quantification Using the Modular Bayesian Approach Based on Gaussian Process, Part 2: Application to TRACE. *Nucl. Eng. Des.* **2018**, *335*, 417–431. [CrossRef]
99. Wu, X.; Xie, Z.; Alsafadi, F.; Kozlowski, T. A Comprehensive Survey of Inverse Uncertainty Quantification of Physical Model Parameters in Nuclear System Thermal-Hydraulics Codes. *Nucl. Eng. Des.* **2021**, *384*, 111460. [CrossRef]
100. Radaideh, M.I.; Borowiec, K.; Kozlowski, T. Integrated Framework for Model Assessment and Advanced Uncertainty Quantification of Nuclear Computer Codes Under Bayesian Statistics. *Reliab. Eng. Syst. Saf.* **2019**, *189*, 357–377. [CrossRef]
101. Radaideh, M.I.; Kozlowski, T. Surrogate Modeling of Advanced Computer Simulations Using Deep Gaussian Processes. *Reliab. Eng. Syst. Saf.* **2020**, *195*, 106731. [CrossRef]
102. Liu, Y.; Dinh, N.T.; Smith, R.C.; Sun, X. Uncertainty Quantification of Two-Phase Flow and Boiling Heat Transfer Simulations Through a Data-Driven Modular Bayesian Approach. *Int. J. Heat Mass Transf.* **2019**, *138*, 1096–1116. [CrossRef]
103. Liu, Y.; Dinh, N. Validation and Uncertainty Quantification for Wall Boiling Closure Relations in Multiphase-CFD Solver. *Nucl. Sci. Eng.* **2019**, *193*, 81–99. [CrossRef]
104. Liu, Y.; Sun, X.; Dinh, N.T. Validation and Uncertainty Quantification of Multiphase-CFD Solvers: A Data-Driven Bayesian Framework Supported by High-Resolution Experiments. *Nucl. Eng. Des.* **2019**, *354*, 110200. [CrossRef]
105. Liu, Y.; Wang, D.; Sun, X.; Liu, Y.; Dinh, N.; Hu, R. Uncertainty Quantification for Multiphase-CFD Simulations of Bubbly Flows: A Machine Learning-Based Bayesian Approach Supported by High-Resolution Experiments. *Reliab. Eng. Syst. Saf.* **2021**, *212*, 107636. [CrossRef]
106. Liu, Y.; Dinh, N.; Sun, X.; Hu, R. Uncertainty Quantification for Multiphase-CFD Closure Relations with a Physics-Informed Bayesian Approach. *Nucl. Technol.* **2023**, *209*, 1–14. [CrossRef]
107. Abulawi, Z.; Hu, R.; Balaprakash, P.; Liu, Y. Bayesian Optimized Deep Ensemble for Uncertainty Quantification of Deep Neural Networks: A System Safety Case Study on Sodium Fast Reactor Thermal Stratification Modeling. *Reliab. Eng. Syst. Saf.* **2025**, *264*, 111353. [CrossRef]
108. Abulawi, Z.; Lim, D.; Garimidi, A.; Liu, Y. Bayesian-Optimized, Feature-Augmented Deep Ensemble for Physics-Guided Critical Heat-Flux Prediction with Uncertainty Quantification. *Ann. Nucl. Energy* **2026**, *230*, 112139. [CrossRef]
109. Fuller, A.; Fan, Z.; Day, C.; Barlow, C. Digital Twin: Enabling Technologies, Challenges and Open Research. *IEEE Access* **2020**, *8*, 108952–108971. [CrossRef]
110. Rasheed, A.; San, O.; Kvamsdal, T. Digital Twin: Values, Challenges and Enablers from a Modeling Perspective. *IEEE Access* **2020**, *8*, 21980–22012. [CrossRef]

111. Aldemir, T. A Survey of Dynamic Methodologies for Probabilistic Safety Assessment of Nuclear Power Plants. *Ann. Nucl. Energy* **2013**, *52*, 113–124. [CrossRef]
112. Endsley, M.R.; Kaber, D.B. Level of Automation Effects on Performance, Situation Awareness and Workload in a Dynamic Control Task. *Ergonomics* **1999**, *42*, 462–492. [CrossRef] [PubMed]
113. Henzinger, T.A. The Theory of Hybrid Automata. In Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, NJ, USA, 27–30 July 1996; pp. 278–292. [CrossRef]
114. Ramadge, P.J.G.; Wonham, W.M. The Control of Discrete Event Systems. *Proc. IEEE* **1989**, *77*, 81–98. [CrossRef] [PubMed]
115. Byun, H.H.; Yim, M.S. Genetic Algorithm-Based Non-Synchronous Unit-Specific Optimal Load-Following Control of Multi-Unit Small Modular Reactor. *Energy* **2025**, *314*, 134091. [CrossRef]
116. Braarud, P.Ø.; Gundersen, A.B.; Langstrand, J.P.; Nystad, E.; McDonald, R.; Odéen, D. Operators' Perspectives on AI Support for Monitoring and Controlling Multi-Unit Small Modular Reactors. In Proceedings of the 2026 1st International Conference on Human Centric Artificial Intelligence (ICHCAI), Halden, Norway, 27–28 May 2026; pp. 1–7. [CrossRef]
117. Scattolini, R. Architectures for Distributed and Hierarchical Model Predictive Control: A Review. *J. Process Control* **2009**, *19*, 723–731. [CrossRef]
118. Jiang, D.; Dong, Z.; Zhu, Y.; Yu, H.; Huang, X. Coordinated Control of the World's First Multi-Modular High Temperature Gas-Cooled Reactor Power Plant HTR-PM. *Nucl. Eng. Technol.* **2026**, *58*, 103940. [CrossRef]
119. Jiang, D.; Dong, Z. Dynamic Matrix Control for Thermal Power of Multi-Modular High Temperature Gas-Cooled Reactor Plants. *Energy* **2020**, *198*, 117386. [CrossRef]
120. Dong, Z.; You, Z.; Lin, X. Multimodular Load Following Coordinated Control of High Temperature Gas-Cooled Reactor Nuclear Cogeneration Plants. *Energy* **2026**, *344*, 140074. [CrossRef]
121. Ndum Ndum, Z.; Lim, D.; Young, C.; Hassan, Y.; Liu, Y. Large Language Model-Assisted Digital Twin for Remote Monitoring and Control of Advanced Reactors. *Prog. Nucl. Energy* **2026**, *192*, 106172. [CrossRef]
122. Lim, J.Y.; Pylorof, D.; Garcia, H.E.; Duraisamy, K. A Digital Twin Framework for Generation-IV Reactors with Reinforcement Learning-Enabled Health-Aware Supervisory Control. *Prog. Nucl. Energy* **2026**, *192*, 106105. [CrossRef]
123. Lin, L.; Athe, P.; Rouxelin, P.; Avramova, M.; Gupta, A.; Youngblood, R.; Lane, J.; Dinh, N. Development and Assessment of a Nearly Autonomous Management and Control System for Advanced Reactors. *Ann. Nucl. Energy* **2021**, *150*, 107861. [CrossRef]
124. Stewart, R.; Trevino, E.; Shields, A.; Heaps, K.; Darrington, J.; Williams, Q.; Pope, C.; Scott, J.; Baker, B.; Palmer, J.; et al. The AGN-201 Digital Twin: A Test Bed for Remotely Monitoring Nuclear Reactors. *Ann. Nucl. Energy* **2025**, *213*, 111041. [CrossRef]
125. Trevino, E.A.; Shields, A.J.B.; Stewart, R.H.; Darrington, J.W.; Scott, J.; Pope, C.; Ritter, C.S. Autonomous Anomaly Detection of Proliferation in the AGN-201 Nuclear Reactor Digital Twin. *Ann. Nucl. Energy* **2025**, *212*, 110990. [CrossRef]
126. Yockey, P.; Erickson, A.; Spirito, C. Cyber Threat Assessment of Machine Learning Driven Autonomous Control Systems of Nuclear Power Plants. *Prog. Nucl. Energy* **2023**, *166*, 104960. [CrossRef]
127. Humayed, A.; Lin, J.; Li, F.; Luo, B. Cyber-Physical Systems Security—A Survey. *IEEE Internet Things J.* **2017**, *4*, 1802–1831. [CrossRef]
128. International Electrotechnical Commission. *Security for Industrial Automation and Control Systems, Part 3-2: Security Risk Assessment for System Design*; Technical Report IEC 62443-3-2:2020; International Electrotechnical Commission: Geneva, Switzerland, 2020.
129. Bainbridge, L. Ironies of Automation. *Automatica* **1983**, *19*, 775–779. [CrossRef]
130. Lee, J.D.; See, K.A. Trust in Automation: Designing for Appropriate Reliance. *Hum. Factors* **2004**, *46*, 50–80. [CrossRef] [PubMed]
131. Hoff, K.A.; Bashir, M. Trust in Automation: Integrating Empirical Evidence on Factors That Influence Trust. *Hum. Factors* **2015**, *57*, 407–434. [CrossRef] [PubMed]
132. Parasuraman, R.; Riley, V. Humans and Automation: Use, Misuse, Disuse, Abuse. *Hum. Factors* **1997**, *39*, 230–253. [CrossRef]
133. Endsley, M.R. Toward a Theory of Situation Awareness in Dynamic Systems. *Hum. Factors* **1995**, *37*, 32–64. [CrossRef]
134. Garimidi, A.; Almukhametov, A.; Abulawi, Z.; Tao, J.; Liu, Y. A Knowledge-Guided Hybrid Expert System for Continuous Industrial Monitoring with Bounded-Context Large Language Model Agents. In Proceedings of the American Nuclear Society Embedded Topical Meeting on Advances in Thermal Hydraulics (ATH), Phoenix, AZ, USA, 15–18 November 2026.
135. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the Opportunities and Risks of Foundation Models. *arXiv* **2021**, arXiv:2108.07258. [CrossRef]
136. Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. Training Language Models to Follow Instructions with Human Feedback. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; Volume 35, pp. 27730–27744.
137. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.T.; Rocktäschel, T.; et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–12 December 2020; Volume 33, pp. 9459–9474.
138. Gao, Y.; Xiong, Y.; Gao, X.; Jia, K.; Pan, J.; Bi, Y.; Dai, Y.; Sun, J.; Wang, M.; Wang, H. Retrieval-Augmented Generation for Large Language Models: A Survey. *arXiv* **2023**, arXiv:2312.10997. [CrossRef]

139. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; Lomeli, M.; Hambro, E.; Zettlemoyer, L.; Cancedda, N.; Scialom, T. Toolformer: Language Models Can Teach Themselves to Use Tools. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023; Volume 36, pp. 68539–68551.
140. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. ReAct: Synergizing Reasoning and Acting in Language Models. In Proceedings of the International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
141. Ji, Z.; Lee, N.; Frieske, R.; Yu, T.; Su, D.; Xu, Y.; Ishii, E.; Bang, Y.J.; Madotto, A.; Fung, P. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.* **2023**, *55*, 248:1–248:38. [CrossRef]
142. Amershi, S.; Weld, D.S.; Vorvoreanu, M.; Fourney, A.; Nushi, B.; Collisson, P.; Suh, J.; Iqbal, S.; Bennett, P.N.; Inkpen, K.; et al. Guidelines for Human-AI Interaction. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019. [CrossRef]
143. Ndum Ndum, Z.; Tao, J.; Ford, J.; Liu, Y. Automating Monte Carlo Simulations in Nuclear Engineering with Domain Knowledge-Embedded Large Language Model Agents. *Energy AI* **2025**, *21*, 100555. [CrossRef]
144. Liu, Y.; Abulawi, Z.; Garimidi, A.; Lim, D. Automating Data-Driven Modeling and Analysis for Engineering Applications Using Large Language Model Agents. *Knowl.-Based Syst.* **2026**, *343*, 115989. [CrossRef]
145. Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Li, B.; Zhu, E.; Jiang, L.; Zhang, X.; Zhang, S.; Liu, J.; et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. *arXiv* **2023**, arXiv:2308.08155. [CrossRef]
146. Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. The Rise and Potential of Large Language Model Based Agents: A Survey. *arXiv* **2023**, arXiv:2309.07864. [CrossRef]
147. Cemri, M.; Pan, M.Z.; Yang, S.; Agrawal, L.A.; Chopra, B.; Tiwari, R.; Keutzer, K.; Parameswaran, A.; Klein, D.; Ramchandran, K.; et al. Why Do Multi-Agent LLM Systems Fail? In Proceedings of the Advances in Neural Information Processing Systems 38, San Diego, CA, USA, 2–7 December 2025; pp. 135676–135729. [CrossRef]
148. Ndum, Z.N.; Tao, J.; Ford, J.; Yim, M.; Liu, Y. A retrieval-augmented, domain-intelligent agentic framework for reliable decision support in safety-critical nuclear engineering. *Reliab. Eng. Syst. Saf.* **2027**, *277*, 113057. [CrossRef]
149. Lee, Y.P.; Cha, J.; Yu, Y.; Kim, S.G. Large Language Model Agent for Nuclear Reactor Operation Assistance. *Nucl. Eng. Technol.* **2025**, *57*, 103842. [CrossRef]
150. Lee, Y.P.; Roy, S.; Kobayashi, K.; Talukder, S.; Abueidda, D.; Koric, S.; Chakraborty, S.; Alam, S.B. Agentic Physical AI toward a Domain-Specific Foundation Model for Energy Systems: A Case Study on Nuclear Reactor Control. *npj Artif. Intell.* **2026**. [CrossRef]
151. Yeom, S.; Lim, K.; Yu, Y. AtomicGPT: A Domain-Adapted Large Language Model for Secure On-Premises Applications in Nuclear Engineering. *Nucl. Eng. Technol.* **2026**, *58*, 104565. [CrossRef]
152. Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; et al. Holistic Evaluation of Language Models. *arXiv* **2023**, arXiv:2211.09110.
153. Kapoor, S.; Stroebl, B.; Siegel, Z.S.; Nadgir, N.; Narayanan, A. AI Agents That Matter. *arXiv* **2024**, arXiv:2407.01502.
154. Cappello, F.; Madireddy, S.; Underwood, R.; Getty, N.; Chia, N.L.P.; Ramachandra, N.; Nguyen, T.D.; Keçeli, M.; Mallick, T.; Li, Z.; et al. EAIRA: Establishing a Methodology for Evaluating LLMs as Scientific Research Assistants. *Int. J. High Perform. Comput. Appl.* **2026**. [CrossRef]
155. Es, S.; James, J.; Espinosa Anke, L.; Schockaert, S. RAGAs: Automated Evaluation of Retrieval Augmented Generation. In Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, St. Julian's, Malta, 21–22 March 2024; pp. 150–158. [CrossRef]
156. Greshake, K.; Abdelnabi, S.; Mishra, S.; Endres, C.; Holz, T.; Fritz, M. Not What You've Signed Up For: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection. In Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, Copenhagen, Denmark, 30 November 2023; pp. 79–90. [CrossRef]
157. Khuwaileh, B.A.; Matesha, P. Establishing Benchmarks for Large Language Models in Nuclear Engineering: A NucBench Evaluation of GPT-4.1, Claude 3.7 Sonnet, and Gemini 2.5 Pro. *Ann. Nucl. Energy* **2026**, *233*, 112223. [CrossRef]
158. Lee, H.; Choi, D.; Kim, B.; Park, H.; Kim, S.G. NRT-Bench: Benchmarking Multi-Turn Red-Teaming of LLM Operator Agents in Safety-Critical Control Rooms. *arXiv* **2026**, arXiv:2606.20408.
159. Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I.D.; Gebru, T. Model Cards for Model Reporting. In Proceedings of the Conference on Fairness, Accountability, and Transparency, Atlanta, GA, USA, 29–31 January 2019; pp. 220–229. [CrossRef]
160. Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J.W.; Wallach, H.; Daume, H., III; Crawford, K. Datasheets for Datasets. *Commun. ACM* **2021**, *64*, 86–92. [CrossRef]
161. McConnell, J.; Ramuhalli, P.; Heim, A. Balance of Plant Modeling and Real-Time Hardware-in-the-Loop Integration with the Microreactor Automated Control System. *Nucl. Technol.* **2026**, 1–8. [CrossRef]
162. Leveson, N.G. *Engineering a Safer World: Systems Thinking Applied to Safety*; MIT Press: Cambridge, MA, USA, 2011.

163. U.S. Nuclear Regulatory Commission. *Human Factors Engineering Program Review Model*; Technical Report NUREG-0711, Revision 3; U.S. Nuclear Regulatory Commission: Washington, DC, USA, 2012.
164. Stouffer, K.; Pease, M.; Tang, C.; Zimmerman, T.; Pillitteri, V.; Lightman, S.; Hahn, A.; Saravia, S.; Sherule, A.; Thompson, M. *Guide to Operational Technology Security*; Technical Report NIST SP 800-82 Rev. 3; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. [CrossRef]
165. International Atomic Energy Agency. *Computer Security Techniques for Nuclear Facilities*; Technical Report IAEA Nuclear Security Series No. 17-T (Rev. 1); International Atomic Energy Agency: Vienna, Austria, 2021.
166. Cardenas, A.A.; Amin, S.; Sastry, S. Secure Control: Towards Survivable Cyber-Physical Systems. In Proceedings of the 28th International Conference on Distributed Computing Systems Workshops, Beijing, China, 17–20 June 2008; pp. 495–500. [CrossRef]
167. Dong, Z.; Pan, Y.; Huang, X.; Dong, Y.; Zhang, Z. Coordinated Control System Design and Verification of HTR-PM Plant. *Nucl. Eng. Des.* **2018**, *329*, 25–33. [CrossRef]
168. U.S. Nuclear Regulatory Commission. *Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants*; Technical Report Regulatory Guide 1.169, Revision 1; U.S. Nuclear Regulatory Commission: Washington, DC, USA, 2013.
169. International Atomic Energy Agency. *Considerations for Deploying Artificial Intelligence Applications in the Nuclear Power Industry*; Technical Report IAEA Nuclear Energy Series No. NR-T-1.26; International Atomic Energy Agency: Vienna, Austria, 2025. [CrossRef]
170. Tabassi, E. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*; Technical Report NIST AI 100-1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2023. [CrossRef]
171. National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*; Technical Report NIST AI 600-1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2024. [CrossRef]
172. Dennis, M.; Lalain, T.; Betancourt, L.; Hathaway, A.; Anzalone, R. *Artificial Intelligence Strategic Plan: Fiscal Years 2023–2027*; Technical Report NUREG-2261; U.S. Nuclear Regulatory Commission: Washington, DC, USA, 2023.
173. U.S. Nuclear Regulatory Commission. Risk-Informed, Technology-Inclusive Regulatory Framework for Advanced Reactors. *Fed. Regist.* **2026**, *91*, 15696–15881.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.