

Online Learning and Self-Adaptation in Reprogrammable-Hardware Applications

Deep-research assessment of the earlier R and S application classes

Audience: system architects, FPGA/embedded-ML engineers, robotics and signal-processing researchers **Research date:** 8 September 2026 **Scope:** applications previously classified R (post-manufacture functional reprogrammability is a requirement) or S (reprogrammability is strongly justified). FPGA means FPGA-class logic unless a more specific device is named. **Research question:** Which R/S applications benefit from machine learning, especially online learning, automatic adaptation, or bounded self-improvement that changes parameters, policies, memory, FPGA configuration, or body shape?

Direct answer

Yes—many R/S applications benefit substantially from machine learning, but the strongest benefit usually comes from updating state, coefficients, model weights, policies, schedules, or a physical configuration while keeping the verified FPGA circuit stable.

The highest-value application groups are:

1. damage-, payload-, terrain-, and morphology-adaptive robots;
2. cognitive SDR, anti-jam communications, and mobile radio resource/beam control;
3. cognitive radar, electronic warfare, adaptive SIGINT, and underwater acoustic links;
4. industrial inspection, predictive maintenance, and drifting process models;
5. autonomous space payloads and adaptive scientific experiments;
6. quantum calibration, feedback, and error-correction control;
7. network-security SmartNICs and changing traffic-control policies;
8. cloud-FPGA workload placement and accelerator selection;
9. edge-AI personalization and continual learning; and
10. adaptive RF front ends, especially digital predistortion and self-calibration.

The best default architecture is a **verified deterministic FPGA data plane plus a learner on a CPU, DSP, AI engine, or bounded on-FPGA training block**. The learner writes parameter/model memory. Dynamic partial reconfiguration (DPR) is appropriate at slower timescales when choosing among signed, pre-validated modules. Autonomous synthesis and installation of a novel bitstream is not a credible fast control mechanism today and is generally unsuitable for safety-critical operation.

Executive findings

- **Online learning and FPGA reconfiguration are different.** A radio that learns a better channel, a robot that infers terrain, or a beamformer that updates complex weights may improve continuously without changing any FPGA configuration bits.
- **The strongest synergy occurs under non-stationarity.** Machine learning adds value when environment, plant dynamics, interference, workload, users, sensors, or hardware condition change in ways that a fixed rule set cannot cover economically.
- **Fast loops should normally update values, not circuits.** Microsecond-to-millisecond adaptation belongs in registers, BRAM, external memory, model state, or a fixed programmable overlay.
- **DPR is a coarse-mode mechanism.** It is useful for swapping complete waveforms, codecs, accelerator variants, or fault-tolerant implementations over milliseconds to minutes, normally from a validated library.
- **New online bitstream generation is a research/design loop.** FPGA place-and-route can take hours; one 2024 study reports that bitstream generation consumed more than 70% of its design iteration time. A major DPR survey also found limited use in deployed systems.
- **Embodiment makes adaptation unusually valuable.** A robot's payload, wear, damage, tool, contact, terrain, and even body geometry alter its dynamics. Real robots have recovered from damage in minutes, while rapid motor-adaptation systems infer changing conditions in fractions of a second.
- **Safety changes the answer.** A learner may optimize within an envelope, but flight-safe logic, medical therapy limits, industrial/nuclear/rail interlocks, cryptographic roots of trust, and hard trading risk limits should remain deterministic, independently monitored, and revertible.
- **"Self-improvement" should be a governed lifecycle, not unrestricted self-editing.** Detect drift, produce a candidate, test in shadow or simulation, approve against constraints, deploy gradually, monitor, and roll back.

1. Definitions and assessment method

1.1 R and S

Label	Meaning in this report	Implication
R	The specification requires the hardware function, protocol, waveform, mission role, or physical form to change after manufacture.	Reprogrammability is part of the product requirement.
S	A fixed ASIC is technically possible, but evolving algorithms, low volume, deterministic latency, parallel I/O, or custom interfaces make FPGA-class hardware strongly justified.	Reprogrammability is an engineering/economic advantage, not always a formal requirement.

1.2 What counts as learning or adaptation

Mode	What changes	Typical methods
Adaptive estimation/control	State estimates, filter taps, plant parameters, controller gains	LMS/RLS, Kalman filtering, adaptive control, system identification, MPC
Online supervised/self-supervised learning	Model weights or prototypes from a stream	online SGD, ridge regression, replay, pseudo-labels, contrastive/domain adaptation
Continual learning	Knowledge accumulates across changing classes/domains while resisting forgetting	replay buffers, regularization, expandable models, drift-triggered updates
Bandit or reinforcement learning	An action policy improves from delayed reward	contextual bandits, Q-learning, actor-critic, model-based RL
Meta/rapid adaptation	A pre-trained policy rapidly infers a new context or uses a few trials	latent-context inference, meta-learning, behavior repertoires
Resource/self-optimization	Scheduling, placement, precision, power, memory, or accelerator choice	Bayesian optimization, surrogate models, RL, constrained search
Morphological adaptation	Tool, limb, stiffness, geometry, module arrangement, or gait/body coordination	morphology-aware policies, co-design, damage recovery, modular reconfiguration

Classic adaptive filters are not automatically “machine learning.” They are included because the user asked about auto-adaptation, and because they often outperform heavier ML in fast, well-specified loops.

1.3 Adaptation layers

Layer	Adapted object	Typical latency	Assessment
A0	Registers, thresholds, LUTs, filter taps, beam weights, calibration, state memory	ns to ms	Preferred for fast deterministic adaptation
A1	Software, model weights, policy, replay/prototype memory	ms to hours	Preferred for genuine online/continual learning
A2	Overlay/CGRA instruction, dataflow, precision, sparsity, tensor shape	us to seconds	Useful when more structural flexibility than A0/A1 is needed
A3	Precompiled full or partial FPGA bitstream	ms to minutes plus validation	Useful for coarse, infrequent mode changes
A4	Newly generated RTL/netlist/bitstream	minutes to hours or longer	Engineering/research loop; normally not an online control action
A5	Physical morphology, stiffness, tool, antenna/metasurface, or modular body	ms to mission phase	Requires controller/model adaptation; does not itself imply an FPGA bitstream change

1.4 Ratings

ML benefit: High means non-stationarity and feedback make learning a natural source of material performance or resilience; Moderate means useful but conditional; Low means ML mainly assists an engineering workflow or a simpler adaptive method is usually preferable; Avoid means autonomous learning should not control the protected function.

Evidence maturity: Field/real system, real-hardware research, simulation/analysis, or inferred engineering fit. A high benefit score is not a claim of deployment maturity.

2. Ranked priorities

Rank	R/S application family	ML benefit	What should adapt	Best layer and timescale	Evidence-based judgment
1	Field, legged, agricultural, mining, disaster, and planetary robots	High	Dynamics context, gait/control policy, footholds, damage compensation, payload/tool model	A0/A1 at ms; A5 at task/mission scale	Best embodiment case. Real hexapod/arm experiments recovered from damage in under two minutes; rapid motor adaptation handles unseen terrain/payload in fractions of a second.
2	Cognitive SDR, tactical/public-safety radio, anti-jam links	High	Channel/frequency, modulation/coding, power, waveform policy, threat belief	A0/A1 per frame; A3 for whole validated waveform	Directly matches a changing spectrum/adversary. Hardware-in-loop and DPR demonstrations exist; learning should command a verified waveform library.
3	Quantum calibration, qubit reset/feedback, QEC control	High	Pulse/control parameters, state classifier, calibration policy, analog-control map	A0/A1 from ns feedback to minutes training	Strong real-hardware evidence: FPGA neural feedback and experiment-in-the-loop RL; deterministic low-latency execution remains essential.
4	RF power-amplifier self-healing, DPD, array calibration	High	Predistorter LUT/weights, bias/gain, impairment model	A0/A1 from samples to minutes	Temperature, aging, and operating point create continuous drift. Updating values is normally sufficient; topology change is exceptional.
5	Industrial inspection and high-speed sorting	High	Defect model, anomaly prototypes, thresholds, illumination/camera calibration	A0/A1 per batch to hours	Raw material, lighting, tooling, and new defect classes cause concept drift; continual learning is valuable if uncertain samples are quarantined/labeled.
6	Cognitive radar, EW, adaptive SIGINT/ELINT	High, conditional	Waveform/bandwidth/frequency, classifier, jammer/countermeasure policy	A0/A1 per pulse/CPI; A3 for validated mode	Learning helps under persistent, structured change but can lose to rules over short horizons. Operational evidence for autonomous learning is much thinner than simulations/labs.
7	Autonomous space payloads and Earth/science observation	High	Event/anomaly model, compression, target/observation policy, payload mode	A1 seconds to months; A3 mission phase	High value because of light-time/downlink limits and unknown environments. Onboard inference and replanning have flown; online learning and autonomous DPR remain tightly governed research.
8	Scientific instruments, beamlines, microscopy, fusion, adaptive optics	High	Calibration, next experiment/shot, controller parameters, trigger/model	A0/A1 at us to minutes; A3 between runs	Feedback is abundant and instrument time is costly. A fixed FPGA inner loop plus host learner is the mature pattern.
9	Network security, traffic classification, SmartNICs	High	Flow/anomaly model, feature/rule tables, mitigation policy	A0/A1 at packet to minute scale; A3 for pipeline variant	Traffic and attacks drift. Line-rate FPGA features/inference plus drift-aware learner is compelling; training is normally outside the packet pipeline.
10	Datacenter FPGA workload consolidation	High	Placement, co-location, resource partition, accelerator choice	A1/A3 seconds to minutes	FARMER demonstrated online throughput modeling plus DPR on a real Alveo U55C while exploring under 0.012% of its design space.
11	5G/6G/O-RAN scheduling, beam and resource control	High	PRBs, slicing, power, handover, beam index/weights, traffic policy	A0/A1 from slots to seconds	Dynamic traffic and channels reward learning. RIC/CPU chooses actions; FPGA/RFSoc consumes coefficients. Per-action bitstream changes are unnecessary.
12	Underwater acoustic links, adaptive sonar, autonomous AUV missions	High/Moderate	Equalizer/waveform/power, channel belief, anomaly model, mission plan	A0/A1 packet to mission; A3 for modem family	Severe time-varying channels and scarce bandwidth favor adaptation. Direct FPGA-online-learning evidence is limited; the FPGA remains the DSP engine.
13	Predictive maintenance and drifting industrial process models	High	Health/anomaly model, soft sensor, maintenance/sampling policy	A1 minutes to days	Wear and process regimes create drift. Strong ML rationale, but actual circuit reconfiguration is rarely needed.
14	Edge AI personalization and continual perception	High/Moderate	User/domain model, embeddings, prototypes, selected accelerator/model	A1 minutes to days; A3 rarely	On-FPGA training is feasible in research prototypes, but memory, energy, labels, privacy, and catastrophic forgetting remain constraints.

3. Comprehensive R/S application matrix

The following matrix covers the broad R/S families found in the earlier catalogue. “No new bitstream” means the application may still need FPGA reprogrammability for product upgrades, but online learning does not require circuit reconfiguration.

3.1 Robotics, vehicles, and industrial systems

Application family	Class	ML benefit	Online/auto-adaptation target	Recommended hardware/layer	Verdict and maturity
Legged/field/search-and-rescue/agricultural /mining/planetary robots	R/S	High	Terrain/context estimate, gait, slip model, damage/payload compensation	SoC FPGA deterministic control and perception; CPU/AI engine learner; A0/A1, optional A5	Strong real-robot research. Rapid context/policy adaptation is safer than unrestricted online RL exploration.
Wheeled/tracked off-road robots and rovers	R/S	High	Traction/slip, terrain cost, steering dynamics, risk map	SoC FPGA sensor/control path plus bounded learner; A0/A1	High benefit in unknown terrain. Maintain a verified mobility envelope and recovery controller.
Manipulators, cobots, changing tools/payloads	S/R	High/Moderate	Inertia/friction/contact model, grasp policy, tool frame, human intent	SoC FPGA/robot controller; A0/A1	Online system identification and residual learning are valuable; safety-rated torque/speed limits stay fixed.
Modular and self-reconfiguring robots	R	High	Body graph, module role, locomotion/manipulation policy, task morphology	Distributed MCU/FPGA nodes; A1/A5; A3 only for module role blocks	Natural fit for morphology-aware/meta-learning. Physical reconfiguration needs topology-aware control, not a novel bitstream each movement.
Soft/continuum robots, exoskeletons, prostheses	S/R	High	Nonlinear kinematic model, stiffness, user intent and personalization	SoC FPGA or MCU/DSP plus sensors; A0/A1/A5	Strong rationale because material/user dynamics drift. Clinical/human-interaction limits require bounded adaptation and rollback.
Warehouse AMRs and mobile manipulation	S	Moderate/High	Map/traffic policy, load/traction model, perception classes	SoC FPGA/NPU; A1	Useful in changing fleets and layouts; conventional fleet optimization may be enough for stable warehouses.
UAV autonomy and adaptive mission payloads	R/S	High but constrained	Wind/aerodynamics, payload, perception, target selection, route policy	SoC FPGA for I/O/control; isolated CPU/AI learner; A0/A1, A3 for payload	Large benefit to mission layer. Primary flight control/safe mode should not self-modify without aviation-grade assurance.
ADAS/autonomous vehicles and sensor fusion	S	High but constrained	Sensor calibration, object/domain model, friction/driver/context policy	Automotive SoC/FPGA prototype/eFPGA; A0/A1	Continuous improvement is valuable, but usually fleet/offline or shadow-validated. Online policy exploration is inappropriate on public roads.
Motor drives, actuators, power converters	S	Moderate	Motor/plant parameters, dead time, load, efficiency map, fault residual	FPGA/DSP; A0 first, small A1 residual if justified	Adaptive control often beats ML on transparency/stability. No new bitstream; independent current/voltage protection.
Industrial machine vision/sorting/additive manufacturing	S/R	High	Defect/anomaly model, new class memory, thresholds, illumination/process domain	SoC FPGA/NPU pipeline; A0/A1; A3 for major pipeline change	Clear concept-drift case with real industrial datasets. Use uncertainty gating and human labels for consequential updates.
Predictive maintenance/condition monitoring	S	High	Health model, anomaly baseline, remaining-life estimate, sampling policy	FPGA feature extraction plus CPU learner; A1	Strong ML benefit but weak need for FPGA topology changes. Drift detection should trigger validation/retraining.
Process control, soft sensors, energy/microgrids	S	High/Moderate	Process model, setpoints, schedules, state/price/load forecasts	FPGA deterministic loop plus supervisory learner; A0/A1	Valuable for time-varying plants; exploration constrained by process safety and stability certificates.
Medical/surgical robotics and therapy systems	S/R	Moderate/High but constrained	Patient/tissue model, tremor/intent, image guidance, treatment planning	SoC FPGA with independent safety CPLD/flash FPGA; A0/A1	Personalization helps; direct autonomous therapy changes need a predetermined, validated change plan and hard limits.

3.2 Communications, radar, sensing, and media

Application family	Class	ML benefit	Online/auto-adaptation target	Recommended hardware/layer	Verdict and maturity
Cognitive/multistandard/tactical/public-safety SDR	R	High	Spectrum belief, channel, frequency, MCS, power, waveform policy	RFSoc/SoC FPGA plus CPU; A0/A1; A3 for complete waveform	One of the best matches. Regulatory and interoperability rules form the action mask.
Satellite/optical communications and ground modems	R/S	High/Moderate	Link state, coding/rate/power, pointing, channel/error model	Rad-tolerant FPGA/RFSoc; A0/A1; signed A3	Long missions and changing link conditions favor learning, but uploads remain authenticated and reversible.
5G/6G/O-RAN, massive MIMO, beam management	R/S	High	Scheduling/slicing, power, handover, beam/codebook/weights	RIC/CPU learner plus FPGA/RFSoc PHY; A0/A1	High value from nonstationary users/channels. Learning changes commands/weights, not the PHY bitstream per slot.
Digital predistortion, crest-factor reduction, RF calibration	S	High	LUT or neural predistorter, PA bias/gain, IQ/array calibration	FPGA/RFSoc; A0/A1	Excellent local-feedback case; on-FPGA RL exists, but conventional adaptation is more mature.
Spectrum monitoring, SIGINT/ELINT, threat warning	R/S	High	Unknown-emitter representation, classifier, channelizer allocation, attention policy	RFSoc fast path plus CPU/GPU learner; A1; optional A3	Continual/open-set learning is attractive because emitter classes evolve. Poisoning and adversarial deception are major risks.
Cognitive/multimode/AESA/passive/SAR/weather radar	R/S	High/Moderate	Waveform, bandwidth, frequency, beam, CFAR/classifier/tracker policy	RFSoc/FPGA; A0/A1; A3 for mode	ML helps in high-entropy, persistent environments; studies show rules may win in short, well-specified horizons.
Electronic attack/adaptive jammer/deception	R	High but adversarial	Countermeasure policy, target/threat belief, waveform choice	Secure RFSoc plus policy processor; A0/A1; signed A3	Strong rationale, sparse open operational evidence. Constrain spectral/emission and friendly-system impacts.
Sonar/hydrophone arrays/acoustic camera/AUV modem	R/S	High/Moderate	Equalizer, beam, waveform, detection, channel belief, mission attention	SoC FPGA plus CPU; A0/A1; A3 for protocol family	Highly nonstationary channel. Real-world learning evidence exists, but FPGA learning itself is uncommon.
Pulsed/FMCW/SPAD/bathymetric/atmospheric LiDAR	R/S	Moderate	ROI/scan pattern, calibration, detection threshold, denoising/domain model	FPGA timing/histogram engine plus CPU/NPU; A0/A1	Often adaptive inference or planning, not online weight learning. No strong case for frequent DPR.
Industrial/scientific/hyperspectral/event/thermal imaging	S/R	High/Moderate	Calibration, exposure, ROI, anomaly/object model, compression	SoC FPGA/adaptive SoC; A0/A1	Valuable where sensors/experiments/domains change; ground truth and forgetting are limiting.
Ultrasound/photoacoustic/HIFU	R/S	Moderate/High	Beam/acquisition parameters, image model, patient/tissue adaptation	FPGA beamformer plus host/AI engine; A0/A1; A3 between modalities	Research loops are promising; clinical autonomous modification requires bounded validation. Therapy interlocks stay fixed.
MRI/NMR research console	R/S	Moderate/High	Calibration, reconstruction/prior, scan/pulse-sequence optimization	SoC FPGA/RFSoc plus GPU/CPU; A0/A1; A3 for major pipeline	High research value; usually optimize parameters/sequences rather than synthesize circuits online.
PET/SPECT/OCT/CT/X-ray/FLIM acquisition	S/R	Moderate	Coincidence/calibration, denoising/reconstruction, acquisition policy	FPGA acquisition pipeline plus CPU/GPU; A0/A1	ML can adapt image quality/dose/acquisition, but safety and clinical drift monitoring dominate.
Broadcast switcher/format converter/AV-over-IP	R/S	Low/Moderate	QoE/bitrate, exposure/colour, routing, failure prediction	FPGA codec/format pipeline; A0/A1; A3 for new codec/standard	Reprogrammability is valuable; online learning is secondary. Major changes are engineered upgrades.
Video encoder/transcoder/live production	S/R	Moderate	Per-title/per-scene rate control, ROI, quality/power policy	FPGA/adaptive SoC plus CPU; A0/A1/A2	ML can improve QoE/efficiency; fixed codec hardware is still preferable at stable high volume.
Microphone arrays, active noise/vibration, digital audio	S	Moderate	Beam/filter taps, source model, acoustic context/effect policy	FPGA/DSP; A0 first, A1 when context classification helps	Adaptive filters are mature; full continual learning or DPR often adds little.

3.3 Networks, computing, storage, and engineering systems

Application family	Class	ML benefit	Online/auto-adaptation target	Recommended hardware/layer	Verdict and maturity
SmartNIC traffic analytics, IDS, botnet/DDoS detection	R/S	High	Flow classifier, anomaly baseline, feature/rule tables, response policy	FPGA/SmartNIC fixed parser/features plus CPU/on-card learner; A0/A1	Strong fit because traffic/attacks drift and line rate matters. Protect training from poisoning; shadow-test mitigations.
Routing, congestion control, load balancing	R/S	High/Moderate	Path, queue, rate and load-balancing policy	Programmable FPGA/NIC/switch data plane plus control-plane learner; A0/A1	Dynamic workloads reward bandits/RL, but stability/fairness and multi-controller oscillation require guardrails.
Cloud/HPC FPGA workload consolidation	R/S	High	Task placement, co-location, resource/power partition, accelerator variant	FPGA plus runtime manager; A1/A3	Direct real-board evidence couples online learning with DPR. A rare case where learned decisions truly select configurations.
Edge-AI accelerator/personalized perception	R/S	High/Moderate	Weights, prototypes, precision/sparsity, model/accelerator choice	SoC FPGA/adaptive SoC; A1/A2; optional A3	On-device FPGA training is demonstrated, but mostly research. Fine-tune a bounded subset and retain a known-good model.
Database acceleration, index/query/buffer tuning	S/R	Moderate	Indexes, join plan, memory/buffer allocation, FPGA operator choice	CPU learner plus FPGA kernels; A1/A3 between workloads	Online tuning can help workload shifts; learning generally manages the system rather than reconfiguring circuits rapidly.
Storage tiering, caching, queues, compression	S/R	Moderate	Placement, cache/queue/readahead, compression algorithm/ratio	CPU/SoC learner plus FPGA kernels; A0/A1; A3 for kernel	Strong systems-optimization rationale, but end-to-end FPGA evidence is immature and workload-specific.
High-frequency trading and exchange gateways	S	Moderate/High but constrained	Short-horizon model/sequence state, quoting/execution parameters	FPGA inference/order path plus CPU trainer; A0/A1	Adaptation can track markets, but hard pre-trade limits and kill switch must be independent. No autonomous bitstream change during trading.
Cryptographic agility/PQC migration	R	Low for learning	Algorithm/key-suite selection is policy; ML may monitor anomalies/side channels	Secure FPGA/eFPGA/CPLD; human-controlled signed A3	ML does not improve the primitive or trust root. Autonomous algorithm/key changes create assurance and attack-surface problems.
ATE, HIL, channel/radar/threat/protocol emulators	R	Moderate	Next stimulus, coverage model, scenario selection, fault hypothesis	FPGA deterministic I/O/model plus host active learner; A0/A1/A3	Active learning can reduce test time; newly generated test logic still needs verification.
Oscilloscope/analyzer/AWG/lock-in/module DAQ	R/S	Moderate	Trigger/anomaly model, adaptive sampling, experiment waveform/sequence	FPGA/RFSoc plus CPU; A0/A1; A3 for new instrument mode	ML improves intelligent acquisition; ordinary reprogrammability remains the main product value.
ASIC prototyping/emulation and HLS/EDA exploration	R/S	High for engineering loop	Partition, placement, directives, precision, architecture/PPA search	Host ML plus FPGA emulator; A1/A4	Excellent ML-assisted design-space use, but it is not deployed online learning. Compilation/timing closure remain slow and must be verified.

3.4 Space, science, and safety-critical infrastructure

Application family	Class	ML benefit	Online/auto-adaptation target	Recommended hardware/layer	Verdict and maturity
Earth observation, onboard science, autonomous event response	R/S	High	Cloud/event/anomaly model, data priority/compression, observation/pointing policy	Rad-tolerant SoC FPGA/adaptive processor; A1; signed A3	High mission value and reduced downlink. Static onboard ML is mature relative to onboard training. Risk-aware planning is active research.
Flexible telecom payload, space SDR/GNSS/navigation	R	High/Moderate	Beam/routing/frequency/bandwidth, link/waveform policy	Rad-tolerant FPGA/RFSoc; A0/A1/A3	Strong reconfiguration case. Learning may optimize parameters; circuit updates remain ground-approved, authenticated, and recoverable.
Mission-phase reuse and fault recovery in space	R	Moderate/High	Resource map, degraded-mode policy, selected redundant/partial module	Rad-tolerant FPGA plus independent configuration controller; A1/A3	ML can diagnose/select a recovery, but a golden image, watchdog, scrubbing, and deterministic safe mode are mandatory.
Radio astronomy, pulsar/FRB/SETI backends	R/S	High/Moderate	RFI model/filter, candidate/trigger model, beam/observation scheduling	FPGA/RFSoc front end plus GPU/CPU; A0/A1; A3 between campaigns	RFI and transient selection benefit from adaptation; false suppression can destroy unique science, so retain raw/audit paths where feasible.
Particle-physics trigger and detector DAQ	R/S	High need, constrained online	Calibration and trigger model updated between fills/runs; inference per event	FPGA inference fixed during acquisition; A1 between runs, signed A3	Drift makes continual learning attractive, but erroneous triggers irreversibly discard events. Use shadow tests and rollback.
Photon/neutron/X-ray detectors and synchrotron/FEL experiments	S/R	High	Calibration, feature/event model, next-shot/measurement setting	FPGA edge/feedback plus host learner; A0/A1/A3 between runs	Good closed-loop experimental value; prevent sample/beam damage with independent aborts.
Accelerator RF, fusion/plasma control, adaptive optics	S	High/Moderate	Plant/vibration/wavefront model, actuator commands, experiment policy	RFSoc/FPGA inner loop plus supervisory learner; A0/A1	Promising real systems/preprints. Verified fallback controller and actuator envelopes are essential.
Quantum pulse/readout, feedback, calibration, QEC	R/S	High	Readout classifier, pulse/control parameters, decoder/policy	FPGA/RFSoc; A0/A1; A3 only when protocol/decoder architecture changes	Among the strongest hardware demonstrations of experiment-in-loop learning and ultra-low-latency FPGA inference.
Primary flight/safe-mode control	R/S platform	Avoid autonomous self-improvement	Only bounded parameter estimation outside protected kernel	Static/locked FPGA, flash FPGA, ASIC or safety CPLD; tightly bounded A0	Learning may advise or operate behind runtime assurance, but the invariant/protected controller should remain verified.
Nuclear, rail, medical-therapy and industrial safety interlocks	S platform	Avoid autonomous self-improvement	Diagnostics may learn; trip thresholds/invariants require governed change	Independent ASIC/CPLD/locked FPGA; learner isolated	Do not let an online learner weaken or rewrite the protection function.

4. Where actual FPGA reconfiguration adds value

4.1 Appropriate uses of A3 dynamic partial reconfiguration

Use a prevalidated partial-bitstream library when the selected module changes materially and infrequently:

- complete SDR waveforms, FEC families, channelizers, or modem protocols;
- radar/sonar mission modes or alternate signal-processing pipelines;
- codec/format or scientific-instrument modes;
- accelerator variants for different neural models, tensor shapes, precision, fault tolerance, power, or battery state;
- cloud-FPGA workload placement/co-location;
- space mission-phase payload functions or a verified degraded/fault-tolerant implementation.

The learner may rank or select the module, but an **adaptation governor** should enforce compatibility, resource, thermal, timing, security, and mission constraints.

4.2 When A0/A1 is better

Do not reconfigure logic merely to change:

- filter coefficients, gain, thresholds, lookup tables, beam weights, pulse parameters;
- model weights, embeddings, prototypes, replay memory, class dictionaries;
- radio channel, power, coding rate, resource block, queue, route, cache size;
- robot state, dynamics estimate, trajectory, policy latent/context, or controller gain.

These changes are faster, easier to verify, less disruptive, and possible on many fixed ASICs as well. The FPGA remains valuable for deterministic parallel execution and future engineered upgrades, but the online learner is using **data programmability**, not circuit programmability.

4.3 Why autonomous A4 bitstream synthesis is rarely justified

An ML system that invents RTL, synthesizes, places/routes, installs, and trusts a new circuit at runtime faces:

- compilation/timing closure on minute-to-hour timescales;
- insufficient test coverage for functional and temporal correctness;
- configuration-interface and bitstream supply-chain attacks;
- difficult rollback after stateful or I/O-facing changes;
- certification invalidation in medical, automotive, aviation, rail, industrial, or space contexts;
- radiation/partial-reconfiguration isolation issues in space; and
- a reward signal that rarely proves safety, interoperability, or absence of side channels.

Overlays and CGRAs are the sensible middle ground: compile a restricted operation/dataflow schedule quickly onto a verified fabric instead of generating unrestricted FPGA topology.

5. Recommended governed-adaptation architecture

1. **Static trust and safety shell.** Boot, authentication, clocks/resets, I/O isolation, watchdogs, thermal/current limits, runtime monitor, and golden recovery image.
2. **Deterministic FPGA/RFSoc data plane.** Streaming sensor/radio/network processing and hard real-time control.
3. **Observable state and parameter memory.** Versioned A0 values and A1 model/policy memory with provenance and checksums.
4. **Learner.** Usually on the SoC CPU, AI engine, GPU, or server; on-FPGA training only when latency, privacy, power, or disconnection justifies it.
5. **Adaptation governor.** Action masks, confidence/uncertainty thresholds, rate limits, resource/thermal limits, and allowed module catalogue.
6. **Shadow evaluation and staged activation.** Compare candidate and incumbent; require minimum evidence before control authority expands.
7. **Runtime assurance.** Independent safety constraints can reject, clip, or override learned actions and transfer control to a verified fallback.
8. **Rollback and audit.** Preserve last-known-good parameters, model, software, and bitstream; log observations, decisions, versions, and outcomes.

5.1 Timescale rule

Timescale	Preferred adaptation	Examples
ns-us	A0 deterministic state/coefficients; fixed inference	QEC feedback, beam weights, motor-current loop, trigger/instrument feedback
ms-s	A0/A1 context or policy inference; bounded bandit action	robot terrain adaptation, radio/radar action, routing, scan ROI
minutes-hours	A1 incremental learning; A2 mapping; selected A3 module	calibration, concept drift, battery/fault mode, workload consolidation
days-mission phase	validated A1 release or signed A3 update	new space payload algorithm, protocol/codec, detector trigger between runs
engineering cycle	A4 ML-assisted design and full verification	new accelerator, HDL/HLS design, ASIC emulation, certified product update

6. Application-specific design recommendations

Robots and morphing bodies

Train a robust policy and dynamics prior offline in simulation; online, infer a compact environment/body context or select from a safe behavior repertoire. Permit slow bounded residual updates only when needed. A physical tool, limb, stiffness, or module change should trigger model/context identification and constraint recalculation. It should not automatically trigger bitstream synthesis. Put joint/current/force/workspace limits in independent logic.

Radios, radar, sonar, and EW

Keep channelization, FFT/FIR, timing, framing, and waveform primitives in FPGA/RFSoc. Let a learner select legal parameters or a prevalidated mode. Use contextual bandits when reward is prompt and actions are discrete; use model-based or constrained RL for longer horizons. Detect nonstationarity and fall back to rules when the episode is too short for learning to converge.

Industrial vision and maintenance

Use FPGA logic for deterministic acquisition and feature/inference throughput; collect uncertain/OOD samples in a bounded buffer. A drift detector should decide when to request labels or retrain. Deploy only after replay against retained old domains to control catastrophic forgetting. Do not allow the learning path to bypass reject/fail-safe logic.

Space and scientific instruments

Separate spacecraft/instrument survival from scientific optimization. Let ML prioritize observations, compress data, identify events, or select experiment settings. Keep thermal/power/pointing/beam-abort constraints in verified monitors. Upload signed models/modules or use onboard adaptation only inside a mission-approved envelope; always retain a golden configuration.

Medical systems

Prefer personalization of reconstruction, detection, or planning over unconstrained therapy control. Specify the permitted changes, data, update method, validation, impact assessment, monitoring, and rollback before deployment. FDA's 2025 PCCP guidance explicitly supports iterative improvement through planned modifications while maintaining safety/effectiveness; it is not permission for arbitrary self-modification.

Networks and datacenters

Put parsers, counters, sketches, feature extraction, and hard line-rate enforcement in the data plane. Train/drift-detect in a control plane with poisoning resistance. Apply rate/route/mitigation updates atomically, measure effect, and roll back. DPR is justified when choosing an accelerator composition—not for each packet-level decision.

7. Risks, limitations, and evidence gaps

- **Catastrophic forgetting:** new domains can erase old capability; replay and regression suites consume memory and may still miss rare cases.
- **Exploration risk:** physical robots, transmitters, markets, patients, and scientific samples can be harmed while an RL agent learns.
- **Weak/ambiguous labels:** a changed signal is not necessarily a fault or attack; delayed outcomes create credit-assignment problems.
- **Adversarial adaptation:** EW, cyber, and markets include opponents who can manipulate observations or rewards; online training expands attack surface.
- **Distribution shift:** a learner can be confidently wrong in precisely the conditions that motivated adaptation.
- **Resource contention:** training competes with deterministic workloads for memory bandwidth, power, and thermal headroom.
- **Certification/version explosion:** each mutable layer multiplies configurations to test and audit.
- **DPR deployment gap:** the FPGA literature contains many demonstrations, but broad operational deployment remains limited.
- **Evidence heterogeneity:** robotics and quantum examples include real hardware; cognitive radar, network control, storage, and morphology co-design contain substantial simulation or small-testbed evidence.
- **Publication bias:** positive adaptive results are easier to publish than failures, regressions, or maintenance burden.

This is a broad evidence synthesis, not a literal enumeration of every possible programmable-hardware product. Benefit rankings are engineering judgments based on demonstrated non-stationarity, feedback, latency/locality, evidence maturity, and ability to govern risk. Vendor performance claims were not used as primary proof of learning benefit.

8. Decision checklist

Before adding online learning to an R/S platform, answer:

1. What changes after deployment: environment, plant, adversary, workload, user, sensor, or hardware health?
2. Is there a trustworthy and timely label/reward?
3. Would a transparent adaptive estimator/controller solve the problem more safely?
4. Does the update require A0, A1, A2, A3, A4, or A5?
5. What is the maximum safe exploration and update rate?
6. Can the candidate run in shadow and be compared with the incumbent?
7. What invariants are enforced independently of the learner?
8. How are data/model/bitstream authenticated and versioned?
9. How quickly can the system roll back to a known-good state?
10. Does the update remain within certification, spectrum, privacy, cybersecurity, and mission constraints?

9. Claim-to-source ledger

Claim	Source support	Confidence and note
DPR is technically distinctive but still limited in deployed systems.	[S1]	High; peer-reviewed survey.
Fresh FPGA compilation is too slow for most online loops; bitstream generation dominated the measured design iteration.	[S2]	High for cited workflow; exact time varies by design/tool.
Online learning can directly select FPGA workload compositions and DPR configurations.	[S3]	High for real Alveo research prototype; not proof of broad production adoption.
On-FPGA online CNN training/personalization is feasible.	[S5], [S6]	Medium/high for prototypes; memory/energy/forgetting remain deployment constraints.
Embedded continual learning faces compute/memory limits, forgetting, and OOD generalization.	[S7]	High; focused peer-reviewed study.
Robots can adapt to damage and rapidly infer changing terrain/payload conditions.	[S8], [S9]	High for real research robots; not general safety certification.
Quantum feedback has real FPGA experiment-in-loop RL evidence.	[S10]	High; peer-reviewed real qubit experiment.
Cognitive radar learning is conditional and can lose to rules on short horizons.	[S11], [S12]	Medium/high; analytical/simulation evidence, limited operational validation.
Industrial inspection and process models face concept drift that incremental learning can address.	[S13], [S14]	Medium/high; industrial datasets and process-model studies.
Space autonomy benefits from adaptation but mission-critical onboard learning is not yet considered ready without risk assurance.	[S15], [S16], [S17]	High for need/risk framing; online-training flight maturity remains low.
FPGA neural feedback can materially improve low-latency network analysis.	[S18], [S19]	Medium/high for prototypes; online adaptation is usually a control-plane function.
Medical AI modifications should be planned, validated, monitored, and impact-assessed.	[S20]	High; current FDA final guidance.
Aviation assurance guidance has historically focused on frozen/offline-trained models rather than unrestricted online learning.	[S21]	High for the cited guidance scope; regulation/guidance continues to evolve.
Deployed AI requires ongoing monitoring for reliability, drift, and unforeseen consequences.	[S22]	High; current NIST report.
FPGA edge ML can support adaptive scientific acquisition, but many demonstrations remain proof-of-concept.	[S23], [S24]	Medium/high.

Claim	Source support	Confidence and note
Cryptographic agility should be governed; ML is not a substitute for approved algorithms or trust roots.	[S25]	High as security architecture judgment grounded in NIST agility guidance.

10. Sources

[S1] Vipin, K. and Fahmy, S. A. "FPGA dynamic and partial reconfiguration: a survey of architectures, methods, and applications." ACM Computing Surveys 51(4), 2018. <https://wrap.warwick.ac.uk/id/eprint/100301/>

[S2] Inayat, K. et al. "FPGA-assisted Design Space Exploration of Parameterized AI Accelerators: A Quickloop Approach." Journal of Systems Architecture 151, 2024. <https://www.sciencedirect.com/science/article/pii/S1383762124001978>

[S3] Montanaro, G., Trovo, F., and Zoni, D. "FARMER: Online-Learning-Based Workload Consolidation on Large FPGAs Accelerated With Dynamic Partial Reconfiguration." IEEE Transactions on VLSI Systems, 2026. <https://re.public.polimi.it/handle/11311/1309163>

[S4] Mahmoud, D. G. et al. "Runtime Replacement of Machine Learning Modules in FPGA-Based Systems." MECO, 2021. https://fount.aucegypt.edu/faculty_journal_articles/2670/

[S5] Venkataramanaiah, S. K. et al. "Efficient and Modularized Training on FPGA for Real-time Applications." IJCAI, 2020. <https://www.ijcai.org/proceedings/2020/755>

[S6] Tang, Y. et al. "EF-Train: Enable Efficient On-device CNN Training on FPGA Through Data Reshaping for Online Adaptation or Personalization." ACM TODAES / arXiv, 2022. <https://arxiv.org/abs/2202.10935>

[S7] Hayes, T. L. and Kanan, C. "Online Continual Learning for Embedded Devices." CoLLAs/PMLR 199, 2022. <https://proceedings.mlr.press/v199/hayes22a.html>

[S8] Cully, A. et al. "Robots that can adapt like animals." Nature 521, 2015. <https://www.nature.com/articles/nature14422>

[S9] Kumar, A. et al. "RMA: Rapid Motor Adaptation for Legged Robots." Robotics: Science and Systems / arXiv, 2021. <https://arxiv.org/abs/2107.04034>

[S10] Reuer, K. et al. "Realizing a deep reinforcement learning agent for real-time quantum feedback." Nature Communications 14, 2023. <https://www.nature.com/articles/s41467-023-42901-3>

[S11] Thornton, C. E. and Buehrer, R. M. "When is Cognitive Radar Beneficial?" 2022. <https://arxiv.org/abs/2212.00597>

[S12] Thornton, C. E. and Buehrer, R. M. "On the Value of Online Learning for Radar Waveform Selection." 2023. <https://arxiv.org/abs/2304.11233>

[S13] "Incremental learning of concept drift in Multiple Instance Learning for industrial visual inspection." Computers in Industry 109, 2019. <https://www.sciencedirect.com/science/article/pii/S0166361519300466>

[S14] "Continual learning for neural regression networks to cope with concept drift in industrial processes using convex optimisation." Engineering Applications of Artificial Intelligence 120, 2023. <https://www.sciencedirect.com/science/article/pii/S0952197623001112>

[S15] NASA JPL. "Risk-Aware Machine Learning for Resilient Space Exploration." <https://www-robotics.jpl.nasa.gov/what-we-do/research-tasks/risk-aware-machine-learning-for-resilient-space-exploration/>

[S16] NASA. "How NASA Is Testing AI to Make Earth-Observing Satellites Smarter." 2025. <https://www.jpl.nasa.gov/news/how-nasa-is-testing-ai-to-make-earth-observing-satellites-smarter/>

[S17] NASA-HDBK-8739.23A, "NASA Complex Electronics Handbook for Assurance Professionals," 2016. <https://s3vi.ndc.nasa.gov/ssri-kb/static/resources/nasa-hdbk-8739.23.pdf>

[S18] Siracusano, G. et al. "Re-architecting Traffic Analysis with Neural Network Interface Cards." USENIX NSDI / Microsoft Research, 2022. <https://www.microsoft.com/en-us/research/publication/re-architecting-traffic-analysis-with-neural-network-interface-cards/>

[S19] "Line-rate botnet detection with FPGA SmartNIC feature extraction and anomaly detection." Computer Networks, 2024. <https://www.sciencedirect.com/science/article/pii/S1389128624006418>

[S20] U.S. Food and Drug Administration. "Marketing Submission Recommendations for a Predetermined Change Control Plan for Artificial Intelligence-Enabled Device Software Functions." Final Guidance, August 2025. <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/marketing-submission-recommendations-predetermined-change-control-plan-artificial-intelligence>

[S21] European Union Aviation Safety Agency. "Concept Paper: First usable guidance for Level 1 and Level 2 machine-learning applications." Issue 02, February 2023. https://www.easa.europa.eu/sites/default/files/dfu/easa_concept_paper_guidance_for_level_1and2_machine_learning_applications_proposed_issue_02_feb2023.pdf

[S22] NIST Center for AI Standards and Innovation. "Challenges to the Monitoring of Deployed AI Systems." NIST AI 800-4, March 2026. <https://doi.org/10.6028/NIST.AI.800-4>

[S23] "Edge machine learning for data acquisition and adaptive experiments at LCLS-II." Machine Learning: Science and Technology, 2024. <https://doi.org/10.1088/2632-2153/ad8ea8>

[S24] "FPGA-based closed-loop control and data acquisition for scientific experiments." 2022. <https://pmc.ncbi.nlm.nih.gov/articles/PMC9665961/>

[S25] NIST. "Considerations for Achieving Crypto Agility: Strategies and Practices." <https://www.nist.gov/publications/considerations-achieving-crypto-agility-strategies-and-practices-0>

[S26] NASA. "Cognitive Anti-jamming Satellite-to-Ground Communications on NASA's SCA_N Testbed." 2018. <https://ntrs.nasa.gov/citations/20190001915>

[S27] "A FPGA-based Fast Converging Digital Adaptive Filter for Real-time RFI Mitigation on Radio Telescopes." 2018. <https://arxiv.org/abs/1805.06376>

[S28] CERN CMS. "Continual Learning in the CMS Phase-2 Level-1 Trigger." CMS-DP-2023-022. <https://cds.cern.ch/record/2859651>

[S29] "Deep Reinforcement Learning on FPGA for Self-Healing Cryogenic Power Amplifier Control." IEEE Open Journal of Circuits and Systems, 2023. <https://ieeexplore.ieee.org/document/10143969/>

[S30] ESA. "Dynamically Reconfigurable Processing Module (DRPM)." https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Onboard_Data_Processing/Dynamically_Reconfigurable_Processing_Module_DRPM

Final conclusion

The R/S applications most likely to benefit from online learning are those in which the world changes faster than engineers can enumerate modes: embodied robots, contested or mobile radio/radar environments, aging analog hardware, industrial concept drift, changing traffic/workloads, distant spacecraft, and experiments whose next action depends on the last measurement.

In those systems, **reprogrammability and learning are complementary but operate at different layers**:

- learning updates A0/A1 values continually;
- overlays provide restricted structural flexibility at A2;
- a governor selects a prevalidated A3 configuration occasionally;
- A4 design generation stays in an offline verification loop; and
- A5 physical reconfiguration is paired with context/model adaptation.

That layered separation delivers most of the benefit of auto-adaptation without turning the FPGA configuration—and the entire safety case—into an uncontrolled learned variable.