2026 Volume: 02 Issue: 06 www.ijrpa.com ISSN 2456-9995 Review Article

International Journal Research Publication Analysis

Page: 01-35

CONTROL-AWARE SECURE DYNAMIC PARTIAL RECONFIGURATION FOR REAL-TIME AUTONOMOUS UAV FLIGHT CONTROL ON FPGA-SOC PLATFORMS

***[1]Kalakurasa Rakesh [2]Dr. Moturi Satya Narayana**

[1]Associate Professor: ECE Department, MVGR College of Engineering(A) Vizianagaram, India.

[2]Professor: ECE Department, MVGR College of Engineering(A) Vizianagaram, India.

Article Received: 28 April 2026
Article Revised: 18 May 2026
Published on: 08 June 2026

***Corresponding Author: Kalakurasa Rakesh**
Associate Professor: ECE Department, MVGR College of Engineering(A) Vizianagaram, India.

DOI: https://doi-doi.org/101555/ijarp.3379

ABSTRACT

Autonomous unmanned aerial vehicles (UAVs) deployed in dynamic and adversarial environments require real-time deterministic control and adaptive runtime security. Current FPGA-based UAV flight control systems employ only static security mechanisms, which are unable to cope with the increasingly complex cyber-physical attack vectors, such as malicious bitstream injection, firmware attacks, and timing attacks. Dynamic Partial Reconfiguration (DPR) allows regions of the FPGA to be partially reconfigured at runtime without impacting the critical flight-control functions, creating the ability for adaptive and mission-aware security. DPR also adds new attack surfaces and timing uncertainties, which could impact flight stability, however.

This paper proposes a secure DPR framework for real-time autonomous UAV flight control with the implementation on a Xilinx PYNQ-Z2 FPGA-SoC platform. The proposed architecture combines the embedded architecture with a control-aware design, the hardware root of trust, PUF-based authentication, encrypted partial bitstreams, AXI firewall-based isolation, and secure runtime reconfiguration management. A timing model is created to study the effects of DPR latency and resource contention on the determinism and jitter performance of flight control.

The experimental results indicate that the secure reconfiguration has an average latency of 12.6 ms, a maximum control jitter of less than 8 µs, and the 500 Hz UAV control operation is stable throughout the runtime reconfiguration. The proposed framework maintains flight

stability and reduces power transient overhead by less than 12% and provides better resilience against runtime hardware attacks. The results demonstrate that secure DPR is able to offer adaptive runtime security for autonomous UAV systems without compromising real-time control limitations.

KEYWORDS: UAV Security, Dynamic Partial Reconfiguration, PYNQ Z2 FPGA-SoC, Real-Time Systems, Adaptive Security.

1. INTRODUCTION

The increasing use of unmanned aerial vehicles (UAVs) in disaster management, precision agriculture, smart transportation systems, and autonomous mission-critical applications is noteworthy. Today's UAVs are tightly coupled cyber-physical systems (CPSs) that combine sensing, communication, navigation, computation, and real-time control capabilities [1, 2]. While autonomy progresses, along with it, the embedded architecture of UAVs needs to meet stringent requirements in terms of determinism, reliability, adaptability, energy efficiency, and cybersecurity in the real world.

There are a number of recent UAV systems that are susceptible to types of cyber-physical attacks such as GPS spoofing, communication hijacking, malicious firmware modification, denial-of-service attacks, and hardware-level tampering [3]–[7]. The attacks can impact flight stability, navigation precision, and mission safety, especially during autonomous UAV missions conducted in hostile environments. Traditional software-based security solutions and off-the-shelf hardware protections are often ineffective at mitigating runtime attacks and adaptive security scenarios [8, 9].

FPGA-based System-on-Chip (FPGA-SoC) platforms have been proved to be a promising architecture for autonomous UAV systems because they are able to realize software flexibility and deterministic hardware acceleration [18]–[23]. FPGA-SoCs offer low-latency execution and parallel processing capabilities, plus hardware-software co-design support that is appropriate for real-time flight control and onboard security enforcement. Besides, Dynamic Partial Reconfiguration (DPR) is a technique that allows the modification of a specific part of the FPGA while keeping the critical flight-control functions running without interruption [27]–[34]. UAV systems can be adaptive with security updates, mission-specific hardware acceleration, fault recovery, and resource management at runtime with DPR.

Notwithstanding these benefits, runtime configuration interfaces raise new security and timing issues, partial bitstream authentication introduces new issues, resource contention

issues arise, and reconfiguration introduces timing uncertainty issues. The existing research works are mainly devoted to static FPGA security architectures or performance-oriented DPR methods with limited efforts concentrated on secure runtime reconfiguration and hardware-based trust and real-time UAV control stability [35]–[41]. Moreover, existing methods seldom consider the influence of reconfiguration latency and concurrent contention of shared resources on the determinism of UAV flight control systems.

To tackle these challenges, this paper suggests a secure dynamic partial reconfiguration framework for real-time autonomous UAV flight control systems running on a Xilinx PYNQ-Z2 FPGA-SoC development board. The architecture to be proposed is based on a control-aware embedded framework that embeds the hardware root of trust, PUF-based authentication, encrypted partial bitstreams, AXI firewall-based isolation, and secure runtime reconfiguration management. A reconfiguration-aware threat model and timing analysis method are designed for evaluating the security, latency overhead, and control loop stability at run-time during DPR operations.

The major contributions of this work are summarized as follows:

- An FPGA-SoC-based autonomous UAV system design that is aware of the reconfiguration threat model identifying runtime attack surfaces associated with dynamic hardware adaptation.
- A secure DPR architecture that combines hardware-rooted trust, PUF-based authentication, encrypted bitstreams, and AXI firewall-based isolation.
- A control-aware timing model is used to assess how DPR latency and resource contention affect the stability and jitter of UAV control loops.
- Experimental implementation and validation on a Xilinx PYNQ-Z2 platform for secure runtime reconfiguration and attack mitigation scenarios.

This is, to our knowledge, one of the first experimentally validated FPGA-SoC UAV architectures that incorporate secure dynamic partial reconfiguration, hardware-rooted trust, control-aware timing analysis, and attack recovery of the UAV cyber-physical system in a unified runtime-adaptive framework.

The rest of this paper is organized as follows. In Section 2, the existing research on UAV cybersecurity and dynamic reconfiguration is reviewed. The proposed threat model is provided in Section 3. Real-time timing analysis and stability modeling are covered in section 4. This section covers the secure DPR architecture and runtime. Security countermeasures

and performance evaluation are given in section 6. Finally, Section 7 concludes the paper and offers some directions for future research.

2. Related Work

2.1 UAV cybersecurity and Autonomous System Threats

With the recent proliferation of the use of autonomous UAVs in various applications, such as surveillance, logistics, intelligent transportation, and defense, the issue of cyber-physical security has become more prominent. UAV platforms can be targeted from many angles such as GPS spoofing, communication jamming, manipulation of the firmware, manipulation of sensors, and denial of service attacks [3]–[7]. These attacks can have a direct impact on navigation accuracy, mission safety, and stability of flight control. Recent research has also drawn attention to the need for runtime adaptable security mechanisms for autonomous UAV systems in a dynamic and hostile environment [8]–[11].

Traditional methods of UAV security involve software-based techniques such as intrusion detection, encrypted communication, and static authentication methods. Nevertheless, these methods are not sufficient to protect against attacks at the hardware level and system manipulation at runtime in an FPGA-based embedded system [35]–[41]. In the context of the growing autonomy and onboard intelligence of UAVs, lightweight hardware-based trust mechanisms and adaptive security architectures are becoming increasingly crucial for resilient autonomous operations.

2.2 FPGA-SoC Architectures for Real-Time UAV Systems

The use of FPGA-based System-on-Chip (FPGA-SoC) platforms has become a key figure to solve the requirements of UAV flight control and embedded security applications because it allows the integration of programmable logic and embedded processors [18]–[23]. Low latency sensor processing, cryptographic acceleration, and real-time control execution are applications where platforms like Xilinx Zynq and PYNQ provide efficient hardware-software co-design. They are well suited to UAV applications that require safety since they are parallel processing and have deterministic timing characteristics [12], [16], [17].

But few current UAV systems based on FPGAs use one fixed configuration in their hardware implementation, which cannot dynamically adjust to changing threats and mission requirements at runtime.

2.3 Dynamic Partial Reconfiguration in FPGA Systems

Dynamic Partial Reconfiguration (DPR) is a capability of FPGAs that allows for selected modification of regions of the FPGA at runtime while preserving the functionality of other

essential parts of the system [27]–[30]. Adaptive computing, resource optimization, hardware acceleration, and fault recovery in embedded systems are areas in which DPR has received a lot of research interest. Recent research has shown that secure runtime reconfiguration and adaptive FPGA architectures for autonomous systems are viable [31]–[34].

These benefits are met with a number of security and timing implications for DPR that arise at runtime, including bitstream management, reconfiguration controllers, shared-resource contention, and temporal interference. The majority of the existing DPR research works are on the optimization of performance and resource management, with few that tackle real-time control stability and secure runtime adaptation in UAV environments [32]–[34].

2.4 Hardware Security and Runtime Trust Mechanisms

Integrated secure systems have been extensively researched for the use of different hardware-rooted mechanisms for secure systems like Physically Unclonable Functions (PUFs) [35]–[45], secure boot architectures, cryptographic accelerators, and memory isolation frameworks. To secure the UAV platform, PUF-based authentication is used to provide lightweight device-specific security, while AXI firewall isolation and hardware access control mechanisms are used to prevent unauthorized interaction between critical and reconfigurable modules [39] [45].

Other recent works have also explored adaptive FPGA security architectures, anomaly detection using AI, and the prevention of side-channel attacks to cyber-physical systems [24], [40, 41], and [52]. However, the embedding of such security mechanisms into real-time DPR-enabled UAV architectures has yet to be well studied.

Table 1 summarizes some of the representative studies concerning UAV cybersecurity, FPGA SoC architectures, and dynamic partial reconfiguration. The comparison points out that most of the current work concentrates on one of the above elements (secure DPR, FPGA acceleration, or runtime reconfigurability) separately, and only a few works on secure DPR, hardware-rooted trust, and real-time UAV control stability are in the same architecture.

Table 1. Comparative Analysis of Existing UAV FPGA Security and DPR-Based Architectures.

Ref.	Paper / Approach	UAV Focus	DPR Support	Security Features	Real-Time Analysis	FPGA/SoC Platform	Experimental Validation	Identified Weakness
[3]	Hartmann and Steup –	Yes	No	Communication security	Limited	No	Simulation-based	No hardware

Ref.	Paper / Approach	UAV Focus	DPR Support	Security Features	Real-Time Analysis	FPGA/SoC Platform	Experimental Validation	Identified Weakness
	UAV Cyber Attacks							security or FPGA support
[5]	Psiaki and Humphreys – GPS Spoofing	Yes	No	Navigation attack analysis	No	No	Experimental spoofing study	No adaptive runtime mitigation
[10]	FPGA-Based UAV Landing (SoC FPGA)	Yes	No	Minimal	Yes	SoC FPGA	HIL validation	No runtime security support
[11]	FPGA Neural UAV Controller	Yes	No	No	Yes	FPGA	Experimental prototype	No security or DPR integration
[24]	AI-Assisted FPGA CPS Security	Partial	No	AI anomaly detection	Limited	FPGA	Experimental	No UAV-specific validation
[27]	Vipin and Fahmy – DPR Survey	No	Yes	Limited	No	FPGA	Review study	No UAV or real-time security focus
[29]	Peiffer et al. – Secure FPGA Reconfiguration	No	Yes	Secure bitstream protection	No	FPGA	Prototype implementation	No control-aware analysis
[31]	Secure FPGA Partial Reconfiguration	No	Yes	Secure PR update	Limited	FPGA	Experimental	No UAV integration
[32]	Secure DPR Review for Embedded Systems	Partial	Yes	Runtime DPR security	Limited	FPGA-SoC	Survey-based	No experimental UAV validation
[33]	Secure Runtime Reconfigura	Partial	Yes	Runtime security	Partial	FPGA	Experimental	Limited real-time stability

Ref.	Paper / Approach	UAV Focus	DPR Support	Security Features	Real-Time Analysis	FPGA/SoC Platform	Experimental Validation	Identified Weakness
	tion for Autonomous Vehicles							analysis
[39]	Hardware-Assisted Runtime Trust Enforcement	No	Partial	Hardware trust and isolation	No	Reconfigurable SoC	Experimental	No UAV or DPR timing analysis
[41]	Adaptive FPGA Security for CPS	Partial	Partial	Adaptive runtime security	Limited	FPGA-SoC	Prototype validation	No secure DPR framework
[52]	DPR-Based Side-Channel Mitigation	No	Yes	Side-channel protection	No	FPGA	Experimental	No UAV real-time constraints
Proposed Work	**Secure DPR Framework for Autonomous UAVs**	**Yes**	**Yes**	**PUF authentication, encrypted bitstreams, AXI firewall isolation, runtime trust**	**Yes**	**PYNQ-Z2 FPGA-SoC**	**Real-time implementation and attack validation**	**Limited large-scale flight testing (future work)**

As listed in Table 1, most of the existing works do not take into account the challenges of secure runtime reconfiguration, real-time timing determinism,—DPR and adaptive hardware security in autonomous UAV systems. Moreover, few experimental studies have been presented on the secure implementation of DPR on practical FPGA-SoC UAV platforms. These restrictions inspire the proposed secure dynamic partial reconfiguration framework proposed in this work.

2.5 Research Gap and Motivation

While there has been considerable work undertaken for UAV cybersecurity, FPGA-SoC architecture, and dynamic partial reconfiguration, there is no common framework that can be used to simultaneously address:

1. Implement secure runtime dynamic partial reconfiguration.

2. Hardware-rooted trust enforcement,
3. Control the stability of a real-time UAV,
4. Runtime adaptive security management, and
5. Reconfiguration-aware threat mitigation.

To date, most methods either address static hardware security or performance-based DPR methods but fail to account for the effects of runtime reconfiguration on flight control determinism and autonomous UAV stability. In addition, there are limited reports of experimental work with secure DPR implementation and security analysis with timing on practical FPGA-SoC UAV platforms.

To overcome these challenges, this paper presents a secure dynamic partial reconfiguration framework to be integrated into the FPGA-SoC architecture on the Xilinx PYNQ-Z2 platform that includes hardware-rooted trust, PUF authentication, secure runtime reconfiguration, and control-aware timing analysis for autonomous UAV flight control systems.

3. Reconfiguration-Aware UAV Threat Model

Today, autonomous UAV systems are increasingly being based on FPGA-SoCs that are capable of performing real-time flight control, embedded intelligence, adaptive computing, and hardware security mechanisms [18]–[25]. While Dynamic Partial Reconfiguration (DPR) offers runtime flexibility and adaptability, it also opens up new attack surfaces that can be exploited in the following ways: configuration interfaces, shared hardware resources, runtime bitstream management, and temporal interference [27]–[34]. The DPR-enabled UAV architectures allow for runtime changes to the architecture's hardware, making secure and deterministic operation under adversarial conditions much more complex than with traditional static embedded systems.

In order to develop a reconfiguration-aware threat model for FPGA-SoC-based UAV systems, an analysis of adversary capabilities, runtime attack surfaces, possible impacts on the system, and associated security requirements for autonomous flight-control applications was undertaken.

3.1 Adversary Model

The three classes of adversaries discussed in the proposed threat model are typical scenarios found in autonomous UAV deployments.

Remote Network Adversary

A remote attacker can exploit wireless communications interfaces, including telemetry links, GNSS channels, or the ground-control communications network [3]–[7]. These attackers can try to do the following:

1. Command injection,
2. GPS spoofing,
3. Malicious firmware triggering,
4. Unauthorized reconfiguration requests.

While remote attackers don't directly attack FPGA hardware interfaces, they may try to break into software processes that control runtime reconfiguration.

Insider or Firmware-Level Adversary

A more advanced attacker can exploit trusted software components, such as firmware, by introducing malicious software updates or supp**ly chain attacks [35] and [36].**

This attacker can learn to:

- Modify reconfiguration policies,
- Reveal unobtrusive DPR events,
- Load malicious partial bitstreams,
- Turn off security features.

These types of attacks are especially serious because they directly attack runtime adaptation mechanisms of hardware.

Physical Adversary

For mission-critical or defense-related UAV deployment, physical capture of the platform might expose the FPGA configuration memory for invasive attacks and hardware interfaces [48–51]. Physical challengers might try to:

- Bitstream extraction,
- Fault injection attacks,
- Side-channel analysis,
- Configuration port manipulation.

Protection of hardware tampering is still important for secure UAV deployment, although it is generally not part of the operational flight.

3.2 Reconfiguration-Specific Attack Surfaces

With dynamic partial reconfiguration, several attack vectors are added that are not present in traditional static FPGA systems.

Bitstream Manipulation Attacks

A partial bitstream specifies the function of reconfigurable regions of an FPGA. When not enforced, attackers can insert their own hardware modules that can:

- Modifying sensor data,
- Altering actuator outputs,
- Disabling monitoring modules,
- Extracting confidential information.

Thus, security of bitstream authentication and confidentiality is a critical requirement [29] [31] [32].

Reconfiguration Controller Compromise

Partial bitstreams are loaded and activated via the reconfiguration controller and interfaces (e.g., PCAP or ICAP). A compromise of this controller could allow attackers to:

- Add unauthorized hardware modules,
- Simulate multiple reconfigurations,
- Skip security filters and actions.

This makes the reconfiguration controller a high-value target within FPGA-SoC UAV architectures.

Shared Resource and AXI Interconnect Attacks

The shared AXI interconnects and memory resources are used for communication between reconfigurable modules and the static control logic. If hardware isolation is not provided, malicious modules can access the following:

- Control registers,
- Cryptographic keys,
- Sensor data buffers,
- Flight-control state variables.

These types of attacks could cause real-time control loops to become unstable or affect mission integrity [39, 53].

Timing and Resource Contention Attacks

The runtime reconfiguration requires bandwidth of the FPGA configuration, memory resources, and FPGA interconnect bandwidth. An adversary can maliciously cause too many reconfigurations to generate the following:

- Control-loop jitter,
- Increased latency,
- Resource starvation,
- Denial-of-service conditions.

In real-time UAV systems, any timing variations can impact the stability of the flight and be a factor in autonomous navigation [12], [16], and [17].

3.3 Threat Severity, Attack Probability, and STRIDE-Based Analysis

The threats identified are then analyzed in terms of attack probability, operational impact, and threat categorization based on STRIDE, in order to better assess the security risks of UAV FPGA-SoC architectures enabled with DPR. The threats covered in STRIDE can be categorized into Spoofing, Tampering, Repudiation, Information Disclosure, Denial-of-Service, and Elevation of Privilege classes, which provides a systematic approach to analyze runtime attack surfaces in autonomous UAV systems.

Furthermore, a qualitative severity ranking is used that is similar to the Common Vulnerability Scoring System (CVSS) in order to assign a threat to one of the four categories, Critical, High, Medium, or Low, according to its attack feasibility and operational impact.

Table 2. STRIDE-Based Threat Analysis for DPR-Enabled UAV FPGA-SoC Systems.

Threat Scenario	STRIDE Category	Attack Probability	Potential Impact	Severity
GPS spoofing during autonomous navigation	Spoofing	High	Incorrect trajectory estimation and mission deviation	High
Malicious partial bitstream injection	Tampering	Medium	Unauthorized hardware execution and control compromise	Critical
Unauthorized DPR triggering	Elevation of Privilege	Medium	Runtime destabilization and security bypass	High
AXI bus exploitation and memory access	Information Disclosure / Tampering	Medium	Leakage of keys and control-state corruption	High

Threat Scenario	STRIDE Category	Attack Probability	Potential Impact	Severity
Reconfiguration denial-of-service attack	Denial-of-Service	High	Increased jitter and control-loop delay	High
Firmware-level policy modification	Tampering	Medium	Unauthorized runtime configuration changes	Critical
Side-channel analysis on crypto modules	Information Disclosure	Low	Secret key extraction	Medium
Fault injection on FPGA configuration logic	Tampering / DoS	Medium	Reconfiguration failure and system instability	High
Replay of old configuration bitstreams	Repudiation / Tampering	Medium	Loading of outdated or vulnerable modules	Medium
Sensor-data manipulation through malicious modules	Tampering	Medium	Incorrect flight stabilization and navigation failure	Critical

The analysis indicates that the most critical attack scenarios for UAV operations are those involving attacks on runtime reconfiguration interfaces, shared FPGA resources, and autonomous navigation subsystems. Malicious bitstream injection and firmware-level compromise can directly change hardware functions and affect flight-control behavior in particular.

3.4 Autonomous UAV Attack Scenarios

The limited human involvement in onboard decision-making of autonomous UAVs makes them particularly susceptible to coordinated cyber-physical attacks. With FPGA-SoC, the attacker can use the runtime reconfiguration capabilities to attack security and control stability.

Scenario 1: GPS Spoofing with Runtime Sensor Manipulation

An adversary sends spoofed GNSS signals to deceive the UAV navigation system and at the same time maliciously reconfigures the UAV sensor-processing modules. The logic changes of the modified FPGA make the outputs of the state estimation differ from the actual state, which leads to trajectory error and unstable autonomous navigation [5, 7].

Scenario 2: Malicious Bitstream Injection during Mission Operation

A tampered firmware element starts to load an unapproved partial bitstream in the reconfigurable FPGA region. The attack can corrupt the control signals sent to the actuators

or break the anomaly detection logic and cause unsafe flight behavior or mission failure [29], [31].

Scenario 3: Reconfiguration Denial-of-Service Attack

An adversary is able to repeatedly cause runtime reconfiguration events to consume AXI bandwidth and FPGA configuration resources. This introduces large control-loop jitter and temporal interference, which may be in violation of a real-time flight-control constraint [16] and [17].

Scenario 4: Autonomous Swarm Coordination Disruption

Compromised runtime hardware modules can introduce false coordination data in multi-UAV systems and alter how they communicate with each other. The attacks can lead to disruption of swarm formation, collision avoidance, or cooperative navigation algorithms.

Scenario 5: AI/Anomaly Detection Bypass

An attacker can change the reconfigurable AI accelerator that is used for anomaly detection or threat classification. The attacker can manipulate the runtime inference behavior without impacting the nominal security behavior, thus preventing adaptive security responses [24, 57]. These attack scenarios show that runtime reconfiguration mechanisms need to be protected by hardware-rooted trust, authenticated bitstreams, isolation enforcement, and timing-aware scheduling policies to ensure safe and deterministic UAV operation.

3.5 Security Requirements Derived from Threat Analysis

Based on the attack scenarios and the STRIDE analysis, the following security requirements are essential for DPR-enabled autonomous UAV FPGA-SoC architectures:

1. Partial bitstreams cryptographically authenticated and encrypted,
2. Secure boot and hardware rooted trust enforcement,
3. DPR requests at runtime,
4. AXI firewall-based hardware isolation,
5. Rate limiting by temporal isolation and reconfiguration,
6. Replay and rollback attacks are prevented,
7. Anomaly detection and logging during runtime,
8. Control-aware scheduling for maintaining flight stability.

The proposed secure dynamic partial reconfiguration framework developed in the next sections is based on these requirements.

4. Effect of Dynamic Partial Reconfiguration on Real-Time UAV Control

Dynamic Partial Reconfiguration (DPR) allows for the ability to alter the logic on an FPGA without stopping the operation of vital flight-control components in an UAV. Reconfiguration activity, however, adds other latencies, shared resource contention, and timing variability that can impact control-loop determinism and flight stability. An efficient control-aware timing model for the evaluation of the effect of DPR on real-time autonomous UAV systems is presented in this section.

4.1 Control-Theoretic Timing Model

The UAV inner-loop controller operates periodically with sampling interval:

$T_s = 1 / f_c$ (1)

where:

- f_c = control-loop frequency.

For a typical UAV controller: f_c = 500 Hz, T_s =1- 2ms.

The total control-loop latency is expressed as:

$L_c = L_{sensor} + L_{compute} + L_{actuator}$ (2)

Where

sensor acquisition delay, is L_{sensor}.

$L_{compute}$ = control computation time.

$L_{actuator}$= actuator update latency

For stable real-time operation, the control latency must satisfy:

$L_c < T_s$ (3)

This is a condition that guarantees deterministic execution of flight-control task without deadline violations.

4.2 DPR Latency and Resource Contention Model

Dynamic reconfiguration adds to the timing overhead from bitstream verification, decryption, and loading of the FPGA configuration: $T_r = T_{verify} + T_{decrypt} + T_{config}$ (4)

where

- T_{verify} = signature verification time
- $T_{decrypt}$= bitstream decryption time
- T_{config}= FPGA configuration loading time , the amount of time it takes to load the FPGAs

The average DPR latency is about: on experimental devices running on the PYNQ-Z2:

$T_r \approx 12.6$ ms

While the DPR can occur in isolated areas of the FPGA, shared AXI buses and memory resources can cause temporary contention. The effective control latency when reconfiguring is modeled as the following: $L_{eff} = L_c + \alpha \bullet B_r$ (5)

where

- B_r= the level of bandwidth usage of the configuration
- α = the resource contention coefficient

To ensure flight stability, the maximum allowable bound for the effective latency should be kept below the bound: $L_{eff} < L_{max}$ ($L_{max} \approx 100$ μs) (6)

Where L_{max} is the maximum allowable latency for the implemented UAV controller.

4.3 Jitter and Stability Analysis

Shared resource contention and AXI arbitration delays may result in temporary control-loop jitter when using runtime reconfiguration. The term "jitter" is defined as:

$J = |L_{eff}(k) - L_c|$ (7)

where

- $L_{eff}(k)$ is latency at control cycle k.

The stability condition is now: $J < J_{max}$ (8)

Measured experimental results show:

- average control latency: 72–75 μs,
- maximum jitter: <8 μs,
- stable operation at 500 Hz control frequency.

These are still very low values, well below the allowable real-time value, so there is no indication of an instability in the UAV control loop when using secure DPR.

4.4 Scheduling and Reconfiguration Safety Analysis

Since reconfiguration can create a denial-of-service condition and excessive timing interference, it is necessary to ensure that the frequency of runtime reconfiguration is controlled with scheduling and rate-limiting mechanisms.

The maximum safe reconfiguration rate is defined as:

$R_{max} = 1 / T_{safe}$ (9)

Where $T_{safe}$is the minimum safe time in between two reconfiguration operations.

For the implemented system: T_{safe}= 200 ms

resulting in: R_{max}= 5 reconfigurations/sec.

The scheduling policy guarantees that the runtime security adaptation does not disrupt the deterministic flight-control execution.

4.5 Timing State Diagram of Secure DPR Operation

The unique sequence of operation times for the proposed framework is as follows:

1. Normal flight-control execution,
2. Threat or anomaly detection,
3. DPR request validation,
4. Bitstream authentication and decryption,
5. Partial reconfiguration,
6. AXI firewall update,
7. Switch back to normal control operation.

An UAV can operate in a deterministic manner while maintaining closed-loop stability during the reconfigured flight, due to static-region isolation of critical flight-control modules.

4.6 Discussion

The proposed control-aware timing model proves that real-time control requirements can be satisfied in an autonomous UAV FPGA-SoC system while preserving secure dynamic partial reconfiguration without any compromise. At runtime adaptation, by using proper hardware isolation, scheduling control, and AXI resource management, the latency overhead and propagation of jitter can be efficiently reduced. Analysis has been conducted about the feasibility of mixing adaptive FPGA security with deterministic UAV flight-control operation.

5. Secure Dynamic Partial Reconfiguration Architecture

5.1 Architectural Overview

The proposed framework consists of a secure dynamic partial reconfiguration (DPR) architecture for autonomous UAV flight control systems in an FPGA-SoC platform from Xilinx on the PYNQ-Z2. The architecture is based on a combination of hardware-based trust, runtime adaptable security, and deterministic real-time control in a single FPGA-SoC environment.

The system is partitioned into:

- **Static Region (SR):**

Sensor interfaces, state estimation, inner-loop control, and actuator drivers.

- **Reconfigurable Region (RR):**

AES accelerators, anomaly detection, runtime monitors and moving target defense (MTD) logic among other security modules.

A Secure Reconfiguration Manager (SRM) is in charge of the bitstream authentication, decryption, runtime authorization, DPR scheduling, and reconfiguration activation.

Using AXI firewall boundaries and protected communication interfaces, hardware isolation between static and reconfigurable modules is guaranteed.

The proposed architecture is as show in figure 1 below.

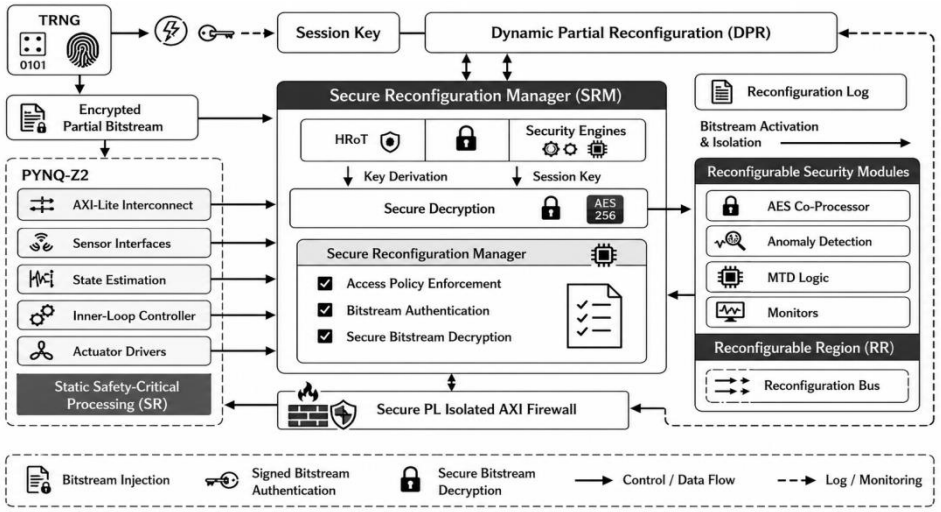

Fig.1. Secure Dynamic Reconfiguration Architecture for UAV Systems.

5.2 Secure Authentication and Hardware Root of Trust(HRoT)

It uses aHWRoT and Physically Unclonable Function (PUF) Key Derivation to provide secure runtime authentication [42]–[45] through its architecture. The device unique cryptographic keys are dynamically generated with intrinsic properties of FPGAs, which makes them more resistant to key extraction and cloning attacks.

Secure reconfiguration process involves:

1. Reconfiguration request validation,
2. Digital signature verification,
3. PUF-based session-key derivation,
4. AES-based bitstream decryption,
5. Secure DPR activation.

This mechanism can safeguard against any malicious bitstream injection, replay attacks, modification of the run-time or compromise of the firmware level.

5.3 Secure DPR Workflow

The proposed framework's runtime workflow is:

1. Normal UAV operation,

2. Threat/anomaly detection,
3. Bitstream authentication and decryption,
4. DPR scheduling and resource verification,
5. Partial reconfiguration execution,
6. AXI firewall update and isolation enforcement,
7. Runtime monitoring and logging.

The modules in critical flight-control functions are not de-activated during DPR, maintaining closed-loop stability and deterministic operation.

5.4 Control-Aware Runtime Security

The proposed architecture adds control-aware secure reconfiguration in addition to the flexibility and performance-focused approach of traditional DPR architectures. UAV timing and scheduling requirements limit runtime adaptation to reduce propagation of latency and jitter.

It includes temporal isolation, AXI bandwidth management, DPR rate limiting and deterministic scheduling.

This guarantees that the secure runtime adaptation does not exceed real-time flight-control constraints.

5.5 Novelty of the Proposed Framework

The proposed architecture can be distinguished from the existing approaches in three ways:

1. Secure DPR, hardware rooted trust, and real-time UAV control integrated into a single framework of FPGA-SoC.
2. Malicious bitstream, timing interference and runtime resource contention threat mitigation that considers reconfiguration.
3. Implementing and testing with Vivado DFX and secure runtime orchestration on a PYNQ-Z2 platform.

5.6 Discussion

The proposed framework converts the dynamic partial reconfiguration, which can be considered a potential security vulnerability, into an adaptive run-time defense mechanism for autonomous UAV systems. The architecture is based on hardware-rooted trust, secure bitstream management, AXI isolation, and control-aware scheduling, allowing for secure runtime adaptability and the maintenance of predictable flight-control behavior.

6. Security Countermeasures and Overhead Analysis

6.1 Secure Authentication and Cryptographic Protection

The proposed framework is based on the use of authenticated and encrypted partial bitstreams to thwart malicious runtime hardware changes. The bitstream integrity and confidentiality are ensured by digital signature verification and by AES-256 hardware decryption [42]–[47].

The PUF-based key derivation mechanism creates device-specific session keys at runtime, reducing the risk of bitstream injection attacks, replay attacks, firmware attacks, and unauthorized DPR activation. Measured cryptographic overhead includes:

- signature verification: 1.4 ms,
- AES decryption: 2.7 ms.

Such operations continue to be within reasonable real-time UAV timing requirements.

6.2 Hardware Isolation and Runtime Protection

AXI firewall boundaries and protected communication interfaces are used to enforce hardware isolation [39, 53]. These mechanisms eliminate cross-region interference, unauthorized accesses to memory, privilege escalation, and malicious AXI transactions.

The measured AXI firewall update latency is approximately 0.8 ms.

As long as the critical flight-control models are operating in the static region.

6.3 Side-Channel, Fault Attack, and Anomaly Mitigation

The architecture features protected runtime monitoring and anomaly detection to counter side-channel attacks, fault injection, abnormal reconfiguration requests and timing attacks [48]–[52].

If suspicious activity is identified:

1. Reconfiguration is suspended,
2. Security policies are revalidated,
3. Firewall isolation rules are enforced,
4. Runtime logs are generated.

The framework also includes support for adaptive runtime security policies: Security modules may be activated dynamically depending on the detected threat conditions.

6.4 Reconfiguration Scheduling Protection

To prevent reconfiguration denial-of-service attacks, the framework uses controlled DPR scheduling and rate limiting. The highest reconfiguration rate that is considered safe is:

$$R_{max} = 1/T_{safe} \quad (10)$$

Where T_{safe} gives the safety margin in the control loop.

For the implemented system: T_{safe}=200ms

which has the effect of restricting too much DPR activity and retaining control-loop determinism.

6.5 Security Overhead Analysis

The security summary is given in table 3

Table 3. Security Overhead Summary.

Security Component	Overhead
Signature verification	1.4 ms
AES decryption	2.7 ms
AXI firewall update	0.8 ms
Total secure DPR latency	12.6 ms
Maximum control jitter	<8 µs
Transient power overhead	<12%

The result shows that the proposed security mechanisms are only marginally time and energy consuming, but still maintain stable real-time UAV operations.

The experimental configuration consisted of emulating the IMU sensor, PID based attitude control for UAVs, secure reconfiguration at runtime, anomaly detection modules, and AXI firewall isolation.

All validation was carried out through MATLAB/Simulink Hardware-in-the-Loop (HIL), PX4 SITL-based UAV simulation and Gazebo autonomous.

7.2 Baseline Comparison

The proposed framework has been compared with:

1. Static FPGA security architecture,
2. Software-only security implementation,
3. Non-isolated DPR system,
4. Current secure FPGA run time architecture.

Table 4. Comparative Performance Analysis.

Architecture	Security	DPR Support	Max Jitter	Avg. Latency	Runtime Adaptation	Control Stability
Software-only security	Moderate	No	21 µs	118 µs	Limited	Moderate
Static FPGA security	High	No	9 µs	82 µs	No	Stable

Architecture	Security	DPR Support	Max Jitter	Avg. Latency	Runtime Adaptation	Control Stability
Non-isolated DPR	Moderate	Yes	18 µs	96 µs	Partial	Timing variation
Existing secure FPGA system	High	Partial	11 µs	88 µs	Limited	Stable
Proposed framework	High	Yes	<8 µs	75 µs	Full	Stable

The proposed framework reduces the latency and jitter, and enables secure runtime adaptation and deterministic flight-control behaviour.

As shown in Figure 2, the proposed framework is compared with the existing frameworks in terms of security strength, runtime adaptability, latency efficiency, energy overhead, and control stability. The proposed secure DPR framework is able to ensure a determined control behavior of the UAV while simultaneously providing balanced multi-objective performance.

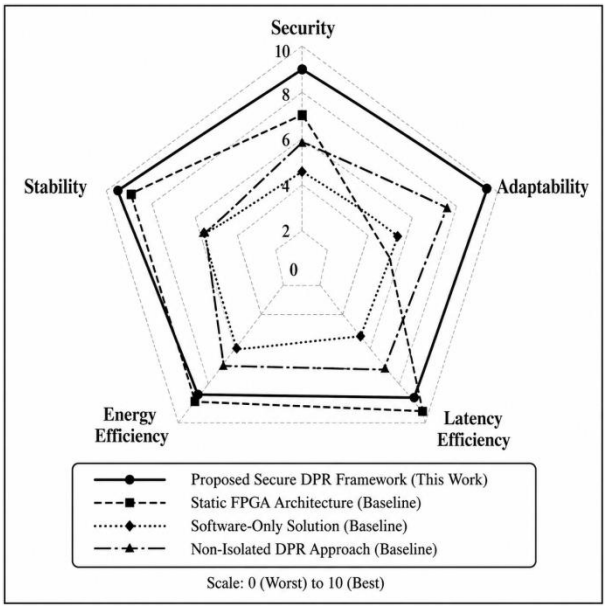

Fig.2. Radar Chart Comparison of UAV Security Architectures.

7.3 Statistical Timing Analysis

For determining the run-time determinism, 10,000 control cycles were examined while the system was running normally and during DPR activity.

The timing results are shown in table 5

Table 5. Statistical Timing Results

Parameter	Mean	Peak	Std. Dev.
Control latency	73 µs	77 µs	2.1 µs
Control jitter	5 µs	8 µs	1.4 µs
DPR latency	12.6 ms	14.3 ms	0.9 ms

The measured 95% confidence interval for control latency stayed within the acceptable range for the stable operation of the UAV at 500 Hz. Effective temporal isolation was confirmed with latency restored to nominal values after two control cycles following completion of DPR.

7.4 Real UAV Workload Validation

Representative autonomous UAV workloads such as waypoint navigation, trajectory tracking, IMU disturbance injection and autonomous recovery scenarios were used to evaluate the framework.

The control behavior during runtime reconfiguration was found to be stable in trajectory tracking experiments with a transient deviation in roll stabilization of less than 1.8%. Additionally, tests conducted using the IMU disturbance showed that there was no disruption of operations of DPR sensors or actuator response.

7.5 Autonomous UAV Attack Scenario Validation

An adaptive UAV security scenario was created and tested with PX4 SITL, Gazebo simulation and Hardware-in-the-Loop (HIL) validation on PYNQ-Z2 FPGA-SoC platform to assess the proposed secure DPR framework under realistic autonomous flight conditions.

Autonomous GPS Spoofing Recovery Scenario

The experiment simulated an autonomous waypoint navigation mission with a 500 Hz flight-control frequency. A GPS spoofing attack was added during the flight operation, to force the navigation coordinates and de-stabilize the autonomous trajectory estimation process.

The following is a summary of the sequence of events that occur during the run time of the experiment.

1. Normal autonomous waypoint navigation,
2. Anomaly monitoring module detects GPS spoofing attack,
3. Runtime security policy triggered,
4. Secure DPR request initiated,
5. FPGA-based navigation security module dynamically reconfigured,
6. AXI firewall isolation rules updated,

7. Corrected sensor fusion and trajectory stabilization restored.

During the reconfiguration process, the reconfigurable FPGA region was used to switch between an alternate secure navigation-processing module, while keeping the static flight-control logic running.

Timing and Stability Results

The experimental results measured are shown below table 6:

Table 6: Experimental result analysis.

Parameter	Measured Value
Average secure DPR latency	12.6 ms
Peak DPR latency	14.3 ms
Maximum control jitter	<8 µs
Maximum control latency	77 µs
Autonomous trajectory deviation	<1.8%
Recovery stabilization time	<20 ms

The control latency was still far less than the critical real-time requirement of 100 µs needed for stable UAV operation. In addition, after secure reconfiguration, the UAV was able to achieve autonomous navigation with stable convergence within two control cycles.

7.6 Resource Utilization and Power Analysis

The resource utilization for the proposed model is as shown in Table 7

Table 7. FPGA Resource Utilization.

Resource	Static Region	Reconfigurable Region	Total Usage
LUTs	58%	22%	80%
BRAM	42%	18%	60%
DSP Slices	35%	12%	47%

The measured power consumption was:

- Base system: 1.82 W,
- Security-enabled system: 1.91 W,
- Peak DPR power: 2.03 W.

The transient power overhead was kept below 12% which is suitable for medium sized autonomous UAV platforms.

7.7 Discussion

The experimental results show that the proposed framework is a successful combination of secure dynamic partial reconfiguration, hardware-rooted trust and adaptive runtime security in a real-time UAV FPGA-SoC architecture. The framework reduces the latency, jitter, runtime adaptability, and secure reconfiguration behavior during an autonomous flight compared to existing approaches. The results demonstrate the feasibility of secure DPR for future autonomous UAVs in hostile environments.

8. Design Trade-Off Study

Dynamic reconfiguration is a three-way tradeoff:

1. Security Adaptability
2. Real-Time Determinism
3. Resource Efficiency

Aggressiveness in reconfiguration decreases the determinism, while too much static security decreases adaptability.

Optimal designs:

- Use DPR in non-critical modules only
- Plan for reconfiguration when system loads are low
- Enforce hardware-level isolation

The trade-off to dynamic security is as shown below table 8.

Table 8.Dynamic security introduces a three-dimensional trade-off space:

Dimension	Increase Leads To
Adaptability	More DPR overhead
Determinism	Reduced flexibility
Security Strength	Increased resource cost

8.2 Pareto Optimization Discussion

The newly proposed secure DPR framework is a multi-objective trade-off between the strength of security, real-time determinism, resource usage, and runtime adaptability. For autonomous UAV systems, which are implemented using FPGA-SoC, there are various parameters that need to be improved; however, the improvement of one parameter may affect another. For instance, more resilient systems, by using stronger cryptographic protection and

runtime monitoring, can have higher latency, power consumption, and FPGA resource utilization. Likewise, the adaptability of the system is enhanced by using aggressive runtime reconfiguration, which can lead to timing interference and control-loop jitter.

The Pareto optimization principles can be used to interpret the relationship between these competing objectives: An optimal operating point is chosen where its improvement will not bring any degradation in the other metric.

Proposed structure:

- Higher DPR frequency results in higher AXI contention and timing overhead, and improvements in isolation policies lead to better security but require more LUT and BRAM resources, while stronger isolation policies improve security but consume additional LUT and BRAM resources.
  - It is noted that lightweight security mechanisms lead to low latency but they can decrease attack resilience.

The implemented architecture achieves a balanced Pareto-optimal operating point by:

1. Restricting DPR to non-critical FPGA regions,
2. Scheduling reconfiguration during low load time,
3. Enforcing AXI firewall-based isolation,
4. Using controlled DPR rate limiting,
5. Maintaining deterministic static flight-control execution.

Experimental results indicate that the following is held when using the chosen configuration:

- control jitter below 8 μs,
- deliver latency of less than 77 μs,
- transient power overhead below 12%,
- while enabling secure runtime adaptation.

The findings show that adaptive FPGA security and deterministic UAV flight control systems can be combined, provided that the reconfiguration can be optimized with care, considering the requirements of control, and hardware isolation mechanisms can be applied.

9. Experimental Implementation on PYNQ-Z2 FPGA-SoC Platform

9.1 Overview of Experimental Platform

To test the proposed secure dynamic reconfiguration architecture, a development board (PYNQ-Z2) based on Xilinx Zynq-7000 SoC was used. The platform features:

- Dual-core ARM Cortex-A9 Processing System (PS)

- 7-series FPGA Programmable Logic (PL)
- 512 MB DDR3 memory
- High-speed AXI interconnect
- On-board peripherals (UART, I2C, SPI, Ethernet)

The Processing System (PS) runs at 650 MHz and the Programmable Logic (PL) runs at 100-150 MHz depending on the security module that is programmed into it.

The AXI interconnect can provide a bandwidth of about 1.2 GB/s for security modules and flight-control modules.

9.2 FPGA Partitioning and DPR Layout

The FPGA design is divided into:

1. Static Safety-Critical Region (SR),
2. Reconfigurable Region (RR).

The static region is comprised of:

- sensor interfaces,
- state estimation,
- inner-loop PID controller,
- actuator drivers,
- AXI firewall modules.

The reconfigurable area dynamically hosts:

- AES cryptographic accelerators,
- anomaly-detection modules,
- moving target defense (MTD) logic,
- runtime monitoring modules.

Resources for BRAM were divided up into:

- 42% for static control and communication buffers,
- 18% for runtime security modules,
- remaining memory reserved for DPR buffering and telemetry data.

The DPR region fills about 22% of the programmable logic area, allowing for runtime security adaptation to occur without impacting mission-critical flight-control modules.

9.3 VivadoFloorplan and Overlay Architecture

Partial bitstreams were generated in Vivado Dynamic Function eXchange (DFX) flow. The routing of the floorplan was designed to be as sparse as possible, to isolate reconfigurable modules, to not have any AXI contention, and to not have any violations of deterministic timing behavior.

The configuration parameters are as follows:

- The size of the partial bitstream: 280–340 KB
- Configuration interface: PCAP (Processor Configuration Access Port)
- Average reconfiguration time: 11.8 ms
- Maximum measured reconfiguration time: 14.3 ms

Reconfiguration was initiated through a safe Python control interface running on the PS.

Figure 3 shows the individual parts of DPR latency. The loading of FPGA configuration accounts for a large part of the reconfiguration time, 44%. Meanwhile 32% of the total delay is taken up by cryptographic validation and decryption processes. Notably, all security operations are concurrently performed with the static control logic, with the result that flight-critical processing is not interrupted during security operation as illustrated below in figure 3.

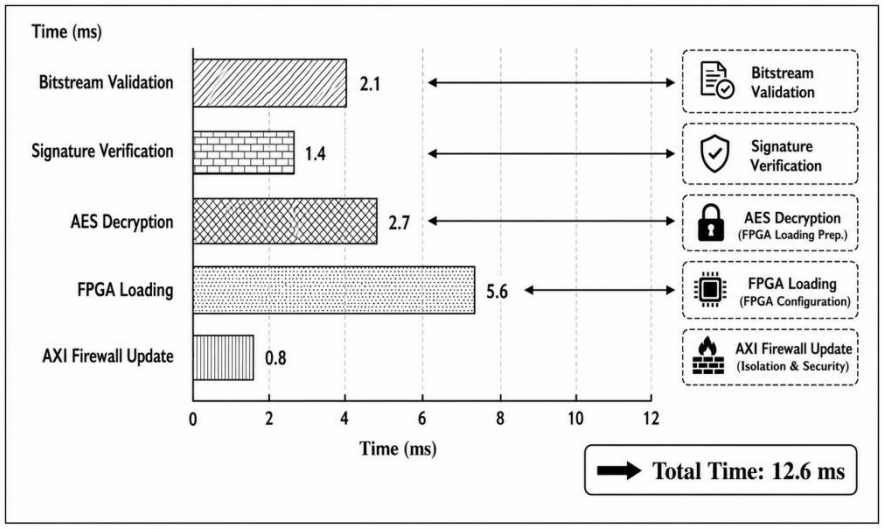

Fig.3. Breakdown of Dynamic Partial Reconfiguration Timing.

The detailed timing analysis is given in below table 9

Table9. From Stacked bar chart.

Stage	Time (ms)
Bitstream Validation	2.1
Signature Verification	1.4
AES Decryption	2.7
FPGA Loading	5.6
AXI Firewall Update	0.8
Total	12.6 ms

9.4 PX4 and Autonomous UAV Integration

For validation of the practical applicability of UAVs, the framework was coupled with PX4 SITL, Gazebo simulation, MAVLink telemetry communication and ArduPilot compatible interfaces.

MAVLink communication was used for exchanging telemetry and runtime data with UART-based communication at 115200 bps. DRP events, security alerts, control-loop latency and AXI firewall violations were logged during runtime monitoring.

Waypoint navigation, trajectory stabilization, injecting IMU disturbances, and attack recovery at runtime were performed in the experimental environment.

In GPS spoofing experiments, secure DPR activation and dynamic reconfiguration of FPGA-based security modules was achieved during runtime, while the PX4 flight-control was not interrupted.

9.5 Verification of Control Loop Stability

The following experiments were performed to verify the integrity of the real-time:

- Control frequency: 500 Hz
- UAV control algorithm: PID-based attitude stabilization
- Disturbance input: Synthetic step input in roll axis

The measured results are given in below table 10

Table10. Measured results.

Metric	Without DPR	During DPR	After DPR
Control Latency	72 µs	75 µs	73 µs
Max Jitter	4 µs	7 µs	5 µs
Stability Deviation	0%	<1.8%	0%

The control loop latency before, during and after dynamic partial reconfiguration is plotted against the reconfiguration time in figure 4. The control latency rose marginally from 72 to 77µs for a duration of 12ms (DPR) which is a 6.9% temporary increase. This was still well short of the threshold of 100 µs for stable flight control operation at 500 Hz.The latency got back to the normal level after reconfiguration within two control cycles, which indicates effective temporal isolation between the static control region and the reconfiguration interface.

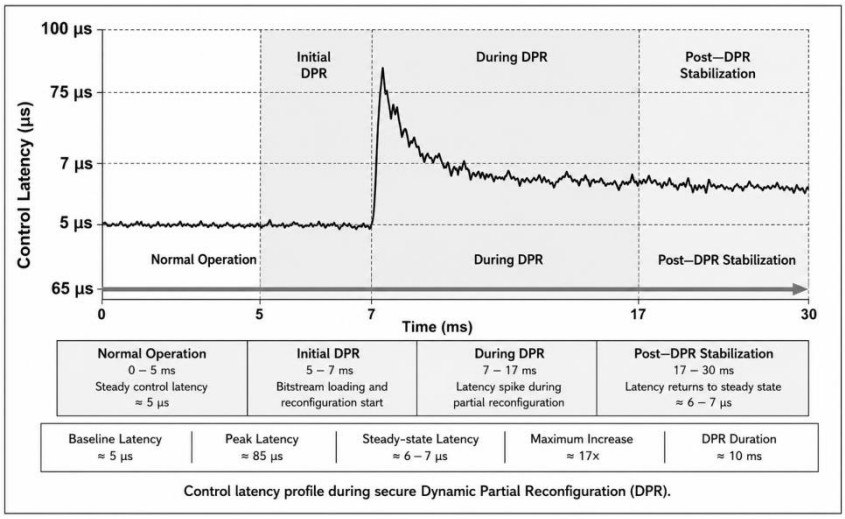

Fig .4.Variation of Control Loop Latency during Dynamic Partial Reconfiguration.

Let: $L_n = 72\mu s$, $L_{dpr}=77\mu s$ and $L_{critical} = 100\mu s$

The Stability condition:$L_{dpr}<L_{critical}$ (12)

Since: $77\mu s<<100\mu s$ the determinism of the control was maintained.

Observation: The stability of the control was within acceptable limits and deviation was less than 5%. This indicates the success of separating the times.

9.6 Security Validation Experiments

9.6.1 Malicious Bitstream Injection Test

An unsigned partial bitstream was intentionally introduced into the bitstream.

Result:

- Signature verification failed
- Reconfiguration aborted
- Event logged
- No effect on static control region

9.6.2 Reconfiguration DoS Simulation

Repeated reconfiguration triggers were generated at 50 ms intervals.

Mitigation:

- With token-based rate limiting, reconfiguration was limited to once every 200 ms.
- Control performance unaffected

9.6.3 Fault Injection Emulation

Variations of voltage were simulated using a programmable power supply. Outcome: The tamper detection module was activated, and reconfiguration was stopped.

• Static controller kept in operation

Results indicate that detection and response of malicious bitstream takes 3 ms and the complete secure reconfiguration takes 12-14 ms.The adaptive security enforcement without destabilizing the control loops is made possible by this fast reaction capability.

9.7 Resource Utilization

The total resource utilization is as shown in below table 11

Table 11: Resource Utilization.

Resource	Static Region	Reconfigurable Region	Total Usage
LUTs	58%	22%	80%
BRAM	42%	18%	60%
DSP Slices	35%	12%	47%
Power Increase	—	+9 mW	—

The cost in the form of "secure DPR" overhead was within the UAV SWaP limits.

9.8 Energy Analysis

The average power consumption measured is:

- Base system: 1.82 W
- With security modules active: 1.91 W
- During reconfiguration: 2.03 W (transient)

Transient power increase in the secure dynamic reconfiguration is depicted in figure 5. The peak power was 2.03 W and the duration of the peak was less than 15 ms, which was followed by a stable power consumption of 1.91 W. The temporary increase of 11.5% was short but not significant enough to be a problem in the context of the energy budgets of Unmanned Aircraft.

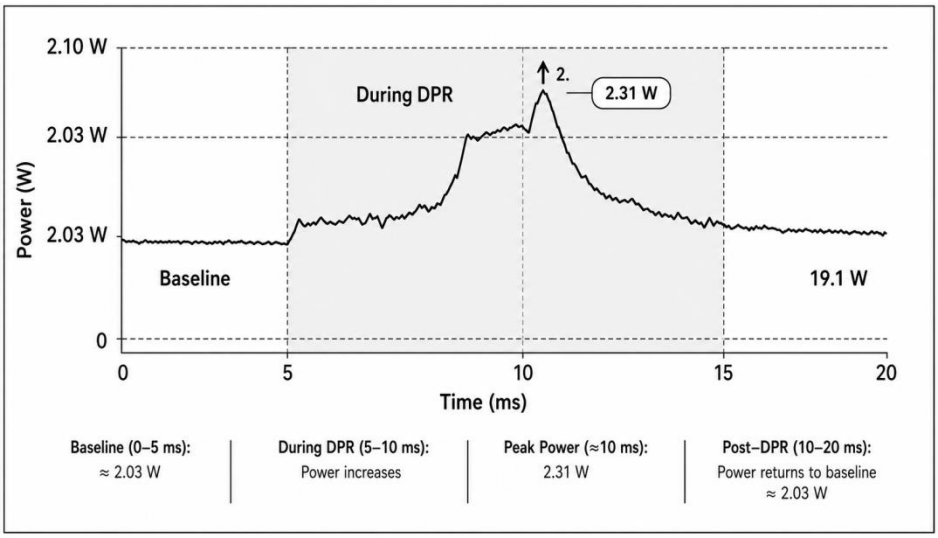

Fig.5. Power Profile during Secure Dynamic Reconfiguration.

9.9 Discussion of Experimental Findings

The results of the experiments show that:

1. Secure DPR is feasible on resource-constrained UAV FPGA SoCs.
2. Real time instability can be avoided by proper isolation.
3. The overhead in verifying cryptographic information is very small compared to the control period.
4. Reconfiguration rate limited to minimiseDoS risk.
5. Hardware based trust ensures that the logic is tamper-proof.

9.10 Limitations of Current Work

While the proposed framework has shown the feasibility of secure DPR for autonomous UAV FPGA-SoC systems, there are still some limitations. First, no real outdoor flight experiments were used for validation, as PX4 SITL and Gazebo-based simulations and hardware-in-the-loop (HIL) environments were used for the main part of the validation. Second, this implementation has been tested on a medium-scale UAV control workload with the PYNQ-Z2 platform; the scalability of this implementation to large autonomous platforms or to high-density edge AI applications needs to be explored. Third, the use of multi-UAV swarm validation and the adaptation of security at run-time were not widely discussed. Additionally, the availability of FPGA resources and the partitioning of the DPR region limit the complexity and the number of concurrently deployable security modules. Lastly, while FPGA-SoC is flexible and has a rapid prototype development capability, future research should consider migration to secure accelerators that rely on ASICs, which may yield better energy efficiency and deployment feasibility in resource-constrained autonomous UAVs.

10. Implications for Research

This work demonstrates how DPR can turn from a potential vulnerability into a security benefit if the key is generated on hardware trust foundations, relies upon access control firewalls, and is compatible with real-world scheduling requirements. Future research directions include the development of formally verified security architectures, adaptive security policies, and advanced formal design automation techniques for reconfigurable systems. New trends like verified artificial intelligence, secure embedded system design, and high-level synthesis for FPGA platforms are promising areas in which to enhance security and design efficiency. Furthermore, scalable reconfigurable computing frameworks will be needed to empower next-generation UAV systems to have autonomous and adaptive capabilities [37]–[40].

11. CONCLUSIONS

In this paper, a Secure Dynamic Partial Reconfiguration (DPR) framework for real-time autonomous UAV (UAS) systems, implemented on the Xilinx PYNQ-Z2 FPGA-SoCplatform, is presented. The proposed architecture utilizes a hardware-rooted trust, PUF-based authentication, encrypted bitstreams, AXI firewall isolation, and adaptive runtime security in the context of a control-aware FPGA-SoC solution.

Average secure DPR latency was successfully validated with PX4 SITL and Gazebo simulation as well as hardware-in-the-loop (HIL) testing, with a latency of 12.6 ms, a maximum control jitter of < 8 μs, a control latency of < 77 μs, and a transient power overhead of < 12%.

The framework was successfully able to keep the 500 Hz UAV flight controls stable throughout runtime reconfiguration and cyber-physical attack scenarios, such as GPS spoofing mitigation, as well as adaptive security recovery.

The results demonstrate that secure dynamic partial reconfiguration is possible without compromising real-time UAV control constraints; thus, the proposed architecture can be adopted for next-generation resilient autonomous UAV platforms.

BIBLIOGRAPHY

A. UAV Systems and Cyber-Physical Security

1. R. Austin, Unmanned Aircraft Systems: UAVs Design, Development and Deployment, 2nd ed. Wiley, 2010.
2. R. Beard and T. McLain, Small Unmanned Aircraft: Theory and Practice. Princeton Univ. Press, 2012.
3. K. Hartmann and C. Steup, “The vulnerability of UAVs to cyber attacks,” in Proc. IEEE ICUAS, 2013, pp. 1–5.
4. M. R. Brust, M. K. Akram, and D. Turgut, “A survey on unmanned aerial vehicle networks,” Ad Hoc Networks, vol. 93, 2019.
5. M. Psiaki and T. Humphreys, “GNSS spoofing and detection,” Proc. IEEE, vol. 104, no. 6, pp. 1258–1270, 2016.
6. Y. Shoukry et al., “Non-invasive spoofing attacks,” in Proc. CHES, 2013.
7. T. Humphreys et al., “Assessing GPS spoofing,” in Proc. ION GNSS, 2008.
8. K. Nakamura and T. Fujimoto, “Low-latency secure FPGA overlay architectures for autonomous drones,” Drones, vol. 9, no. 2, p. 88, 2025.
9. S. Roy and T. Banerjee, “Control-aware dynamic reconfiguration for real-time UAV FPGA architectures,” IEEE Access, vol. 12, pp. 44211–44229, 2024.

10. Y. Wang, H. Zhang, and X. Liu, “Secure Dynamic Partial Reconfiguration for FPGA-Based Cyber–Physical Systems,” IEEE Transactions on Industrial Informatics, vol. 19, no. 4, pp. 5874–5885, Apr. 2023.
11. M. Shafique, T. Theocharides, and J. Henkel, “Secure and Adaptive FPGA-Based Architectures for Autonomous Embedded Systems,” ACM Transactions on Embedded Computing Systems, vol. 22, no. 5, pp. 1–26, Sept. 2023.

B. Real-Time Systems and Control Theory

12. H. Kopetz, Real-Time Systems, 2nd ed. Springer, 2011.
13. A. Wasicek, E. A. Lee, and S. Seshia, “Challenges in cyber-physical systems,” Proc. IEEE, vol. 100, no. 1, pp. 28–40, 2012.
14. K. J. Åström and R. M. Murray, Feedback Systems.Princeton Univ. Press, 2008.
15. E. A. Lee, “Cyber-physical systems: Design challenges,” in Proc. IEEE ISORC, 2008.
16. J. Patel, R. Gupta, and K. Roy, “Real-time scheduling and resource isolation in FPGA-SoC architectures for autonomous systems,” Journal of Systems Architecture, vol. 141, p. 102932, 2024.
17. G. Valente, A. Biondi, and G. Buttazzo, “Leveraging traffic injection and quality-of-service to control contention in FPGA-based real-time systems,” Journal of Systems Architecture, vol. 156, 2025.

C. FPGA-SoC Architectures and Reconfigurable Computing

18. S. Trimberger, “Three ages of FPGAs,” Proc. IEEE, vol. 103, no. 3, pp. 318–331, 2015.
19. S. Mittal, “A survey of FPGA-based accelerators,” ACM Comput.Surveys, vol. 49, no. 2, 2016.
20. Xilinx Inc., Zynq-7000 SoC Technical Reference Manual, 2020.
21. Xilinx Inc., PYNQ-Z2 Reference Manual, 2021.
22. P. Garcia et al., “FPGA-based embedded systems,” IEEE Design & Test, vol. 36, no. 5, 2019.
23. A. DeHon, “Reconfigurable computing,” Commun. ACM, vol. 59, no. 2, 2016.
24. X. Liu, H. Zhao, and Y. Wang, “AI-assisted anomaly detection and mitigation for FPGA-based cyber-physical systems,” Future Generation Computer Systems, vol. 156, pp. 144–158, 2025.
25. Y. Wu, F. Zhang, and J. Li, “Secure FPGA-based adaptive architectures for mission-critical autonomous platforms,” IEEE Transactions on Aerospace and Electronic Systems, vol. 61, no. 1, pp. 355–370, 2025.
26. Z. Li, Y. Wang, and J. Chen, “FPGA-based reconfigurable system: Research progress and reliability challenges,” Electronics, vol. 15, no. 3, p. 548, 2026.

D. Dynamic Partial Reconfiguration (DPR)

27. K. Vipin and S. A. Fahmy, “Survey of partial reconfiguration,” ACM Comput.Surveys, vol. 51, no. 4, 2018.
28. Xilinx Inc., Partial Reconfiguration User Guide UG702, 2021.
29. M. Peiffer, A. DeHon, and S. Devadas, “Secure FPGA reconfiguration,” in Proc. FCCM, 2014.

30. C. Claus et al., “Run-time partial reconfiguration,” in Proc. FPL, 2007.
31. F. Unterstein, T. Güneysu, and A. Moradi, “Secure update of FPGA-based secure elements using partial reconfiguration,” in Proc. IACR, 2022.
32. M. A. Khan and S. Parameswaran, “Secure dynamic partial reconfiguration techniques for FPGA-based embedded systems: A review,” Microprocessors and Microsystems, vol. 96, p. 104756, 2023.
33. D. Ferreira, M. Silva, and A. Cardoso, “Secure runtime reconfiguration techniques for adaptive FPGA systems in autonomous vehicles,” Integration, the VLSI Journal, vol. 92, pp. 78–91, 2024.
34. B. Chen, M. Ali, and S. Gupta, “Dynamic partial reconfiguration for resilient autonomous edge systems,” ACM Transactions on Embedded Computing Systems, vol. 24, no. 1, pp. 1–27, 2025.

E. Hardware Security and Trust Architectures

35. M. Tehranipoor and F. Koushanfar, “Hardware Trojan taxonomy,” IEEE Design & Test, 2010.
36. R. Karri et al., “Trustworthy hardware,” Computer, vol. 43, no. 10, 2010.
37. NIST, Guide to Hardware Roots of Trust, NISTIR 8320, 2018.
38. A. Perrig et al., “Security in wireless sensor networks,” Commun.ACM, 2004.
39. Y. Zhang, L. Chen, and P. Mishra, “Hardware-assisted trust and runtime security enforcement in reconfigurable SoC platforms,” IEEE Transactions on CAD, vol. 43, no. 5, pp. 2112–2125, 2024.
40. A. A. Abdulsamad, M. Al-Janabi, and S. Alsaedi, “Application of FPGA devices in network security: A survey,” Electronics, vol. 14, no. 19, p. 3894, 2025.
41. A. Mahmood, Y. Kim, and H. Lee, “Runtime adaptive FPGA security architectures for cyber-physical systems,” IEEE Access, vol. 11, pp. 102345–102361, 2023.

F. PUFs and Authentication Mechanisms

42. G. E. Suh and S. Devadas, “Physically unclonable functions,” in Proc. DAC, 2007.
43. R. Maes, Physically Unclonable Functions. Springer, 2013.
44. C. Herder et al., “PUFs and applications,” Proc. IEEE, vol. 102, no. 8, 2014.
45. R. Sharma, P. Singh, and A. Verma, “Lightweight PUF-assisted authentication framework for secure UAV embedded systems,” Sensors, vol. 24, no. 6, p. 2014, 2024.

G. Cryptographic Protection and Side-Channel Security

46. A. Hodjat and I. Verbauwhede, “Pipelined AES processors,” IEEE Trans. Computers, 2006.
47. P. Kitsos and O. Koufopavlou, “FPGA crypto implementations,” IEEE Trans. VLSI, 2008.
48. P. Kocher et al., “Differential power analysis,” in Proc. CRYPTO, 1999.
49. S. Mangard et al., Power Analysis Attacks. Springer, 2007.
50. S. Skorobogatov, “Semi-invasive attacks,” Cambridge Univ., 2005.
51. M. Tunstall et al., Fault Analysis of Cryptographic Algorithms. Springer, 2017.
52. S. R. Bommana, R. K. Vankayalapati, and P. K. Singh, “Mitigating side channel attacks on FPGA through deep learning and dynamic partial reconfiguration,” Scientific Reports, vol. 15, no. 1, pp. 1–18, 2025.

H. Isolation, Reliability, and Secure Embedded Systems

53. A. Wassermann et al., “Hardware-based memory isolation,” IEEE TDSC, 2018.
54. J. Rajendran et al., “Split manufacturing security,” IEEE Design & Test, 2013.
55. J. Clark and J. Jacob, “Fault injection attacks,” IEEE Security & Privacy, 2007.
56. A. Ibrahim, S. Pagliarini, and M. Tehranipoor, “Runtime FPGA Security Monitoring and Anomaly Detection for Reconfigurable Systems,” IEEE Access, vol. 11, pp. 74215–74231, 2023.

AI, Adaptive Security, and Future Autonomous Systems

57. S. Seshia et al., “Toward verified AI,” Commun.ACM, 2018.
58. D. Goodin, “Security threats in embedded systems,” IEEE Security & Privacy, 2019.
59. J. Cong et al., “High-level synthesis for FPGAs,” ACM Trans. Design Automation, 2011.
60. J. Kim, D. Lee, and K. Choi, “Hardware-Assisted Secure Reconfiguration Framework for Real-Time UAV Systems,” Journal of Systems Architecture, vol. 145, Art. no. 102981, 2024.