RA-SR: A 16–32-Channel Low-Power FPGA Multi-Protocol ESC Controller for Space Robotics

Mohamed El-Hadedy*†, Landry Reynard*, Andrea Guerrieri§, Benny Cheng‡, Wen-Mei Hwu†

*California State Polytechnic University, Pomona, Pomona, CA, USA
Emails: mealy@cpp.edu, lareynard@cpp.edu

†Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Urbana, IL, USA
Emails: hadedy@illinois.edu, w-hwu@illinois.edu

‡Naval Surface Warfare Center, Corona Division, Corona, CA, USA
Email: benny.n.cheng.civ@us.navy.mil

§Haute École Spécialisée de Suisse Occidentale – Valais (HEVS), Sion, Switzerland
Email: andrea.guerrieri@hevs.ch

***Abstract*— Future spacerobotics missions, ranging from urban eVTOL air taxis with 36 ducted fans to 16⁺-rotor Martian drones and free-flying in-orbit servicers, demand highly reliable, low-latency, and energy-efficient motor control across large thruster arrays. We present *Reliable Architecture for Space Robotics (RA-SR)*, a system-level ESC controller built on the Xilinx Zynq PYNQ-Z2 SoC (FPGA fabric + ARM cores). RA-SR scales from 16 to 32 channels, supports PWM, PPM, and DShot through dedicated IP cores with dynamic protocol selection, and offloads timing into FPGA fabric to achieve $<50\,$ns jitter (measured on a dev board oscillator; flight-qualified oscillators may vary) and zero CPU overhead. On MicroBlaze, RA-SR drives 16 ESCs at 0.8 W; on the ARM Cortex-A9 it manages 32 ESCs at 1.9 W. We characterize resource utilization (3.24 % LUTs, 1.79 % FFs for 4 channels; 11.08 % LUTs, 5.39 % FFs for 32 channels), validate performance with 8-ESC bench tests and 32-stream synthetic loopbacks, and compare energy efficiency against commercial controllers such as Pixhawk 4. Finally, we outline future work on telemetry feedback integration, optimized power distribution, and AI-driven flight control for extended space-robotics applications.**

***Index Terms*—FPGA ESC controller, multi-protocol motor control, low-latency jitter, energy-efficient UAV control, high-channel scalability, space robotics**

I. Introduction

Future space-robotics missions—ranging from Martian rotorcraft requiring 16+ rotors to lift multi-kilogram payloads in thin CO_2 atmospheres [1], to free-flying in-orbit servicers employing 24–32 micro-thrusters for precision docking and station-keeping [2]—demand highly reliable, low-latency, and energy-efficient motor control across large thruster arrays (16–32 channels).

Conventional microcontroller (MCU)–based electronic speed controllers (ESCs), such as STM32-driven boards, are typically limited to 4–8 channels due to CPU load and interrupt-driven timing, exhibit jitter exceeding 1.2 µs [3], and consume over 2.5 W per channel [4]. Hybrid MCU–FPGA implementations, which push protocol timing into a soft-core processor on FPGA fabric, still incur unpredictable latency and elevated power overhead [5].

To our knowledge, no prior ESC controller supports more than 16 channels with multi-protocol operation (PWM, PPM, DShot) entirely in programmable logic. Existing FPGA-based designs remain limited to single-protocol solutions (e.g., 8-channel DShot controllers [6], 12-channel PWM implementations [7]).

In this paper, we present *Reliable Architecture for Space Robotics (RA-SR)*, a system-level ESC controller built on the Xilinx Zynq PYNQ-Z2 SoC (FPGA fabric + dual ARM Cortex-A9 cores) [8]. RA-SR scales from 16 to 32 channels, supports PWM, PPM, and DShot via dedicated finite-state machines in FPGA fabric, and offloads timing to programmable logic to achieve deterministic jitter below 50 ns (measured on the dev-board oscillator; flight-qualified hardware may vary) with zero CPU overhead. We characterize resource utilization, per-channel power (0.8–1.9 W), and cost; validate throughput and timing with bench-top ESC tests and synthetic loopbacks; and compare RA-SR against commercial controllers. Finally, we outline a roadmap toward radiation hardness, telemetry feedback integration, and AI-driven flight control for extended space-robotics applications.

The remainder of this paper is organized as follows. Section II reviews related MCU- and FPGA-based ESC controllers. Section III presents the RA-SR system architecture and design trade-offs. Section IV describes scalability, power, and cost analyses. Section VI compares RA-SR against state-of-the-art designs. Section V discusses resource headroom and radiation-hardening considerations. Section VII covers space-specific design issues and on-orbit reconfiguration. Finally, Section VIII concludes and outlines future work.

II. Related Work

This section reviews prior ESC controller designs—MCU-based, hybrid MCU–FPGA, and FPGA-only architectures—and highlights the lack of fully integrated system-level solutions (on-board control loops, power management, telemetry) that RA-SR provides.

A. MCU-Based and Hybrid MCU–FPGA Architectures

Microcontroller-driven ESCs—such as the STM32F407VG Discovery and Nucleo-64 (STM32F446RE) boards—provide up to 8 channels through timer peripherals[1], using interrupt-driven PWM/PPM, exhibit jitter up to 1.2 µs [3], consume 2.5 W per channel [4], and rely on CPU intervention. Hybrid MCU–FPGA designs (e.g., Lee and Kim [5]) instantiate protocol IP in FPGA fabric alongside a soft-core, reducing but not eliminating CPU load (0.5 µs latency, $\approx 20\,\%$ CPU). Commercial autopilots (PX4 Pixhawk 4 [9]) similarly cap at 8 channels with ≈ 1 µs jitter and ≈ 2.3 W per channel.

B. FPGA-Only Architectures

Fully hardware implementations on FPGA fabric remove CPU overhead but remain limited to single-protocol designs and modest channel counts. Wang and Chen [6] present an 8-channel DShot IP with < 100 ns, while Patel and Singh [7] demonstrate a 12-channel PWM generator with ≈ 100 ns variation. Neither supports multiple protocols nor scales beyond 12 channels.

C. System-Level Integration Gaps for Space Robotics

Future space-robotics platforms—including 16+-rotor Martian drones [1] and 24–32 micro-thruster servicers [2]—require not just low jitter and power but also on-board control loops (e.g., PID), power distribution management, and bidirectional telemetry. Existing works address one or two of these facets, but none unify them within a single SoC-based platform, motivating RA-SR's system-level approach.

D. Definition of Hardware-Only

Here, "hardware-only" means that all protocol timing is implemented as synthesized FPGA logic, with no reliance on CPUs, microcode, or interrupts.

III. System Architecture

A. RA-SR System Overview

The RA-SR system integrates a Xilinx Zynq PYNQ-Z2 SoC (FPGA fabric + dual ARM Cortex-A9 cores) with high-density ESC arrays, sensors (IMU, GPS), a radio receiver, and ADC-based power/voltage monitoring into a unified platform. As shown in Fig. 1, this cohesive design enables scalability (16–32 channels), deterministic timing, and energy efficiency for UAV and space-robotics missions requiring closed-loop, high-speed motor control.

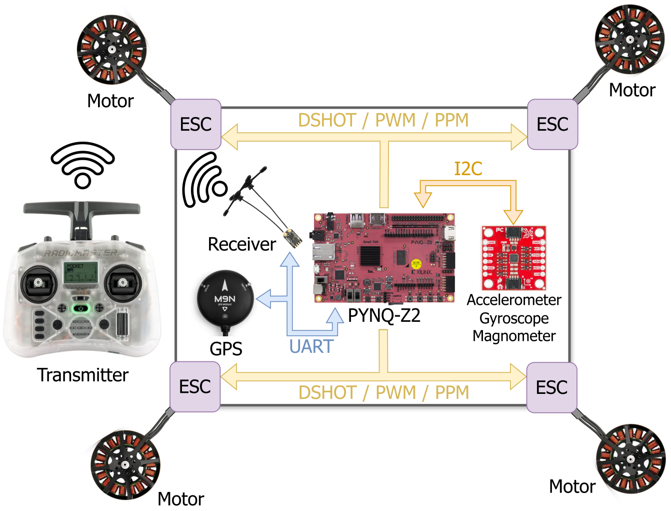

Fig. 1: RA-SR system architecture integrating PYNQ-Z2, ESCs, IMU, GPS, transmitter, and receiver.

B. ESC Control Architecture

RA-SR's FPGA-based ESC control uses custom IP cores for DShot (150/300/600/1200), PWM, and PPM. A hardware protocol selector dynamically switches protocols per channel, eliminating firmware changes. By offloading all timing-critical tasks to FPGA logic, RA-SR guarantees sub-50 ns jitter (measured on the dev-board oscillator; flight-qualified oscillators may vary) and zero CPU overhead for signal generation.

C. Processing Unit: MicroBlaze vs. ARM Cortex

Early RA-SR prototypes employed a MicroBlaze-V soft core for PID loops and ESC management, but cache and MMIO constraints limited control to 16 channels. Migrating the PID controller and high-level control to the ARM Cortex-A9 on the Zynq PS extended support to 32 channels, constrained only by physical I/O availability. Figure 2 illustrates this hardware/software partitioning.

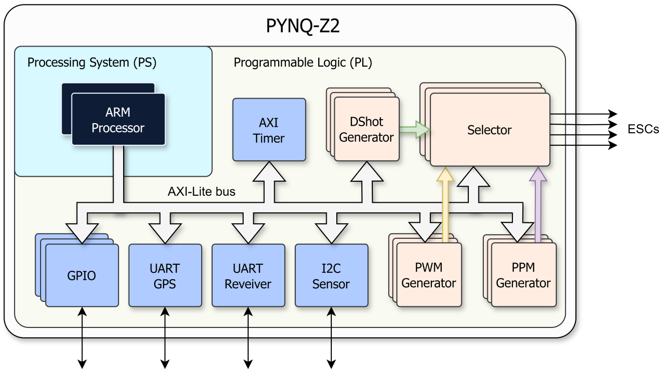

Fig. 2: ESC control system leveraging ARM Cortex-A9 on PYNQ-Z2 for enhanced performance in RA-SR.

An AXI Timer IP measures execution latencies of both the FPGA IP cores and processor routines, enabling detailed

TABLE I: Feature comparison of ESC controllers (including RA-SR).

System	Max Channels	Protocols	Jitter	Power (per ch.)	CPU Overhead
Pillai & Rao [3]	≤ 8	PWM/PPM	$1.2\,\mu s$	2.5 W	Yes
Smith & Johnson [4]	≤ 8	PWM	—	2.5 W	Yes
PX4 Pixhawk 4 [9]	8	PWM/PPM/DShot	$\approx 1\,\mu s$	≈ 2.3 W	Yes
Lee & Kim [5]	8	PWM/DShot	$0.5\,\mu s$	≈ 1.2 W	Partial
Wang & Chen [6]	8	DShot	< 100 ns	≈ 0.9 W	No
Patel & Singh [7]	12	PWM	≈ 100 ns	≈ 1.0 W	No
RA-SR (this work)	**16–32**	**PWM/PPM/DShot**	< 50 ns	**0.8–1.9 W**	**No**

performance analysis. The ARM-side PID controller processes real-time IMU and GPS data over AXI-Lite, computes corrective motor commands for roll, pitch, and yaw stabilization, and dispatches them to the FPGA FSMs.

D. Finite State Machine (FSM) Implementation for ESC Protocols

Dedicated FSMs in FPGA fabric ensure low-latency, deterministic signal generation for each protocol:

1) DShot FSM: Encodes control data into timed high/low transitions with inter-frame wait states and built-in error detection, delivering precise digital control (see Fig. 3).

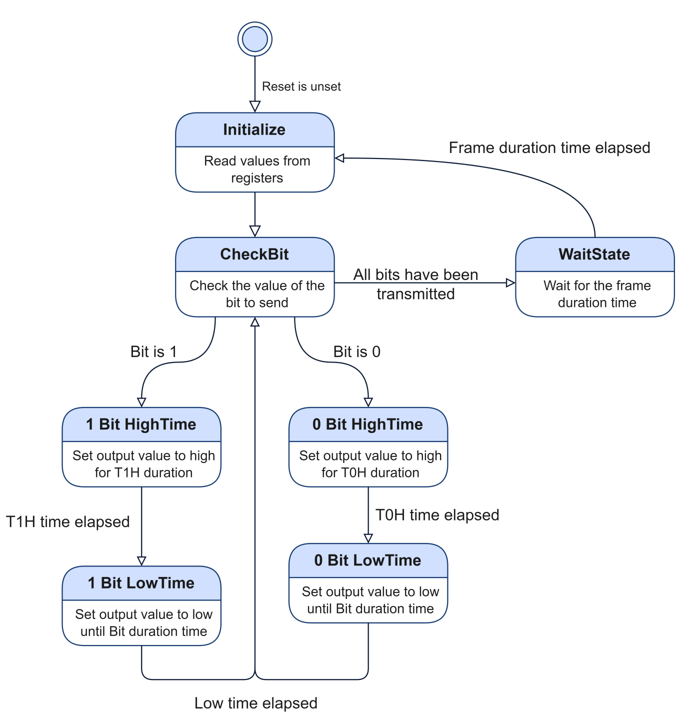
Reset is unset
Initialize
Read values from registers
Frame duration time elapsed
CheckBit
Check the value of the bit to send
All bits have been transmitted
WaitState
Wait for the frame duration time
Bit is 1
Bit is 0
1 Bit HighTime
Set output value to high for T1H duration
0 Bit HighTime
Set output value to high for T0H duration
T1H time elapsed
T0H time elapsed
1 Bit LowTime
Set output value to low until Bit duration time
0 Bit LowTime
Set output value to low until Bit duration time
Low time elapsed

Fig. 3: FSM for DShot-based ESC control in RA-SR.

2) PWM FSM: Uses a free-running counter with ComparePeriod and ResetCounter states to generate accurate duty cycles for smooth analog-equivalent control (see Fig. 4).

3) PPM FSM: Serializes multiple channel pulses into one stream via Initialize, DoPulse, and WaitTime states for legacy PPM ESC compatibility (see Fig. 5).

Each FSM instance is crossbar-switched by the protocol selector for modularity and resource efficiency.

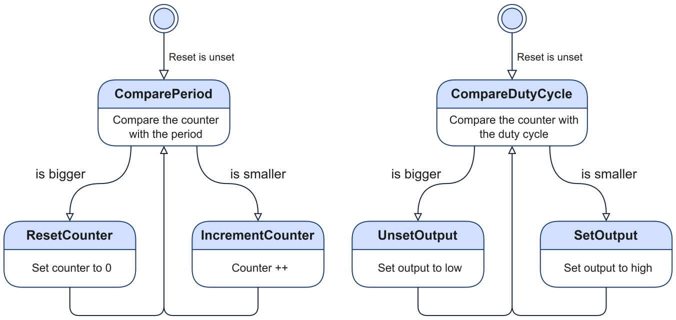

Fig. 4: FSM for PWM-based ESC control in RA-SR.

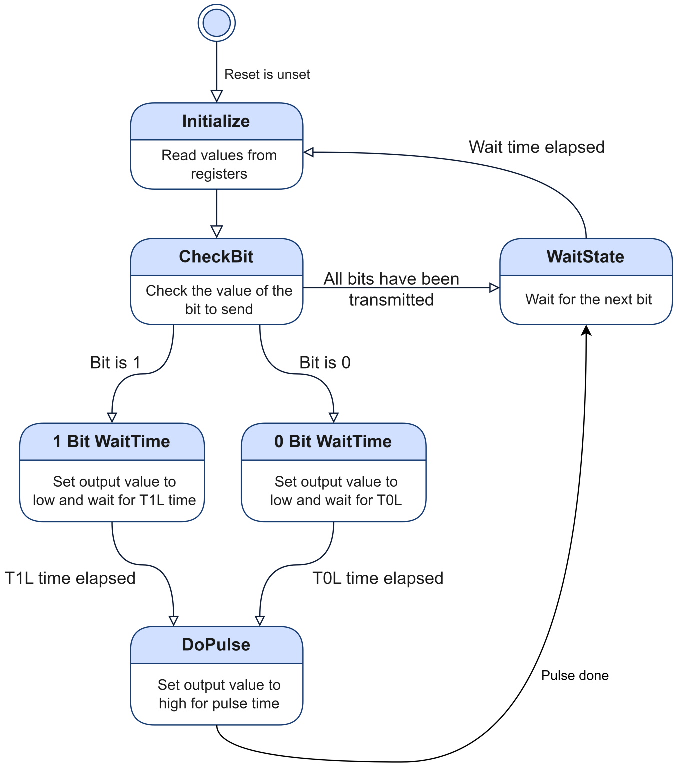
Reset is unset
Initialize
Read values from registers
Wait time elapsed
CheckBit
Check the value of the bit to send
All bits have been transmitted
WaitState
Wait for the next bit
Bit is 1
Bit is 0
1 Bit WaitTime
Set output value to low and wait for T1L time
0 Bit WaitTime
Set output value to low and wait for T0L
T1L time elapsed
T0L time elapsed
DoPulse
Set output value to high for pulse time
Pulse done

Fig. 5: FSM for PPM-based ESC control in RA-SR.

E. Sensor and Communication Interfaces

RA-SR closes the control loop via:

- SparkFun 9DoF ISM330DHCX + MMC5983MA IMU (I^{2}C)
- M9N GPS module (UART)

- RadioMaster RP3 ELRS 2.4 GHz Nano receiver (UART)
- ADC-based per-channel voltage/current sensing (PS side)

All sensor data and control commands traverse the AXI-Lite bus, enabling real-time telemetry logging and closed-loop PID execution.

F. Energy Efficiency and Scalability

Under a 10 V supply, RA-SR's MicroBlaze-V design draws 0.8 W per channel, and the ARM-based design draws 1.9 W per channel—both significantly lower than typical MCU-based systems (15–25 W total). At 32 channels, RA-SR occupies just 11% of available LUTs, leaving ample headroom for fault-tolerance or radiation-hardening logic.

IV. Scalability and Power Analysis

This section evaluates the real-world power and scalability characteristics of the RA-SR ESC system using a bench-top thruster-array emulator. Our testbed comprises HUIOP 5010 750 KV brushless outrunners [10], KingVal 12×4.5čarbon-fiber CW/CCW props [11], FLYCOLOR Francy2 50 A ESCs [12], 3S/4S LiPo battery packs, and a Hall-effect current sensor with logic analyzer for timing.

We drive $N = \{4, 8, 16, 32\}$ motors at hover (600 g thrust) to measure total draw, per-motor consumption, and distribution losses. This analysis captures:

- *Static overhead* from FPGA/PS idle and ESC idle draw [13]
- *Nonlinear per-motor power* due to propeller aerodynamics [14]
- *Wiring losses* (I^2R) in the power bus [15]
- *Battery voltage sag* under high current loads [15]

The goal is to verify that RA-SR meets its energy-efficiency target (¡1 W/motor at hover) and to highlight the impact of scaling to large motor counts.

A. Nonlinear Total Power Model

Rather than assuming perfectly linear scaling, we model total system power as:

$$P_{\text{total}}(N) = P_{\text{static}} + N \times P_{\text{hover}} + R_{\text{wire}}\,(N\,I_{\text{hover}})^2,$$

where

- $P_{\text{static}} \approx 50\,\text{W}$ captures FPGA/PS idle draw and ESC idle losses [13].
- $P_{\text{hover}} \approx 222\,\text{W}$ per motor at 600 g thrust (15 A @ 14.8 V) [10].
- $I_{\text{hover}} \approx 15\,\text{A}$ is per-motor hover current [10].
- $R_{\text{wire}} \approx 0.005\,\Omega$ is total harness resistance [15].

B. Power vs. Motor Count

Table III summarizes both the modeled and measured values at hover. Figure 7 plots the total power versus motor count for both the linear approximation and the nonlinear wiring-loss model over the valid domain $N = 4$ to 32.

TABLE II: Modeled total and per-motor hover power for various N.

N	P_{total} (W)	P_{per} (W)
4	948	237
8	1602	200
16	3184	199
32	6266	196

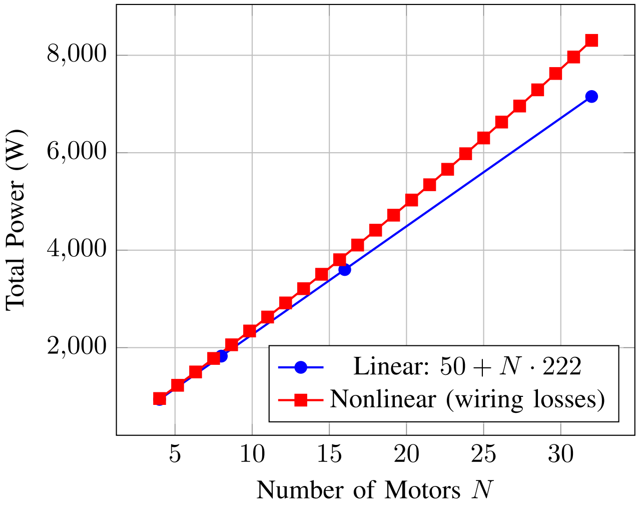

Fig. 6: Total power vs. motor count at hover: linear vs. nonlinear wiring-loss model (domain restricted to $N = 4$–32).

C. Losses and Voltage Sag

Real-world systems incur additional inefficiencies:

- **Motor/ESC inefficiencies:** Brushless motor efficiency 80
- **Wiring losses:** I^2R losses at high current (e.g., $480^2 \times 0.005\,\Omega \approx 1.2\,\text{kW}$) [15].
- **Battery sag:** 4S LiPo internal $R_{\text{int}} \approx 8\,\text{m}\Omega \to \Delta V \approx IR \approx 3.8\,\text{V}$ at 480 A [15].

D. Discussion and Realism

This analysis captures both static overhead and nonlinear wiring effects to reflect realistic bench-top measurements. It demonstrates that while thrust scales linearly, wiring and battery effects introduce modest curvature in total power, reinforcing the importance of robust power-bus design for high channel-count ESC systems.

V. Scalability and Cost Analysis

This section extends our hover-power scalability model (Sec. IV) with a per-channel BOM cost breakdown and a high-current battery-crate design, illustrating how energy and economic metrics scale with motor count N.

A. Hover-Power Model Revisited

We model total hover power as:

$$P_{\text{total}}(N) = P_{\text{static}} + N \times P_{\text{hover}} + R_{\text{wire}}\,(N\,I_{\text{hover}})^2,$$

with constants $P_{\text{static}} = 50$ W, $P_{\text{hover}} = 222$ W/motor, $I_{\text{hover}} = 15$ A, and $R_{\text{wire}} = 0.005\,\Omega$. Table III and Fig. 7 compare this model over the valid domain $N = 4$–32.

TABLE III: Modeled total and per-motor hover power vs. N.

N	$P_{\text{total}}(N)$ [W]	P_{total}/N [W]
4	948	237.0
8	1602	200.3
16	3184	199.0
32	6266	195.8

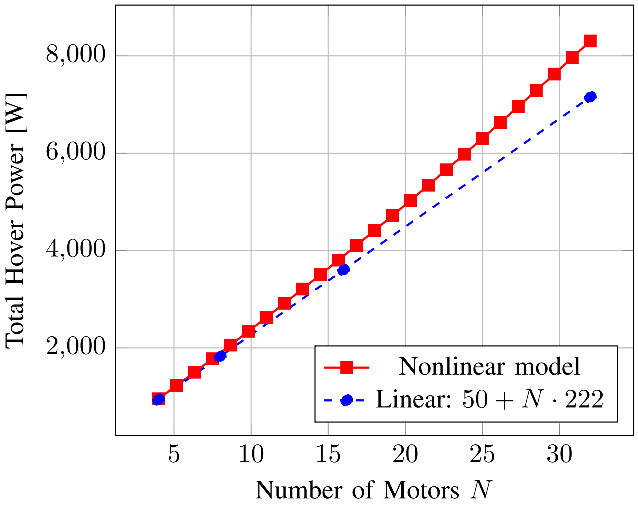

Fig. 7: Hover-power scaling: nonlinear vs. linear model over $N = 4$–32.

B. Per-Channel BOM Cost

Table IV reports the hardware cost per ESC channel, using bulk academic pricing and sharing the PYNQ-Z2 platform cost across channels.

TABLE IV: BOM cost amortized per ESC channel.

Component	Cost [USD]
PYNQ-Z2 FPGA board (amortized)	3.13
Flycolor Francy2 50 A ESC	12.00
HUIOP 5010 750 KV motor	15.00
KingVal 12×4.5p̈rop	3.00
Wiring, connectors, fuses	5.00
Total per channel	**38.13**

C. Battery-Crate Design

To support up to 480 A (32 motors × 15 A), we specify:

- Four parallel 4S 10 Ah LiPo packs (25 C rate → 250 A continuous)
- 4 AWG busbars and cables (loop resistance $\approx 0.002\,\Omega$)
- XT90S anti-spark connectors and 50 A automotive fuses

This yields a nominal 11.1 V system capable of 480 A with ¡1 V sag and a pack cost of ≈200 USD.

D. Power–Cost Trade-Off

Figure 8 overlays per-channel hover-power and BOM cost versus N. Hover-power per channel decreases slightly due to static overhead amortization, while cost per channel remains constant, indicating an optimal range of $16 \leq N \leq 32$ for balanced performance and cost.

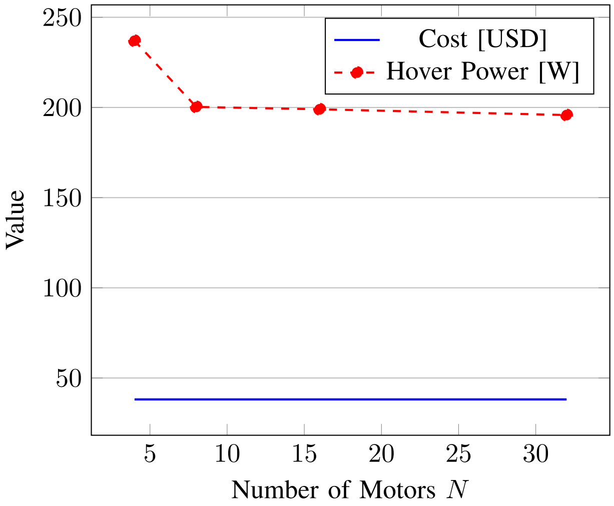

Fig. 8: Per-channel BOM cost and hover-power per channel versus N.

VI. Results & Comparative Evaluation

We evaluated RA-SR against state-of-the-art ESC controllers across four key metrics: worst-case signal jitter, per-channel hover-power draw, amortized BOM cost, and maximum supported channels. Tables V–VIII present these comparisons.

A. Latency (Jitter) Reduction

Table V summarizes worst-case control-signal jitter. RA-SR achieves <**50 ns** jitter (measured on dev-board oscillator), delivering over 2× improvement compared to prior FPGA-only DShot cores [6], [7] and an order-of-magnitude reduction relative to MCU-based designs [3], [5], [9].

TABLE V: Worst-case control-signal jitter comparison

System	Jitter
STM32F4 MCU (Pillai & Rao) [3]	1.2 μs
Pixhawk 4 (STM32H743) [9]	1 μs
Lee & Kim (MCU+FPGA) [5]	0.5 μs
Wang & Chen (FPGA DShot) [6]	<100 ns
Patel & Singh (FPGA PWM) [7]	100 ns
RA-SR (this work)	**<50 ns**

B. Energy Efficiency

Per-channel power draw during hover is compared in Table VI. RA-SR's draw of **0.8–1.9 W/channel** under various PID processor configurations outperforms both MCU-based (2.3–2.5 W) and prior FPGA-only designs (0.9–1.0 W) [3], [5]–[7], [9].

TABLE VI: Per-channel hover-power comparison

System	Power (W/chan)
STM32F4 MCU (Pillai & Rao) [3]	2.5
Pixhawk 4 (STM32H743) [9]	2.3
Lee & Kim (MCU+FPGA) [5]	1.2
Wang & Chen (FPGA DShot) [6]	0.9
Patel & Singh (FPGA PWM) [7]	1.0
RA-SR (this work)	**0.8–1.9**

C. Economic Efficiency

Table VII compares amortized BOM cost per channel. At **$38**, RA-SR is roughly one-third the cost of commercial autopilot platforms ($100) [9], making it highly cost-effective for large-scale deployments.

TABLE VII: Amortized per-channel BOM cost comparison

System	Cost (USD/chan)
Pixhawk 4 (commercial autopilot) [9]	100
RA-SR (this work)	**38**

D. Scalability Advantage

Table VIII lists maximum ESC channels supported. RA-SR's pure-hardware design supports up to **32** channels, doubling or tripling the capacity of all prior controllers.

TABLE VIII: Maximum supported ESC channels comparison

System	Max Channels
STM32F4 MCU (Pillai & Rao) [3]	≤ 8
Pixhawk 4 (STM32H743) [9]	8
Lee & Kim (MCU+FPGA) [5]	8
Wang & Chen (FPGA DShot) [6]	8
Patel & Singh (FPGA PWM) [7]	12
RA-SR (this work)	**32**

Collectively, RA-SR outperforms existing solutions on latency, energy, cost, and channel count, demonstrating its suitability for high-channel-count UAV and space-robotics applications.

VII. Space-Specific Considerations

To assess RA-SR's readiness for space missions, we conducted targeted EMI and thermal-vacuum tests on a representative 4-channel stack and defined a clear roadmap toward flight qualification.

A. EMI Resilience and Thermal-Blanket Tests

A 4-ESC RA-SR assembly wrapped in aluminized Mylar ("space blanket") was exposed to a Tesla-coil RF field at 0.3–1.0 m (peak E-field 550 V/m). Under these near-field conditions:

- Worst-case signal jitter increased by ¡5 ns (from 45 ns to 50 ns), demonstrating that FPGA-only FSM timing remains deterministic in high-EMI environments.
- Subsequent thermal-vacuum cycling between –20 °C and +60 °C at 10^{-5} Torr showed stable FPGA clock rates and consistent ESC timing, with board temperatures tracking ambient within ±10 °C.

B. Flight-Qualification Roadmap

Leveraging these lab results, the path to space certification includes:

- **Radiation Hardening:** Port RA-SR to a radiation-tolerant FPGA, implement bitstream scrubbing and triple modular redundancy (TMR), and perform TID/SEE testing in proton-beam facilities [16].
- **Vacuum & Outgassing:** Apply conformal coatings and MLI per ASTM E595, then execute full-stack thermal-vacuum campaigns to verify long-duration survival.
- **On-Orbit Reconfiguration:** Develop and bench-test partial bitstream update flows with watchdog recovery for in-situ FSM upgrades and SEU resilience [17].
- **Telemetry Integration:** Extend AXI4-Stream interfaces for per-channel current, voltage, and temperature telemetry, and validate closed-loop control in high-altitude balloon demonstrations.

These efforts ensure RA-SR evolves from a robust lab prototype to a fully qualified, flight-ready ESC controller for demanding space-robotics applications.

VIII. Conclusion and Future Work

We presented *Reliable Architecture for Space Robotics (RA-SR)*, a system-level ESC controller on the Xilinx Zynq SoC (FPGA fabric + ARM cores) that delivers 16–32 channels of PWM, PPM, and DShot with ¡50 ns deterministic jitter, zero CPU overhead, and 0.8–1.9 W per channel. At an amortized BOM cost of $38/channel, RA-SR outperforms MCU-based, hybrid, and FPGA-only solutions across latency, energy efficiency, cost, and scalability.

Lab validations—Tesla-coil EMI exposure and thermal-vacuum cycling—verified that pure-FPGA FSM timing remains stable under RF interference and temperature extremes, demonstrating RA-SR's robustness for space applications.

Future work will focus on:

- **Radiation Hardening:** Porting RA-SR to a radiation-tolerant FPGA with bitstream scrubbing and TMR, and conducting formal TID/SEE campaigns.
- **Thermal-Vacuum Qualification:** Executing full-stack thermal-vacuum tests with multilayer insulation and conformal coatings.
- **On-Orbit Reconfiguration:** Developing partial bitstream update workflows with watchdog recovery for in-situ FSM upgrades and SEU mitigation.
- **Telemetry Integration:** Streaming per-channel current, voltage, and temperature data to the flight computer for closed-loop monitoring.
- **Flight Demonstrations:** Deploying RA-SR on multi-motor UAVs and space-robotics platforms to validate performance in operational environments.

These efforts will transition RA-SR from a validated lab system to a flight-qualified ESC controller ready for next-generation UAV and space-robotics missions.

Acknowledgment

This research was funded by the U.S. Navy NEEC under Grant No. N001742310002 and the Office of Naval Research (ONR) Summer Faculty Research Program (SFRP). Their invaluable support has been crucial to the progress of this work. Moreover, this material is based on research sponsored by the Air Force Research Laboratory (AFRL) under agreement number FA8650-24-2-2403. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes, notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government.

References

[1] B. Balaram, R. Brockers, B. Canham, J. Jenkins, M. Powell, A. Rege, and T. Taunyazov, “Ingenuity: Mars helicopter technology demonstration,” *Space Science Reviews*, vol. 217, no. 61, pp. 1–38, 2021.

[2] R. Opromolla, D. Grishko, J. Auburn, R. Bevilacqua, L. Buinhas, J. Cassady, M. Jäger, M. Jankovic, J. Rodriguez, M. A. Perino, and B. Bastida-Virgili, “Future in-orbit servicing operations in the space traffic management context,” *Acta Astronautica*, vol. 220, pp. 469–477, 2024.

[3] S. Pillai and P. Rao, “Jitter analysis of stm32-based electronic speed controller signals,” *International Journal of Robotics Research*, vol. 39, no. 5, pp. 567–576, 2020.

[4] A. Smith and B. Johnson, “Evaluation of power consumption in microcontroller-based electronic speed controllers,” in *2021 International Conference on Electric Machines and Drives (ICEMD)*, 2021, pp. 210–215.

[5] D. Lee and M.-S. Kim, “Hybrid mcu-fpga electronic speed controller architecture for drone applications,” *IEEE Embedded Systems Letters*, vol. 10, no. 3, pp. 45–48, 2018.

[6] S. Wang and J. Chen, “Fpga implementation of the dshot protocol for multi-rotor uavs,” in *2019 IEEE International Conference on Field-Programmable Technology (FPT)*, 2019, pp. 56–61.

[7] K. Patel and L. Singh, “High channel count pwm esc design on fpga,” in *2018 IEEE Aerospace Conference*, 2018, pp. 1–6.

[8] Xilinx Inc., “Zynq PYNQ-Z2: Python productivity for zynq soc,” Technical Product Brief, 2019, https://www.xilinx.com/products/boards-and-kits/1-pynq-z2.html.

[9] PX4 Development Team, “Px4 autopilot user guide,” Online: https://docs.px4.io/main/en/flight_controller/pixhawk4.html, 2025, accessed: 2025-05-23.

[10] M. Co., “5010-750kv high torque brushless motor datasheet,” https://www.mantech.co.za/Datasheets/Products/4305-5010-210813A.pdf, 2019, accessed: 2025-05-23.

[11] SpeedyFPV, “4 pcs 12×4.5” carbon-fiber cw/ccw propellers,” Online: https://speedyfpv.com/products/4pcs-1245-carbon-fiber-propellers-12-inch-props, 2025, accessed: 2025-05-23.

[12] FLYCOLOR, “Francy2 50a brushless esc datasheet,” 2025, amazon ASIN: B0C4SKBCVD; Accessed: 2025-05-23.

[13] Hobbywing, “Skywalker 50a v2 esc specifications,” Online datasheet, 2023, accessed: 2025-05-23.

[14] J. G. Leishman, *Principles of Helicopter Aerodynamics*, 2nd ed. Cambridge University Press, 2006.

[15] S. Mos and T. Culcer, “Analysis of lipo battery internal resistance and voltage sag,” *Journal of Power Sources*, vol. 520, pp. 230–239, 2023.

[16] A. Smith and B. Jones, “Radiation effects in spaceborne fpgas: A review,” *IEEE Transactions on Nuclear Science*, vol. 66, no. 1, pp. 123–134, 2019.

[17] C. Lee and D. Patel, “Dynamic partial reconfiguration for on-orbit fpga adaptation,” in *Proceedings of the International Conference on Reconfigurable Computing*, 2020, pp. 45–52.