Intelligence in Robotic Computing: Agile Design Flows for Building Efficient and Resilient Autonomous Machines

Zishen Wan[1*], Vijay Janapa Reddi[2], Arijit Raychowdhury[1]
[1]Georgia Institute of Technology, Atlanta, GA [2] Harvard University, Cambridge, MA *zishenwan@gatech.edu

1 PROBLEM AND MOTIVATION

The next ubiquitous computing platform, after personal computers and smartphones, is likely to be autonomous nature, such as drones, robots, and self-driving cars, which have been moving from mere concepts in labs to permeate almost every aspect of our society such as transportation, delivery, manufacturing, and agriculture [21, 37]. The continuous proliferation of autonomous machines depends critically on efficient and resilient computing substrates, driven by higher performance and safety requirements and the miniaturization of machine form factors. However, state-of-the-art autonomous machines are becoming increasingly complex, imposing hefty design efforts, latency, energy consumption, and reliability challenges.

Despite the high demand for scalable, efficient, and resilient autonomous machine computing (AMC), many fundamental problems remain unsolved in this area. Firstly, autonomous machines consist of complex cyber-physical components, resulting in a huge design space that hinders the scalability of computing architectures [8, 17]. Secondly, autonomous machines are typically resource-constrained and need to operate in dynamic changing environments in real-time, requiring extremely efficient and adaptive computing substrates [19, 33]. Moreover, autonomous systems need to operate resiliently to ensure their functional safety in the presence of faults, while nowadays resiliency solutions lead to higher overheads in performance, energy, and silicon cost [11, 12]. These challenges raise a surging need to develop more agile, efficient, and robust AMC solutions.

In this work, we present a holistic solution to facilitate the development of scalable, efficient, adaptive, and reliable AMC. We explore various methods for automatic design space exploration, software-hardware co-design, and efficiency-resilience co-optimization in three research lines. These methodologies can be seamlessly integrated and synergistically advance the ubiquitous application of next-generation agile, efficient, and reliable autonomous systems.

2 BACKGROUND AND RELATED WORK

Autonomous machines such as drones, robots, and self-driving cars are poised to become integral to our daily lives. Early design efforts have focused on developing accurate autonomy algorithms for perception, localization, and planning [22] to improve mission performance. Recently, autonomous machine computing (AMC) has attracted extensive research interests in system and architecture areas, given their growing demand for efficient, real-time, and resilient computing substrates [6, 8, 26]. Fig. 1 shows the AMC design targets, including agile and scalable design space exploration, efficient and adaptive computing hardware architecture, and resilient and robust system optimization. Due to the high complexity of autonomous machines, coupled with size-weight-and-power constraints, and their safety-critical nature in real-world deployment,

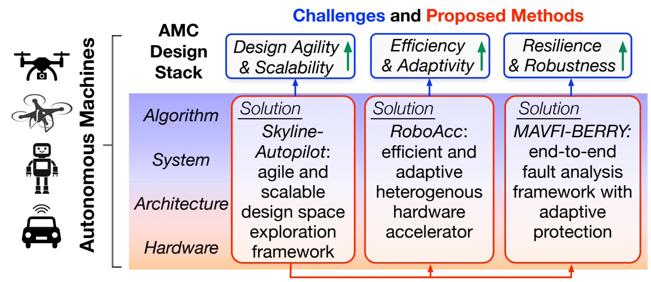


Figure 1: Challenges in current autonomous system design stack and our proposed across-stack software-hardware co-design solutions for agile, efficient, and resilient autonomous machine computing.

the potential of AMC will not be fully unleashed without careful optimization of scalability, efficiency, and resilience.

2.1 Early-Stage Design Space Exploration

During the early design stages, autonomy algorithms and hardware configurations will be jointly explored with other cyber-physical components to derive the design target which can achieve the optimal system performance. Previously, Hadidi *et al.* [8] quantified the unmanned aerial vehicle (UAV) design space and its various components needed for autonomous operations. Park *et al.* [25] explored the UAV design space for large-scale delivery services. Gables [9] provided insights into the optimization effort required to maximize the compute throughput for a given workload. However, autonomous machines are extraordinarily complex and diverse systems, where compute is just one component among many involving sensors, autonomy algorithms, onboard compute, and body dynamics. Existing models typically focus on individual components and cannot handle the vast design space of autonomous machines. To enable scalable AMC, there is a high demand for automatic design space exploration and agile domain-aware design flow to achieve optimal design configurations and maximize system performance.

2.2 Efficient Autonomous Machine Computing

Given an AMC system target, the second stage is to design an efficient AMC architecture. Specifically, the aim is to accelerate the computation-intensive compute kernels so that the design can achieve low-latency processing to meet AMC real-time requirements. Previously, Palossi *et al.* [24] proposed a specialized architecture for end-to-end learning autonomy paradigm running on a nanorobot platform. Navion [26] is a hardware accelerator for improving visual-inertial-odometry kernel in nano UAVs for autonomous exploration tasks. eSLAM [20] presented an FPGA-based hardware accelerator for autonomous machine real-time ORB-SLAM task [23]. Though the compute speed is already orders-of-magnitude higher than software running on CPU, their flexibility and adaptability are inadequate for practical autonomous machine computing due to the increasingly diverse autonomy algorithms, exacerbating resource constraints and dynamically changing operating environments.

2.3 Resilient Autonomous Machine Computing

Equipped with computational efficiency, the ability of an autonomous machine to tolerate or mitigate against errors, such as environmental conditions, sensor, hardware, and software faults, is essential to ensure its functional safety. Recently, Google [10] and Meta [5] have shown that potential compute and memory failures pose safety threats to the computing system at scale. Moreover, recent high-profile tragedies [1] heighten the urgent need for building reliable autonomous machine systems. To evaluate system reliability, prior analysis [13, 28] explored kernel vulnerability on GPUs. To improve system reliability, previous solutions [1, 6] have adopted "one-size-fits-all" nature where they typically use the same protection scheme, such as modular redundancy or anomaly detection, throughout the entire autonomous machines. However, these fault analyses lack an end-to-end perspective, and existing resiliency solutions make fundamental trade-offs between resiliency and performance overhead, leaving a large room for cross-layer efficiency-resilience co-optimization to design resilient and robust autonomous machines.

2.4 Our Solutions

Overall, existing studies still fail to provide agile, efficient, and resilient AMC designs. Hence, better AMC design flow and advanced efficiency-resilience co-optimization are still in great demand. Therefore, we propose a holistic AMC design solution to help build scalable, efficient, adaptive, and resilient autonomous machines with the following methodologies seamlessly integrated,

- **Agile and Scalable AMC Design Methodology**: *for the first time, the AMC design space is automatically navigated across the entire cyber-physic system stack.* We propose a systematic AMC system-on-chip (SoC) design framework with domain-aware characterization tool. An automated design space exploration method is proposed to further boost the performance of generated designs by 2-3× [15–18].
- **Efficient and Adaptive AMC Architecture**: *reconfigurable spatial-aware computing is considered in AMC architecture design for the first time.* Hardware-aware algorithms and processing-efficient hardware are proposed to achieve considerable efficiency improvement under dynamic environments [2, 7, 19, 33, 34, 38].
- **Resilient and Robust AMC Optimization**: we propose an end-to-end AMC reliability analysis and improvement framework, *which is the first to enable adaptive and cost-effective protection in safety-critical autonomous machines.* Our end-to-end framework precisely captures fault propagation in AMC, and the adaptive protection scheme boosts the AMC reliability by 3× with 1% less overhead [11, 29, 30, 32, 35, 36].

As shown in Fig. 1, the proposed cross-layer solutions focus on different AMC design stages and targets, synergistically advancing the ubiquitous applications of autonomous machine computing.

3 APPROACH AND UNIQUENESS

In this study, we present a holistic solution to enable agile, efficient, and resilient AMC, including automatic design space exploration `Skyline-AutoPilot` for improving design agility and scalability, reconfigurable accelerator `RoboAcc` for efficient and adaptive AMC, and lightweight protection scheme `BERRY` with end-to-end fault analysis frameworks `MAVFI`-family for improving AMC resilience.

3.1 `Skyline-Autopilot`: Agile AMC Design Flow

Autonomous machines are complex systems with sensors, autonomy algorithms, and onboard computing, posing a huge design space of 10^{18} configurations with cross-product effects on overall performance [16]. Previous efforts [8] quantify the system stack but fail to provide a solution to navigate this design space to produce optimal designs. Thus, we focus on answering the following critical questions: 1) how to evaluate the role of computing in complex cyber-physical autonomous machines and determine the optimal design, and 2) how to intelligently navigate the huge AMC design space and systematically design domain-specific system-on-chips (DSSoCs) for a rapidly evolving domain like autonomous machines [15–18].

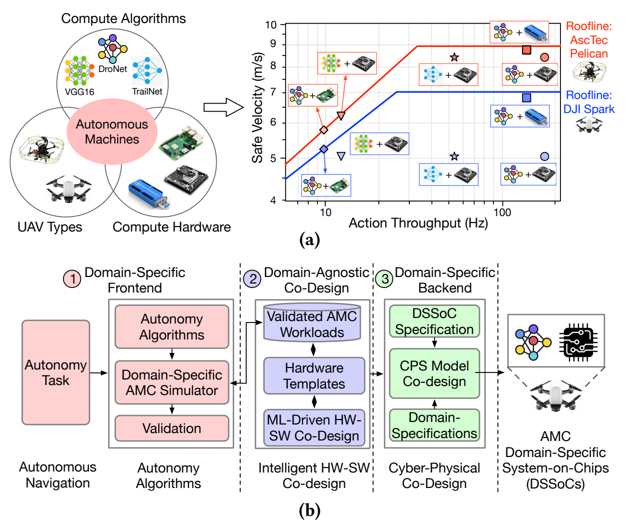


Figure 2: `Skyline-AutoPilot` [15–18] proposes (a) bottleneck analysis model for complex AMC systems, and (b) automated design space exploration framework for agile and scalable AMC.

Proposed AMC System Characterization Tool: To evaluate the role of computing in complex autonomous machines, we propose a novel bottleneck analysis model, `Skyline` [16, 18], for designing optimal computing systems for UAVs, as shown in Fig. 2a. `Skyline` provides insights by exploring the fundamental relationships between various components in autonomous machines, such as sensor, compute, and body dynamics, and quantifying the bottlenecks, which can aid a system architect in understanding the optimal compute design or selection for autonomous machines. The `Skyline` model is experimentally validated using real UAVs in real-world flight tests and is available as an interactive web-based tool [16].

Proposed Agile and Scalable AMC Design Framework: To intelligently navigate the huge AMC design space, we propose `AutoPilot` [17], a systematic methodology for automatically designing DSSoCs for autonomous UAVs (Fig. 2b). Integrated with `Skyline` model, `AutoPilot` offers a complete solution for automatically navigating the AMC design space using machine learning and performing co-design across the entire system stack, including sensors, autonomy algorithms, onboard compute, and dynamics, to maximize the end-to-end autonomous machine performance. `AutoPilot` also supports the ASIC flow and generates a layout of the floor-planed accelerator, which can be used to tape out the final hardware chip [15]. `AutoPilot` generates optimal DSSoC designs that consistently outperform off-the-shelf hardware and demonstrate 2-3× higher performance compared with state-of-the-art robotic accelerators.

Overall Contributions: We propose an automatic and intelligent design framework to break through the AMC agility and scalability. It is the first time that complex AMC design space has been systematically explored. We move beyond the compute-isolated design and propose an efficient characterization tool to navigate the cyber-physical design space. We also propose an automatic design flow with domain-specific insights to generate optimal design configurations. Our agile and highly scalable `Skyline-AutoPilot` framework consistently generates optimal DSSoCs outperforming state-of-the-art hardware accelerators in end-to-end performance for diverse deployment scenarios, and provides a design template for broader classes of autonomous machine applications.

3.2 `RoboAcc`: Efficient Adaptive AMC Hardware

Autonomous machines typically have strict real-time and energy requirements and operate in constantly changing environments, raising a surging need for more efficient and adaptive computing under diverse scenarios. However, prior AMC architectures [20, 26] lack adaptability for dynamic surroundings, and are designed for specific algorithms to accommodate the worst case, leading to wasteful computation at run time and suboptimal efficiency. Instead of static design, we focus on two aspects: (1) how to systematically find autonomy algorithm bottleneck and lucrative acceleration target, and (2) how to intelligently adapt AMC in dynamic environments to achieve better efficiency via software-hardware co-design [2, 7, 19, 33, 34, 38].

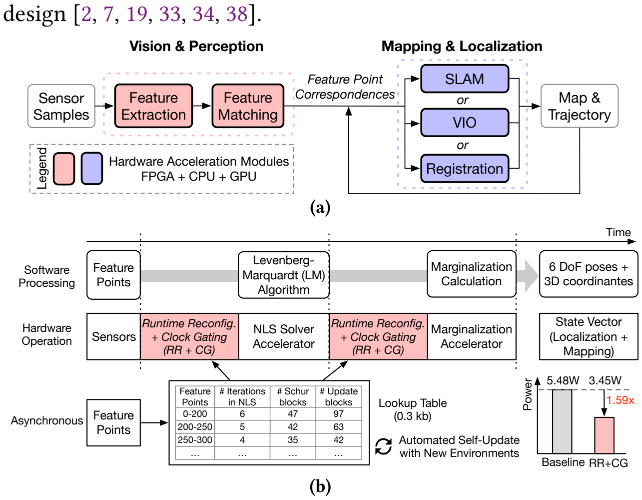


Figure 3: `RoboAcc` [2, 7, 19, 33, 34, 38] features (a) unified autonomy algorithm framework and (b) runtime-reconfigurable hardware accelerator for efficient and adaptive AMC.

Proposed Unified Autonomy Algorithm Framework: To determine the lucrative acceleration target, we propose an algorithm framework that flexibly adapts to different operating environments and systematically characterizes the end-to-end latency of the compute pipeline [38]. To provide a unified architecture to efficiently support these components in one system, we propose to capture the general patterns and shared common blocks across the primitive algorithms [37]. We reveal that the vision frontend is typically the bottleneck while the localization backend exhibits high latency variations. Our proposed unified algorithm framework provides insight in AMC performance characterization and hardware acceleration.

Proposed Runtime-Reconfigurable Accelerator: To efficiently adapt to dynamic environments, we propose `RoboAcc` [7, 19, 38], a reconfigurable heterogeneous AMC accelerator that can continuously adjust computational resources at run time according to the surroundings to save power while sustaining performance and accuracy for autonomy tasks. `RoboAcc` adopts hardware-friendly robotic algorithms, software-friendly sparsity, data flow, and memory access patterns, as well as software-hardware co-design techniques to reduce energy consumption and improve throughput. We further propose to modularize the AMC kernel design by building optimized hardware blocks and efficiently map Robot Operating System (ROS) computational graphs on the silicon substrate, with ROS node acceleration and a better ROS-SoC interface [33].

Contributions: Our proposed unified AMC algorithm framework systematically characterizes the end-to-end latency of autonomy pipeline to efficiently support AMC hardware design. Our proposed runtime-reconfigurable design techniques address the unique challenges posed by autonomous machines in continuously interacting with dynamic surroundings and intelligently adjusting computing resources. Our efficient accelerator co-designs hardware-aware algorithms and processing-efficient hardware, unlocking the capability to deliver high performance, efficiency, and adaptive AMC.

3.3 `MAVFI-BERRY`: Resilient AMC Optimization

With improved AMC efficiency and adaptability, ensuring operational reliability is another critical step in intelligent AMC solutions, which requires efficient methods to analyze, improve and optimize AMC resilience. However, prior resilience analysis [28] lacks end-to-end evaluation and protection solutions [1] incur large performance overhead. Thus, we focus on answering the following critical questions: (1) how to precisely capture end-to-end fault propagation and efficiently evaluate the resilience of various AMC kernels, (2) how to significantly improve AMC resilience with little overhead, and (3) how to further advancing energy savings while ensuring safety via performance-efficiency-resilience co-optimization [11, 29, 30, 32, 35, 36].

Proposed End-to-End Resilience Analysis Framework: Unlike traditional computing systems, autonomous machines are complex cyber-physical systems, and evaluating the resilience of individual components of compute/control is devoid of a cross-stack perspective. This may limit reliability solutions and miss error propagation. To efficiently tackle the AMC resilience challenge, we propose an end-to-end resilience analysis framework, `MAVFI` [11], that can analyze the fault tolerance of autonomous machines to various types of compute and memory failures. To accelerate the fault injection

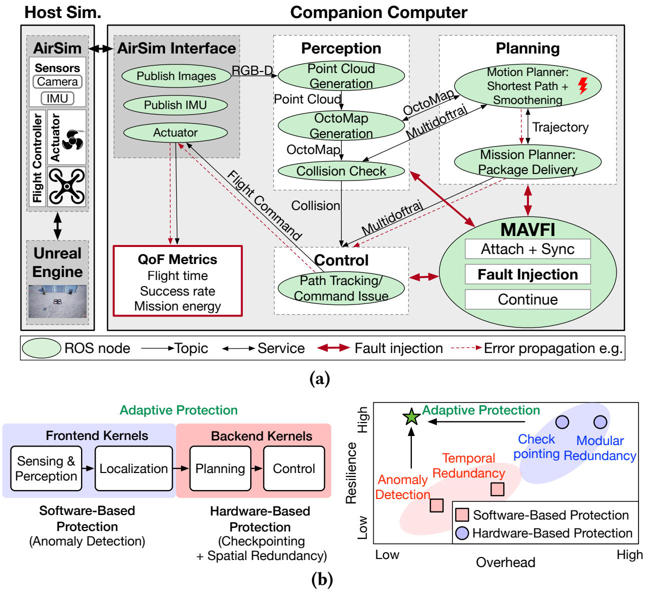


Figure 4: MAVFI-VPP-BERRY [11, 29, 30, 32, 35, 36] proposes (a) end-to-end resilience analysis framework and (b) adaptive protection with performance-efficiency optimization for resilient AMC.

(FI) campaign and efficiently identify safety-critical scenarios, we propose an intelligent multi-phase hierarchical FI scheme [35]. We integrate MAVFI into hardware-in-the-loop simulators, and further propose a complete solution to evaluate the end-to-end resilience in both physical model-based and learning-based autonomy pipelines in single-agent RL-FI [29] and multi-agent scenarios FRL-FI [30]. MAVFI-family resilience evaluation frameworks are portable to any Robot Operating System and provide AMC resilience insights.

Proposed Lightweight Adaptive Protection: To provide high protection coverage with little cost, we propose an adaptive vulnerable proportional protection (VPP) design [32] that exploits the inherent robustness variations in the autonomous machine system. Building upon MAVFI-family tool, for the first time, we reveal that different nodes in the autonomous machines differ significantly in their inherent robustness, where front-end is generally more robust while the back-end is less so. In stark contrast to the existing "one-size-fits-all" strategy [1, 6] that uniformly applies the same protection strength to all tasks, VPP paradigm dynamically attributes the protection budget, be spatially or temporally, inversely proportional to the inherent robustness of an autonomous task/algorithm. VPP achieves 3× resilience improvement with little overhead.

Proposed Robust Low-Voltage Computation: Lowering the operating voltage is a powerful means of efficient computing, however, it can also result in on-chip failures that are detrimental to the safety and performance of autonomous machines. To co-optimize efficiency and resiliency, we propose BERRY [36], a robust learning framework for autonomous machines under low-voltage operation. BERRY supports robust learning, both offline and on-board processor, and for the first time, demonstrates the practicality of robust low-voltage AMC that leads to high energy savings in both compute-level operation and system-level quality-of-mission. BERRY can be applied in conjunction with VPP, leading to order-of-magnitude energy reduction in AMC while maintaining system reliability.

Overall Contributions: Our resilience analysis framework MAVFI-family introduces intelligent fault injection to enable accurate end-to-end fault tolerance characterization of autonomous machines. Our lightweight adaptive protection scheme VPP efficiently exploits the inherent robustness variations to improve AMC resilience with little overhead. Our robust learning technique BERRY supports robust low-voltage computation. From the analysis framework to the protection design paradigm, our proposed methodologies co-optimize the efficiency, performance, and resilience of AMC.

4 RESULTS AND CONTRIBUTIONS

4.1 Skyline-AutoPilot: Agile AMC Design Flow

Skyline-AutoPilot achieves fast AMC full-system design space exploration and consistently generates optimal design configurations across deployment scenarios in an agile and scalable way.

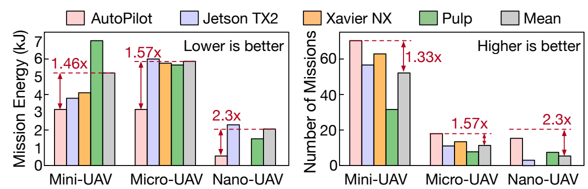


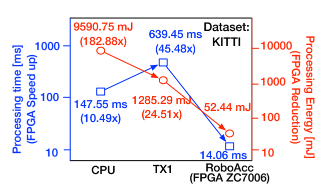
Figure 5: Skyline-AutoPilot quickly explores huge design space and consistently generates optimal AMC designs across scenarios.

Holistic AMC System Characterization. We use Skyline to demonstrate the need for holistic AMC full-system co-design to achieve maximum overall performance. High-performance onboard compute does not necessarily translate to high overall AMC performance due to cross-product effects. Miniaturization of autonomous machines puts greater emphasis on full system co-design. We experimentally validate Skyline using real UAVs and the error is between 5.1% to 9.5% compared to real-world mission tests [16, 18].

Fast AMC Design Space Exploration with Optimal Configuration. AutoPilot can navigate the huge AMC design space of 10^{18} configurations within 1 hour, outperforming prior solutions by 1000×. We show that design configurations generated by AutoPilot consistently outperforms general-purpose hardware and dedicated accelerators across different deployment scenarios, increasing the number of missions on average by 2.25× over baselines [15, 17] (Fig. 5). We demonstrate the need for automated flows to simplify the design process for autonomous cyber-physical systems.

4.2 RoboAcc: Efficient Adaptive AMC Hardware

RoboAcc achieves efficient and reconfigurable spatial-aware AMC under dynamic environments and demonstrates the significant energy efficiency and real-time performance improvements.



KITTI Dataset	RoboAcc Speedup		RoboAcc Energy Reduc.	
	Over CPU	Over TX1	Over CPU	Over TX1
FPGA ZC706	10.49x	45.48x	182.88x	24.51x
Kintex-7 Series	8.27x	35.82x	196.09x	26.28x
Virtix-7 Series	12.71x	55.08x	188.60x	25.28x

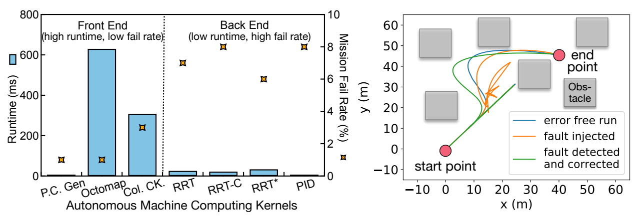
Figure 6: RoboAcc efficiently processes AMC kernels in real-time and is adaptive to dynamically changing environments.

Hardware-Efficient and Real-Time AMC. We demonstrate that, with improved hardware-friendly algorithms and optimized hardware architecture, RoboAcc achieves 10.49× speedup and 183× energy reduction over CPU on autonomous navigation task, as well as >5× better performance against prior robotic accelerators (Fig. 6). Based on RoboAcc, we also release the first deep-dive AMC book [21].

Adaptive AMC under Dynamic Environments. RoboAcc is dynamically optimized at runtime to adapt to different surroundings and save power while maintaining accuracy. Evaluated on outdoor environments, the runtime reconfigurable method further enables 1.59× power reduction with <0.01cm localization accuracy degradation.

4.3 MAVFI-BERRY: Resilient AMC Optimization

MAVFI-BERRY accurately evaluates AMC end-to-end resilience and enables adaptive AMC protection with low performance overhead.



(a) **(b)**

Figure 7: MAVFI-BERRY accurately evaluate end-to-end AMC resilience and significantly improve AMC resilience with low overhead.

Accurate AMC Resilience Evaluation. We integrate end-to-end resilience evaluation framework MAVFI on both autonomous vehicle and drone systems, and for the first time, we reveal the performance-robustness trade-offs in AMC. The front-end (sensing, localization, perception) has higher resilience but with higher latency and energy consumption, while the back-end (planning, decision-making) is more vulnerable to errors but with low runtime [32] (Fig. 7a).

Cost-Effective AMC Resilience Improvement. We validate adaptive protection design VPP with various environments on Intel CPU and Nvidia TX2 platforms. The system reliability is improved by 3× and the failure cases can be fully recovered in best-case scenarios [11, 29, 30] (Fig. 7b). Compared with traditional redundancy-based solutions, our lightweight application-aware protection scheme incurs <1% overhead with 18.5% further energy reductions when integrated with BERRY robust low-voltage operation.

4.4 Research Impacts

We present a cross-layer autonomous machine computing design solution to help build scalable, efficient, adaptive, and resilient autonomous machines. We delve into a range of methodologies encompassing automatic design space exploration, software-hardware co-design, and efficiency-resilience co-optimization, where they can be seamlessly integrated and synergistically advance the ubiquitous application of autonomous systems. Our research and endeavor in the AMC area have led to 1 book [21] in Synthesis Lectures on Computer Architecture and several first-authored [7, 11, 19, 29–38] and co-authored [2–4, 14–18, 27] publications in premier system/design automation/computer architecture/circuit journals and conferences. We believe our contribution to AMC can benefit a broad range of next-generation drone/vehicle/robot/mixed-reality system design and cognitive AI applications.

ACKNOWLEDGEMENT

We are grateful for the fruitful collaboration, support, and inspiration from Srivatsan Krishnan, Kshitij Bhardwaj, Thierry Tambe, David Brooks, Gu-Yeon Wei, Aleksandra Faust, Paul Whatmough, Shaoshan Liu, Bo Yu, Yiming Gan, Yuhao Zhu, Maximilian Lam, Sharad Chitlangia, Yu Wang, Yu-Shun Hsiao, Tianyu Jia, Abdulrahman Mahmoud, Karthik Swaminathan, Nandhini Chandramoorthy, Pin-Yu Chen, Aqeel Anwar, Ashwin Lele, Brian Crafton, Muya Chang, Samuel Spetalnick, Jong-Hyeok Yoon, Cong (Callie) Hao, Qiang Liu, Yanjun Zhang, etc.

REFERENCES

[1] P. Bannon, G. Venkataramanan, D. D. Sarma, and E. Talpes. 2019. Computer and redundancy solution for the full self-driving computer. In *2019 IEEE Hot Chips 31 Symposium (HCS)*. IEEE Computer Society, 1–22.

[2] K. Bhardwaj, Z. Wan, A. Raychowdhury, and R. Goldhahn. 2023. Real-Time Fully Unsupervised Domain Adaptation for Lane Detection in Autonomous Driving. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1–2.

[3] M. Chang, A. S. Lele, S. D. Spetalnick, B. Crafton, S. Konno, Z. Wan, A. Bhat, W.-S. Khwa, Y.-D. Chih, M.-F. Chang, et al. 2023. A 73.53 TOPS/W 14.74 TOPS heterogeneous RRAM In-memory and SRAM near-memory SoC for hybrid frame and event-based target tracking. In *2023 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 426–428.

[4] B. Crafton, Z. Wan, S. Spetalnick, J.-H. Yoon, W. Wu, C. Tokunaga, V. De, and A. Raychowdhury. 2022. Improving compute in-memory ECC reliability with successive correction. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*. 745–750.

[5] H. D. Dixit, S. Pendharkar, M. Beadon, C. Mason, T. Chakravarthy, B. Muthiah, and S. Sankar. 2021. Silent data corruptions at scale. *arXiv preprint arXiv:2102.11245* (2021).

[6] Y. Gan, P. Whatmough, J. Leng, B. Yu, S. Liu, and Y. Zhu. 2022. Braum: Analyzing and protecting autonomous machine software stack. In *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 85–96.

[7] T. Gao, Z. Wan, Y. Zhang, B. Yu, Y. Zhang, S. Liu, and A. Raychowdhury. 2021. IELAS: An ELAS-based energy-efficient accelerator for real-time stereo matching on FPGA platform. In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 1–4.

[8] R. Hadidi, B. Asgari, S. Jijina, A. Amyette, N. Shoghi, and H. Kim. 2021. Quantifying the design-space tradeoffs in autonomous drones. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 661–673.

[9] M. Hill and V. J. Reddi. 2019. Gables: A roofline model for mobile socs. In *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 317–330.

[10] P. H. Hochschild, P. Turner, J. C. Mogul, R. Govindaraju, P. Ranganathan, D. E. Culler, and A. Vahdat. 2021. Cores that don't count. In *Proceedings of the Workshop on Hot Topics in Operating Systems*. 9–16.

[11] Y.-S. Hsiao, Z. Wan, T. Jia, R. Ghosal, A. Mahmoud, A. Raychowdhury, D. Brooks, G.-Y. Wei, and V. J. Reddi. 2023. Mavfi: An end-to-end fault analysis framework with anomaly detection and recovery for micro aerial vehicles. In *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1–6.

[12] S. Jha, S. Banerjee, T. Tsai, S. K. Hari, M. B. Sullivan, Z. T. Kalbarczyk, S. W. Keckler, and R. K. Iyer. 2019. Ml-based fault injection for autonomous vehicles: A case for bayesian fault injection. In *2019 49th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. IEEE, 112–124.

[13] S. Jha, S. S. Banerjee, J. Cyriac, Z. T. Kalbarczyk, and R. K. Iyer. 2018. Avfi: Fault injection for autonomous vehicles. In *2018 48th annual ieee/ifip international conference on dependable systems and networks workshops (DSN-W)*. IEEE, 55–56.

[14] S. Krishnan, M. Lam, S. Chitlangia, Z. Wan, G. Barth-maron, A. Faust, and V. J. Reddi. 2022. QuaRL: Quantization for Fast and Environmentally Sustainable Reinforcement Learning. *Transactions on Machine Learning Research* (2022).

[15] S. Krishnan, T. Tambe, Z. Wan, and V. J. Reddi. 2021. Autosoc: Automating algorithm-soc co-design for aerial robots. *arXiv preprint arXiv:2109.05683* (2021).

[16] S. Krishnan, Z. Wan, K. Bhardwaj, N. Jadhav, A. Faust, and V. J. Reddi. 2022. Roofline model for uavs: A bottleneck analysis tool for onboard compute characterization of autonomous unmanned aerial vehicles. In *2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 162–174.

[17] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, S. Neuman, G.-Y. Wei, D. Brooks, and V. J. Reddi. 2022. Automatic Domain-Specific SoC Design for Autonomous Unmanned Aerial Vehicles. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 300–317.

[18] S. Krishnan, Z. Wan, K. Bhardwaj, P. Whatmough, A. Faust, G.-Y. Wei, D. Brooks, and V. J. Reddi. 2020. The sky is not the limit: A visual performance model for cyber-physical co-design in autonomous machines. *IEEE Computer Architecture Letters* 19, 1 (2020), 38–42.

[19] Q. Liu, Z. Wan, B. Yu, W. Liu, S. Liu, and A. Raychowdhury. 2022. An energy-efficient and runtime-reconfigurable fpga-based accelerator for robotic localization systems. In *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 01–02.

[20] R. Liu, J. Yang, Y. Chen, and W. Zhao. 2019. eslam: An energy-efficient accelerator for real-time orb-slam on fpga platform. In *Proceedings of the 56th Annual Design Automation Conference 2019*. 1–6.

[21] S. Liu, Z. Wan, B. Yu, and Y. Wang. 2021. Robotic computing on fpgas. *Synthesis Lectures on Computer Architecture* 16, 1 (2021), 1–218.

[22] M. Mohanan and A. Salgoankar. 2018. A survey of robotic motion planning in dynamic environments. *Robotics and Autonomous Systems* 100 (2018), 171–185.

[23] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics* 31, 5 (2015), 1147–1163.

[24] D. Palossi, A. Loquercio, F. Conti, E. Flamand, D. Scaramuzza, and L. Benini. 2019. A 64-mw dnn-based visual navigation engine for autonomous nano-drones. *IEEE Internet of Things Journal* 6, 5 (2019), 8357–8371.

[25] S. Park, L. Zhang, and S. Chakraborty. 2016. Design space exploration of drone infrastructure for large-scale delivery services. In *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 1–7.

[26] A. Suleiman, Z. Zhang, L. Carlone, S. Karaman, and V. Sze. 2019. Navion: A 2-mw fully integrated real-time visual-inertial odometry accelerator for autonomous navigation of nano drones. *IEEE Journal of Solid-State Circuits* 54, 4 (2019), 1106–1119.

[27] T. Tambe, E.-Y. Yang, Z. Wan, Y. Deng, V. J. Reddi, A. Rush, D. Brooks, and G.-Y. Wei. 2020. Algorithm-hardware co-design of adaptive floating-point encodings for resilient deep learning inference. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[28] T. Tsai, S. K. S. Hari, M. Sullivan, O. Villa, and S. W. Keckler. 2021. Nvbitfi: Dynamic fault injection for gpus. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 284–291.

[29] Z. Wan, A. Anwar, Y.-S. Hsiao, T. Jia, V. J. Reddi, and A. Raychowdhury. 2021. Analyzing and improving fault tolerance of learning-based navigation systems. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 841–846.

[30] Z. Wan, A. Anwar, A. Mahmoud, T. Jia, Y.-S. Hsiao, V. J. Reddi, and A. Raychowdhury. 2022. Frl-fi: Transient fault analysis for federated reinforcement learning-based navigation systems. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 430–435.

[31] Z. Wan, B. Crafton, S. Spetalnick, J.-H. Yoon, and A. Raychowdhury. 2022. RRAM-ECC: Improving Reliability of RRAM-Based Compute In-Memory. In *13th Annual Non-Volatile Memories Workshop (NVMW)*.

[32] Z. Wan, Y. Gan, B. Yu, S. Liu, A. Raychowdhury, and Y. Zhu. 2023. VPP: The Vulnerability-Proportional Protection Paradigm Towards Reliable Autonomous Machines. In *Proceedings of the 5th International Workshop on Domain Specific System Architecture (DOSSA)*. 1–6.

[33] Z. Wan, A. Lele, B. Yu, S. Liu, Y. Wang, V. J. Reddi, C. Hao, and A. Raychowdhury. 2022. Robotic computing on fpgas: Current progress, research challenges, and opportunities. In *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 291–295.

[34] Z. Wan, A. S. Lele, and A. Raychowdhury. 2022. Circuit and system technologies for energy-efficient edge robotics. In *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 275–280.

[35] Z. Wan, K. Swaminathan, P.-Y. Chen, N. Chandramoorthy, and A. Raychowdhury. 2022. Analyzing and Improving Resilience and Robustness of Autonomous Systems. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 1–9.

[36] Z. Wan, K. Swaminathan, P.-Y. Chen, N. Chandramoorthy, V. J. Reddi, and A. Raychowdhury. 2023. BERRY: Bit Error Robustness for Energy-Efficient Reinforcement Learning-Based Autonomous Systems. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.

[37] Z. Wan, B. Yu, T. Y. Li, J. Tang, Y. Zhu, Y. Wang, A. Raychowdhury, and S. Liu. 2021. A survey of fpga-based robotic computing. *IEEE Circuits and Systems Magazine* 21, 2 (2021), 48–74.

[38] Z. Wan, Y. Zhang, A. Raychowdhury, B. Yu, Y. Zhang, and S. Liu. 2021. An energy-efficient quad-camera visual system for autonomous machines on fpga platform. In *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*. IEEE, 1–4.