AutoX: Automating Algorithm SOC Co-Design for Aerial Autonomous Machines

HARVARD
John A. Paulson
School of Engineering
and Applied Sciences

Srivatsan Krishnan, Thierry Tambe, Zishen Wan

Introduction:

- Aerial robotics is a cross-layer, interdisciplinary field that spans from algorithm design to all the way to choice of hardware for the onboard compute.
- End-to-End learning methods based on a neural network model show great promise to replace the traditional 'Sense-Plan-Act' paradigm to control the aerial robot.
- In End-to-End learning, the neural network model directly processes on the sensor data to generate actuation commands to control the aerial robot.

Challenges:

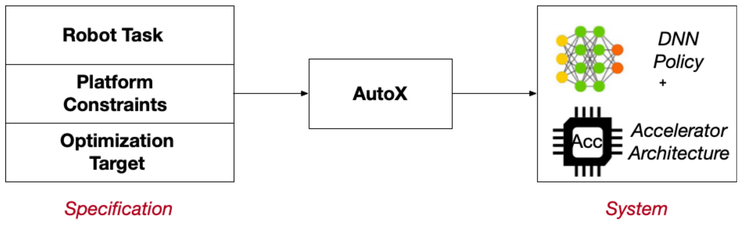
- No standard neural network architecture (Policy) for aerial robot control.
- The neural network model is in the critical path which determines the closed-loop control frequency.
- For agile maneuvers, the closed loop control frequency has to be as high. For instance, for a 30 FPS camera, the processing of End-to-End policy must be at least 33ms.
- Since the power envelope for onboard compute is constrained, the processing of End-to-End policy must also be energy efficient.

Problem Statement:

Given the interdisciplinary nature of the problem, what are the various infrastructure components needed to systematically co-design algorithm and onboard compute for a given aerial robot task.

AutoX:

To study the complex domain systematically starting from robotics task all the way exploring hardware accelerator solution for aerial robotics, we propose AutoX, an agile algorithm-SOC co-design platform. The following figure shows the high-level overview of AutoX:

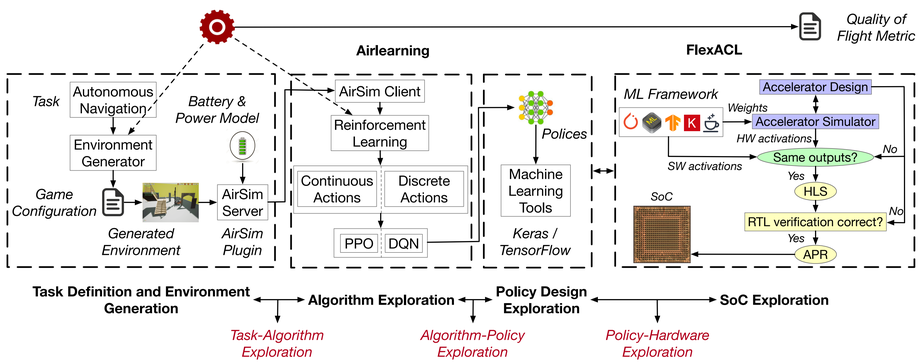


In AutoX, the user inputs the task and platform constraints and the framework will determine the best neural network policy and the best accelerator template that can deliver the performance within the specified platform constraints.

Infrastructure:

AutoX infrastructure has two major components namely "Air Learning" and "FlexACL". Air Learning is a simulation framework to develop end-to-end learning-based policies for aerial robotics tasks. An example task includes navigation of aerial robotic in a densely clustered environment.

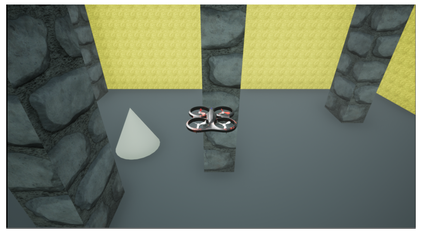
Another major component is FlexACL, which is a modular accelerator template based on SystemC + HLS flow. It generates RTL from a C++ code producing a hardware accelerator template with AXI interface which can be plugged as an IP into an SOC.



Experiment Methodology:

The robot task is end-to-end navigation in a cluttered environments with obstacles. To that end, we use the following Air Learning Environment Settings:

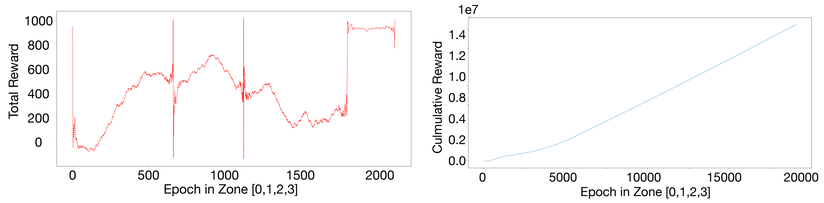
Parameters	Value
Arena Size	[25 m, 25m , 20m]
Static Obstacles	[1, 5]
Seed	Random
Goal Position	Random



We use Deep Q-Network as the end-to-end learning algorithm. The reward function that we use is defined as:

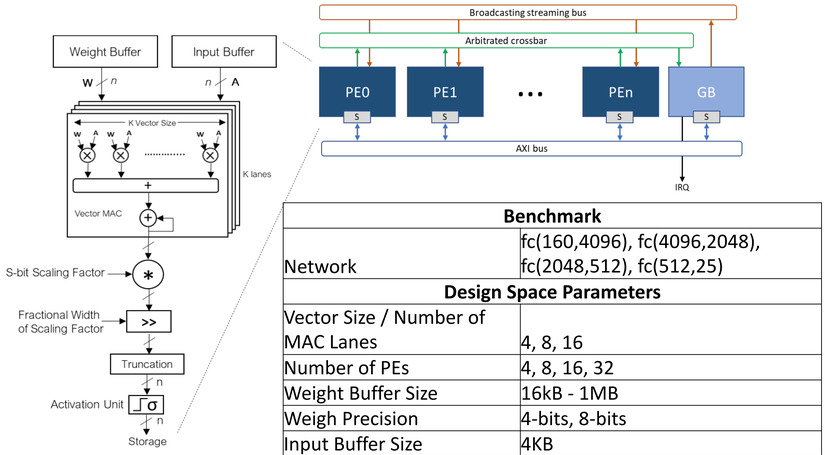
$$R = 1000*\alpha - 100*\beta - Dg - Dc*\delta - 1$$

The input is 1x 160 vector that comprises of depth sensor, position and other IMU data. The neural network architecture is a simple three-layer MLP with 4096, 2048, and 512 hidden states. The final layer has 25 hidden states that corresponds to the action space to control the aerial robot.



FlexACL Accelerator Template

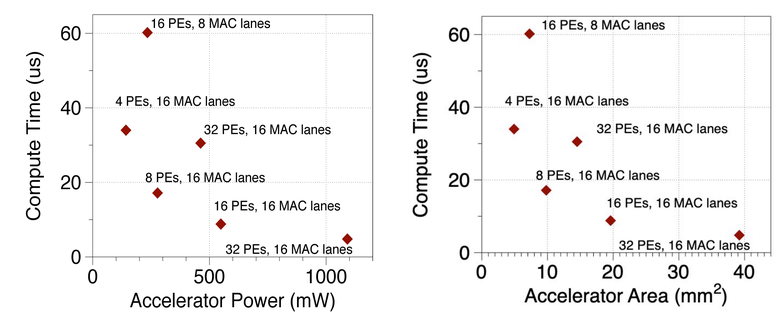
The FlexACL architecture template is shown below with a close-up of its parameterizable integer-based processing element (PE). The accelerator candidates are produced by sweeping the design parameters shown in the table.



Benchmark	
Network	fc(160,4096), fc(4096,2048), fc(2048,512), fc(512,25)
Design Space Parameters	
Vector Size / Number of MAC Lanes	4, 8, 16
Number of PEs	4, 8, 16, 32
Weight Buffer Size	16kB - 1MB
Weigh Precision	4-bits, 8-bits
Input Buffer Size	4KB
Global Buffer Size	4KB
Frequency	300MHz

Experimental Results:

The performance, power and area of a few generated FlexACL accelerator candidates are shown below:



Example of a generated placed-and-route layout of the 4PEs-16Mac lanes accelerator candidate