14.1 A 65nm 1.1-to-9.1TOPS/W Hybrid-Digital-Mixed-Signal Computing Platform for Accelerating Model-Based and Model-Free Swarm Robotics

Ningyuan Cao, Muya Chang, Arijit Raychowdhury

Georgia Institute of Technology, Atlanta, GA

Artificial swarm intelligence, inspired by biological studies of insects, ants and other organisms, present an emerging computing paradigm, where seemingly simple elements interact with each other to collectively solve challenging problems. In particular, swarm robotics, where multiple robots co-ordinate in real-time to solve diverse problems such as pattern-formation, cooperative reinforcement learning (RL), path-planning etc. [1], find extensive uses in exploration, reconnaissance and disaster relief. This is partly motivated by the robustness of swarm dynamics to failures and malfunctions of individual robots. Successful hardware demonstrations of neuro-inspired algorithms on edge-devices [2-6] is now leading to the emergence of intelligence and control in swarms as the next frontier. Although certain swarm algorithms rely on real-time learning (e.g., cooperative RL) representing a *model-free* approach, many powerful algorithms that have been developed over the past two decades (e.g., pattern formation) rely on a mathematical structure and represent a more traditional *model-based* approach. The next generation of swarm hardware needs to support both of these approaches. In this paper, we identify the commonalities and shared compute primitives across a variety of *model-based* and *model-free* swarm algorithms and present a unified, fully-programmable, energy-efficient and scalable platform capable of real-time swarm intelligence.

While *model-based* applications (pattern formation and coordinated obstacle avoidance) require calculation of vector fields, *model-free* applications (multi-robot predator-prey, patrolling and exploration) rely on RL-based training of neural-network (NN) models (Fig. 14.1.1). The fact that both of these approaches require linear algebraic kernels, as well as non-linear processing units (e.g., trigonometric units in vector fields and non-linear activation in NNs), motivates us to develop a common computing platform to support both. The principal algorithmic approaches and computing primitives are summarized in Fig. 14.1.1.

Figure 14.1.2 illustrates the architecture of the test-chip fabricated in 65nm CMOS. It features a 16KB data-cache, 27B instruction cache, a processing unit and peripheral circuits for control, data-load and write-back. The test-chip interfaces with a raspberry-Pi platform consisting of integrated sensors (inertial sensors and ultrasonic distance sensors) and LoRa (Long Range) radios for decentralized, peer-to-peer communication among mobile robotic vehicles in a swarm. The processing unit consists of two principal modules (1) a non-linear function evaluator (NFE) and (2) a linear processing unit (LPU). We realize a variety of relevant non-linear functions of input (x) using a piecewise linear approximation, where each piece is characterized by a slope (g_{ref}) and an offset (y_{ref}). The number of linear segments of each function depends on the number of inflection points and the (g_{ref}, y_{ref}) for each segment is stored in the data cache as look-up table (LUT). Data-volume is reduced by aliasing periodic functions to their fundamental ranges and comparing x with the period boundaries, as illustrated in Fig. 14.1.2. The test-chip provides support for 7 non-linear functions determined by the control signal, PU[2:0]. Architecturally, the chip supports high-level functions, which are scanned in as 8b instructions to the instruction cache. Each instruction is decoded and the relevant control signals and data read from the data cache are transmitted to the processing unit. Once an output is generated, it is written back to the data cache, enabling a sequential execution of tasks. The large data cache allows *model-free* algorithms and a 3-layered neural network for RL for up to 20 agents.

Swarm algorithms are characterized by dynamically varying environments and need to support various swarm sizes. A detailed study of *model-based* and *model-free* algorithms reveals that as the swarm sizes increase, vector dimensions, as well as the range of operand values required for correct execution also increase. This is shown in Fig. 14.1.3 for path-planning and predator-prey problems with 2 and 20 agents. The required bit-precision as a function of swarm size and algorithm is summarized. This motivates the design of an energy-scalable LPU capable of easy reconfiguration and high energy efficiency across the range. Time-domain mixed-signal (TD-MS) MACs [2] have been shown to offer voltage-scalability and energy-benefits compared to digital counterparts for lower

data-width. However, with increasing operand size, the energy/MAC increases non-linearly and surpasses digital logic, as shown in the energy-map and corresponding table. To address this issue, we propose a hybrid-digital-mixed-signal (HDMS) circuit, where a 5b TD-MS multiplier (4 data bits and 1 sign bit represented in the signed magnitude format [2]) is nested within a digital shift-add loop, such that execution is purely TD-MS for bit-width<=5b and hybrid of TD-MS and digital (5b TD-MS followed by a sequence of digital shift and adds) for $6 \leq$ bit-width ≤ 8. This creates an energy map with high efficiency across the range and results in 81% (for 3b operations) to 31% (for 8b operations) average energy/MAC reduction compared to digital implementations. The LPU consists of a 3-stage pipeline and supports 6 linear operations of one or multiple operands with varying latencies.

The TD-MS 5b multiplier kernel consists of a digital-to-pulse converter (DPC) which converts operand (A) to a delay ($A.T_0$) via two matched digital-to-time-converters (DTC) followed by an XOR, as shown in Fig. 14.1.4. The DPC output gates a digitally-controller-oscillator (DCO) with a control word (B), which produces a frequency ($B.F_0$). The DCO clocks an 8b counter to produce ($A.B.T_0.F_0$). We match F_0 and T_0 by using the same logic gates and the output captures $A.B$. Even if there is a mismatch between F_0 and T_0, the output gets scaled and the algorithms is robust against such scaling. The TD-MS inverter can be further calibrated via analog control signals (DCO_BL and DCO_BH). The TD-MS kernels allow 3-to-5b reconfiguration without any overhead and easy interface to digital memory. Further, the digital shift-and-add circuits allow seamless transition from 6-to-8b. Measured responses of the DCO and DPC show acceptable linearity of less than 1.2 lsb (worst-case) at V_{CC}=1.0V and V_{CC}=0.6V.

Figure 14.1.5 illustrates the measured F_{MAX} and logic-power dissipation showing functionality down to 0.36V and a peak logic power of 3.2μW (1.9μW) for 8b (5b) operation. Measured energy/op shows excellent energy-resolution scalability reaching a peak of 0.22pJ/MAC (at 3b) and 1.76pJ/MAC (at 8b). We measure the average arithmetic energy efficiency as a function of V_{CC} and record 9.1TOPS/W (3b) to 1.1TOPS/W (8b). We plot the energy break-down of the computation unit where the LPU (NFE) consumes 88% (12%) of the logic power. The power distribution across the various blocks of the LPU are shown.

We benchmark the test-chip and note competitive figures-of-merit for an emerging application. The test-chip is mounted on a robotic car (Fig. 14.1.6) and interfaces with a Raspberry-Pi, motor-controllers, sensors and LoRa radios. We implement 4 template swarm algorithms (*model-based*: path planning and pattern formation, and *model-free*: predator prey and joint exploration). In Fig. 14.1.6 we demonstrate (1) two robots (R1 and R2) capture two targets (T1 and T2) collaboratively, (2) three robots starting from three random locations converge to a triangular pattern, and (3) cooperative RL where the number of iterations required to perform obstacle avoidance reduces with the number of training examples. We further benchmark the energy/task and the number of actions taken per second for the template problems, illustrating a wide diversity and complexity of applications. The die-shot and the chip-characteristics are shown in Fig. 14.1.7.

References:
[1] M. La et. al., "Multirobot Cooperative Learning for Predator Avoidance," *IEEE Trans. on Control Systems Technology*, vol. 23, no. 1, pp. 52-63, 2015.
[2] A. Amravati et. al., "A 55nm Time-Domain Mixed-Signal Neuromorphic Accelerator with Stochastic Synapses and Embedded Reinforcement Learning for Autonomous Micro-Robots," *ISSCC*, pp. 124-125, 2018.
[3] B. Moons et. al., "Envision: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable Convolutional Neural Network processor in 28nm FDSOI," *ISSCC*, pp. 246-247, 2017.
[4] J. Sim et al., "A 1.42TOPS/W Deep Convolutional Neural Network Recognition Processor for Intelligent IoE Systems," *ISSCC*, pp. 264-265, 2016.
[5] S. Choi et. al., "A 9.02mW CNN-Stereo-Based Real-Time 3D Hand-Gesture Recognition Processor for Smart Mobile Devices," *ISSCC*, pp. 220-221, 2018.
[6] Y. Chen et. al., "Eyeriss: An Energy-Efficient Reconfigurable Accelerator for Deep Convolutional Neural Networks," *IEEE JSSC*, vol. 52, no. 1, pp. 127-138, 2017.
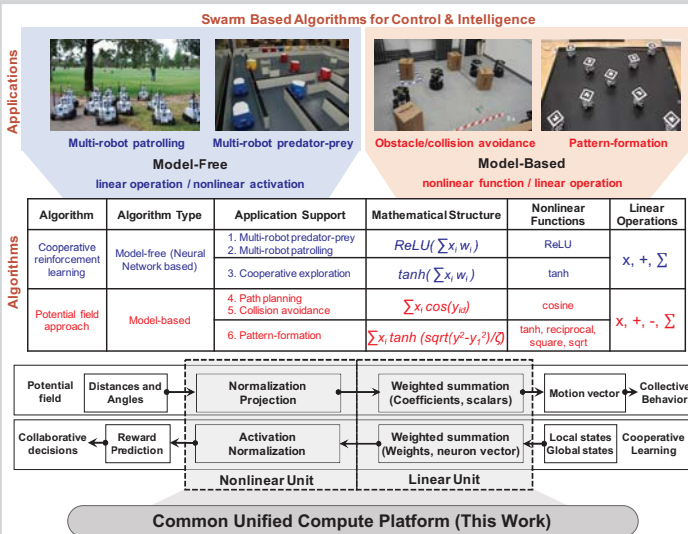
Figure 14.1.1: A common unified compute platform supporting model-based and model-free algorithms in swarm robotics.
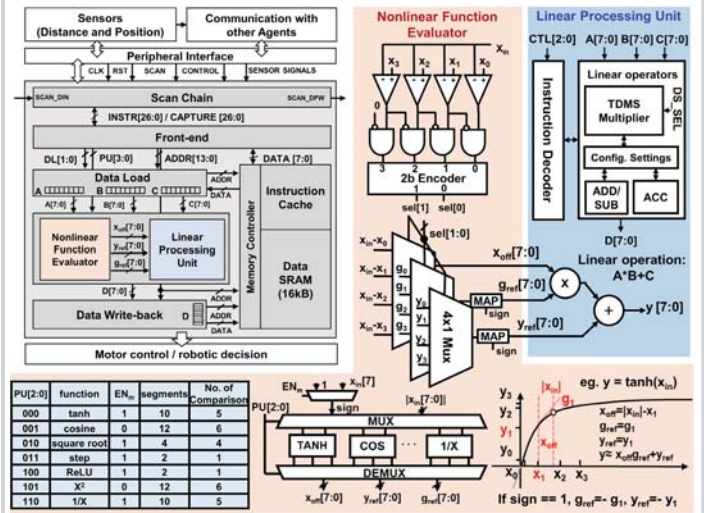

Figure 14.1.2: System architecture illustrating on-chip memory, compute unit and peripheral circuits.
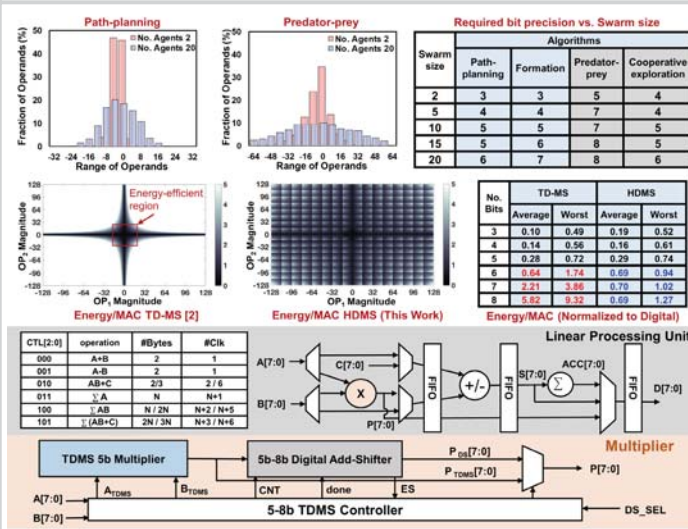

Figure 14.1.3: Hybrid-digital-mixed-signal LPU provides energy-resolution scalability across a variety of algorithms and swarm sizes.
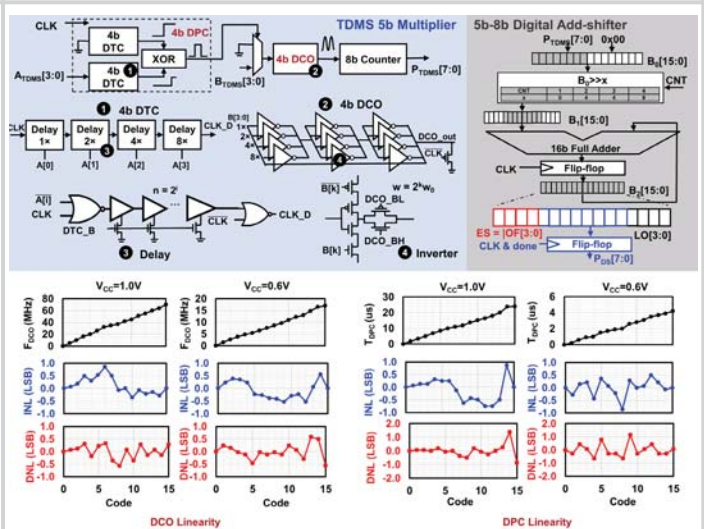

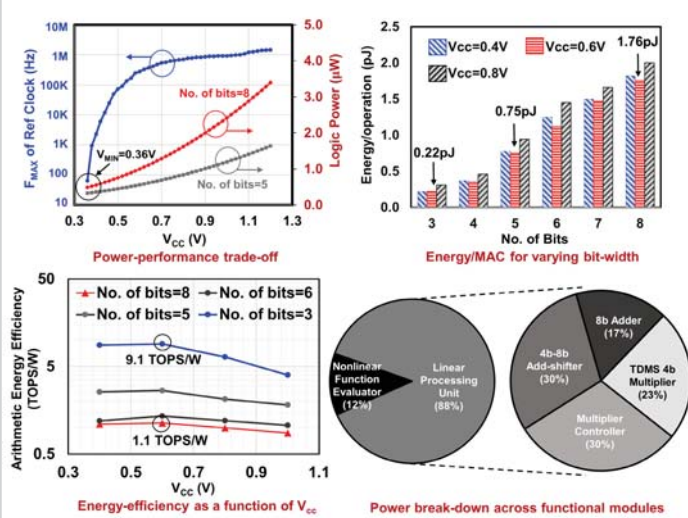Figure 14.1.4: HDMS-based LPU illustrating various design components and measured linearity of DCO and DPC.

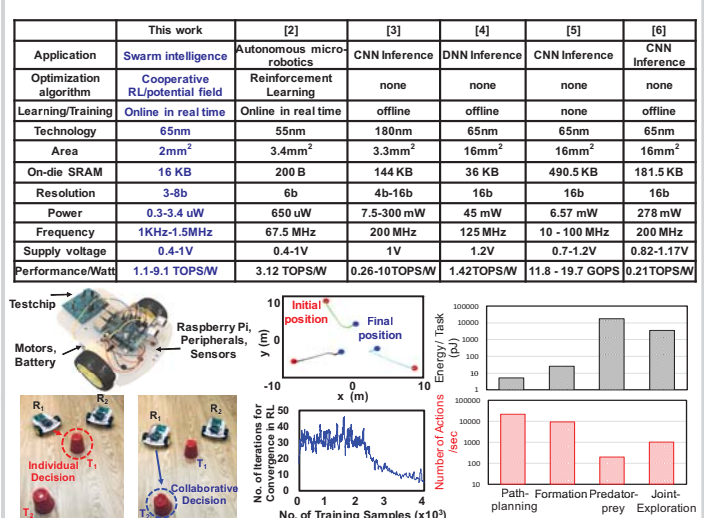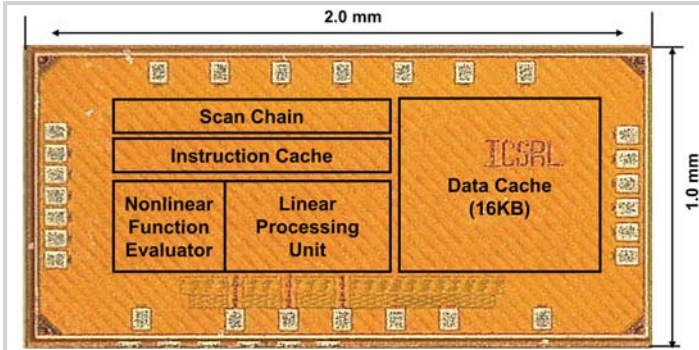Figure 14.1.5: Measured power, performance and energy-efficiency.

Figure 14.1.6: Benchmarking and application to mobile swarm robotics.

14

Chip Characteristics	
Technology	65nm 1P9M CMOS
Die area	1mm*2mm
Testing interface	QFN package
Pin Count	28

Figure 14.1.7: Die shot and chip characteristics.