DISSERTATIONES
TECHNOLOGIAE
UNIVERSITATIS
TARTUENSIS
82

IMAN DADRAS

Low power neural network-based control and actuation solutions for insect-scale robots

DISSERTATIONES TECHNOLOGIAE UNIVERSITATIS TARTUENSIS
82

DISSERTATIONES TECHNOLOGIAE UNIVERSITATIS TARTUENSIS
82

IMAN DADRAS

Low power neural network-based control and actuation solutions for insect-scale robots

UNIVERSITY OF TARTU
Press
1632

Institute of Technology, Faculty of Science and Technology, University of Tartu, Estonia.

The dissertation was accepted for the commencement of the degree of Doctor of Philosophy in Physical Engineering on May 17, 2024, by the Joint Council of the Doctoral Program of Engineering and Technology of the University of Tartu.

Supervisors: Alvo Aabloo, PhD
Professor of Polymeric Materials, Materials Science
Institute of Technology, University of Tartu,
Tartu, Estonia

Jaan Raik, PhD
Tenured Full Professor
Department of Computer Systems,
School of Information Technologies,
Tallinn University of Technology,Tallinn, Estonia

Reviewer: Masoud Daneshtalab, PhD
Professor, Intelligent Future Technology,
University of Mälardalen, Sweden

Opponent: Haralampos-G. Stratigopoulos, PhD
Professor, Director of research CIAN -
Analog and Digital Integrated Circuit
Sorbonne University, UPMC-LIP6, France

Commencement: Auditorium 121, Nooruse 1, Tartu, Estonia,
at 12.15 on June 27th, 2024

Publication of this thesis is granted by the Institute of Technology, Faculty of Science and Technology, University of Tartu.

ISSN 2228-0855 (print)
ISBN 978-9916-27-548-1 (print)
ISSN 2806-2620 (pdf)
ISBN 978-9916-27-549-8 (pdf)

University of Tartu Press
http://www.tyk.ee/

In the loving memory of my mother
To my beloved wife, Fatemeh
To my dearest sister, Azar
And to my supportive father, Mahmoud

ABSTRACT

Insect-sized robots have an immense potential for applications in various fields, such as remote inspection of hard-to-reach places, data gathering in environmental monitoring, search and rescue operations after natural disasters, and space and deep-sea exploration. Their small size requires minimal material, making them cost-efficient and easy to produce and operate in swarms. Hence, they are ideal for tasks that demand vast coverage of an area if they operate autonomously.

However, achieving power and control autonomy in insect-scale robots is still a challenging task. The piezoelectric actuators utilized in flying miniature robots require a considerable amount of power, which exceeds the minuscule payload capacity of insect-scale robots. Additionally, the electronics necessary for control autonomy not only consume a significant amount of power but also add to the weight of the robot. Therefore, insect-scale robots are tethered to supply their power and control, significantly reducing their field of operation.

This thesis proposes an innovative way to enable insect-scale robots to achieve power autonomy by using alternative actuator types and novel computation circuits that consume less power. Ionic electroactive polymer-based soft actuators are a feasible low-power option for the locomotion of insect-scale robots, which can help reduce power consumption by allowing the robot to crawl slowly instead of flying. We model these actuators from a robotic perspective in the Laplace domain. We also use a hybrid network-hardware codesign approach to design new application-specific integrated circuits for convolutional neural network acceleration, enabling the minuscule robot to perform visual control. These designs incorporate new hardware architectures, network structures, and computation techniques that help to reduce power consumption. One of the accelerators presented in this thesis performed classification with less than 1.5 nW per image. The results of this thesis pave the way for future research and development on autonomous insect-scale robots.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Acronyms

ADC Analog-to-Digital Converter. 45, 47–52, 55, 61, 63, 64, 67–69, 71, 74, 77

AI Artificial Intelligence. 18, 20–24, 26, 30, 34, 45, 65, 86

AIMC Analog In Memory Computing. 7, 8, 10, 45–58

APE Analog Processing Elements. 8, 10, 51, 52, 54–60

ASIC Application-Specific Integrated Circuit. 45, 46, 63, 64, 66

BNN Binary Neural Network. 66, 76

CIS CMOS Image Sensor. 66, 67

CMOS Complementary Metal-Oxide-Semiconductor. 18

CNN Convolutional Neural Networks. 8, 32, 33, 45, 65–68, 70, 71, 74, 76–80, 82, 86

CNT- Carbon NanoTube. 30

CP Conducting or Conjugated Polymer. 30

DAC Digital-to-Analog Converter. 8, 50, 51, 57, 59, 67, 68, 71, 77

DEA Dielectric Elastomer Actuators. 29

FC Fully-Connected. 65, 79, 81–85

FE Feature Extractor. 79, 84

FFCNN fully-fusible CNN. 8, 12, 78–82, 86

FMC FPGA Mezzanine Card. 54

HAMR Harvard Ambulatory MicroRobot. 29

IC Integrated Circuit. 18

IEAP Ionic ElectroActive Polymer. 10, 17, 18, 21, 22, 24, 29–31, 36, 43

IMC In-Memory Computing. 33, 46

IPMC Ionic Polymer–Metal Composites. 30

IPN Interpenetrating Polymer Network. 30

ISR Insect-Scale Robots. 17

LPC Low Pin Count. 54

LWCNN LightWeight Convolutional Neural Network. 8–11, 69, 70, 75, 77, 78, 82–86

MEMS Micro-ElectroMechanical Systmem. 24

ML Machine Learning. 30

NVM Non-Volatile memories. 47

PCM phase-change memory. 47

PEDOT:PSS Poly (3,4-ethylenedioxythiophene) polystyrene sulfonate. 30, 36

PWM Pulse Width Modulation. 48–52, 57, 58

RRAM Resistive Random Access Memory. 47

SBC Single Board Computer. 83

SoC System on Chip. 45, 46, 48, 51, 54, 62, 83

LIST OF ORIGINAL PUBLICATIONS

Publications included in the thesis

1. **I. Dadras** et al., "Modeling and Experimental Analysis of the Mass Loading Effect on Micro-Ionic Polymer Actuators Using Step Response Identification," in **Journal of Microelectromechanical Systems**, vol. 30, no. 2, pp. 243-252, April 2021, doi: 10.1109/JMEMS.2021.3060897.
   Author contributions: Experiment planning, equation and model development, and authoring the main part of the manuscript.

2. **I. Dadras**, M. H. Ahmadilivani, S. Banerji, J. Raik and A. Abloo, "An Efficient Analog Convolutional Neural Network Hardware Accelerator Enabled by a Novel Memoryless Architecture for Insect-Sized Robots," 2022 11th **International Conference on Modern Circuits and Systems Technologies (MOCAST)**, Bremen, Germany, 2022, pp. 1-6, doi: 10.1109/MOCAST54814.2022.9837551. **IEEE International Conference on Modern Circuits and Systems Technologies (MOCAST)**, 2022.

   Author contributions: AI network architecture, analog circuit design, and authoring the core manuscript.

3. **I. Dadras**, G. M. Sarda, N. Laubeuf, D. Bhattacharjee and A. Mallik, "AIMC Modeling and Parameter Tuning for Layer-Wise Optimal Operating Point in DNN Inference," in **IEEE Access**, vol. 11, pp. 87189-87199, 2023, doi: 10.1109/ACCESS.2023.3305432

   Author contributions: Conducting experiments and analyses, formulating mathematical expressions, constructing mathematical models, authoring core manuscript

4. **I. Dadras**, S. Seydi, M. H. Ahmadilivani, J. Raik and M. E. Salehi, "Fully-Fusible Convolutional Neural Networks for End-to-End Fused Architecture with FPGA Implementation," 2023 30th **IEEE International Conference on Electronics, Circuits and Systems (ICECS)**, Istanbul, Turkiye, 2023, pp. 1-5, doi: 10.1109/ICECS58634.2023.10382831.

   Author contributions: Conceiving the central concept, AI network architecture, and authoring the core manuscript.

1. INTRODUCTION

A robot, in its most general terms, is an entity with intelligence embodied in a physical form, sensing and manipulating its environment. Thus, robotics is a multidisciplinary field of research by definition. It necessitates different electronic and mechanical subsystems for sensing, intelligence, manipulation, and locomotion in a minimal approach. Other sciences, such as biology and chemistry, can also be required for robot operation in specific environments.

Since the introduction of Shakey [1], the first general-purpose mobile robot, in the 1960s, robots have taken many different form factors and used various locomotion principles. We will see some interesting examples of them in the next sections. Insect-Scale Robots (ISR), categorized by their small size, are emerging as a fascinating section of robots promising revolutionary applications.

1.1. Motivation

Miniaturization in robotics is opening a new field of study with unprecedented opportunities: insect-scale robotics. Terrestrial [2], aquatic [3], and aerial [4] insect-scale robots have been developed recently. They range from one to a few centimeters in size and ten milligrams to tens of grams in weight. Their prospective applications include confined space inspection, swarm robotics for search and rescue, environmental monitoring, and space or deep-sea exploration.

While piezoelectric actuators are widely adopted in the field for their high bandwidth and substantial force excursion, their huge power consumption (300 mW [5]) has hampered insect-scale robot power autonomy. Piezoelectric actuators enable insect-scale robots to fly. However, their bulky DC-DC converters and batteries exceed the robot's payload capacity, necessitating a tether to an external power source, reducing their autonomy. Ionic ElectroActive Polymer (IEAP) actuators present attractive alternatives due to their lower power consumption and operating voltage, leading to insect-scale robots' power autonomy. An autonomous IEAP-based worm-like robot is showcased in [6]. In addition to lower power consumption, it boasts exceptional flexibility and bio-compliance requirements. These characteristics make IEAP actuators particularly well-suited for applications in fields like minimally invasive surgery in medicine and search and rescue operations in confined spaces. Nonetheless, it is worth noting that IEAP actuators do have certain limitations, such as a reduced capacity for generating high forces, low frequency, and imprecise control. Nevertheless, ongoing research continually addresses these challenges and expands the range of applications where IEAP actuators can be effectively employed.

As the human body is more than just muscles, a robot is more than its mechanical components. A successful robot necessitates a brain—an electronic control module—and eyes in the form of sensors to accompany its actuators. However, recent advancements in insect-scale robotics have predominantly focused on enhancing actuator technology, which has resulted in tethered robots or those with limited autonomy. Due to undeveloped proper electronics and power-hungry actuators, the robots rely on an external power and control unit source.

This observation may appear counterintuitive in the context of the relentless scaling of Complementary Metal-Oxide-Semiconductor (CMOS) technology and the increasing number of transistors on an Integrated Circuit (IC), often described as Moore's Law: the number of transistors on a chip with a certain area doubles every two years [7]. This trend is illustrated in Fig. 1. While scaling has brought numerous benefits in terms of reducing the size, weight, and power consumption of chips, it is most advantageous when circuitry is highly integrated, minimizing IC pads, interconnections, and power losses. However, the computational demands of Artificial Intelligence (AI) algorithms, one of the most potent tools in robotics, remain high and require significant hardware resources, even as technology nodes become smaller.

Incorporating highly integrated electronics into insect-scale robots empowered by low-power actuators can solve the challenges of limited autonomy and propel us toward developing autonomous small robots that can be utilized across a wide range of applications. This research seeks to bridge the gap between advanced actuator technologies and the electronic control systems required for fully autonomous AI-powered insect-scale robots by tackling the following challenges:

1. **Lack of Sufficient Model for IEAP Actuators:** There is no comprehensive model for the IEAP actuators. This leads to a lack of information on their maximum payload, which limits the electronics and power supply form factor. The model is also essential for robot control.
2. **Shortage of Electronics for Sensory and Control Circuitry:** Insect-scale robots lack the necessary electronic designs for sensory and control circuitry. This gap poses a fundamental obstacle to their autonomy, as sensors and control systems are essential for navigation, data collection, and intelligent decision-making.
3. **Challenges in Implementing AI Algorithms:** While AI algorithms are powerful tools for robotics, their implementation on insect-scale robots presents significant challenges. The hardware requirements for AI algorithms can be too demanding, both in terms of processing power and energy consumption, for electronics that can fit

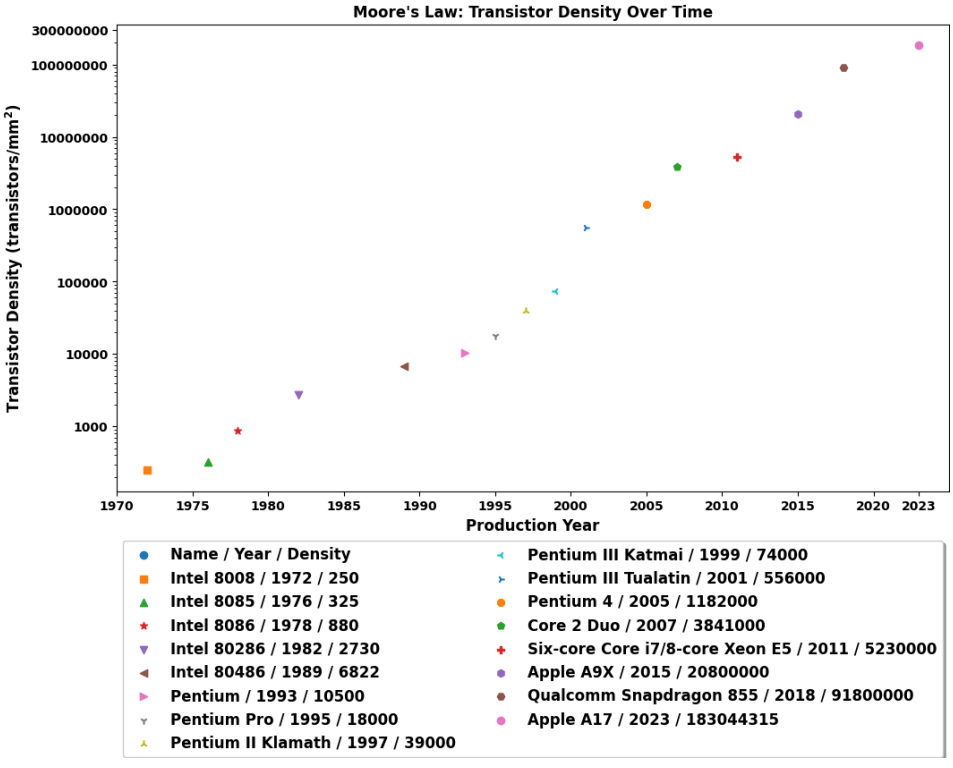


Figure 1. The figure demonstrates the Moore's law. The number of transistors in a chip doubled every two years. However, the power per area remained constant. Thus, a chip performing the same operation became twice smaller and low-power each other year. Despite these advancements, insect-scale robot electronics still hinder its autonomy.

within the constraints of insect-scale robots. These challenges must be overcome to harness the full potential of AI in these small-scale robotic systems.

1.2. Problem Formulation

Developing different levels of abstraction has been one of the indispensable tools that allowed us to advance complex technologies. The engineers can focus on some levels of abstraction and avoid being overwhelmed with many intricate details from other levels. However, it can preclude cross-level optimization and co-designs, benefiting the system's top-level performance [8]. Aside from the hierarchical partitioning, there is horizontal partitioning between systems for locomotion, control, sensing, and power in insect-scale robots [9]. A clear top-level view of all these sections can lead to an efficient robot.

In this thesis, I look into different levels of abstraction of various subsystems to find opportunities for new co-designs, which improve performance and efficiency, in order to approach an autonomous insect-scale robot. The control and electronic system is the primary focus of the thesis. Another significant section of the thesis explores robot locomotion. Power and sensing systems are not studied directly. However, a constant quest for lower power consumption in electronic system design and power analysis in locomotion deals with power system restrictions remotely. Moreover, the actuators for the locomotion system suggested in this thesis consume much less power and operate on lower voltages compared to their piezoelectric counterparts. The designed electronic system also contains signal-processing circuitry for handling the sensory outputs.

A heavy burden on the power subsystem is the high voltage requirement for conventional actuators in the field of insect-scale robots [5]. The thesis will investigate alternative actuators with lower operating voltages in cross-subsystem optimization. The power consumption is another limiting factor. It increases the power source weight beyond the robot payload capacity. Electronics, sensing, and mechanical subsystems share in the power consumption. The share of electronics can be reduced by a cross-abstraction-level design in which the algorithm, hardware architecture, and transistor level are co-optimized to achieve lower power consumption.

Fig. 2 Shows the insect-scale robot's different subsystems with an analogy with an actual bee. The power management system suffers from limited capacity, which is a consequence of the robot's minuscule dimensions. Moreover, it must provide higher voltage levels for the operation of the locomotion system through a sizeable DC-DC converter.

The locomotion subsystem in insect-scale robots requires actuators. Conventionally, piezoelectric actuators are deployed. Nevertheless, their high

power consumption and voltage requirements burden the power management system. Therefore, our first research question is defined as follows:

RQ1: What are the alternative small-size actuators for insect scale robots that operate at lower voltages and consume less power than piezoelectric actuators? The question is addressed in chapter 3.

Finally, the control and intelligence system can be realized by electronic chips. The limitations on power and form-factor are yet valid and put our the next research questions in the area of efficient computing:

RQ2: What is the transferable knowledge from the fields of edge AI and other small robots that can be used in insect scale robots? This question is addressed in chapter 4.

RQ3: What is the best computing paradigm for AI accelerator in insect scale robots in terms of area and power consumption? The question is discussed in chapter 5.

RQ4: Can the AI algorithm be redesigned for optimum efficiency in insect-scale robot applications? Chapter 6 answers this question.

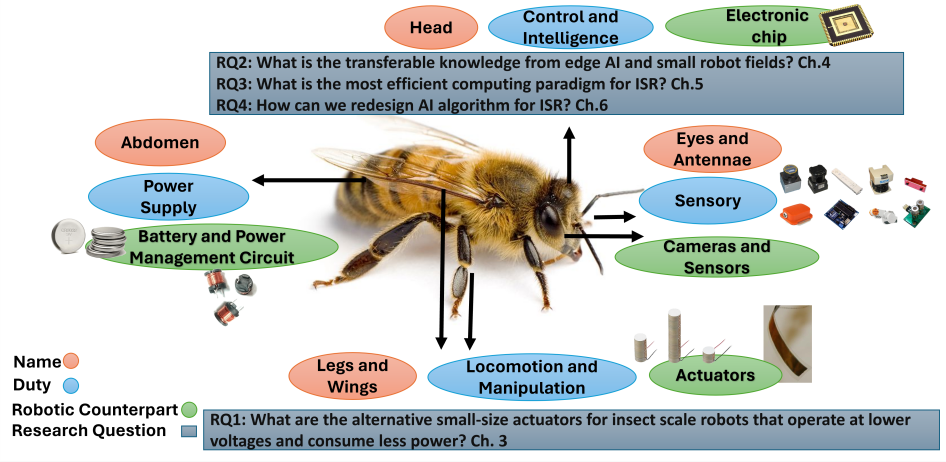


Figure 2. Different subsystem of an insect-scale robot and their analogy with an actual bee. The research questions in each section are provided.

1.3. Contribution of the Thesis

This dissertation focuses on (1) modeling IEAP actuators with a focus on their payload capability and control purposes as a low-power alternative for conventional insect-scale robotic actuators, (2) harnessing alternative computing schemes like analog in-memory computing to deploy AI on very small sizes, (3) revisiting AI hardware architecture for reducing the processor size, and (4) revisiting the AI algorithm with the new hardware architecture to adopt a hardware friendly version of the algorithm.

- The dissertation develops a brand-new comprehensive model for Ionic Electroactive Polymer (IEAP) actuators, focusing on their payload capability and control applications. The model characterizes actuators with step responses, providing insights into payload limitations and circuitry requirements. This research led to the publication of **publication I**.
- Analog computing is utilized to increase AI processor efficiency. Also, a novel analog AI accelerator is characterized to see the impact of analog computation on accuracy. This research has been disseminated by **publications II and III**.
- The hardware accelerator and AI algorithm are revisited to increase the efficiency and reduce the form factor by innovations in both levels of abstraction. **Publications II and IV** include this contribution.

1.3.1. Thesis structure

In the next chapter, the necessary background is delivered. It overviews insect-scale robots and IEAP actuators and provides the information required for comprehending this dissertation. Afterward, in chapter 3, we look at IEAP actuators. It develops a linear lumped-element model in the Laplace domain incorporating the payload effects. The model can be applied for IEAP actuators control in insect-scale robotics. Chapter 4 investigates a state-of-the-art AI accelerator chip, DIANA. DIANA is used in drone-size robots with payload limitations. We use DIANA as a case study for the newest techniques for onboard control with power and payload restriction in robotics. Taking advantage of the study in chapter 4, a brand-new hardware AI accelerator is designed especially for insect-scale robots in chapter 5. The accelerator is compact and realized in the analog domain for efficiency and size. Chapter 6 presents a novel AI algorithm and a hardware architecture that complement each other for more efficiency and area compactness. Eventually, chapter 7 summarizes the key findings, discusses their implications, and suggests areas for future research.

2. BACKGROUND AND PRELIMINARIES

This thesis is multidisciplinary. Therefore, it includes a vast collection of concepts and terminology from robotics, soft actuators, and AI hardware accelerators. This section provides the preliminary information necessary to understand the thesis's remainder.

2.1. Insect-Scale Robots

Insect-scale robots are a fascinating area of research that has been gaining traction in recent years. These robots are designed to mimic the behavior and capabilities of insects, which are known for their agility, resilience, and efficiency. This section will discuss different aspects of insect-scale robots, including their scales, applications, and varieties.

2.1.1. Scale and Applications

Conventional robots are bigger than tens of centimeters. Examples of conventional robot categories, in terms of form factor and locomotion, include humanoid, quadrupedal, and wheeled robots. They are much larger in size and have different design requirements than insect-scale robots. For example, humanoid robots are designed to mimic human-like movements and have a wide range of applications such as entertainment, education, and healthcare [10]. Quadrupedal robots are designed to move on four legs and have applications such as search and rescue operations, military operations, and exploration [11]. Wheeled robots are designed to move on wheels and have applications such as transportation, logistics, and surveillance [12].

In contrast, insect-scale robots are a unique class of robots designed to mimic insects' behavior and capabilities. These robots are typically less than a few cm in size [13]. Fig. 3 visualizes the insect scale robots in contrast to some other categories. Insect-scale robots have numerous applications due to their small size. They can enter confined spaces where humans or larger robots cannot [14]. These applications include gas leak detection in pipelines, search and rescue in disaster response, and crop monitoring for smart agriculture [14]. Fig. 4 provides examples of insect-scale robots and their applications.

2.1.2. Subsystem Architecture of Insect-Scale Robots

As with any other complex device, insect-scale robots are composed of different subsystems. As we discussed, cross-subsystem optimization can enhance overall performance and efficiency. This section introduces different subsystems and their duties and the interrelation between them.

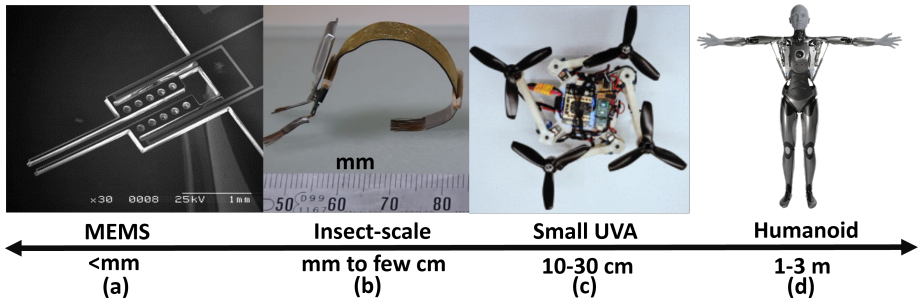


Figure 3. Examples of robots with different form factors. The spectrum continues to more than 100m for passenger jets. **a: Thermally actuated microgripper** can displace up to 262 μm. This MEMS robot is applied in cell manipulation. The general applications of robots on this scale are micro-object manipulation and material characterization [15]. **b: IEAP insect-scale robot:** A centimeter-scale robot is designed to mimic the locomotion of an inchworm. This robot is constructed using an IEAP laminate actuator, capable of generating high electrically induced strain and high bending modulus. The IEAP laminate comprises activated carbon-based electrodes, an ionic liquid as an electrolyte, and gold foil as current collectors. The laminate is pre-shaped into an arched form to facilitate unidirectional bending. This power-autonomous, microprocessor-controlled robot can crawl on a smooth surface in the open air, demonstrating the potential of IEAP for use in autonomous miniature soft robotics [6]. **c: Quadropter:** A quadrotor with an innovative adaptive morphology. This design, featuring a frame with four independently rotating arms that fold around the main frame, allows for dynamic adaptability during flight. The drone employs an optimal control strategy that adjusts to the drone's morphology in real-time, ensuring stable flight under all conditions. The adaptive morphology is versatile in various tasks such as negotiating narrow gaps, inspecting vertical surfaces closely, and object grasping and transportation. Potential applications of this technology could include search and rescue operations, surveillance, infrastructure inspection, and delivery services, where maneuverability and adaptability are crucial. [16] **d: Ameca** is a humanoid robot that is a cutting-edge platform for AI development and human-robot interaction. It boasts lifelike motion, advanced facial expression capabilities, and a cloud-connected focus. The design is reliable, modular, upgradable, and developer-friendly. Its modules can operate independently or together in the complete shape and can be accessed globally. Ameca finds applications in AI and machine learning development, human-digital interaction studies, and as a high-tech demo for events or visitor attractions. It has been utilized by robotics labs, science centers, and artists worldwide [17].

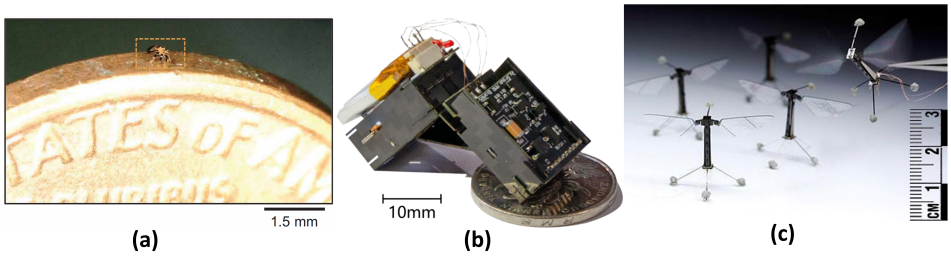


Figure 4. Examples of insect scale robots. **a: Submillimeter-scale terrestrial robot** showcases the potential of miniaturization in robotics. It is constructed from both organic and inorganic materials, supporting mechanical and optical functionalities. The robot's manufacturing procedure exploits controlled mechanical buckling to create complex, three-dimensional structures. It can perform various modes of locomotion and manipulation, from bending and twisting to linear/curvilinear crawling, walking, turning, and jumping. This robot represents an advancement in the field of micro-robotics, with potential applications in micro/nanomanufacturing, minimally invasive surgery, and sensing [18]. **b: S2worm** is an untethered insect-scale inchworm robot. It weighs 4.34 g and spans 1 cm in length. S2worm is equipped with a custom-designed onboard control system and a high-voltage boost converter to provide the driving signal for the piezoelectric bending actuator. Thanks to the novel transmission mechanism with two degrees of freedom designed based on screw theory, S2worm shows high mobility, such as high crawling speed (27.4 cm/s, 6.7 body length/s) and small turning radius (1.7 cm, 0.4 body length/s). The S2worm holds the following advantages: small size, untethered, high mobility, and low energy consumption. It is promising for application in planetary exploration, earthquake search, and constructing insect-scale multi-robot systems [2]. **c: RoboBee**, a flying microrobot developed by the Wyss Institute at Harvard University, is Inspired by the biology of a bee. The RoboBee measures about half the size of a paper clip, weighs less than one-tenth of a gram, and flies using "artificial muscles" made of piezoelectric actuators. The RoboBee has potential uses in crop pollination, search and rescue missions, and surveillance, as well as high-resolution weather, climate, and environmental monitoring. The development of the RoboBee is broadly divided into three main components: the Body, Brain, and Colony [4].

Locomotion and Mechanical Subsystem. This subsystem has the following duties:

- Integrate the different parts of the robot
- Provide locomotion for the robot [19]
- Adjust the position and angle of other parts if necessary (such as the angle of a camera) [20]
- Manipulate the environment [21]

The mechanical subsystem should include a chassis and actuator for locomotion and may have extra actuators and grippers for internal movements or manipulation. The locomotion actuator must exert enough force to move the weight of the robot while consuming low power that can be delivered with a small power source. The actuator's operating voltage can also limit the power subsystem if it is more than the available source level [4]. The subsystem's weight has to be small so that the overall weight of the robot remains low enough for the actuators. The locomotion system should also be controllable with the control and electronic subsystem.

Sensory Subsystem. This subsystem may be equipped with a variety of sensors that gather data about the robot's external environment. These sensors may include visual [22], auditory[23], and tactile sensors, among others. The sensory data is relayed to the control subsystem, where it is used to inform decision-making processes. The weight and power of the sensory subsystem can burden the locomotion and mechanical subsystem and power management subsystem, respectively, as their budgets are limited.

Control and Intelligence Subsystem. The control and intelligence subsystem processes the information from the sensory subsystem. It can include electronic circuitry for controllers and processors and uses AI algorithms for processing the sensory data [22]. It then makes decisions upon the data and controls the other subsystems, especially the mechanical subsystem, to put the robot on the correct trajectory, for example. The power and weight of this subsystem are also a burden on the power and mechanical subsystems that should be kept low by high integration and optimized designs.

Power Management Subsystem. The power subsystem provides the device with the energy it needs for its tasks. Aside from a power source, it may have some circuitry to pre-condition the voltage before delivering it to parts requiring different voltage levels or frequencies e.g. DC-DC converters. The main constraint on the power subsystem is its weight [24]. They are different power management systems according to their power source, which are discussed in the next subsection.

All these subsystems collaborate with each other and need to be optimized with a co-designed approach. Fig. 5 depicts the subsystems, their interrelations, and some possible components.

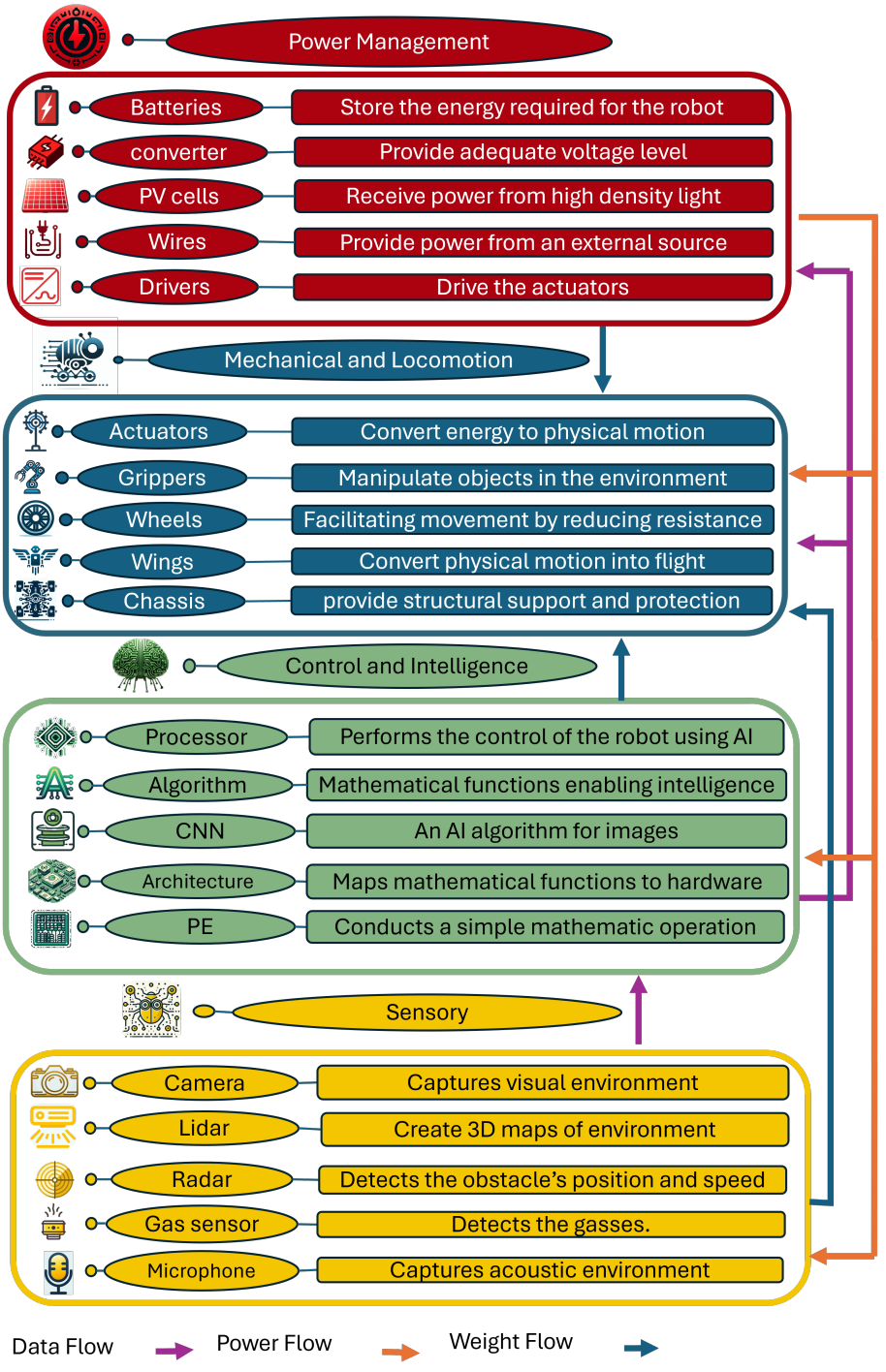


Figure 5. Four subsystems of an insect-scale robot, their interrelations, possible components, and duties. The data is acquired in the sensory subsystem and processed in the control and intelligence core. Then, data are sent to power and locomotion subsystems as commands. The power center provides power for all other three subsystems. the mechanical and locomotion subsystem bears the weight of the whole robot.

2.1.3. Power Management System

The design of the power management system in insect-scale robots is a demanding task due to its small form factor. In fact, this system is a limiting factor when it comes to the robot's autonomy. This subsection introduces some common power management systems in the field.

Tethered robots. These robots depend on an external source for their power. A famous example of these robots is [4]. While its deployment is straightforward, it puts restrictions on the robot's autonomy, flexibility, and maneuverability.

Wirelessly-Powered Robots. The energy is delivered wirelessly, e.g. a laser beam to the robot and then is converted to electricity, e.g. via a photovoltaic cell. Robofly [24] is an example of this power management system. The main drawbacks here are the complicated implementation and high power consumption of the off-board power system and the restricted operation region due to the need for a direct line of sight between the laser and photovoltaic cell.

Combustion. The idea behind using combustion in insect-scale robots is the higher energy density in fuels (22.7 MJ/Kg for methanol) than in batteries (1 MJ/Kg). A robot [25] shows longer and more powerful actuation in an insect-scale robot running on methanol compared to its battery-powered counterparts. However, exhaust and safety concerns, as well as implementation difficulties, hinder the robot's application in some fields, such as non-invasive surgeries. Moreover, the sensory and control circuitry and ignition system require a secondary electrical power management system that overloads the robots.

Battery-Powered Robots. Batteries can power mechanical and electrical parts and do not disturb the robot's flexibility or its field of operation. However, batteries have low power density. Their high weight precludes flight in aerial robots. Moreover, their low voltage necessitates applications of DC-DC converters for some actuators, such as piezoelectric transducers. The highest power density in batteries is reported to be 0.711 mW.h/g [26], which is not enough for aerial insect-scale robots with limited payload and power-hungry piezoelectric actuators [19]. It is worth mentioning that the mass of batteries coming in small packages is dominated by sealing material as the surface-to-volume increases [19]. Thus, even reaching 0.711 mW.h/g is impossible for small batteries. Nevertheless, a terrestrial insect-scale robot is reported to achieve untethered movement for about 200 m with a 35 mA.h 3 V battery [2].

Therefore, when batteries are a promising solution for power management systems providing autonomy and flexibility, the robot's power consumption should be minimized in order to keep the battery size feasible and operation time reasonable.

2.1.4. Actuation Mechanism

Insect-scale robots employ a variety of actuation mechanisms, each with unique working principles. These mechanisms can be broadly categorized into three types: piezoelectric actuators, Dielectric Elastomer Actuators (DEA), and IEAP actuators.

Piezoelectric Actuators. Piezoelectric actuators are a popular choice for insect-scale robots due to their high-frequency operation and respectable power densities [27]. However, they have some limitations. Individual actuations are low-force and low amplitude and must be operated near resonance to achieve suitable performance [27].They also require high voltages and consume relatively much power [5]. Despite these challenges, piezoelectric actuators have been successfully implemented in various insect-scale robots.

For instance, the Harvard Ambulatory MicroRobot (HAMR) uses piezoelectric actuators for its legged locomotion [28]. Another example is the flapping-wing robot, which uses a high-voltage power electronics unit to supply an oscillating signal to piezoelectric actuators. This system can modulate wing thrust, generating the forces and torques required for controlled flight [29].

In another study, researchers manipulated the moving trajectories of insect-scale soft robots by applying different driving electrical voltage frequencies to piezoelectric actuators [30]. This demonstrates the versatility and potential of piezoelectric actuators in the field of insect-scale robotics. However, more research is needed to overcome the limitations and fully exploit the potential of these actuators.

One of the significant challenges with piezoelectric and DEA actuators in insect-scale robots is the requirement for high operating voltages. This need for high voltage can pose a problem, especially considering the size constraints and power supply limitations in insect-scale robots [29]. It can also complicate the design of the robot's electronics due to the required DC-DC converter and increase the overall weight and complexity of the robot. Therefore, while piezoelectric actuators offer many advantages, addressing the high voltage and power requirement is a crucial aspect of ongoing research in this field.

2.2. IEAP actuators

IEAP actuators are a class of smart materials that can perform sensing or actuating functions by controlling the movement of cations and anions in the active layer [31]. They can deform under low voltage stimulation and generate electrical signals when undergoing mechanical deformation due to ion redistribution [31]. Their working principle is based on volume change in the electrodes due to the insertion and extraction of counter ions into the

polymer matrix [32]. Fig. 6 provides a graphical introduction to the IEAP actuators.

2.2.1. Different Types of IEAP Actuators

Various types of IEAP actuators have been developed, including Conducting or Conjugated Polymer (CP), Ionic Polymer–Metal Composites (IPMC), Carbon NanoTube (CNT-) or bucky gel polymers, and Interpenetrating Polymer Network (IPN) [33]. Each type has its unique characteristics and applications.

PEDOT:PSS Actuators. Among these, Poly (3,4-ethylenedioxythiophene) polystyrene sulfonate (PEDOT:PSS) based IEAP actuators have gained significant attention. They exhibit excellent stability under ambient conditions and are expected to improve the actuation performance of IEAP actuators [39]. Applying a PEDOT:PSS film to an IEAP actuator results in synergistic coupling between the thin nanostructured electrodes and the polymer, improving stability and preventing detachment during long-term actuation [39].

2.2.2. IEAP actuators in insect-Scale robotics

The scalability, malleability, low voltage operation, and sense-actuation integration make IEAP actuators particularly useful in insect-scale robotics. Due to their low operating voltage and power, they are biocompatible and also do not require a voltage converter.

Comparing IEAP inch-worm [6] and S2worm [2] is particularly useful. Both robots aimed to mimic the locomotion of an inchworm: crawling. S2worm used a piezoelectric actuator and achieved higher speed. However, IEAP inchworm used the malleability of its material to reduce the number of required mechanical components. In addition, its lower operation voltage freed the robot from the DC-DC converter requirement. These two reduced IEAP inchworm bill of materials and complexity compared to S2worm.

However, there is no universal model of IEAP actuators that can easily be tailored according to the sample to obtain information about their capabilities, such as payload and frequency. Such a model is also needed in the control loop. In addition, the low-frequency response of IEAP actuators may exclude them from high-speed applications like aerial robots. A precise and easy-to-use model is needed to assess the abilities of IEAP actuators in insect-scale robotics.

2.3. Machine Learning and Convolutional Neural Networks

AI is a generic term for all computer programs designed to emulate human intelligence. Machine Learning (ML) is a subfield of AI that involves the development of algorithms and statistical models that enable computers

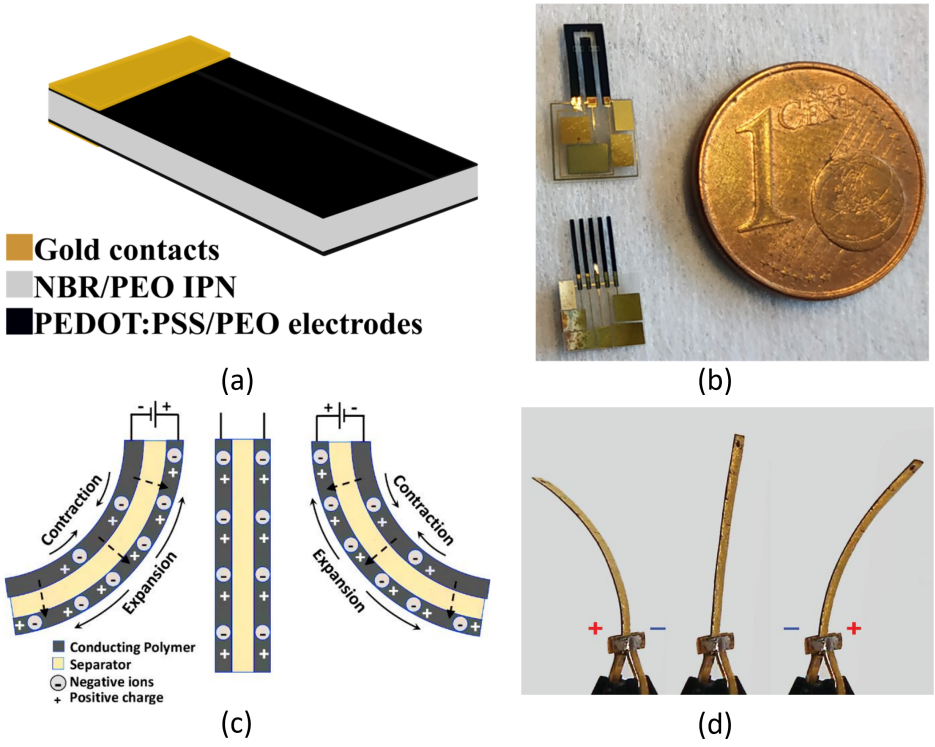


Figure 6. Graphical introduction to IEAP actuators. **a: Structure of IEAP actuator:** The sandwich structure of IEAP consists of a separator (gray) between two electrodes (black). Layers of gold are utilized to improve the contacts. The specific actuator shown here has a clean-room compatible and reproducible fabrication process facilitating the usage and miniaturizing of the actuator [34, 35]. **b: Scale of the actuator with integrated contacts:** A technique for contact integration, accompanied by the clean room fabrication process, shrinks the actuators and reduces their manual handling. The automation not only helps to reduce the length of the actuators but also makes them thinner and increases their speed. These actuators can be used as microgrippers with different numbers and shapes of fingers. It is possible to integrate strain sensing and actuation with adjacent fingers [36]. **c: IEAP actuator working principle**, IEAP actuators comprise conducting polymers and a separator. They operate based on the principle of ion migration under the influence of an electric field, which causes a change in the shape of the actuator. The ions in the separator are attracted to the electrode with the opposite charge, causing an expansion on that electrode. The lack of ions in the other electrode leads to a contraction. Thus, the actuator bends to the electrode with the same charge as ions. The direction and magnitude of the curvature are controlled by the sign and magnitude of the applied voltage, respectively [37]. **d: IEAP actuators in different states:** an IEAP actuator is placed on a fixture. Different voltages, from -3 V to 3 V, are applied to the actuator. The curvature is changing according to the applied voltage. The yellow color of the actuator is due to the gold coating. The coating reduces the longitude resistance of the actuator and provides the whole actuator with consistent electric field and curvature [38].

to improve their performance in tasks through experience. Various types of machine learning algorithms, such as supervised, unsupervised, semi-supervised, and reinforcement learning, exist in the area. Besides, deep learning, part of a broader family of machine learning methods, can intelligently analyze data on a large scale [40].

2.3.1. Convolutional Neural Networks

Convolutional Neural Networks (CNN) are a type of Deep Learning neural network architecture commonly used in Computer Vision [41, 42]. CNNs are primarily used to solve difficult image-driven pattern recognition tasks [41, 42]. CNNs are made of layers of artificial neurons called nodes. Some of these nodes are functions that calculate the weighted sum of the inputs and return an activation map [41]. This is the convolution part of the neural network [41].

Convolutional layers apply filters to the input image or the activation map of the previous layer to extract features. Pooling layers, divided into Max and Average Pooling, downsample the activation map by selecting the maximum or average value of a section to reduce computation. Fully connected layers make the final prediction [41]. The network learns the optimal filters' weights through backpropagation and gradient descent [41]. Fig. 7 and Fig. 8 provide more information on CNNs.

2.3.2. Memory Bottleneck in Edge Device Inference

Edge devices, such as smartphones and IoT devices, are increasingly being used to perform inference on pre-trained CNNs. However, these devices often face a significant challenge known as the "memory bottleneck" problem [43, 44].

The memory bottleneck problem arises due to the limited memory resources and bandwidth available on edge devices. When performing inference on large pre-trained models, memory limitations become the throughput bottleneck for the inference pipeline, limiting the execution speed. This is particularly problematic for real-time applications where low latency is required [43].

One approach to mitigate this problem is distributing the inference across multiple edge devices[43]. This involves partitioning the pre-trained model into smaller parts that can be spread across the devices. Each device then performs inference on its part of the model and sends its computed result to a subsequent device. This approach can increase throughput and decrease per-device compute load [43].

However, even with this distributed approach, the memory bottleneck remains a significant challenge. Also, it is not feasible for applications with a single isolated device. The Google Edge TPU, for example, has been

found to operate significantly below its peak computational throughput and theoretical energy efficiency due to its memory system being a large energy and performance bottleneck [44]. The one-size-fits-all design of such accelerators ignores the high degree of heterogeneity both across different neural network models and across different layers within the same model, leading to these shortcomings [44].

To address these issues, researchers have proposed new acceleration frameworks incorporating multiple heterogeneous edge machine learning accelerators [44]. These accelerators cater to the characteristics of a particular subset of neural network models and layers. During inference, for each layer, the framework decides which accelerator to schedule the layer on, considering both the optimality of each accelerator for the layer and layer-to-layer communication costs [44].

In-Memory Computing and Dataflow Engineering. In-Memory Computing (IMC) and dataflow engineering have emerged as promising solutions to the memory bottleneck problem in edge device inference [45, 46]. IMC is a non-von Neumann paradigm that has recently established itself as an auspicious approach for energy-efficient and high throughput hardware for deep learning applications [45]. One prominent application of IMC is performing matrix-vector multiplication in O(1) time complexity by mapping the synaptic weights of a neural network layer to the devices of an IMC core [45].

In the context of CNN, activation maps must be fetched from memory multiple times during the computation process, which requires a significant amount of bandwidth [46]. Dataflow engineering can help optimize this process by improving the efficiency of the calculation method, reducing the excessive read/write times and execution steps of CNN calculation circuits [46].

Despite these advancements, further research is needed to develop more effective strategies for overcoming the memory bottleneck problem in edge device inference.

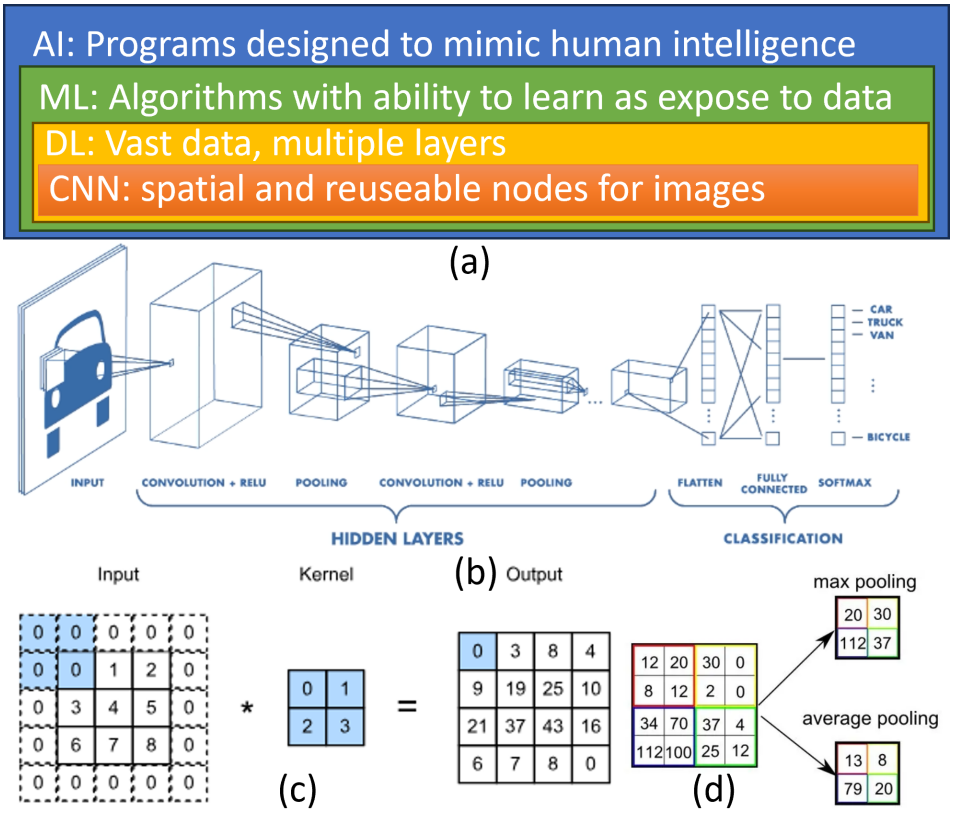
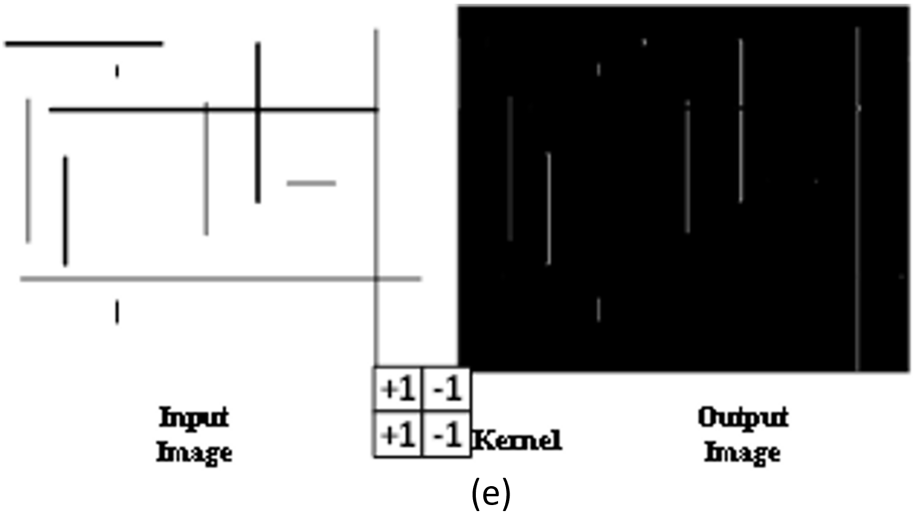


Figure 7. Graphical introduction to CNN part one. **a: CNN position in AI:** AI is an umbrella term for programs designed to mimic human intelligence. ML is a subset of AI with the ability to learn from data. Deep learning is a branch of ML dealing with big datasets and with several different layers. CNN uses spatial operations to find certain features in different locations of an image. It also reduces the number of parameters by reusing them in different areas of an image. **b: CNN structure:** Input image goes through a network of convolutions and poolings. As the picture is processed further, the activation map deepens, but its width and length shrinks. At the last layer of this network, the activation map is flattened and handed over to a network of fully-connected layers. Fully-connected layers connect each input to all outputs with a specific weight. A non-linear function (here ReLU) is applied after each layer (except pooling) to increase the order of the network. Image taken from [47]. **c: Convolution operation:** A kernel (filter) applies to an activation map from the upper left corner. It calculates a weighted sum of the activations it is applied to. Then, the kernel slides right and performs the same operation again. When the kernel reaches the image's rightmost, the kernel slides down and starts from the leftmost. The number of activations the kernel slides between two subsequent operations is the stride. The stride in this figure is one. If the stride were two, the output dimension would be two by two, and odd columns and rows would be skipped. The activations with dots are padding. They are added to keep the input and output dimensions equal. It is called the "same" padding. Convolutions without padding are called "valid". Image is taken from [48]. **d: Pooling:** Pooling are kernels that move like Convolutions. However, they downsample an activation map by selecting the maximum or average value of the activations in an area. Sride is usually equal to kernel size. Image is from [49].



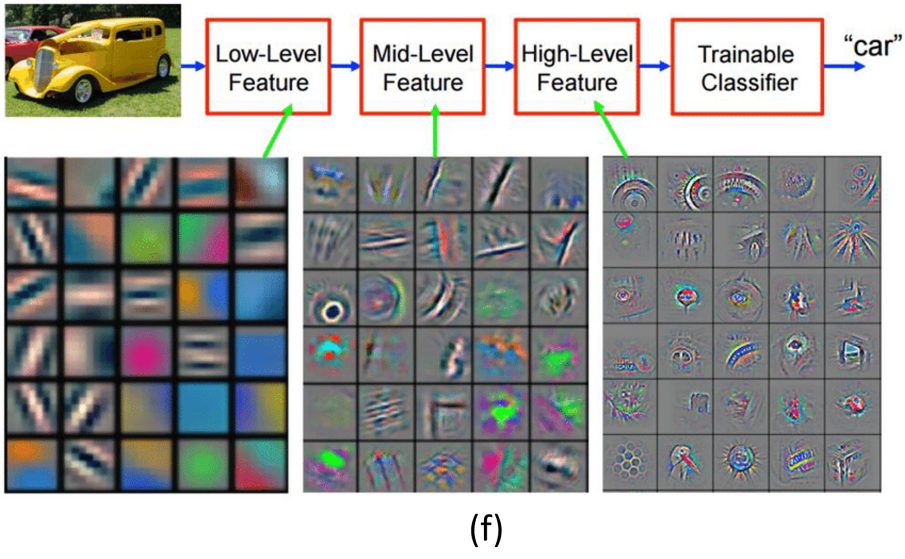


Figure 8. Graphical introduction to CNN part two. **a: CNN feature extractions:** The operation of a single layer CNN is visualized. It captures features (here, vertical lines). It ignores other features (here, horizontal lines and cross sections. The captured features become more complicated as the networks become deeper. **b: CNN visualization:** Features captured by CNN become more complex, from lines to simple shapes to complex patterns, as the network grows deeper [50].

3. IEAP ACTUATOR MODELING

This chapter introduces a comprehensive novel linear model of IEAP actuators in the frequency domain, considering the loading effect. The frequency domain is chosen for its distinct advantages. Firstly, it allows for a more effective model application, particularly because it aligns with the frequency approach used in control theory for controlling robot actuators. This alignment enhances the model's utility in practical applications. Secondly, the frequency domain simplifies the generation of actuator-specific models, making it a versatile tool for various robotic applications. Importantly, the validity of this model is not just theoretical; it is corroborated by a series of rigorous experimental results, providing a robust foundation for the discussions and analyses that follow. This chapter, therefore, serves as a critical link between theory and practice, paving the way for further exploration and understanding of IEAP for its potential applications in insect scale robotics.

The modeling approach in this chapter is universal for IEAP actuators. However, we specifically used PEDOT:PSS actuators in experimental verification. PEDOT:PSS/PEO actuators are selected in this chapter due to their auspicious position for insect-scale robotic applications. The following are some of this actuator's outstanding characteristics for miniaturized robotics:

- ability to operate in air [51]
- reversible and fast switching [52]
- biocompatibility [53]
- long life cycle [54]
- low operating voltage (<3 V) [55]
- clean-room compatible fabrication process [54]

The clean-room fabrication automates the production, reduces the handling, and shrinks the actuator's size. Thinner actuators operate faster due to lower capacitance and Young's modulus. The actuator material and its automated fabrication technique are discussed in the first paper of this thesis [35].

3.1. Model presentation

We divided our actuator gray-box lumped model into electrical, chemical, and mechanical subsystems. Figure 9 explains the model visually. First, we deal with the electrical subsystem transfer function. R_e is insignificant and can be neglected. Ignoring R_e, R_cc becomes parallel to the voltage source and nullified in the double-layer capacitance charge calculation. From the circuit theory, we can drive 3.1.

$$\frac{Q(s)}{V(s)} = \frac{\frac{1}{R}}{\frac{1}{RC}} \tag{3.1}$$

In the equation above, s is a complex number frequency parameter.

The chemical subsystem transfer function remains as α in the Laplace domain. The time-domain equation for the mechanical model is taken from [56] and shown in 3.2.

$$\kappa(t) = \frac{1}{EI}[q(t) - \lambda e^{-\lambda t} \int_0^t q(\tau) e^{\lambda \tau} d\tau] \tag{3.2}$$

κ is curvature, I is the second moment of inertia, and $\lambda = \frac{E}{\eta}$ denotes the rate of relaxation. A Laplace transform from 3.2 provides the mechanical subsystem transfer function.

$$\frac{K(s)}{M(s)} = \frac{\frac{1}{EI}s}{s + \lambda} \tag{3.3}$$

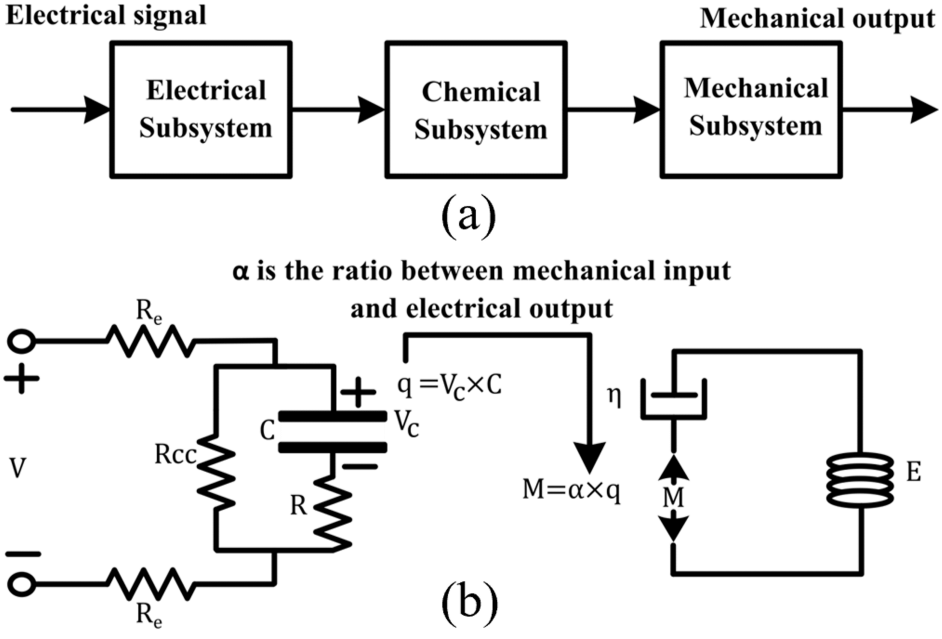


Figure 9. The actuator is modeled with three subsystems. The input is a voltage applied in the electrical subsystem. The electrical part is modeled with electrode resistors (R_e), double-layer capacitance (C), leakage resistance (R_{cc}), and series resistance (R). The electrical subsystem output is the charge stored (q) in the capacitor. The chemical subsystem, then, transforms this charge into a mechanical bending moment (M) with the ratio of α. From the mechanical perspective, the actuator is modeled with a cantilever beam consisting of a spring (E, Young's modulus, and a damper (η).

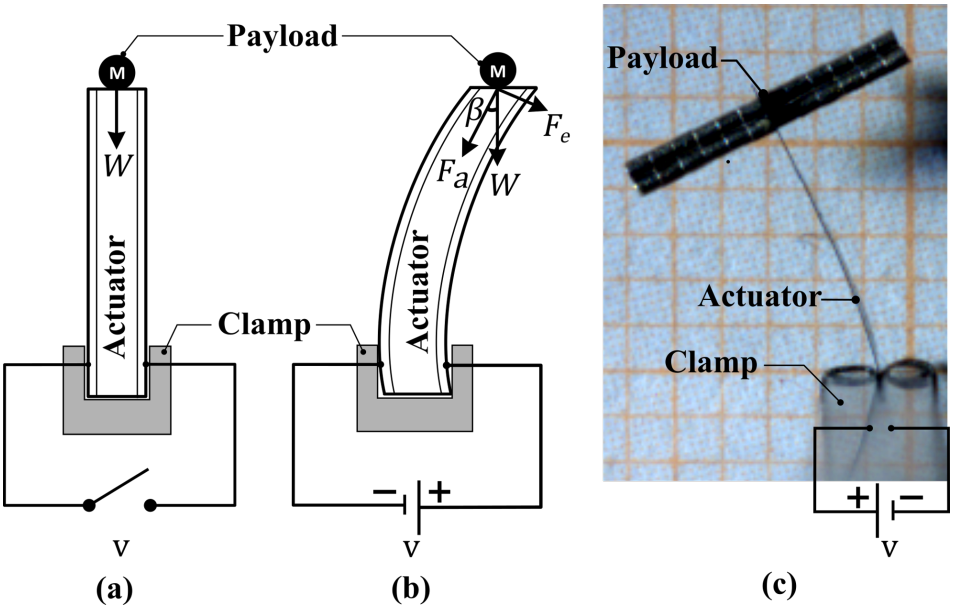


Figure 10. a) The actuator is in the neutral position. The effective component of the load weight is zero in this state. b) The actuator bent. There is an effective component of the weight (f_e). f_e value is proportional to the curvature. This force makes a bending moment that itself increases the curvature. c) The payload implementation in action in our experiment. The actuator responds to 2 V_{pp} input with 12.6 mg of magnets as load.

3.1.1. Loading effect

To incorporate the payload effect in our model, we assume that the actuator is placed perpendicular to the ground and load is added to its tip. In the neutral position (curvature = 0), the payload weight component in the direction of movement (tangent to the curve, hereafter referred to as effective force (f_e)) is zero. As the actuator bends and curvature increases, this component increases and causes a new mechanical bending moment in addition to the chemically induced one. This results in a positive feedback loop over the mechanical subsystem, where more curvature leads to more bending moment, which in turn leads to even more curvature. The setup is illustrated in Figure 10.

As the bending moment is related to f_e, it is essential to find the value of f_e. In Figure 10, you can see that $F_e = Wsin(\beta)$. If we approximate the actuator with an arc with a constant curvature (κ) equal to the average actuator curvature, β equals the arc's central angle, $\beta = \kappa l$, Where l is the actuator length. Thus,

$$F_e = mgsin(\kappa l) \tag{3.4}$$

m is the payload mass and g is the earth's gravitational constant.

The moment implied by a point force, F_e, at the tip of a cantilever

beam is $M = F_e d$, and d is the distance from the load. To adhere to our assumption that the actuator makes a perfect arc, we wish to substitute the moment from the point force, which increases towards the actuator's fixed end, with a constant moment that results in the same deflection. This moment is $M = 23F_e l$.

Figure 10.c) illustrates that the effective actuator length, from the center of the load to the fixture, is less than its geometric length. We call this property effective length and show it with l_e. Thus, the equivalent constant moment is obtained from 3.5.

$$M = \frac{2}{3} F_e l_e \tag{3.5}$$

From 3.4 and 3.5 we have:

$$M_e = \frac{2}{3} l_e mgsin(\kappa l) \tag{3.6}$$

The mechanically induced bending moment, M_e, input of the mechanical subsystem, is related to κ, the mechanical subsystem's output, through a *sin* function. Therefore, *sin* is in the feedback loop. As we develop a linear model, the *sin* is linearized, as shown in Figure 11. We linearized the sine into five regions. In each region, the sine function is approximated with a line. The y-intercept of the line equation is constant in each region and is dealt with as a mechanical input. The gradient changes with κ. The slope with which the gradient is related to κ is the feedback component.

Deploying 3.6 and linearization from Figure 11, we can develop Table 1 that provides the mechanical moment in each region and divides it into mechanical input and feedback. Putting the Table 1, and subsystem transfer functions into the model in Figure 9, the load-included model in Figure 12 is obtained.

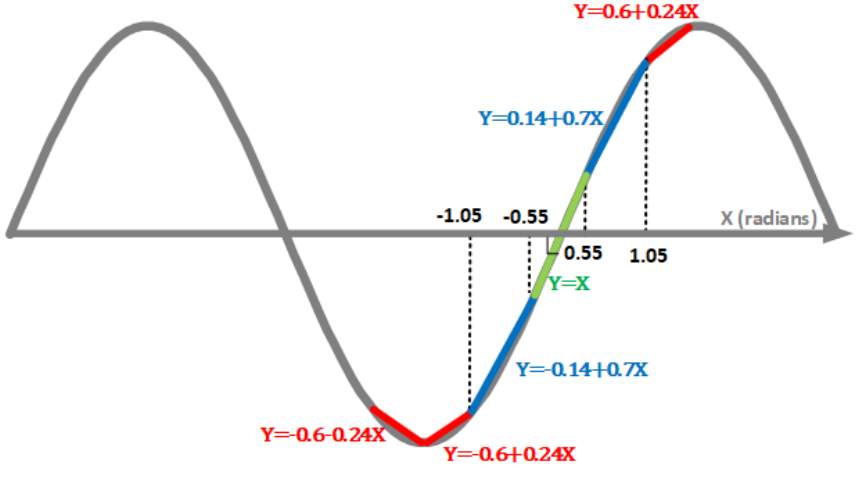


Figure 11. Sine function linearization.

Table 1. FEEDBACK (FB) AND MECHANICAL INPUT (m_l) FOR DIFFERENT LINEARIZED CURVATURE REGIONS

angle (radian, κl_e)	$M_e(N.m)$	FB $(N.m^2)$	m_l $(N.m)$
$\kappa(t)l_e < -1.05$	$mg\frac{2l_e}{3}(-0.6 + 0.24\kappa l_e)$	$0.24mg\frac{2l_e}{3}$	$-0.6mg\frac{2l_e}{3}$
$< -1.05\kappa(t)l_e < -0.55$	$mg\frac{2l_e}{3}(-0.14 + 0.7\kappa l_e)$	$0.7mg\frac{2l_e}{3}$	$-0.14mg\frac{2l_e}{3}$
$< -0.55\kappa(t)l_e < +0.55$	$mg\frac{2l_e}{3}\kappa l_e$	$mg\frac{2l_e}{3}$	0
$0.55 < \kappa(t)le < -1.05$	$mg\frac{2l_e}{3}(+0.14 + 0.7\kappa l_e)$	$0.7mg\frac{2l_e}{3}$	$+0.14mg\frac{2l_e}{3}$
$1.05 < \kappa(t)l_e$	$mg\frac{2l_e}{3}(+0.6 + 0.24\kappa l_e)$	$0.24mg\frac{2l_e}{3}$	$+0.6mg\frac{2l_e}{3}$

3.2. Actuator Identification

To identify the different values of the model parameters, we put the actuator under two step voltages with $2V_pp$ and $3V_pp$ and fit their responses to a second-order linear time-invariant (LTI) system. The experimental and fitted second-order responses are depicted in Figure 13, and the second-order response equation with its Laplace transform transfer function is given in 3.7.

$$\kappa(t) = 181 - 90e^{-1.8t} - 265e^{-0.3t}\frac{K(s)}{V(s)} = \frac{63(s + 0.52)}{(s + 0.3)(s + 1.8)} \tag{3.7}$$

The zero and the low-frequency pole in 3.7 come from the mechanical and the high-frequency pole from the electrical subsystem. However, the distinction between mechanical and electrical gains necessitates implementing the payload effect. Mechanical gain is $\frac{1}{EI}$. Nevertheless, large actuation

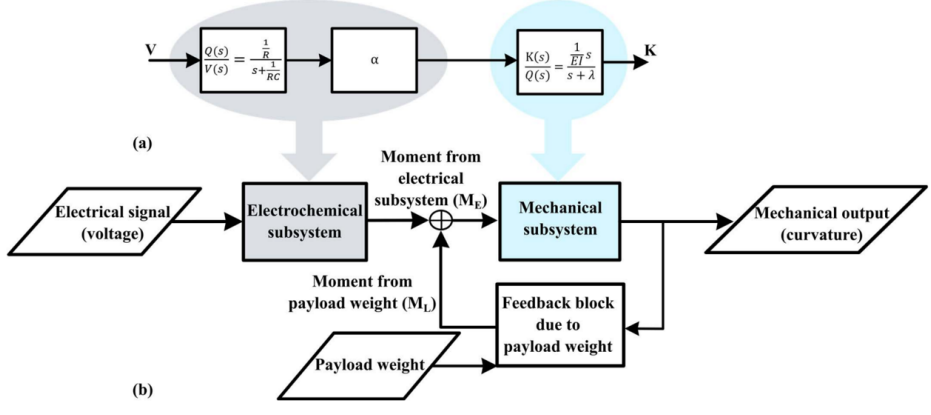


Figure 12. The payload-aware model incorporates the payload weight as a new mechanical input (moment) and a feedback component from its output, curvature.

Table 2. VALIDATION OF THE PROPOSED MODEL

	$m(mg)$	l_e(mm)	Voltage (V)	κ_{pp}	$Error_{pp}$	$\%Error$
a	3.15	6.20	2	243	25	10
b	3.15	6.20	3	248	5	2
c	6.30	6.70	2	260	37	14
d	6.30	6.70	3	263	11	5
e	12.60	6.27	2	428	60	14
f	12.60	6.27	3	504	75	14

and applied electrical voltage change Young's modulus value [57]. Hence, we used an empirical technique to acquire the mechanical gain. In this technique, we use the average or most popular voltage-payload combination and fit the mechanical gain into the equation to minimize the error. The mechanical gain for the under-test actuator obtained $4.86 \times 10^7 N^{-1}m^{-2}$.

The step response identification method facilitates the actuator-specific model development as only simple step response analysis is required. This method is advantageous in applications like swarm insect scale robots when many actuators are to be modeled.

3.3. Model Verification

We conducted experiments with six combinations of voltages and payload weights to verify the model. The experimental results and the model results for these combinations are shown in Figure 14. Table 2 summarizes the combinations and reports their error. The error stays less than 15% even for the worst combinations.

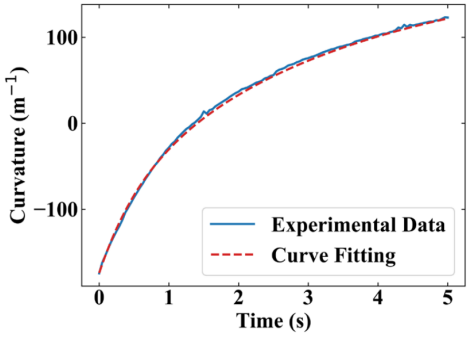


Figure 13. Actuator step response is properly fitted the second-order response

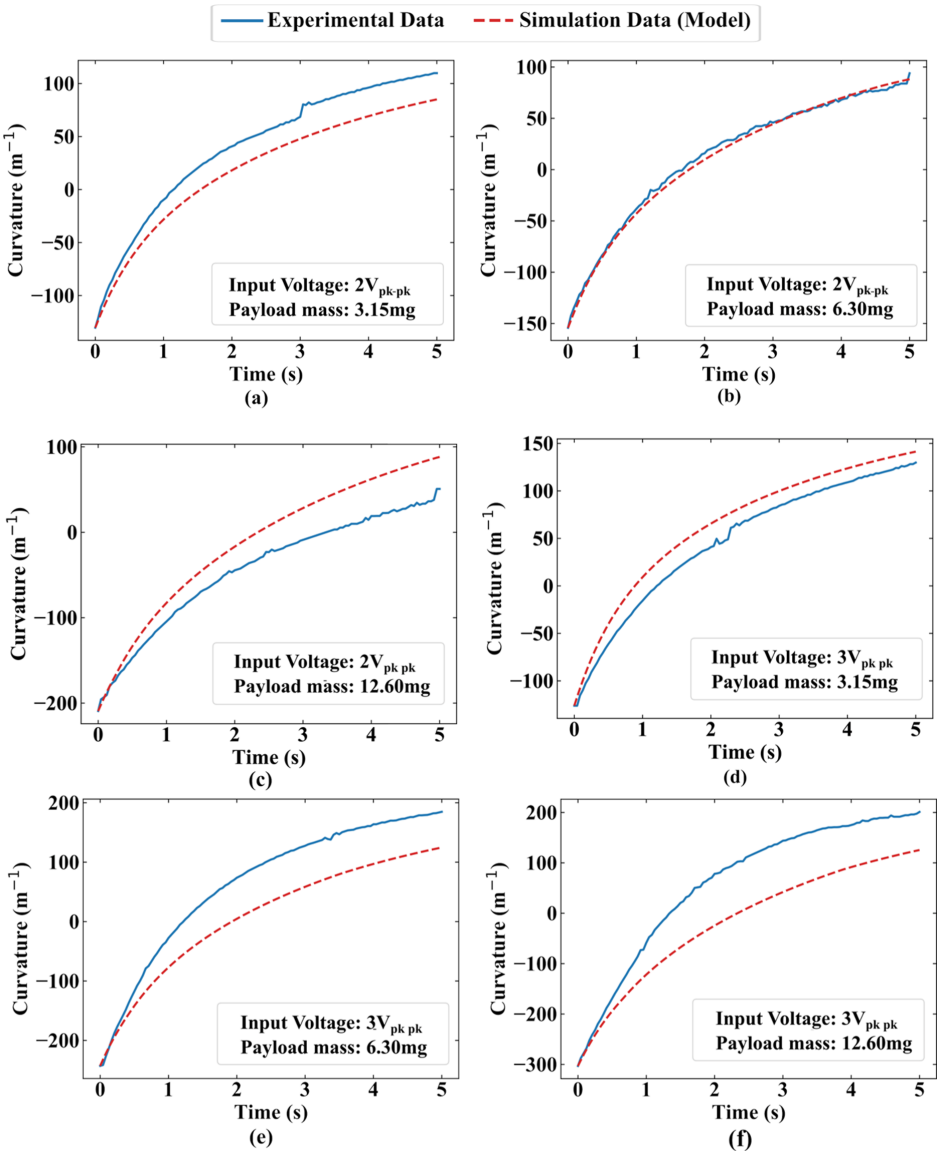


Figure 14. six various combinations of voltage-payload are examined to compare the model and experimental results. The mechanical gain is optimized for combination (b). Thus, the minimum error is observed there. The error increases as the payload or voltage differs from combination (b) points due to non-linearity in the mechanical system for large actuation or voltage changes.

3.4. Stability and maximum payload

The maximum allowed payload is translated to our model as the biggest load for which the actuator is stable in region three. We carried out a zero-pole stability analysis in this region. The results showed that the actuator

is in marginal stability for a payload of 20 mg. So, 20 mg is the maximum limit for the actuator payload capacity. Figure 15 shows the actuator zero and poles in the s-plane. The mechanical subsystem poles move toward each other as the payload increases. For a payload equal to 20 mg, the actuator is marginally stable, and after that, it is unstable. Instability is interpreted as the payload is more than the actuator force, and it falls.

3.5. Robot morphology with IEAP actuators

The payload capacity for the under-test actuator in this paper is 20 mg. This restriction seems to be strict. However, the force required to move the payload is significantly less if it is located on a wheeled loading platform. The required force depends on rolling resistance, bearing, surface, and many other parameters. Nevertheless, as a rule of thumb, 1.8 N force is required to move 1 Kg payload stably. So, if we assume that our actuator can exert about 0.2 N (enough to lift 20 mg of payload), it can move about 100 mg of payload on a wheeled platform. It is also possible to use more than one actuator to move each wheeled loading platform. Figure 16 shows an example possible structure for an insect-scale robot consisting of a loading platform (pink) driven with two IEAP actuators (black). The robot can go forward or turn right and left.

3.6. Chapter Conclusion

In this chapter, we developed a payload-aware model for IEAP actuators. This model is practical due to its easy actuator-specific model development with step response that facilitates model generation. Each unknown IEAP actuator can be modeled with this technique by only measuring its step response without any information about its electro-chemo-mechanical properties. Also, the model Laplace-domain representation is consistent with control theory and electrical analysis.

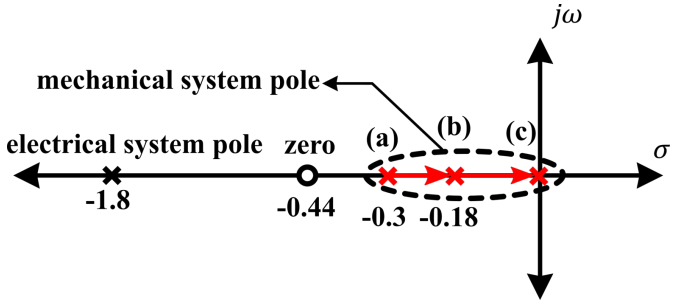


Figure 15. System stability analysis: location of the actuator poles and zero. Mechanical subsystem pole for (a) an unloaded actuator, (b) a payload mass = 12.6 mg, and (c) marginal stability, calculated with a payload mass = 20 mg.

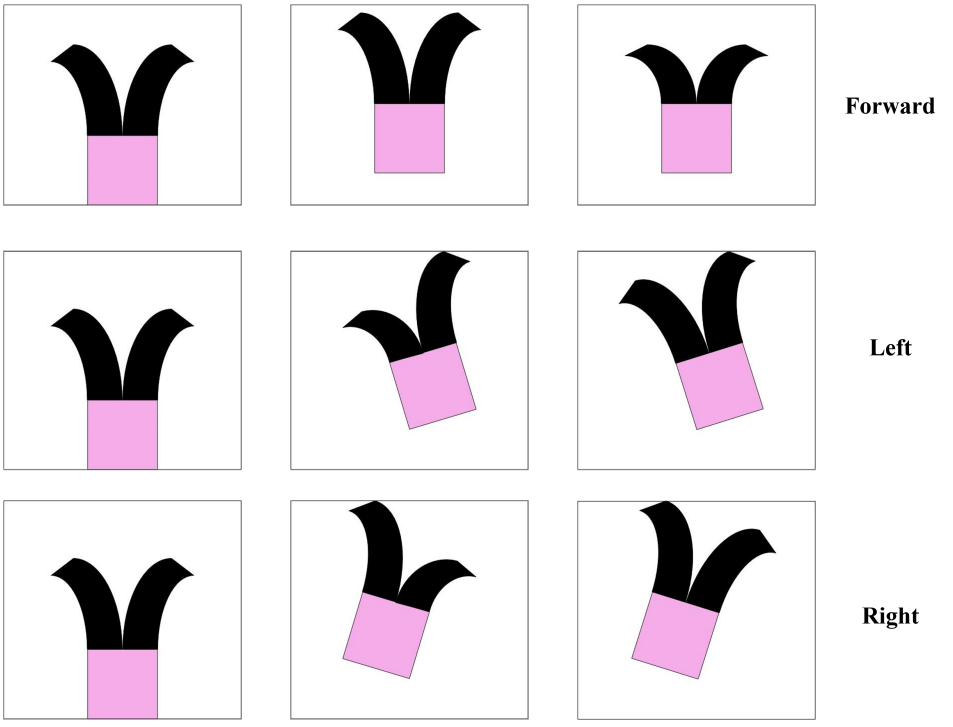


Figure 16. insect-scale robot structure example with a loading platform and two IEAP actuators

The results portray a good potential in IEAP actuators to be used as micro-manipulators or carry very small payloads (≈20mg). The results are also promising for the application of IEAP actuators in insect-scale robots as drivers for wheeled loading platforms where the weight of the payload is not directly on the actuator. Figure 16 shows one possible configuration in which two IEAP actuators are used as drivers to carry a wheeled platform.

4. ANALOG IN MEMORY COMPUTING (AIMC) WITH DIANA CHIP CASE STUDY

Today, no onboard AI-based vision platform exists for insect-scale robots. Restricted payload, power, and area have been prohibitive for the embarkation of required electronics on the miniaturized robots, restricting their autonomy, field of operation, and applications. Nevertheless, Studying AI accelerators for applications that face similar constraints, albeit to a lesser extent, offers valuable insights for insect-scale AI-based visual control circuitry. Numerous Application-Specific Integrated Circuit (ASIC)s have been developed for edge inference engines, employing innovative architecture and computation concepts to enhance efficiency and adhere to the limitations [58][59]. It is imperative to study and advance these cutting-edge designs further to meet the stringent requirements of insect-scale robots. Analog In Memory Computing (AIMC) is one such paradigm that offers a wide range of applications and significant improvements in efficiency[60].

AIMC has been utilized in CNN edge inference engines to solve the memory bottleneck problem and increase efficiency [61]. However, AIMC Analog-to-Digital Converter (ADC)s restricted resolution imposes quantization of output activations that can reduce the accuracy without meticulous optimization [60]. A study conducted output quantization calibration and obtained configurations with which low-resolution ADCs did not affect the accuracy. The configurations were layer-specific. Therefore, a real-time quantization adjustment was required [60][62]. AIMC output quantization is adjusted by controlling analog gain, entangling it with analog parameters and nonlinear functions. AIMC dynamic output quantization control without interrupting its operation has been an unsettled problem until now. This chapter introduces a technique for imposing output quantization configurations obtained from calibration processes on AIMC through circuit parameters setup. The technique permits on-the-fly quantization adjustments, enabling layer-wise calibration that increases achievable network accuracies on AIMC platforms. As a case study, we deployed the method on the AIMC macro of an artificial intelligence (AI) inference engine System on Chip (SoC) platform with a RISC-V processor and hybrid DIgital-ANAlog accelerators (DIANA). We related its controllable circuit parameters with the quantization configuration in a look-up table. This case study has noteworthy side benefits in identifying platform limitations due to nonlinearities and design imperfections. These limitations are investigated, and design advice that is transferable to future AIMC designs is provided to avoid imperfections such as mismatch, bias voltage drop, and interconnect delay. In addition, the study of output quantization from different levels of abstraction leads to design guidelines to facilitate dynamic quantization control during the application phase.

4.1. Introduction

Artificial Intelligence (AI) has been recognized as "the new electricity" for its potential to revolutionize the industry [63]. It demonstrated vast applicability in various domains, from natural language processing (NLP) [64, 65], image classification and object recognition [66, 67] to stock market trading [68, 69]. In computer vision applications, convolutional neural networks (CNNs) showed outstanding ability due to their spatial kernels [70].

CNNs require a high computational load indicating parallelization possibility. A vast effort exists to fully harness parallel computing architectures for higher efficiency [71]. GPUs [72], FPGAs [73], and Application-Specific Integrated Circuit (ASIC)s [74] can leverage a higher level of parallelism than conventional processors. Nevertheless, the development cost and time, along with the performance and efficiency, increase from GPUs to ASICs. In specific applications where energy and speed constraints are limited, ASICs are the only viable solution. Research is moving towards more efficient and accurate systems to accelerate CNN at the edge. A very promising acceleration method consists of computing MAC operations in the analog domain directly in memory cells.

The unmet need for efficiency by digital computing in edge devices caused a resurgence in analog computing. Analog computing can be exploited to increase efficiency at the cost of accuracy reduction [75]. The analog domain represents a number by a single signal without resolution restriction and performs MAC operations with one device per input [76]. This characteristic makes analog accelerators strong candidates for applications with limited energy budgets. AIMC combines analog efficiency with in-memory computing (IMC) to overcome the memory wall bottleneck by merging processor and memory units, pushing energy efficiency by orders of magnitude [77]. However, device nonlinearities, mismatches, and noise impact the analog computation's accuracy. On top of that, analog circuits lack flexibility as their behavior, as well as the data flow, are fixed at design time. Thus, an analog macro engineered for a particular workload or required precision may not be efficient when the requirements change. CNNs show high resilience to errors and reduced parameter precision but with limitations and different output sensitivity to different layers in the network [78]. These observations promise high accuracy and efficiency with a hybrid digital-AIMC accelerator that can split the workload in agreement with accuracy and efficiency requirements.

Along this line, the DIANA SoC [61] integrates three cores in a complete system: a RISC-V CPU, a digital accelerator, and an AIMC-based accelerator[79]. The RISC-V processor controls the system and allocates the workload among the two accelerators. The AIMC accelerator is designed to achieve high utilization and efficiency with moderate accuracy for layers

with a high number of channels. Layers with fewer parallelization possibilities and more severe sensitivity to accuracy are assigned to the digital accelerator. This structure allows DIANA to achieve high efficiency without a decrease in network accuracy by allocating the execution of different layers in the digital or analog core. DIANA's AIMC macro is used as a case study in this chapter while analyzing the applicability of the technique presented here for other AIMC platforms.

The AIMC paradigm can be implemented with various cell technologies. Non-Volatile memories (NVM) form dense crossbar arrays to perform parallel MAC operations [80]. NVMs are enabled with emerging technologies such as Resistive Random Access Memory (RRAM), [81] phase-change memory (PCM) [82], and spintronics [83]. However, NVM's technological drawbacks, like read and write non-idealities [84], low reliability, and temperature dependency [80], make the design of AIMC macro challenging and motivate designers towards more standard technologies such as CMOS-based SRAMs [85, 86, 87, 88]. This last type of cell is the one used in DIANA.

ADCs are essential parts of AIMCs. They convert voltages proportional to the MAC operation results to digital data. Consequently, they quantize the output activations to fewer levels than the MAC operation result requires. At this stage, careful calibration is required to reach an accuracy comparable with the baseline [60]. The quantization configurations are set by selecting circuit analog parameters. A methodology is missing to link the quantization configurations obtained from thorough optimizations [60] to physical circuits. This method should determine the mechanism by which the quantization parameters, obtained at the software level, are imposed on AIMC. It can be a modeling technique that connects the circuit-level parameters to quantization configurations.

There are some efforts on AIMC modeling. Spetalnick, S. et al. [89] combine system and circuit models and simulations to analyze the SRAM AIMC design space and spot efficiency gains and losses. Kein, J. et al. [90] integrate an AIMC cell model to gem5-x simulator for full-system simulation in the design phase. However, to the authors' knowledge, there is no model that selects the circuit parameters according to output quantization calibration.

The lack of a quantization imposition technique has hindered the optimal use, in terms of computation accuracy, of the AIMC. Moreover, non-idealities of the analog circuit should be known for a correct accuracy evaluation and compensation strategy. The characterized non-idealities can also be mitigated in future AIMC designs.

In this chapter, we contribute to the AIMC paradigm with the following developments:

- A technique is developed to link the AIMC circuit analog parameters

to its output quantization. It allows on-the-fly implementations of layer-wise quantization calibrations like [60] on AIMCs, significantly increasing the achievable classification accuracy on the platform. The study of the output quantization mechanism also gives guidelines for AIMC designs to better exploit the ADC output range.

- The method is applied to DIANA's AIMC macro as a case study. As a result, a linear model of the DIANA's AIMC is developed. The model can translate the quantization parameters obtained from calibration or training to controllable circuit parameters. The Controllable parameters are an external bias voltage and a programmable Pulse Width Modulation (PWM) unit time. Thus, the quantization configuration is set by adjusting these parameters in a real-time manner. A look-up table summarizes the model and eases its application.
- Important non-idealities for accuracy are characterized and modeled. Methods are proposed to avoid, compensate, and in future designs, improve non-idealities.

Section II discusses the technique required to impose the quantization parameters on AIMC output. Section III briefly introduces the DIANA's AIMC macro. This SoC is used in the rest of the chapter as an example to apply the suggested method. Section IV presents the experimental results. Section V implements the method on DIANA as a model, and section VI concludes the chapter.

4.2. AIMC output quantization control through circuit parameters

Quantization is used to reduce the computational cost of CNN and match the edge applications restrictions [91]. The quantization is applied to input activations and weights to reduce their bit precision to b_a and b_w, respectively. The output of a convolution should ideally be coded with l_o levels:

$$l_o = (2^{b_a} - 1) \times (2^{b_w} - 1) \times n_a + 1 \tag{4.1}$$

where n_a is the number of accumulations in the MAC operation. Some works reported re-quantization at the activation layer to directly produce quantized input activations for the next layer [92][93]. This output quantization is necessary for AIMCs due to their restricted ADC resolutions. As an example, a layer with 16 7-bit input channels and 3 by 3 kernels with 2-bit weights would need 11-bit ADCs according to (4.1) that are prohibitively power-hungry [94].

The output quantization should be calibrated according to the CNN model and activation distribution in each layer or even channel in order to achieve high network accuracy[91]. Calibration is the process of determining

the clipping range $[\alpha, \beta]$, a range that data out of it will be mapped to its limits before quantization.

There are different calibration approaches. A straightforward way is to set the clipping range to the maximum and minimum values of the to-be-quantized data. This method increases the dynamic range and reduces the resolution. So, other approaches, like using percentile [95] and optimizing the data loss [96], take a smaller range to increase resolution mitigating the effect of outliers. Another method is to learn the quantization parameters during the training [93][60].

Laubeuf. N et al. [60] conducted research on output quantizations with AIMCs. They showed improvement in accuracy with a layer-wise output quantization calibration over the network-wide counterpart. Their work used a DIANA-like AIMC macro and did a Pytorch simulation to show dynamic output quantization control with adjusting Pulse Width Modulation (PWM) unit time. Accuracies on par with the baseline are achieved for Resnet-20 on CIFAR-10 and Resnet-18 on ImageNet after output quantization optimization. The chapter showed the importance of AIMC output quantization calibration and its enforcement possibility via circuit parameters selection.

However, results from [60] are not directly applicable to AIMCs, because first, their assumptions in the simulation are different from the actual chip structure. Second, they only analyzed the unit time adjustment for a fixed bias voltage value. The bias voltage can be used for fine-tuning the quantization parameters in DIANA, as its values are continuous, unlike discrete unit time values. Also, using the combination of bias voltage and unit time increases the designers' degree of freedom, so they can optimize the chip also for power and performance vs. accuracy [61], for example. And third, the nonlinear behavior and second-order effects of AIMC are neglected in their linear Pytorch model. There is a gap between their high-level study and the low-level AIMC circuits. To fill the gap, it is required to investigate the quantization from both network and circuit-level perspectives to unveil the output quantization mechanism in AIMCs.

The ADC thresholds in AIMCs are usually uniform and symmetric. Therefore, the output quantization is also uniform and symmetric from the high-level network perspective. Under this assumption, the scale factor is defined as a floating point number with which the data multiplies before discretization. As it should convert a data from $[-\beta, \beta]$ to $2^b - 1$ levels, scale factor can be calculated with:

$$S = \frac{2^b - 1}{2\beta} \tag{4.2}$$

in which b is the quantization (ADC) bitwidth. The quantization output

(O) with this scale factor is then obtained as:

$$O = int(S \times O_{mac}) \tag{4.3}$$

Where O_{mac} is the MAC operation result.

From the circuit perspective, there is a gain (A_v) that determines the voltages at the ADC input (V_{adc}) proportional to the MAC operation result.

$$V_{adc} = A_v \times O_{mac} \tag{4.4}$$

This voltage is then converted to digital at the ADCs according to the ADC quantization steps (δ_{adc}).

$$O = int(\frac{V_{adc}}{\delta_{adc}}) \tag{4.5}$$

Comparing (4.4) and (4.5) with (4.3) the scale factor from the circuit perspective is

$$S = \frac{A_v}{\delta_{adc}} \tag{4.6}$$

Therefore, there are two ways to control the AIMC output quantizer, adjusting ADC quantization steps or AIMC analog gain. ADC quantization steps are usually optimized and fixed according to the voltage dynamic range and ADC bit precision. On the other hand, it is preferred to support quantization dynamic control via analog gain.

To control the quantization via analog gain, one or more parameters to change the analog gain have to be devised during the design. These parameters should be able to change easily to set different quantization set-ups while executing different layers. In addition, the controllable analog gain range should be adequately wide to support different possible quantization configurations. The effect of the gain-controlling parameters on other performance figures should also be taken into account. Because the gain-controlling parameters have effects on other performance figures, it is beneficial to have more of these parameters. This gives more flexibility to optimize the affected performance by tuning the correct parameter for specific applications. Moreover, the relationship between the parameters and quantization should be defined clearly. Due to analog devices' higher-order effects and non-idealities, it usually cannot be done analytically. Thus, a measurement and characterization campaign may be needed.

The parameters that can be used for analog gain control are as diverse as AIMC structures. For example, in the memristor-based [97][98] architectures, the memristor value and the Digital-to-Analog Converter (DAC) gain are related to the analog gain. Memristor values are programmable, and there is a possibility to consider a scale on them. In designs with PWM

DAC [99][100], the PWM unit time is a potential parameter to be easily programmed to control the gain and modulate the quantization. In DIANA, there are PWM unit time and current limiting transistor bias voltage for this purpose. The fact that DIANA SoC uses a combination of two parameters that one, e.g. bias voltage, has a non-linear relationship with the analog gain, and the other is a commonly used parameter in time-domain AIMC makes DIANA a good example to be a case study in this chapter.

This section developed a technique for controlling the AIMC output quantization. The relationship between gain and quantization was defined, and it was shown that the analog gain is preferred to set the quantization parameters. Hence, gain-controlling parameters should be utilized in AIMC to dynamically adjust quantization for each layer. In the next section, DIANA's AIMC will be introduced to be used as an example for this technique. The parameters involved in its analog gain will be identified, and their relationship with quantization parameters will be studied.

4.3. DIANA'S AIMC Macro

This section first briefly describes the AIMC macro implementation integrated into DIANA. Then, it focuses on the AIMC output quantization parameters control. It finds the circuit parameters that can modulate the output quantization. The effects of these parameters on quantization will be modeled in the following sections.

4.3.1. AIMC macro structure

The macro is an 1152x512 array of Analog Processing Elements (APE). When the macro is fully utilized, 1152 7-bit activations are converted to PWM signals. Then, each is fed to all 512 APEs in a row. APEs multiply activations by ternary weights (+1, 0, -1) stored in two standard 6T SRAM cells. The product is accumulated at summation lines, which connect the APEs in the same column. Finally, 512 6-bit ADCs convert voltage on the summation lines to digital. Fig. 17 illustrates the simplified diagram of the AIMC.

Fig. 18 shows the transistor-level schematic of an APE. PWM DAC produces two active-low signals, Act- and Act+. Each signal is used to modulate the activations with the corresponding signs. Act+ and Act- are connected to the source of two transistors. Two SRAMs store weight (W+) and negated weight (W-); each is connected to the gate of two transistors with different PWM signals at sources. The drains of the transistors with concordant sign signals (W+ and Act+ or W- and Act-) go to the positive summation line; Sum+, and the other two (discordant signs) go to the negative summation line; Sum-.

At the beginning of a processing cycle, summation lines are pre-charged to VDD. For non-zero weights and activations, one transistor turns on and determines the connected summation line and, consequently, the product sign. PWM width dictates the magnitude of the product. Outputs of APEs are in the form of current. They discharge the connected summation line proportional to the pulse width. Thus, accumulation is conducted at the summation lines. The readout circuit deals with Sum+ and Sum- as differential signals and sends the results to ADCs.

Two current-limiting transistors connect each APE to the summation lines. It adds more flexibility to the design and mitigates the channel length modulation effect. The bias voltage of these transistors (hereafter V_{bcs} or bias) and PWM unit time determine the quantization parameters and control the resolution and dynamic range of the output activations.

4.3.2. DIANA's AIMC output quantization

The ADCs' voltage thresholds are fixed. Therefore, output activations (O) are related to the summation line voltages (V_{adc}) as (4.5). The summation line voltage is proportional to the result of the MAC operation. We define the unit voltage (V_u) as the summation line voltage corresponding to the result of a MAC equal to one. With the assumption that the cell currents are DC, as the summation lines are capacitive, the summation line voltage is equal to:

$$V_u = \frac{t_u . I_{cell}(V_{bcs})}{C_{line}} \tag{4.7}$$

I_{cell} is the cell current that is a function of bias voltage. t_u and C_{line} are respectively unit time and summation line capacitance. V_{adc} corresponding

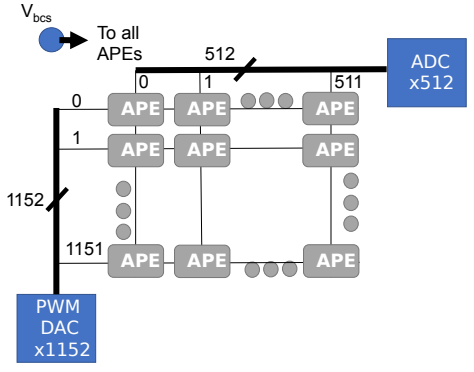


Figure 17. Block diagram of the AIMC macro; each of 1152 activations goes to 512 APEs in a row (horizontal lines). Outputs of APEs are accumulated in summation lines (vertical lines)

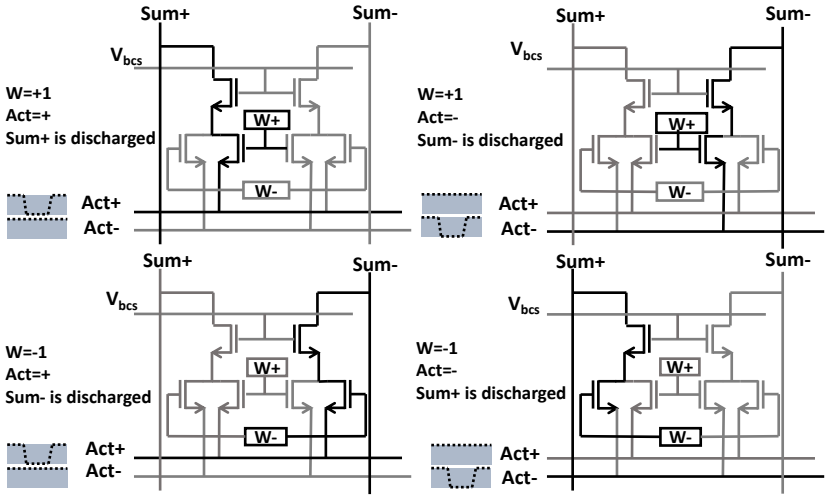


Figure 18. AIMC macro transistor-level schematic; active paths are shown with black lines. Activation and weight signs determine the product sign. The result magnitude is controlled by that of the activation.

to other MAC operation results are proportional to the unit voltage:

$$V_{adc} = V_u.O_{mac} \tag{4.8}$$

combining (4.5), (4.7), and (4.8), we have:

$$O = int(\frac{t_u.I_{cell}(V_{bcs})}{\delta_{adc}.C_{line}}.O_{mac}) \tag{4.9}$$

With a comparison between (4.3) and (4.9) scale factor is obtained.

$$S = \frac{t_u.I_{cell}(V_{bcs})}{\delta_{adc}.C_{line}} \tag{4.10}$$

C_{line} and δ_{abc} are fixed. Thus, quantization scale factor control is possible through bias voltage and unit time. The bias voltage is applied to DIANA externally, and the unit time can be changed among 16 values by programming a register runtime.

Unit time and bias voltage have different effects on DIANA performance. So the designer has the freedom to make tradeoffs and optimize these two parameters with respect to each other to achieve the desirable quantization setup and overall performance. For example, unit time controls the AIMC's total cycle time and speed. Unit time affects power consumption more than bias voltage does [61]. However, it is shown in the next section that small

unit times may cause scheduling problems. Therefore, it is possible to make tradeoffs on speed, power, and scheduling sanity with these two parameters.

In this section, we introduced the DIANA's AIMC circuit and showed the relation between output quantization configurations and unit time and bias voltage. The AIMC quantization control can now be achieved by a model that connects bias voltage and unit time to the output quantization parameters to enable the implementations of optimization techniques like [60]. The relationship between quantization and circuit parameters is not straightforward as it is nonlinear and correlated. For example, bias voltage affects MOSFET switching time, which changes the effective unit time. Therefore, it is important to obtain the model via experiment rather than theory.

The following section will present the experimental results. These results will be used to develop the model and also to provide guidelines for the best chip setup and design improvements.

4.4. Experimental Results

The characterization campaign aims to incrementally model the AIMC behavior in order to connect output quantization to circuit parameters. It also shows the non-idealities. Information on non-idealities can be used for compensation or improvement of the next AIMC generations.

In the experiment setup, DIANA is installed on a custom motherboard. A ZedBoard™ Zynq®-7000 ARM/FPGA SoC development board is connected to the motherboard via FPGA Mezzanine Card (FMC) Low Pin Count (LPC) connector. The ZedBoard performs the DIANA's booting procedure and provides the clock signal. A PC programs and loads the inputs and weights into the chip through a JTAG interface. The PC reads the results from DIANA through the same interface. The motherboard and DIANA's power, as well as the bias voltage, is provided by two NI PXIe-4145 4-channel source-measure units mounted on a NI PXIe-1088 chassis. Fig. 19 shows the diagram of the experiment setup.

The first investigation examines the accumulation function linearity. The summation linearity is crucial because it allows the AIMC model to break down into the addition of small models of APEs using the additive property.

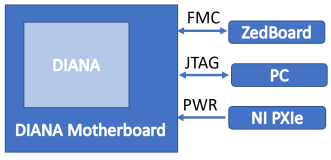


Figure 19. Measurement setup diagram; ZedBoard boots the chip, PC programs and reads the results, and NI PXIe powers DIANA

Then, mismatches between APEs are evaluated. Two design imperfections, bias voltage drop and interconnect delay, are investigated later. These three experiments give chip users guidelines to avoid non-ideality effects and provide chip designers with suggestions for improvement.

The last experiment shows that nonlinearity error is a function of the output rather than the input. This observation is utilized to develop the linear and fine-grain models combination. Eventually, the results of this part are used to plan an experimental exploration of unit time and bias that leads to a linear model in section IV.

4.4.1. Accumulation linearity

AIMC performs multiplications inside APEs and accumulations at the summation lines. If the accumulation operation is linear, the AIMC model decomposes to the addition of APEs models by utilizing the additivity property. So, before modeling APEs in section 4.5, we must show that the accumulator is linear.

The number of activations is increased in an experiment to analyze the linearity of addition. Fig. 20 shows the AIMC output for different numbers of activations. In this specific example, non-zero activations are set to 5, while weights are all positive (1). Fixed APEs inputs isolate the experiment from APE's nonlinearities. However, their mismatches still reflect in the results. In each experiment, 128 activations are added. The set of iterations is then repeated for ten values of unit time ([50ns: 140ns, 10ns]). The bias voltage in this experiment is 0.61.

Fig. 20 visualizes the linearity of the accumulation operation up to output saturation. As the ADCs output range is [-31: +31], the AIMC output is saturated on 31. A part of the nonlinearity is rooted in the APEs mismatches, as shown in the following subsection.

Thus, the rest of the experiments try to demystify the APEs, assuming accumulation at the summation lines is linear.

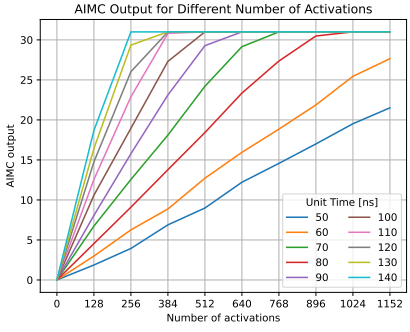


Figure 20. Addition linearity; the number of activations and addends is linearly proportional to the output.

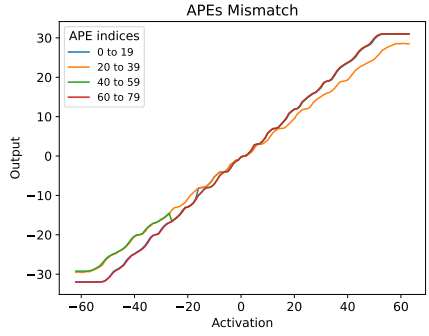


Figure 21. APE mismatch; the outputs of the experiments with the same inputs but on different APEs are slightly off due to the mismatch.

4.4.2. APE mismatch

Dealing with device mismatches is a burdensome challenge for analog designers, and DIANA's AIMC is no exception. Mismatches occur due to differences in devices that are designed identically. Coping with mismatches after tape-out is not possible. Their analysis needs a statistical approach that is out of the scope of this chapter. Fig. 21 illustrates the mismatches as differences in the AIMC output for different APEs with the same activation and weight. Here, 20 APEs are examined in each experiment, as for a single APE, the output might be weak and noisy.

Mismatches should be avoided as much as possible to achieve good linearity. The best stage to deal with mismatches is during the layout design. Matching guidelines can be found in analog layout books including [101].

4.4.3. BIAS voltage drop

Experiments show that, for bigger workloads, the sensitivity of the summation lines with higher indices decreases. An experiment isolated the effect by feeding the whole array with equal activations and weights. Fig. 22 illustrates the output of the summation lines. The output decreases for summation lines with higher indices.

The loading effect is related to the drop in the current limiter transistor bias voltage in the AIMC macro. Although MOSFET gates do not ideally draw any current, an array of more than one million long-channel transistors introduces big gate current leakage in scaled FDSOI technologies due to thin oxide and direct tunneling effect [102].

The design level fix is to add more bias voltage contact on the die in distant locations. At the application level, it is better if high utilization is avoided. Otherwise, post-processing compensation is necessary for chip users.

4.4.4. Interconnect delay

The AIMC output is quantized in the range [-31,31]. So, the output is saturated when it reaches 31. However, experiments show that the outputs become saturated sooner when the low-index APEs are utilized. The early saturation is shown in Fig. 23. The effect is only visible for low-index APEs and is proportional to unit time; the output becomes saturated in lower values for smaller unit times.

Interconnect delay causes early saturation. The control and timing unit (CTU) is located at the bottom of the AIMC array, closer to the high-index APEs. There is an interconnect delay (t_{id}) for the PWM DAC enable signal from CTU to the top of the AIMC array where low-index APEs and their DACs are located. Thus, these DACs start their operations later, leading to a delay in low-index APEs' PWM signals. When unit time decreases, the total AIMC cycle time decreases proportionally while the delay remains constant. So, for small unit times and big activations, a big part of the PWM signal does not overlap with the AIMC active time and is ineffective. That leads to an early saturation. Fig. 24 shows the timing diagram in a) high- and b) low-index APEs. Due to interconnect delays, for big activations, the end of the PWM signal does not overlap with the AIMC active time. So, the output is saturated as increases in activation just increase the futile part of the PWM signal.

Hopefully, it is possible to lengthen the AIMC active window by setting DIANA's control registers. So, the users should make sure to set these registers correctly while using small unit times. However, some buffers on control signals improve the design, so increasing active window time will be unnecessary. This will improve the chip's speed and power.

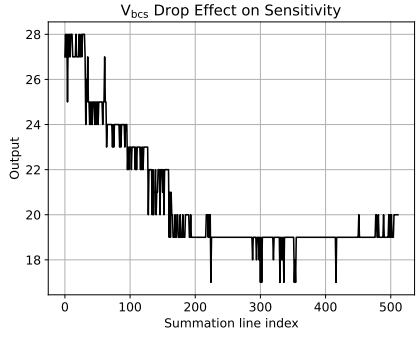


Figure 22. Current limiting transistor bias voltage (V_{bcs}) drop causes a sensitivity reduction for high-index summation lines.

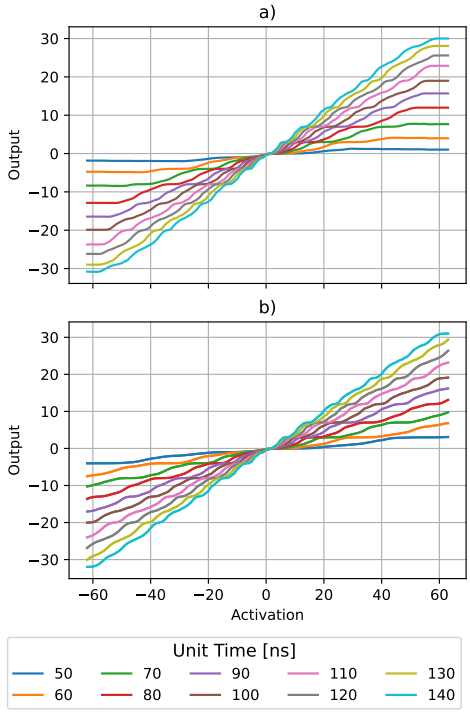


Figure 23. Large outputs linearity; a) early saturation happens for low-index APEs, and b) high-index APEs stay linear in [-31, +31].

4.4.5. Errors as a function of output

Fig. 25 shows that error is a function of the output for different unit times and bias voltages. It does not include errors rooted in voltage drop or interconnect delay. It is utilized in the next section to develop linear and fine-grain model serialization.

This section delivered observations used in the next section to develop a linear model of AIMC, the non-idealities that one should take into account during the application of the chip, and an observation that will be used for model nonlinearity adjustment. The following section will model the AIMC macro with a look-up table that translates the quantization parameters to unit time and bias voltage.

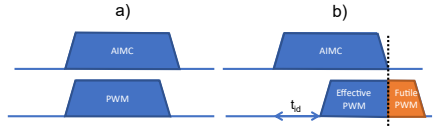


Figure 24. Timing diagram of PWM and AIMC in a) high- and b) low-index APEs. PWM signal slides off the AIMC active time due to interconnect delay and causes early saturation.

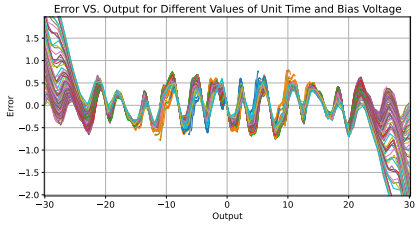


Figure 25. Error is a function of the expected linear output for different unit times and bias voltages.

4.5. Model

This section proposes an approach to include nonlinearities in a linear AIMC model. Then, we present a model for the DIANA's AIMC macro that connects its circuit parameters with scale factors and quantization configurations. At the end of the section, the model is used to evaluate the DIANA's AIMC in the implementation of output quantization calibrations from the literature.

4.5.1. Nonlinearity adjustment for linear model

In the previous section, the error of a linear model was presented as a function of the output. Thus, one can model the AIMC macro as a linear and nonlinearity adjustment model in series if voltage drop and interconnect delay errors are neglected. This approach is depicted in Fig 26.

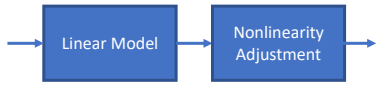


Figure 26. Linear model can be followed with a fine-grained sample-specific model for fine adjustment.

The nonlinearity adjustment is sample-specific and cannot be achieved generally.

4.5.2. Model presentation

The characterization was planned as an exploration of DIANA's operating points; bias voltage and unit time as their effects on quantization are shown in Section 4.2. Bias voltage has values between 0.5V to 0.8V with steps of 0.01V. The DAC conversion unit time ranges between 50 ns to 200 ns with 10 ns step granularity. The experiment is conducted on 40 APEs with the highest indices in each summation line. APEs with high indices are selected to avoid the interconnect delay effect. Forty APEs are used to average out the spatial variations due to mismatches while avoiding the voltage drop effects resulting from overloading. These effects should be eluded by the user with suggestions provided in the previous section or compensated in the post-processing.

The output quantization is fixed to 6-bit symmetric and uniform by design. Only the quantization scale factor and clipping range can be set during the application phase. We couple each combination of unit time and bias voltage with a quantization scale factor. For this purpose, activations are swept from -63 to +63 for Each operating point. The output is normalized by the number of APEs. A line is fitted into the normalized output versus activation graph using linear regression. As weights are one, the slope of the fitted line is the scale factor. The linear regression standard error is also calculated and provided as a measure of accuracy in the look-up table. The user can apply this error along with the observations from the previous section to favor one combination over others.

If the clipping range is obtained from quantization calibration, the scale factor can then be converted to clipping range $[-\beta, \beta]$ by the following equation.

$$\beta = \frac{2^6 - 1}{2S} \tag{4.11}$$

The minimum value for the scale factor in the look-up table is 0.00044, and the maximum is 0.0477. This means clipping ranges between [-660, 660] and [-71590, 71590] are possible. MAC operations results conducted on DIANA's AIMC can be in the range of [-72576, 72576] (1152x63) that almost fit inside the wider clipping range. The smallest possible step size equals 20.6.

TABLE 3 is a part of the model look-up table. Let's assume the calibration leads to a layer with a scale equal to 0.01. If the layer can be fitted in high-index APEs, interconnect delay is insignificant. Thus, the user can select combination 1, which has a smaller unit time and consequently less power consumption and higher speed [14]. However, small unit times should be accompanied by large AIMC active windows for layers that utilize the low-index APEs to mitigate the interconnect delay error. Combination 5 is a good choice to reduce the interconnect delay errors at the cost of lower speed and higher power consumption for layers that use low-index APEs if increasing the AIMC active window is not an option.

4.5.3. DIANA's output quantization capability for accepting calibrated parameters

To validate our method, we use quantization parameters from [60] to execute ResNet-20 for CIFAR10 and ResNet-18 for ImageNet on AIMC. These parameters are obtained via two quantization calibration approaches; network-wide and layer-wise.

In the network-wide quantization, they selected a single scale factor for the whole network by which the network accuracy is maximized. The scale factors were 0.031 and 0.018 for Resnet-20 and Resnet-18, respec-

Table 3. MODEL's LOOK-UP TABLE EXAMPLE

Nr.	Scale	Bias Voltage	Unit Time [ns]	Standard Error
1	0.010264	0.76	80	$2.16e^{-5}$
2	0.010265	0.63	130	$2.08e^{-5}$
3	0.010379	0.69	100	$2.18e^{-5}$
4	0.010419	0.72	90	$2.17e^{-5}$
5	0.010494	0.62	140	$2.05e^{-5}$

tively. These scale factors were implementable by DIANA with our model. Nevertheless, they showed that with layer-wise quantization, in which the quantization setup is unique in each layer, a 0.5% higher network accuracy is achievable (90.1% for Resnet 20 and 64.7% for Resnet18). Therefore, we analyze layer-wise quantization in more detail.

For each layer, we use the look-up table to generate the optimal unit time and bias voltage for efficiency in DIANA. The first observation is that some scale factors are beyond the hardware capabilities (e.g., need a higher unit time). There are different solutions to this problem: during design, one can either allow a wider range of legal unit times or make changes at the circuital level, such as reducing the summation line capacitance. Increasing the ADC resolution is also an option that comes with the power cost. At runtime, it is possible to execute the computation with half the required scale factor and multiply the result by two in the digital domain. There is a post-processing SIMD unit in DIANA that can be utilized for this purpose. Using post-processing is a power-efficient and straightforward solution. However, it only mimics the scale factor of the calibrated quantization setup, and its step size is twice bigger, which can degrade the accuracy. To avoid this degradation by consuming more power, a scaling unrolling scheme can increase the obtainable scale factor; if weights and activations are unrolled twice, it is like a gain of two, and all scale factors in the look-up table are doubled. Finally, it could be possible to constrain the neural network training to lower scales, but the viability of this last option is out of the scope of our work.

To further assess the benefit of using AIMC for ML workloads, we measured the performance and power consumption for the different workloads in the ResNet20 and ResNet18, changing the unit time. Fig. 27 shows the relationship between efficiency/performance and unit time. Longer pulse widths result in higher power consumption and a slower computation cycle. The dots on the line represent the mapped scale factors for different layers. The dots on the extreme right, marked with a star, exceed the macro operating limits and have been mapped with the highest gain possible.

In the plots, we reported peak performance and efficiency from the digital core of DIANA [61] and TinyVers [103], a digital SoC that targets extreme edge and efficient inference, to compare analog and digital computation paradigms for real workloads. When considering efficiency, the DIANA analog macro dominates its digital counterparts: only in limited cases TinyVers is comparable with AIMC but with two orders of magnitude degradation in performance. Focusing on performance, TinyVers is bounded by a 10 MHz clock and low throughput, while DIANA digital core shows better performance than analog on the early, small layers from ResNet20. The comparison shows the performance, efficiency, and accuracy dilemma. The digital platforms exchange performance and efficiency, TinyVers in favor of efficiency, and DIANA in favor of performance. AIMC achieved higher figures in both merits, however, by sacrificing deterministic computation accuracy. We also noticed that unit time affects performance and efficiency for the analog macro in different degrees; while performance can only degrade by a factor of 2, efficiency can decrease by one order of magnitude when unit time increases over its legal values.

The developed model can translate the calibrated output quantization parameters into DIANA's AIMC's operating points; bias voltage and unit time. Applying output quantization calibration is essential to achieve high accuracies. It is shown that DIANA can implement network-wide calibrations reported in the literature [60]. However, The current design needs

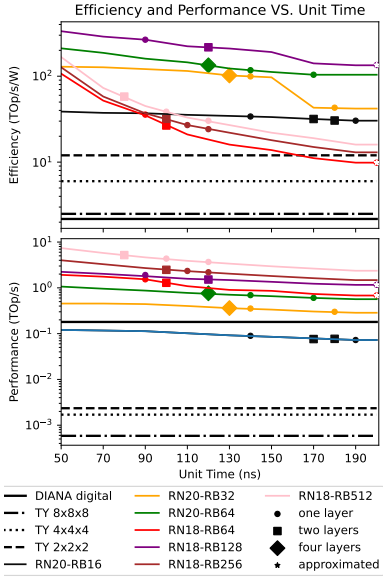


Figure 27. Efficiency and performance of DIANA's AIMC for different layer structures along with the digital baselines and mapped output quantization operating points from [60].

minor changes to support the higher scale factors that are required for layer-wise quantizations that offer more accuracy. The overall standard error of the model is always below $3.4e^{-4}$, suggesting good linearity of the DIANA's AIMC. However, a nonlinearity adjustment model is proposed that can be utilized to study AIMC variability impact or in training or compensation.

4.6. Chapter Conclusion

The primary purpose of this chapter was to bridge the gap between theoretical works on AIMC output quantization calibration and the practical difficulties of working with AIMC analog circuits. Hardware imposition of optimized quantization parameters is important for achieving high accuracy. The aim is fulfilled by studying the quantization from both network and circuit perspectives. The analog gain in AIMCs should be controlled in order to set the quantization parameters. As a case study, the method is applied to DIANA's AIMC output quantization calibration. We coupled the calibrated quantization parameters with the chip's operating points that determine its analog gain in a look-up table. Thus, a dynamic quantization control on DIANA for implementing layer-wise quantization calibration is possible by using the look-up table.

It is also learned from the case study that more AIMC analog gain range improves the control over quantization parameters, enables more quantization implementations, and increases the achievable accuracies.

This chapter also spots the design improvement points in DIANA and suggests solutions. More bias voltage contacts solve the voltage drop problem, and early saturation is remedied by interconnect delay reduction. These minor fixes can benefit the chip performance by a significant amount. The improvement points can be important for other AIMC designs to avoid the trial and error phase.

DIANA is tailored for drones, which are considerably larger than insect-scale robots. This chapter has demonstrated that the majority of issues with DIANA arise when a tuning parameter is used to broaden the chip's application range. While this is essential for drones, it may not be necessary for insect-scale robots. These robots are intended to be produced in swarms, be cost-effective, and have a short lifespan. As a result, the need for flexibility in expanding the application range is not required. A network-specific CNN inference engine can improve performance and efficiency by optimizing the circuit for a single network and eliminating auxiliary blocks and functionalities designed to increase flexibility. Furthermore, data transfer accounts for a significant portion of the chip's time and power. In a network-specific ASIC design, it is feasible to avoid weight data transfer by hardcoding them into design parameters. Finally, ADCs are the bottleneck in AIMC designs, although this chapter has offered techniques to optimize these blocks with calibration.

Drawing from the lessons learned from DIANA, we will design a new network-specific CNN inference engine ASIC in the following chapter. The ASIC design is optimized to process a specific network with the highest possible efficiency. The network's weights are embedded in the chip's transistor aspect ratios, eliminating the need for weight data transfer. The data is managed in the analog domain and transferred between layers without being written to memory. Therefore, there is no need for ADCs as the data remains in analog format. The innovative chip shows promising results in meeting the stringent requirements of insect-scale robots.

5. ANALOG ACCELERATOR FOR INSECT-SCALE ROBOTS

The previous chapter showed the power of analog computation to minimize the size and power of CNN accelerators. However, scalability and one solution for all applications are not analog designers' cups of tea. Analog circuits are not programmable, and the panacea approach does not tap the full capability of analog computation. A network-specific accelerator is the optimum analog solution for insect-scale robots with strict payload limitations.

In this chapter, after reviewing the literature on smart insect-scale robots, we design an analog accelerator with new designs in architecture and transistor levels. The designs prioritize efficiency over programmability. So, the accelerator will be suitable for insect-scale robot applications. Considering their small material and fragility, the insect-scale robots will be disposable or have a short life span. Therefore, programmability is not only hindering the miniaturizing accelerators but may also be unnecessary. However, the design has some levels of programmability and flexibility at the Fully-Connected (FC) layer.

5.1. Introduction

Visual perception constitutes 90% of human brain input [104], and machine vision is proven to be a disruptive technology in robotics [105]. Convolutional Neural Networks (CNN) are used as a solution to perform machine vision tasks adapted from the Artificial Intelligence (AI) domain for image classification problems [106, 107]. It is utilized in robots' locomotion control for applications such as obstacle avoidance [108], target detection [109], foothold selection [110], and trajectory planning [111]. However, CNN-based onboard locomotion control has only been deployed for relatively large-scale robots [112] owing to the high-power and area requirements of CNN processors and the low payload capacity of insect-sized robots [113, 62]. This poses a pressing need to reshape CNN processors for insect-sized robots in terms of size, weight, and power (SWaP) cost.

5.1.1. Existing solution

Off-board CNN-based visual control. Recent work [112] has demonstrated the utility of CNN for visual control at the insect scale. A custom-built low-weight vision sensor is mounted on a flapping wing insect-sized robot. The images are classified using CNN implementation off-board to make the robot recognize and repeatably move toward flower images and away from predator images. nevertheless, owing to the computationally expensive [114]

CNN algorithms and the payload and power constraints of the robot, the system is unable to accommodate onboard computation, restricting its field of operation. Small payload capacity in the order of few hundreds of milligrams in insect-sized drones [113] and tens of milligrams in ionic electroactive polymer (IEAP)-based robots [62] is prohibitive for power and control autonomy.

Non-CNN smart control solutions. An insect-sized robot with extended payload capacity [113] is shown to have enough payload for either power or sensor autonomy, not both. By shrinking the processor and reducing the power, Application-Specific Integrated Circuit (ASIC) hardware accelerators can improve both control and power autonomy in compliance with small robots' low power budget (100 μW to 100 mW for the whole system [115]). An autonomous 10-cm glider (MicroGlider) is demonstrated in [22]. MicroGlider has an audio-based guidance system assisted by an optic flow ASIC processor. BrainSoC [116] is a central controller designed for controlling insect-scale flapping-wing robots. It uses hardware accelerators for edge sharpening and optical flow. In [115], a Binary Neural Network (BNN) hardware accelerator is reported to have potential applications in insect-sized drones. To the authors' knowledge, no CNN hardware accelerator has been reported for onboard control of insect-sized robots.

Efficient CNNs with other intended applications for repurposing. References [117, 118, 119] report low power integrations of CNN/BNN first layers with CMOS Image Sensor (CIS) for always-on devices. Although a camera and a CNN's first layer on the same chip reduce the energy-hungry interchip data transfer, a higher level of integration is required to comply with the restricted requirements of insect-scale robots. Reference [120] shows a CIS integrated with a full analog convolutional processor for an always-on image sensor. However, although this reference exploits analog computation compactness, using capacitors as memory components deteriorates its performance and hinders its repurposing for insect-sized robots.

Memory wall problem in analog computation. Analog computation represents each pixel with a single signal (voltage or current) and performs multiply-accumulate (MAC) operations with approximately one transistor per input bit. This feature empowers analog computation compared to its digital counterpart, which needs several gates for each operation. Despite this advantage, interlayer memories in existing CNN architectures impede the widespread utilization of analog processors. In a feed-forward CNN, each layer's input is the previous layer's output. Thus, the output features of a layer should await the completion of the rest of the features' map in memory. This leads to massive memory walls between layers. Analog designers can realize these architectures either by several power-hungry conversions between analog and digital domains to store features in digital interlayer memories or by implementing slow and error-prone analog mem-

ories [120, 121], of which neither is appropriate. Pipeline architecture [122] has been proposed to minimize the memory walls. However, if the output is analog, it still needs to be either converted into the digital domain to be stored in the memory, or designers are required to tackle the hassles of analog memory [123].

5.1.2. Chapter contribution

As the problem arises at the architecture level, a solution should be sought by an interlevel design in which the algorithm and architecture are selected or designed for analog computation.

Hybrid design approach. In this work, with a hybrid bottom-up and top-down design approach, a new architecture is tailored for an optimal analog computational performance (bottom-up) in which memory is omitted completely. Then, an algorithm is selected to keep the architecture reasonably sized (bottom-up) by avoiding overlaps between receptive fields of the output map features while achieving the application requirements (top-down).

Memoryless architecture. The proposed architecture is completely memoryless, i.e., no storing components such as capacitors are used as they slow down the system by introducing time constants to the circuit, as well as increasing processing time and energy consumption per image. Therefore, designs such as [120] that do not have a memory block but use capacitors to store data after each clock cycle are not considered to be memoryless.

Novel low-power analog circuits. The convolutional processor in this chapter, based on the new memoryless architecture and novel low-power analog circuitries ($<$1.5 nW/image), fits the low power requirement and complies with the restricted power budget of insect-sized robots.

Our contributions towards the first autonomous insect-sized robot that will have a single-chip control unit, including a full CNN inference engine, controller, and CIS are as follows:

• We proposed a new architecture termed Fully-Fused that omits the need for intermediate memories and ADC/DACs and significantly lessens the requirements on output ADC.

• Novel circuitries are proposed to realize the Fully-fused architecture, among them a dual-purpose input DAC/convolution circuitry, which performs both operations with about one transistor per input bit.

• It is shown that the proposed analog convolutional processor is four orders of magnitude more efficient than [120], achieving 46 TOPSPW without sacrificing accuracy.

The rest of the chapter is organized as follows: Section 5.2 presents the novel convolutional processor architecture. Then, circuit topology for each layer is proposed in Section 5.3. Section 5.4 explains the experimental setup. The interpretation of the results is discussed in Section 5.5, followed by concluding remarks in Section 5.6.

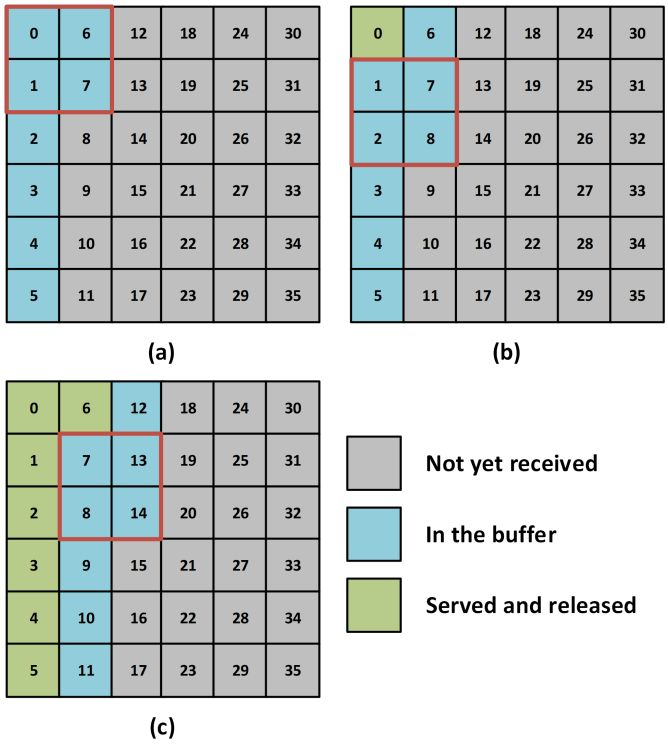


Figure 28. Pipeline architecture (a) enough pixels in the buffer for one kernel operation (red box). (b) kernel operation is done, pixel number 0 is released from memory to free space for pixel number 8. (c) the process has progressed for six clocks. (Taken from [122])

5.2. Architecture and Algorithm

5.2.1. Architecture

With contemporary computer architecture, interlayer memory walls are responsible for the significant area and power dissipation in CNN accelerators [124]. Pipeline architecture was devised to shrink the memory walls between the intermediate layers [122]. However, it fails to ease the ADC/DAC requirements. For an ADC/DAC-free accelerator, the circuit should completely lose the intermediate memories.

Pipeline architecture. In the pipeline architecture, the features are stored in a buffer awaiting the rest of the array. When there are sufficient features for a kernel operation in the next layer, the accelerator conducts the operation. Then, the accelerator discards any used features that are not involved in forthcoming operations to free up space for new features from the previous layer (Fig. 28).

Fully-fused architecture. The architecture proposed in this chapter is based on two additional aspects of pipeline architecture. First, kernel op-

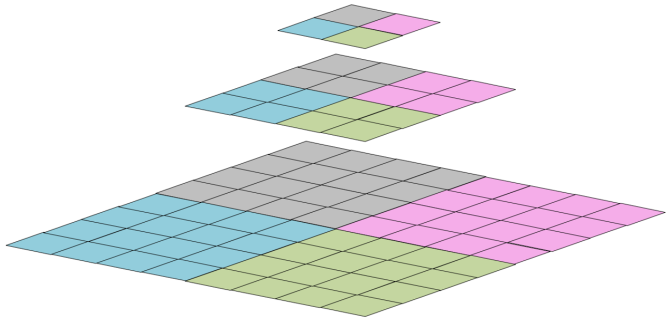


Figure 29. Proposed fully-fused architecture. Bottom: input layer. Center: intermediate layer. Top: output layer.

erations are spatial. It means that the order of pixels in the input register matters. For example, in Fig. 28, if the order of incoming pixels changes to 0, 1, 6, and 7, the number of required buffer registers is reduced to 4. Moreover, when only 4 pixels need to be accessed simultaneously, they could be obtained concurrently with parallel computation instead of using memory.

Therefore, the first layer throughput is selected equal to the receptive field of the first feature in the convolutional processor output. Then, the accelerator conducts parallel computations and provides the intermediate layers with the required input to produce one feature at the final output. The throughput then moves to the receptive field of the second output feature and so forth.

Fig. 29 shows the fully-fused architecture for a small convolutional network with one intermediate layer with stride and kernel size 2×2. As only one output is produced at each clock, the requirements on the output ADC are also reduced.

5.2.2. LWCNN algorithm

This paper adapts the LightWeight Convolutional Neural Network (LWCNN) algorithm [120] according to the applied image resolution due to the reasons stated below:

- The fully-fused architecture has the best efficiency for algorithms in which the stride and kernel size are equal in each layer. In this condition, the receptive fields of output features share no pixel. If receptive fields share pixels, a bigger throughput is needed to prevent redundant calculations.
- The algorithm consists of only four layers and conforms to the SwaP cost for insect-sized robotics. The algorithm has been tested successfully [120].

  The modified LWCNN algorithm is illustrated in Fig. 30. The convolutional processor consists of two convolutional and two pooling layers. The kernel in each layer is 2×2 with a stride of 2. The last pooling

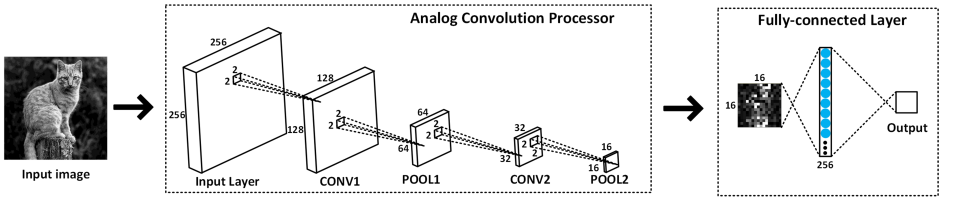


Figure 30. LWCNN layer and kernel dimensions abstract visualization.

output is 256 features which are processed in the fully-connected layer for binary classification.

5.3. Novel analog circuits for CNN accelerator

Two different convolution circuits, two versions of a maximum pooling circuit, and a fully-connected unit are used to realize the LWCNN algorithm with the fully-fused architecture. The input circuitry performs both digital to analog conversion and convolution. The second layer is a modified voltage-mode MAX circuit. Layer three conducts convolution with a differential pair-like circuit. And the last layer is again a voltage-based MAX. The fully-fused architecture requires simultaneous readiness of input for each layer. Shift registers transfer input pixels to the convolution layer. Then, 64 convolution blocks in the first layer (CONV1) provide inputs for 16 first pooling layer blocks (POOL1). The output of 16 POOL1 blocks goes to 4 CONV2 blocks that feed the last layer, POOL2, which produces one output that is converted to digital and handed to the digital fully-connected layer. The top-level topology is depicted in Fig. 31.

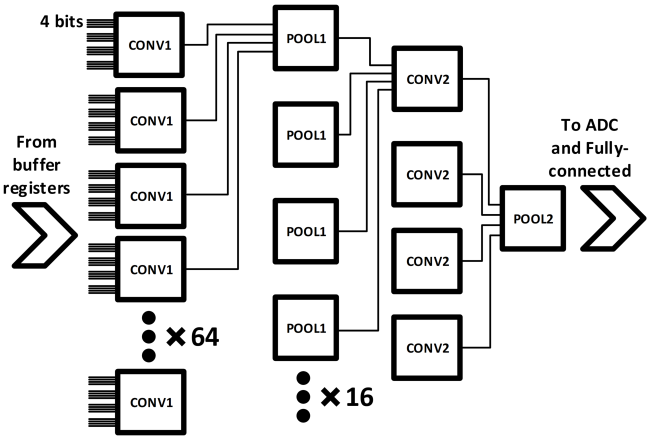


Figure 31. Top level topology of analog convolutional network.

5.3.1. Dual-purpose DAC/convolution input circuitry

As mentioned, the power consumption of ADC/DAC blocks is the primary bottleneck in realizing CNN analog accelerators. Whereas the new fully-fused architecture omits the intermediate ADC/DAC and reduces the output ADC requirement to one pixel per clock, the proposed input layer precludes the input DAC as the conversion is carried out concurrently with the convolution.

Mathematical background. The input layer circuitry is designed considering that DAC and convolution functionally resemble each other. For DAC, each bit is multiplied by its weight; then, all the products are summed up together according to (5.1).

$$P = \sum_{i=0}^{N-1} b_i W_{bi} \tag{5.1}$$

In (5.1), b_i is the bit value, W_{bi} is the bit weight, N is the number of bits used to represent each pixel,and P is the pixel value.

Similarly, convolution results are the summation of all pixel-weights' products as shown in (5.2):

$$C = \sum_{i=0}^{K-1} P_i W_{Pi} \tag{5.2}$$

where C is the convolution output, W_{Pi} is the pixel weight, and K is the number of pixels in a kernel.

By factorization, it is possible to multiply each bit with the product of bit and pixel weights and then add all the results together, directly obtaining convolution output:

$$C = \sum_{i=0}^{(N-1)(K-1)} b_i W_{ni} \tag{5.3}$$

W_n is $W_n \times W_b$

Circuit realization. The first convolution block is implemented using current sources switched by bit values. For each bit, a current source transistor, e.g., a simple common-source NFET, is placed. The aspect ratios of current sources are set proportional to W_n of the corresponding bit. The current sources for bits with negative and positive pixel weights are normally-on (gates are connected to the $\bar{Q}$ output of the last flip-flop of input image shift register) and off (connected to Q output), respectively. Drain currents of all transistors go to the load transistor, generating a proportional voltage. Hence, when all bits are zero, there is a neutral voltage (corresponding to zero) at the output. As any bit turns one, its current source turns on for

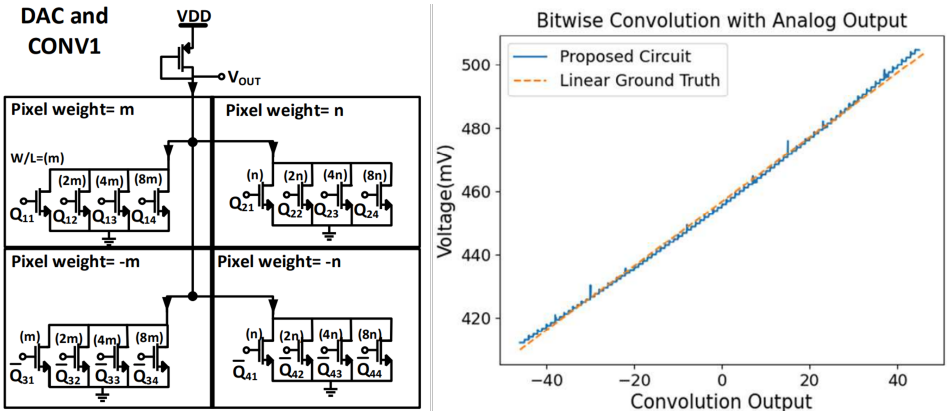


Figure 32. Transistor-level schematic of the input layer circuitry and its output voltages for all possible convolution values.

bits in pixels with positive weights and off for bits in pixels with negative weights. So, the output voltage changes proportional to W_n.

Fig. 32 shows the transistor-level implementation of the convolution block with a kernel size of four 4-bit input pixels. The first and second digits in the Q subscript indicate the pixel and bit numbers, respectively. This figure also illustrates the convolution block output. With four 4-bit data with weights of [-2, -1, 1, 2], the output can have 92 values. The results are almost linear. It is worth mentioning that in the case of a Relu activation function, the negative values are disregarded, leading to an even more linear output. According to the architecture and the algorithm, 64 instances of this block are needed in the input layer. Thus, this part highly influences the power and area of the circuit.

5.3.2. Pooling circuitry

Pooling (POOL1 and POOL2) blocks are modified versions of the widely used voltage-mode MAX pooling circuit [120] (see Fig. 33). However, as in this paper, the first pooling output should be assigned directly to a differential pair-like circuit; it is essential to have an adequate DC component for biasing the next stage and have small swinging to maintain the following circuit in the saturation region. Also, the impedance at the output node should match the one at the corresponding node of the input branches. Therefore, three transistors are placed at the output to add more degrees of freedom in the first pooling layer to meet these requirements (Fig. 33). The second pooling layer is a standard voltage-mode MAX circuit [125] and has only one NMOS in the output branch. The waveforms of the output pooling block are depicted in Fig. 6. The output follows the maximum input voltage performing the pooling operation.

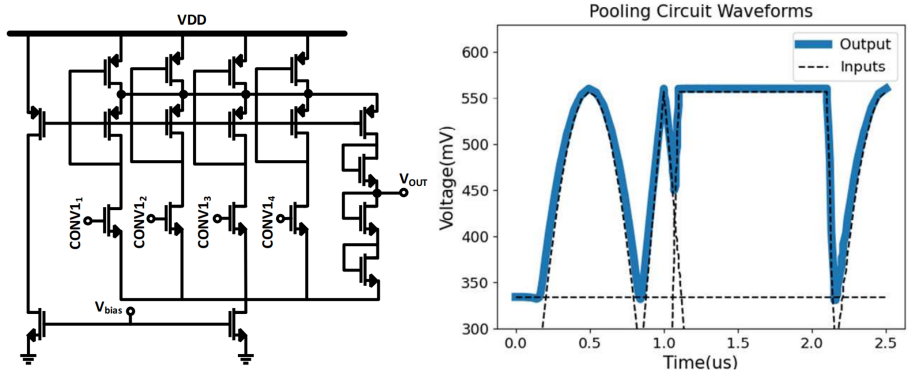


Figure 33. The modified voltage-mode MAX circuit and its input and output waveforms

5.3.3. Second Convolution

A differential pair configuration with two parallel transistors on each side forms the CONV2 blocks. The schematic diagram is shown in Fig. 34. NFET aspect ratios and subsequently transconductances are set proportional to the weights. NMOS transistors corresponding to positive and negative weights are shown on the right and left-hand sides, in Fig. 34, respectively. Therefore, they add or subtract a proportionate current on the PMOS load transistor. Inputs/output waveforms are illustrated in Fig. 34. The Output slope in each instance is equal to the weighted sum of the slope of inputs.

5.3.4. Fully-connected

The digital fully-connected layer is placed after the convolutional processor. This layer conducts weighted-sum operations on the analog part output to categorize images into two classes.

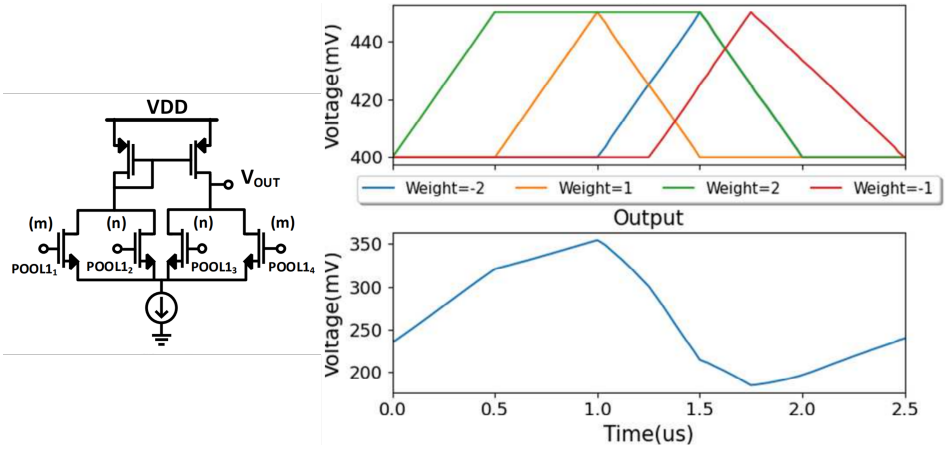


Figure 34. Second convolution layer Schematic diagram and waveform.

At each clock cycle, the convolutional processor via an ADC provides the fully connected layer with one feature. The feature is multiplied by its corresponding weights and added to the values of two registers attributed to each class. At the last clock of each image, the fully connected layer determines the image's class according to the registers' values and resets the registers.

5.4. Experimental Setup

The transistor-level implementation of the proposed circuit is simulated in TSMC 40nm technology using Cadence Design Suite together with the Spectre Simulator. Shift registers for input images and the fully-connected layer are implemented in VHDL and included in the design to evaluate the final performance and accuracy of the LWCNN with proposed analog convolutional circuitry for face detection.

The training is carried out by a dedicated script written in Python. A dataset of 2700 gray-scale images (resolution 256×256) consisting of 1000 images of human faces (randomly taken from LFW dataset [126]) and 1700 images of cats and dogs (randomly selected from [127] is used. For power and memory considerations, all images, in both training and inference phases, weights, and the ADC are 4-bit.

The obtained weights are embedded into the VHDL of the fully-connected layer. The inference is conducted in the Cadence Design Suite. Four hundred fifty images (150 humans and 300 animals from the aforementioned datasets) are analyzed to attain the system's accuracy.

5.5. Results

The results of the convolutional network are assessed in two ways. First, the output of the proposed circuit for an image is compared with its ground truth obtained with a Python script. The comparison result shows good conformity: the simulated results and ground truth are visualized side by side in Fig. 35. It also shows the error distribution. The error for 188 pixels out of 256 is less than 5%. The mean square error (MSE) of the normalized values is calculated as 0.007. The error does not affect the final accuracy of the system.

Then, according to the previous section, the convolutional network is simulated along with shift registers at the input and an ADC and fully-connected layer at the output to evaluate and compare the CNN system performance with cutting-edge accelerators. Table 4 compares the proposed accelerator with state-of-the-art CNN processors [117, 118, 119, 120] and [128]. This table provides a thorough comparison of different works

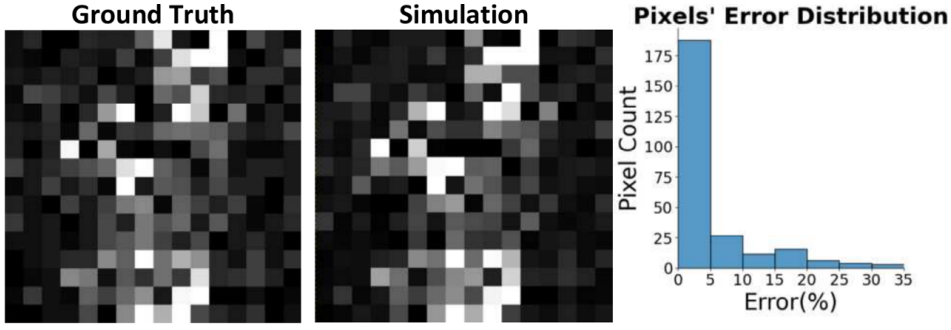


Figure 35. Visualization of simulation results and ground truth with pixels' error distribution .

regarding their characteristics of the design (technology, implemented circuit, and supply voltage) as well as their applications. It also reports the accuracy of the neural networks with their image resolutions and power and efficiency of implemented analog circuitry. The efficiency is measured using a criterion called Tera (MAC) Operation Per Second Per Watt (TOPSPW), which is the number of (MAC) operations is done in a processor normalized by power and time to give an unbiased comparison. Shaded columns represent partial analog implementations of the convolutional processor while this work and [120] realize a complete analog convolutional processor. References [117] and [119] consume less power than our work. However, they implement only one layer of network, and the efficiency criterion shows 5-9 times improvement, normalized by network size. In addition, the efficiency of our work is 23365 times, and the power consumption is 23.9 times better than [120], which used the same LWCNN algorithm.

	[118]	[119]	[120]	[129]	[121]	This work
Technology	Samsung 65 nm	180 nm	180 nm	45 nm	Dongbu 110 nm	TSMC 40 nm
Application	Face recognition	Classification	Classification	Feature extraction	Face detection	Face detection
Implementated analog circuitry	First CNN layer	First CNN layer	First BNN layer	Kernal filter and relu function	2 layers of CONVs and POOLs	2 layers of CONVs and POOLs
Supply (v)	1.2	0.5	1.8	1	3.3	0.9
Network accuracy	96.18%	92.2%	98.3%	-	89.3%	92.2%
Resolution	320×240	128×128	32×32	120×120	160×120	256×256
Power consumption	10.17-18.75 μ W	117 μW	5.9 μW	11.44 mW (per kernel)	1.12 mW	97 μ W
Efficiency (TOPSPW)	5.18-9.06	9.08	8.23	0.1	0.002	46.73

Table 4. PERFORMANCE COMPARISON∗The shaded works did not report complete convolutional processors

5.6. Chapter Conclusion

The chapter presented a convolutional processor for CNN hardware acceleration based on a novel architecture. The proposed Funnel architecture omits the need for memories and dedicated ADC/DAC stages within the convolutional processor. A dual-purpose input layer for the convolutional processor was designed to satiate the accelerator with DAC while also performing convolution with almost a single transistor per input bit. The circuit performance was compared to that of existing accelerators and showed competitive performance with significantly less power consumption ($<$1.5 nW per image) than provided by the state-of-the-art. The achieved computational efficiency in terms of TOPSPW was four orders of magnitude (more than 20,000 times) higher than the previous implementation of the LWCNN algorithm. This extremely low power consumption and efficiency promise to empower insect-sized robots with machine learning and modern robotic solutions.

Despite all the advantages of analog computing, mainstream processors are in the digital domain due to flexibility, stability, and easy prototyping. Thus, the next chapter will try to design a digital accelerator based on the same memory-less architecture of this chapter. It increases flexibility and reduces the prototyping cost and time by sacrificing efficiency. Insect-scale robots vary in shape and application and require different solutions. The LWCNN in this chapter is efficient for binary classification tasks but cannot handle more complex problems. The next chapter will also extend LWCNN and introduce fully-fusible CNNs. An extendable class of CNNs that can be fused end-to-end.

6. FFCNN EXTENSION AND FPGA IMPLEMENTATION

This section is based on the fourth paper in the publications [129]. It is meant to be comprehensive on its own. However, more details can be found in the article. In the previous chapter, I demonstrated the efficiency of a memoryless CNN analog accelerator. The accelerator gained much power and area efficiency from the memoryless architecture and analog computing. However, its extensibility, amenability to design automation, and resilience for fabrication mismatches are questionable.

6.1. Introduction

The memoryless architecture of the analog accelerator is very efficient. However, this funnel-shaped architecture is enabled by the special CNN algorithm. While the CNN performed just well for a face detection task, it comes up short for more complex applications. Thus, to achieve its maximum value, funnel architecture should be extendable.

Another limiting factor is associated with the inherent limitations of analog design. Although analog computing has been demonstrated to be highly efficient, its development cost is high, and it is prone to mismatches and second-order effects. Analog computer superiority over their digital counterparts in device numbers starts vanishing when the device size and performance tradeoff comes to the calculation, especially in smaller technology nodes. Also, analog computers do not enjoy sandbox environments like FPGAs, which are necessary for rapid prototyping. Therefore, while analog computing is highly efficient for insect-scale robots, a digital equivalent for funnel architecture can be beneficial for reducing production time and cost and prototyping.

Here, I extract the characteristics of LWCNN from the previous chapter and mix them with innovative techniques to achieve a scalable CNN class that can be fused end to end. Then, I develop a fully-fusible CNN (FFCNN) for the Cifar-10 [130] dataset that reaches an acceptable accuracy/network size ratio. Moreover, I prove the efficiency of digital fully-fused architecture over conventional architecture by developing the LWCNN with both architectures on an FPGA.

6.2. Fully-Fusible CNN

This section extracts the essence of the LWCNN from the previous chapter and exploits it to develop the FFCNNs. FFCNNs are a class of CNN that are extendable and memory-less implementable. They empower us to tackle more demanding tasks while solving the memory bottleneck problem.

Fig. 36 uses an analogy to explain why conventional CNNs are not optimum for memoryless architectures and why there is a need for FFCNNs. Overlaps between convolution windows prevent an optimal implementation of layer fusion as output activations share part of their receptive field. The layer fusion technique can only handle the shared part at the extra expense of storage or recalculation. On the other hand, CNN models with non-overlapping receptive fields can be fused end-to-end without any extra computation power. They are also most efficient in terms of the economical use of parameters [131]. In addition, these models are known to be strongly convex near the global optima [132], and their convergence is guaranteed [133]. Therefore, there is an opportunity here to redesign not only the hardware architecture to exploit the inter-layer dimension of CNNs and omit the intermediate memory but also the CNN model structure to optimize the new layer-fused hardware implementation and alleviate the storage-recalculation dilemma.

6.2.1. FFCNN Class

FFCNNs are a class of CNN that can be fully fused during the hardware implementation without needing storage or recalculation of the intermediate activations in the Feature Extractor (FE). Therefore, FFCNNs have two characteristics: the kernel size and stride must be equal, and each activation map can be processed by only one kernel. These characteristics seem to freeze FFCNNs and prevent their growth. However, we take two reasonable assumptions to unfreeze the progress with FFCNN.

The first assumption is that the image is read from memory. Thus, the image is already stored and available upon request without any storage or recalculation expenses. This allows us to apply multiple kernels with arbitrary strides on the input layer.

Second, There are accumulator registers at the end of the FE. These registers enable a multiple-accumulation operation, e.g., convolution or average pooling, at the last layer of the FE with no restriction on stride or kernel numbers. This happens because the activation is multiplied by the corresponding weight and goes to the tied accumulator registers. There, accumulations are done upon the availability of other operands.

These two assumptions help us to scale FFCNNs up by increasing branches. The input image process by N kernels. The resulting activation map of each kernel starts a branch. Branches are then processed by single non-overlapping kernels. The last layers of branches are stacked over each other. Now, conducting a convolution or average pooling between the corresponding activations of branches is possible before sending them to FC. This last operation is important to keep the dimension of the FC low while increasing the number of branches.

We design an FFCNN for the CIFAR-10 dataset with these assumptions in the next subsection.

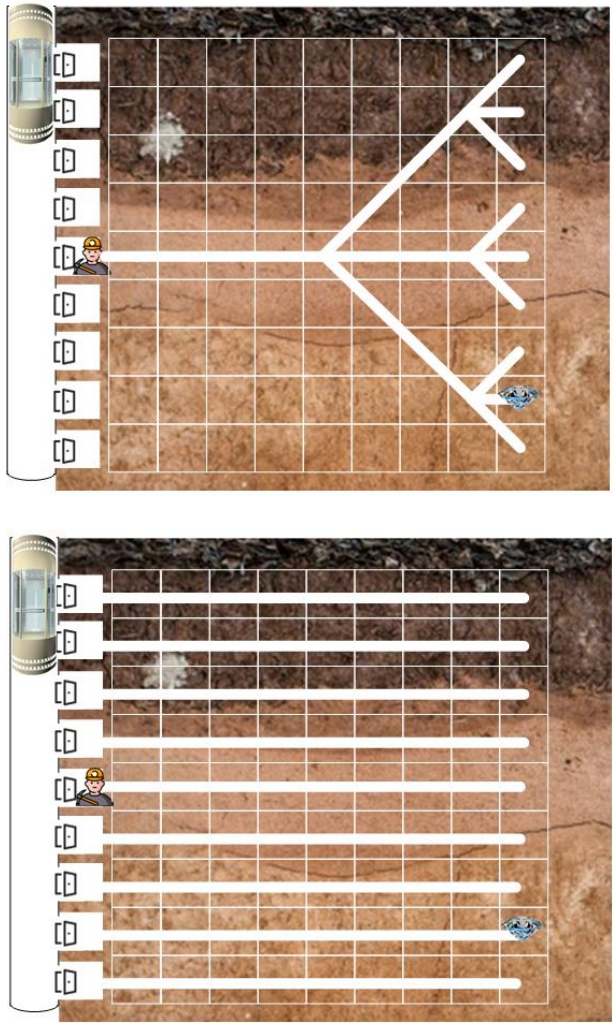


Figure 36. A miner game analogy helps us to understand the difference between FFCNNs and other CNNs. Assume there is a miner game in which your quest is to find a gym hidden in the last column of a nine-by-nine area with the fewest moves. You can start from each row without using any move with an elevator (mimicking the availability of the input image upon request without extra expenses). Then, you can go straight or diagonal. Each diagonal move costs 1.4 times more than that of the straight one. You fail if you reach the ninth column without finding the gym. Assume there is a save option (a surrogate for memory) so you can start over from the point you saved before after each failure, and your total moves add to the previous ones. It makes sense to go to some central bases (feature maps), save the state, and start over from those bases after each failure. The above image shows an optimum map (CNN) for this scenario (conventional computer architecture). However, what if there is no save option (memoryless)? The player should start from the beginning after each failure. The above map is not optimum anymore for this scenario (memoryless computer architecture) because it does not use the shortest path to the column nine squares from the beginning. The bottom map with parallel paths (branches) is optimum now.

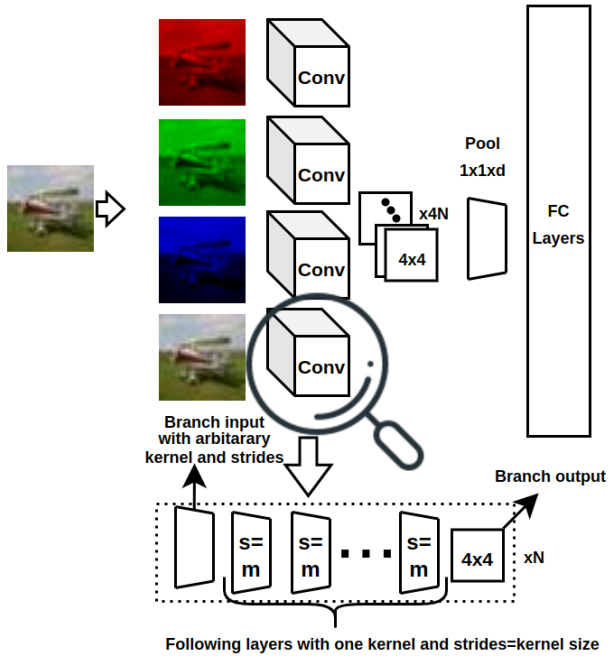


Figure 37. CNN with fully-fusible feature extraction; All kernels are square, and, Aside from the first layer, all kernel sizes and strides are equal.

6.2.2. FFCNN for Cifar-10

With an empirical approach, we learned that separately processing RGB channels increases the achievable accuracy by 0.5%. It also reduces the parameters and operations in the input layer by a factor of 3. Thus, we divided the number of branches by 4; three divisions are applied on each RGB channel and one on the full image.

Trial and error also taught us that non-identical branch structures can improve the FFCNNs' accuracy. It means that the kernel size in corresponding layers of different branches is not equal. We interpret these results as kernels with a certain size are suitable for extraction of a specific feature. So, various branch structures help to extract a wider range of features.

Therefore, we obtained the model structure in Fig. 37. Each channel, as well as the three-channel image, is processed by N parallel branches with different structures. The first layers have a stride of one. The following layers have strides equal to their kernel sizes. The outputs of branches stack over each other, and an average pooling over corresponding activations from different branches reduces the activation map dimension by d before handing it to FC layers.

It is worth mentioning that Fig. 37 abstracts the network structure, and it is different from its fully-fused implementation. If the network inference were implemented in hardware with fully-fused architecture, each activation

from branches output would be calculated with the layer fusion technique as described earlier. The activation would then be divided by four and goes to the tied accumulator registers to the FC layer input activations.

We used 128 branches, so N=32. There are eight different kernel sizes for the first layer [5, 6, 7, 11, 12, 13, 14, 15], each followed by four specific branch structures, resulting in 32 unique branches. Table 5 summarizes the branch structures. Padding is "valid" for all structures except structure 2, which has "same" padding for the sake of dimension conformity.

Table 5. FFCNN Structure

Layers	2	3	4
Structure 1	2×2 CONV	2×2 CONV	2×2 CONV
Structure 2	3×3 CONV	3×3 CONV	NA
Structure 3	2×2 CONV	4×4 CONV	NA
Structure 4	4×4 CONV	2×2 CONV	NA

The network achieved 78.79% accuracy with only 95552 parameters and 97660 FLOPs. Considering that the network can be fully fused, it shows excellent results for efficiency in edge devices. The accuracy is acceptable for the network size. However, this is a pioneer FFCNN. There is a further area for improvement

6.3. Fully-fused LWCNN

In this section, we implement a fully fusible CNN (LWCNN from the previous chapter) with a fully fused architecture and the conventional layer-by-layer architecture. Then, comparing these two implementations demonstrates the advantages of fully fused implementation of FFCNNs. This work differs from layer fusion implementations and is necessary for this thesis for two reasons. First, thanks to FFCNN, **layer fusing a CNN end to end is possible without recalculation**. In the previous works, the storage-recalculation trade-off hampers the deepening of layer fusion. Moreover, recalculation deteriorates the efficiency and performance, damaging some layer fusion advantages. FFCNNs, with network-level modifications, alleviate these barriers. Second, our end-to-end fused architecture **includes the FC layer**; the activations go through the first FC layer and are stored in an accumulator register.

6.3.1. LWCNN model

The LWCNN model architecture used in this paper is depicted in Fig. 38. Each convolution (CONV) and max pooling layer has one 2x2 kernel with a stride of two. The Dataset compromises 2700 256×256 grayscale images, of which 1000 contain a human face, randomly taken from the LFW dataset [134]. The grayscale image stays in one channel and halves in dimension from one layer to the next. Finally, activations are flattened and go to

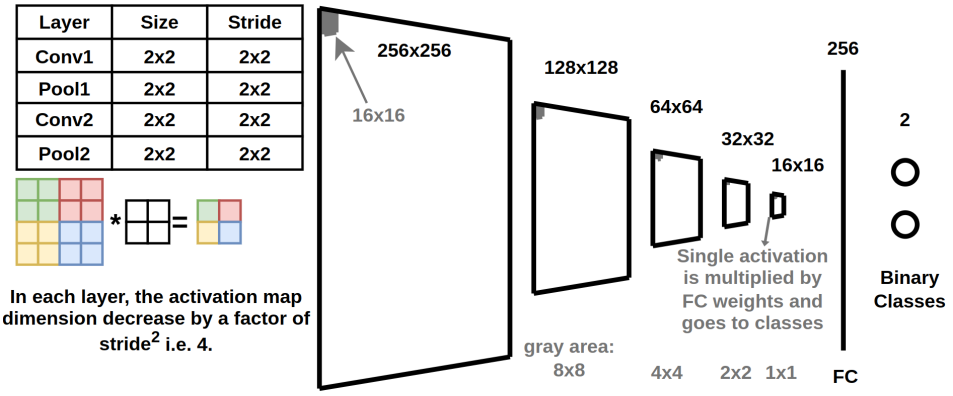


Figure 38. Model architecture of LWCNN; the gray area is the receptive field of the first activation in FC. The area slides right and then down without overlapping for the subsequent FC activations.

an FC layer, categorized into two classes. The network has a real-world application in always-on image sensors [135].

LWCNN Data Flow for Layer Fusion. The first activation in the FC layer projects a 16×16 receptive field on the top right of the input layer. The receptive field is the gray area in Fig. 38. These 256 activations should parallelly be processed to produce 16 input activations for layer Conv2. The process is repeated by processing 16 activations in Conv2 and providing one activation for the FC layer. The activation is multiplied by its corresponding weights and goes to two accumulator registers assigned to each class. The gray window then moves 16 pixels to the right to produce the second FC activation. This technique omits the memory arrays between all layers, including convolution and FC.

The absence of overlap between the receptive fields of different FC activations, i.e., equal stride and kernel size and single kernel for each layer, enabled the layer fusion without storage or recalculation. In the next subsection, we optimize LWCNN for activations and FC weights bit-width and implement it on an FPGA.

6.3.2. Bit-width Optimization and FPGA Implementation of Fused LWCNN

In this section, we first optimize the bit-with of different data in LWCNN for accuracy, efficiency, and utilization. Then, we implement the optimized network on an FPGA. We use a Z-turn high-performance Single Board Computer (SBC) built around the Zynq-7020 All Programmable System-on-Chip (SoC), including XC7Z020CLG400-1 FPGA.

Bit-Width Optimization. This part presents the bit-width optimization conducted on LWCNN for hardware design and implementation. First, we explore design space over various bit-widths for output activations of

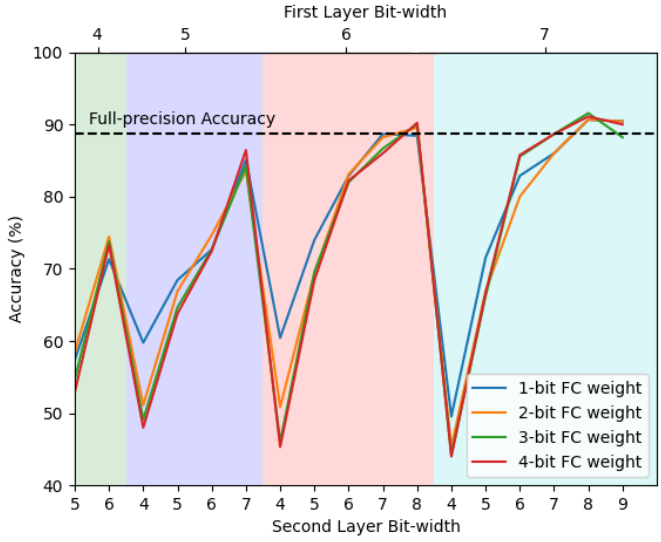


Figure 39. Accuracy exploration of LWCNN for different bit widths of CONV layers' activation outputs and weights in FC. Each line indicates the accuracy achieved by an FC weight bit-width for the different first-layer and second-layer activation bit-widths. The first layer's bit-widths are indicated on the top and divided by colors, while the second layer's bit-widths are on the bottom axis.

CONVs and FC weights in a Python simulation. Thereafter, we implement the selected designs of LWCNN in VHDL for FPGA optimization regarding power consumption and resource utilization.

The inputs of LWCNN are 4-bit integers, and CONV filters are integers between *-2* and *+2*. Fig 39 illustrates the accuracy for all 17 possible bit-width combinations for output activations of CONV layers while FC weights are quantized in 1 to 4 bits. It can be observed that, first, reducing FC weights down to one bit does not remarkably affect the accuracy, and second, some configurations achieve full-precision accuracy.

Thereafter, six configurations with the highest accuracies are selected for VHDL implementation to compare them in power consumption and resource utilization. All chosen configurations have binary FC weights, as this does not degrade the accuracy and notably boosts the performance. In VHDL, the FE is implemented in combinational logic processing one 16×16 array of an image in one clock cycle and outputs a single activation which is stored in a register.

FC layer multiplies the activation with its corresponding weights for each class. The outputs then go to accumulator registers assigned to their tied classes. The classification result is determined by the register with a higher value. All configurations are synthesized in Xilinx Vivado V2021.2 for the mentioned FPGA to obtain power and utilization reports.

Fig. 40 indicates the classification error, power, and resource utilization

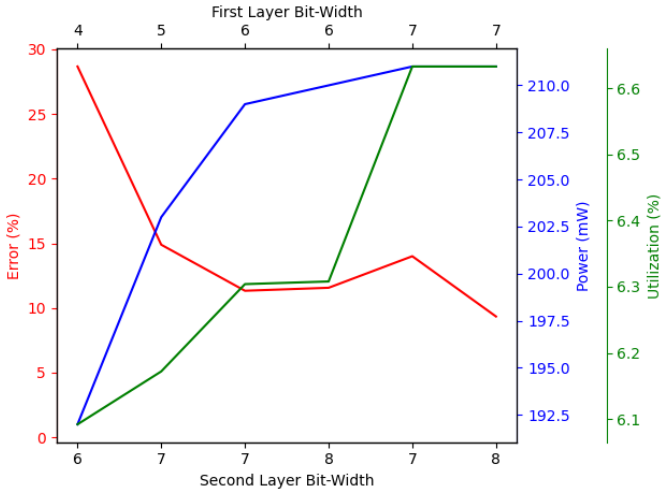


Figure 40. Power, utilization, and error for configurations with acceptable accuracy

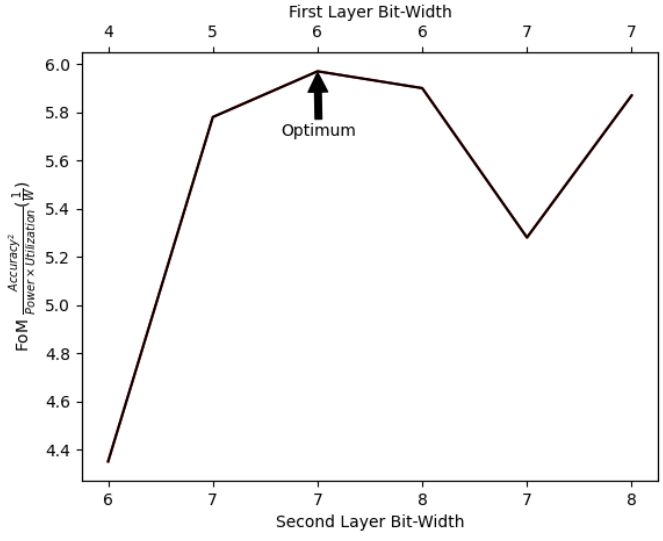


Figure 41. Figure of merit comparison of selected configurations

for implemented configurations. To select the optimal design point, a Figure of Merit $FoM = \frac{accuracy^2}{power \times utilization}$ is defined as an indicator of overall efficiency. Fig. 41 compares the FoM for all configurations. As a result, we implement and run the configuration with 6, 7, and 1 bit-width for first and second layers CONV output activations and FC weights, respectively, on the mentioned SCB. The accuracy of the selected LWCNN is 88.67%, the same as the full-precision LWCNN.

FPGA Implementation. In this part, the LWCNN implementation results are presented. We have designed a full implementation of the inference of the optimized LWCNN for the Z-turn board.

We also implemented the conventional architecture that includes memory components for activation maps as a comparison baseline. The conventional architecture finishes the processing of one layer to completion and stores its entire output activation map in memory before starting the next layer. Considering the inefficiency of utilizing off-chip memory, first, we considered utilizing on-chip memories. However, the available BRAM on the FPGA was 36 Kb while LWCNN requires 103 Kb on-chip memory (considering the number of output activations of each layer). Therefore, we designed it by the use of LUTs of the FPGA.

In our LWCNN design, we have leveraged DMA and DDR through AXI interconnect to transfer and store the input images to FPGA. The results of synthesis, implementation, and execution of the inference for one image on the board are reported in Table 6. As observed, the conventional architecture cannot fit on the FPGA. Thus, the execution time is not obtained. Whereas, the fused LWCNN is implemented on the board. DMA takes the main execution time for transferring the input image. As indicated, our Fused LWCNN requires significantly less resource utilization, namely, 204.1 times fewer LUTs, and 7.1 times fewer registers of the FPGA.

Table 6. Results of FPGA implementation

	LUT	Register	Memory	Power	Execution time
Fused LWCNN	9.2%	4.53%	1.43%	1.698 W	1.384 sec
Conventional Architecture	1878%	32.39%	1.43%	1.920 W	ND

6.4. Chapter Conclusion

With a network-hardware co-design approach, this chapter modified the network and hardware so that end-to-end fusion of a CNN became possible without storage or recalculation of intermediate activations. On the network side, we contributed by introducing a new class of CNNs, FFCNNs, that empowered fully-fused architectures in hardware. In hardware, we redesigned the last fused layer, enabling FFCNNs to grow further.

As support for our approach, we showcased a fully-fused implementation of an LWCNN. The fully-fused implementation was able to shrink the size of the hardware to fit it on our board without off-chip memory, which was impossible with the conventional architecture. We also displayed that FFCNNs are scalable. We designed an FFCNN for the CIFAR-10 classification task that achieved 78.79% accuracy with a model size suitable for edge application. This pioneering design promises more advanced FFCNNs will increase the capability of edge AI by rectifying the off-chip memory problem that eventually can be used in restricted size and weight applications such as insect-scale robotics.

7. CONCLUSION

Insect-scale robots are tiny machines that mimic the capabilities of insects, such as flying, jumping, and crawling. They have numerous applications in swarm robotics, including search and rescue, confined place inspection, and space or deep-sea exploration. Due to their small form factor and their potential low fabrication cost, they can revolutionize the field of robotics by enabling new modes of locomotion, sensing, communication, and intelligence.

Despite the potential applications, insect-scale robots are still under development. There are many areas that need more research, such as the design, fabrication, power, and control of these tiny machines. One of the promising technologies for crawling insect-scale robots is the use of IEAP-based soft actuators, which are bio-compatible, easy to manufacture, compliant with confined environments, and capable of actuation-sensing integration. However, there is not enough information and operational models of them in the literature regarding their incorporation into robots. Additionally, the control autonomy of insect-scale robots is limited by the power and area of the electronic controller, which poses significant challenges for achieving autonomous and long-range locomotion.

This thesis proposed a new method for IEAP actuator modeling. The lumped element approach modeled the IEAP actuators as an LTI system in the Laplace domain, compatible with control and circuit theory. The model has the following advantages:

- The LTI Laplace model reduces the computational cost and complexity. The computational power of insect-scale robots is limited by their restriction on payload and power.
- The model is easy to obtain for different actuators. We only need to analyze the actuator step response to acquire its actuator-specific model. IEAP actuators are slightly different from one another due to their fabrication procedure, and they are needed in large quantities in a swarm application. Therefore, the easy development of an actuator-specific model is very important for IEAP actuators in insect-scale robots.
- The model incorporates the loading effect. As the actuator in robots rarely operates unloaded, the loading effect is of utmost importance. The loading effect is also linearized and obtained empirically to ease its application.

The model of our under-test actuator shows that it is able to move a very small payload. Nonetheless, we spot robot morphologies in which the payload is carried in a wheeled loading platform, and multiple actuators drive the platform to increase the loading capacity.

In the control part, we tried to design a low-power and small convolutional neural network hardware accelerator to comply with the strict limitations of insect-scale robots. We reviewed the AIMC paradigm in the literature of other low-power applications. Studying SOA AIMC ASICs, we spotted the opportunities and challenges in reducing power consumption. Using these insights, we came up with the fully-fused architecture that fuses the network layers to each other to remove the memory usage between the layers. The fully-fused architecture is enabled by an innovation in the network end, FFCNNs. We designed two accelerators based on this architecture and network-hardware codesign.

Our first accelerator used the analog domain. The fully-fused architecture eliminated the need for energy-hungry ADCs and DACs between the layers. The design was simulated at the transistor level and reached 92% accuracy with four orders of magnitude higher efficiency. Later, we developed the second accelerator on an FPGA. We implemented two accelerators with fully-fused and conventional architectures on an FPGA to compare their performances. The results showed the superiority of our fully-fused architecture in terms of utilization and power, regardless of the implementation method. The fully-fused architecture is a result of our network-hardware codesign approach. The network-end counterpart of this architecture is FFCNNs. FFCNNs have a network structure that allows complete omission of inter-layer memory usage.

In this thesis, we covered a vast area of the multidisciplinary field of insect-scale robots. We investigated the alternative materials and concepts for their actuator design. However, we focused mostly on the on-board visual control of the miniaturized robots. The network-hardware codesign approach in this thesis exploited any opportunities on each side to optimize the system, considering the constraints and trade-offs on the other side. This approach achieved high efficiency and performance for the insect-scale robots. As the next steps, one can design the robot body according to our IEAP actuator model and integrate it with an ASIC designed with the novel FFCNN and fully-fused architecture presented in this chapter. This would pave the way for the realization of autonomous insect-scale robots with onboard intelligent visual control.

BIBLIOGRAPHY

[1] Artificial Intellgence Center. “Shakey the robot”. In: (1984).

[2] Yide Liu et al. “S2worm: A Fast-Moving Untethered Insect-Scale Robot With 2-DoF Transmission Mechanism”. In: *IEEE Robotics and Automation Letters* 7.3 (2022), pp. 6758–6765. DOI: 10.1109/LRA.2022.3176435.

[3] Yufeng Chen et al. “Hybrid aerial and aquatic locomotion in an at-scale robotic insect”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 331–338. DOI: 10.1109/IROS.2015.7353394.

[4] Robert J. Wood. “The First Takeoff of a Biologically Inspired At-Scale Robotic Insect”. In: *IEEE Transactions on Robotics* 24.2 (2008), pp. 341–347. DOI: 10.1109/TRO.2008.916997.

[5] Mario Lok et al. “A power electronics unit to drive piezoelectric actuators for flying microrobots”. In: *2015 IEEE Custom Integrated Circuits Conference (CICC)*. 2015, pp. 1–4. DOI: 10.1109/CICC.2015.7338392.

[6] Indrek Must et al. “Ionic and capacitive artificial muscle for biomimetic soft robotics”. In: *Advanced Engineering Materials* 17.1 (2015), pp. 84–94.

[7] Gordon E. Moore. “Cramming More Components Onto Integrated Circuits”. In: *Proceedings of the IEEE* 86.1 (1998), pp. 82–85.

[8] Haldun M. Ozaktas. “Levels of Abstraction in Computing Systems and Optical Interconnection Technology”. In: *Optical Interconnections and Parallel Processing: Trends at the Interface*. Ed. by Pascal Berthomé and Afonso Ferreira. Boston, MA: Springer US, 1998, pp. 1–18. ISBN: 978-1-4757-2791-3. DOI: 10.1007/978-1-4757-2791-3_1. URL: https://doi.org/10.1007/978-1-4757-2791-3_1.

[9] S.B. Niku. *Introduction to Robotics: Analysis, Control, Applications*. Wiley, 2020. ISBN: 9781119527626. URL: https://books.google.ee/books?id=rLfADwAAQBAJ.

[10] G Di Cesare et al. “How attitudes generated by humanoid robots shape human brain activity”. In: *Scientific Reports* 10.1 (2020), p. 16928.

[11] Marco Hutter et al. “Anymal-a highly mobile and dynamic quadrupedal robot”. In: *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2016, pp. 38–44.

[12] Mingfang Du. “Overview of Autonomous Vehicle”. In: *Autonomous Vehicle Technology: Global Exploration and Chinese Practice*. Springer, 2022, pp. 1–15.

[13] Neil Savage. *Keeping it simple*. 2022.

[14] Yogesh Madhavrao Chukewad. *RoboFly: Towards Autonomous Flight of a Multimodal Insect-Scale Robot*. University of Washington, 2020.

[15] Belen Solano and David Wood. “Design and testing of a polymeric microgripper for cell manipulation”. In: *Microelectronic Engineering* 84.5-8 (2007), pp. 1219–1222.

[16] Davide Falanga et al. “The Foldable Drone: A Morphing Quadrotor That Can Squeeze and Fly”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 209–216. DOI: 10.1109/LRA.2018.2885575.

[17] July 2023. URL: https://www.engineeredarts.co.uk/robot/ameca/.

[18] Mengdi Han et al. “Submillimeter-scale multimaterial terrestrial robots”. In: *Science Robotics* 7.66 (2022), eabn0602.

[19] Noah T Jafferis et al. “Untethered flight of an insect-sized flapping-wing microscale aerial vehicle”. In: *Nature* 570.7762 (2019), pp. 491–495.

[20] Vikram Iyer et al. “Wireless steerable vision for live insects and insect-scale robots”. In: *Science robotics* 5.44 (2020), eabb0839.

[21] Jihyun Ryu et al. “Paper robotics: Self-folding, gripping, and locomotion”. In: *Advanced Materials Technologies* 5.4 (2020), p. 1901054.

[22] Robert J. Wood et al. “Design, fabrication and initial results of a 2g autonomous glider”. In: *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.* IEEE, 2005, pp. 1870–1877. DOI: 10.1109/IECON.2005.1569190.

[23] Nirupam Roy. “Owlet: Insect-Scale Spatial Sensing With 3D-printed Acoustic Structures”. In: *GetMobile: Mobile Comp. and Comm.* 25.2 (Sept. 2021), pp. 14–20. ISSN: 2375-0529. DOI: 10.1145/3486880.3486884. URL: https://doi.org/10.1145/3486880.3486884.

[24] Johannes James et al. “Liftoff of a 190 mg Laser-Powered Aerial Vehicle: The Lightest Wireless Robot to Fly”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 3587–3594. DOI: 10.1109/ICRA.2018.8460582.

[25] Cameron A. Aubin et al. “Powerful, soft combustion actuators for insect-scale robots”. In: *Science* 381.6663 (2023), pp. 1212–1217. DOI: 10.1126/science.adg5067. eprint: https://www.science.org/doi/pdf/10.1126/science.adg5067. URL: https://www.science.org/doi/abs/10.1126/science.adg5067.

[26] Quan Li et al. “A 700 (Wxh)/kg Rechargeable Pouch Type Lithium Battery”. In: *Chinese Physics Letters* 40.4 (2023), p. 048201.

[27] Cameron A Aubin et al. “Powerful, soft combustion actuators for insect-scale robots”. In: *Science* 381.6663 (2023), pp. 1212–1217.

[28] Yufeng Chen et al. “Controllable water surface to underwater transition through electrowetting in a hybrid terrestrial-aquatic microrobot”. In: *Nature communications* 9.1 (2018), p. 2495.

[29] Johannes James and Sawyer Fuller. “A high-voltage power electronics unit for flying insect robots that can modulate wing thrust”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 7212–7218. DOI: 10.1109/ICRA48506.2021.9561869.

[30] Jiaming Liang et al. “Manipulating the Moving Trajectory of Insect-Scale Piezoelectric Soft Robots by Frequency”. In: *2019 IEEE 32nd International Conference on Micro Electro Mechanical Systems (MEMS)*. 2019, pp. 1041–1044. DOI: 10.1109/MEMSYS.2019.8870751.

[31] A. Maziz, A. Simaite, and C. Bergaud. “Ionic Electrochemical Actuators”. In: *Polymerized Ionic Liquids*. The Royal Society of Chemistry, Sept. 2017. ISBN: 978-1-78262-960-3. DOI: 10.1039/9781788010535-00456. URL: https://doi.org/10.1039/9781788010535-00456.

[32] Yanjie Wang and Takushi Sugino. “Ionic Polymer Actuators: Principle, Fabrication and Applications”. In: *Actuators*. Ed. by Constantin Volosencu. Rijeka: IntechOpen, 2018. Chap. 3. DOI: 10.5772/intechopen.75085. URL: https://doi.org/10.5772/intechopen.75085.

[33] Raphael Neuhaus et al. “Integrating Ionic Electroactive Polymer Actuators and Sensors Into Adaptive Building Skins – Potentials and Limitations”. In: *Frontiers in Built Environment* 6 (2020). ISSN: 2297-3362. DOI: 10.3389/fbuil.2020.00095. URL: https://www.frontiersin.org/articles/10.3389/fbuil.2020.00095.

[34] Laurent J Goujon et al. “Flexible solid polymer electrolytes based on nitrile butadiene rubber/poly (ethylene oxide) interpenetrating polymer networks containing either LiTFSI or EMITFSI”. In: *Macromolecules* 44.24 (2011), pp. 9683–9691.

[35] Iman Dadras et al. “Modeling and Experimental Analysis of the Mass Loading Effect on Micro-Ionic Polymer Actuators Using Step Response Identification”. In: *Journal of Microelectromechanical Systems* 30.2 (2021), pp. 243–252. DOI: 10.1109/JMEMS.2021.3060897.

[36] Lauréline Seurre et al. “Demonstrating Full Integration Process for Electroactive Polymer Microtransducers to Realize Soft Microchips”. In: *2020 IEEE 33rd International Conference on Micro Electro Mechanical Systems (MEMS)*. 2020, pp. 917–920. DOI: 10.1109/MEMS46641.2020.9056371.

[37] Mohsen Annabestani and Mahdi Fardmanesh. “Ionic electro active polymer-based soft actuators and their applications in microfluidic micropumps, microvalves, and micromixers: a review”. In: *arXiv preprint arXiv:1904.07149* (2019).

[38] Karl Kruusamäe et al. “Self-sensing ionic polymer actuators: a review”. In: *Actuators*. Vol. 4. 1. MDPI. 2015, pp. 17–38.

[39] Minjeong Park et al. “Fast and stable ionic electroactive polymer actuators with PEDOT: PSS/(Graphene–Ag-Nanowires) nanocomposite electrodes”. In: *Sensors* 18.9 (2018), p. 3126.

[40] Iqbal H Sarker. “Machine Learning: Algorithms, Real-World Applications and Research Directions”. In: *SN Computer Science* 2.160 (2021).

[41] Saad Al-Azawi and Tareq Abed Mohammed. “Understanding of a convolutional neural network”. In: *2017 International Conference on Engineering and Technology (ICET)*. IEEE. 2017.

[42] Keiron O’Shea and Ryan Nash. “An Introduction to Convolutional Neural Networks”. In: *arXiv preprint arXiv:1511.08458* (2015).

[43] Arjun Parthasarathy and Bhaskar Krishnamachari. “DEFER: Distributed Edge Inference for Deep Neural Networks”. In: *arXiv preprint arXiv:2201.06769* (2022).

[44] Amirali Boroumand et al. “Google Neural Network Models for Edge Devices: Analyzing and Mitigating Machine Learning Inference Bottlenecks”. In: *Proceedings of the ACM/IEEE International Symposium on Microarchitecture* (2021).

[45] Martino Dazzi et al. “Accelerating Inference of Convolutional Neural Networks Using In-memory Computing”. In: *Frontiers in Computational Neuroscience* 15 (2021).

[46] Jun-Ying Huang et al. “In-Memory Computing Architecture for a Convolutional Neural Network Based on Spin Orbit Torque MRAM”. In: *Electronics* 11 (8 2022), p. 1245.

[47] Mayank Mishra. *Convolutional Neural Networks, explained*. Sept. 2020. URL: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939.

[48] Aston Zhang et al. *Dive into Deep Learning*. https://D2L.ai. Cambridge University Press, 2023.

[49] Sumit Saha. "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way". In: *Saturn Cloud Blog* (2023). URL: https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/.

[50] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer. 2014, pp. 818–833.

[51] Alexandre Khaldi et al. "Conducting interpenetrating polymer network sized to fabricate microactuators". In: *Applied Physics Letters* 98.16 (2011).

[52] Charlotte A Cutler, Mohamed Bouguettaya, and John R Reynolds. "PEDOT polyelectrolyte based electrochromic films via electrostatic adsorption". In: *Advanced materials* 14.9 (2002), pp. 684–688.

[53] Hao He et al. "Biocompatible conductive polymers with high conductivity and high stretchability". In: *ACS applied materials & interfaces* 11.29 (2019), pp. 26185–26193.

[54] Alexandre Khaldi et al. "Patterning highly conducting conjugated polymer electrodes for soft and flexible microelectrochemical devices". In: *ACS applied materials & interfaces* 10.17 (2018), pp. 14978–14985.

[55] Elisabeth Smela. "Conjugated polymer actuators for biomedical applications". In: *Advanced materials* 15.6 (2003), pp. 481–494.

[56] Veiko Vunder, Andres Punning, and Alvo Aabloo. "Mechanical interpretation of back-relaxation of ionic electroactive polymer actuators". In: *Smart Materials and Structures* 21.11 (2012), p. 115023.

[57] Tan N Nguyen et al. "Nonlinear dynamic modeling of ultrathin conducting polymer actuators including inertial effects". In: *Smart Materials and Structures* 27.11 (2018), p. 115032.

[58] Pilsung Kang and Athip Somtham. "An Evaluation of Modern Accelerator-Based Edge Devices for Object Detection Applications". In: *Mathematics* 10.22 (2022), p. 4299.

[59] Wenbin Li and Matthieu Liewig. "A Survey of AI Accelerators for Edge Environment". In: *Trends and Innovations in Information Systems and Technologies*. Springer. 2020.

[60] Nathan Laubeuf et al. "Dynamic quantization range control for analog-in-memory neural networks acceleration". In: *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 27.5 (2022), pp. 1–21.

[61] Pouya Houshmand et al. "DIANA: An End-to-End Hybrid DIgital and ANAlog Neural Network SoC for the Edge". In: *IEEE Journal of Solid-State Circuits* 58 (1 Jan. 2023), pp. 203–215. ISSN: 0018-9200. DOI: 10.1109/JSSC.2022.3214064.

[62] Iman Dadras et al. "Modeling and Experimental Analysis of the Mass Loading Effect on Micro-Ionic Polymer Actuators Using Step Response

Identification". In: *Journal of Microelectromechanical Systems* 30.2 (2021), pp. 386–396. DOI: 10.1109/JMEMS.2021.3060897.

[63] Andrew Ng. *AI is the new electricity*. O'Reilly Media, 2018.

[64] Huifeng Zhu et al. "CMOS Image Sensor Data-Readout Method for Convolutional Operations with Processing Near Sensor Architecture". In: *2018 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS 2018* (2019), pp. 528–531. DOI: 10.1109/APCCAS.2018.8605660.

[65] Thanveer Shaik et al. "A Review of the Trends and Challenges in Adopting Natural Language Processing Methods for Education Feedback Analysis". In: *IEEE Access* 10 (2022), pp. 56720–56739. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3177752.

[66] Sebastien Herbreteau and Charles Kervrann. "DCT2net: An Interpretable Shallow CNN for Image Denoising". In: *IEEE Transactions on Image Processing* 31 (2022), pp. 4292–4305. ISSN: 1057-7149. DOI: 10.1109/TIP.2022.3181488.

[67] Lu Yang et al. "Hier R-CNN: Instance-Level Human Parts Detection and A New Benchmark". In: *IEEE Transactions on Image Processing* 30 (2021), pp. 39–54. ISSN: 1057-7149. DOI: 10.1109/TIP.2020.3029901.

[68] Xiao Zhong and David Enke. "Predicting the daily return direction of the stock market using hybrid machine learning algorithms". In: *Financial Innovation* 5 (1 Dec. 2019), p. 24. ISSN: 2199-4730. DOI: 10.1186/s40854-019-0138-0.

[69] Chen Wu et al. "Low-precision Floating-point Arithmetic for High-performance FPGA-based CNN Acceleration". In: *ACM Transactions on Reconfigurable Technology and Systems* 15 (1 Mar. 2022), pp. 1–21. ISSN: 1936-7406. DOI: 10.1145/3474597. URL: https://dl.acm.org/doi/10.1145/3474597.

[70] Zewen Li et al. "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects". In: *IEEE Transactions on Neural Networks and Learning Systems* 33 (12 Dec. 2022), pp. 6999–7019. ISSN: 2162-237X. DOI: 10.1109/TNNLS.2021.3084827.

[71] Yunxiang Hu, Yuhao Liu, and Zhuovuan Liu. "A Survey on Convolutional Neural Network Accelerators: GPU, FPGA and ASIC". In: IEEE, Jan. 2022, pp. 100–107. ISBN: 978-1-7281-7721-2. DOI: 10.1109/ICCRD54409.2022.9730377.

[72] Jongmin Jo, Sucheol Jeong, and Pilsung Kang. "Benchmarking GPU-Accelerated Edge Devices". In: IEEE, Feb. 2020, pp. 117–120. ISBN: 978-1-7281-6034-4. DOI: 10.1109/BigComp48618.2020.00-89.

[73] Mahreen Zainab et al. "FPGA Based Implementations of RNN and CNN: A Brief Analysis". In: IEEE, Nov. 2019, pp. 1–8. ISBN: 978-1-7281-4682-9. DOI: 10.1109/ICIC48496.2019.8966676.

[74] Diksha Moolchandani, Anshul Kumar, and Smruti R. Sarangi. "Accelerating CNN Inference on ASICs: A Survey". In: *Journal of Systems Architecture* 113 (Feb. 2021), p. 101887. ISSN: 13837621. DOI: 10.1016/j.sysarc.2020.101887.

[75] Marian Verhelst and Boris Murmann. "Machine Learning at the Edge". In: *NANO-CHIPS 2030: On-Chip AI for an Efficient Data-Driven World*. Ed. by Boris Murmann and Bernd Hoefflinger. Cham: Springer International

Publishing, 2020, pp. 293–322. ISBN: 978-3-030-18338-7. DOI: 10.1007/978-3-030-18338-7_18.

[76] Iman Dadras et al. "An Efficient Analog Convolutional Neural Network Hardware Accelerator Enabled by a Novel Memoryless Architecture for Insect-Sized Robots". In: IEEE, June 2022, pp. 1–6. ISBN: 978-1-6654-6717-9. DOI: 10.1109/MOCAST54814.2022.9837551.

[77] Donghyuk Kim et al. "An Overview of Processing-in-Memory Circuits for Artificial Intelligence and Machine Learning". In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 12.2 (2022), pp. 338–353. DOI: 10.1109/JETCAS.2022.3160455.

[78] Sangheon Kwon et al. "Measuring error-tolerance in SRAM architecture on hardware accelerated neural network". In: *2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. 2016, pp. 1–4. DOI: 10.1109/ICCE-Asia.2016.7804818.

[79] I. A. Papistas et al. "A 22 nm, 1540 TOP/s/W, 12.1 TOP/s/mm 2 in-Memory Analog Matrix-Vector-Multiplier for DNN Acceleration". In: IEEE, Apr. 2021, pp. 1–2. ISBN: 978-1-7281-7581-2. DOI: 10.1109/CICC51472.2021.9431575. URL: https://ieeexplore.ieee.org/document/9431575/.

[80] Kaushik Roy et al. "In-Memory Computing in Emerging Memory Technologies for Machine Learning: An Overview". In: *2020 57th ACM/IEEE Design Automation Conference (DAC)*. 2020, pp. 1–6. DOI: 10.1109/DAC18072.2020.9218505.

[81] Hong-Yu Chen et al. "Resistive Random Access Memory (RRAM) Technology: From Material, Device, Selector, 3D Integration to Bottom-Up Fabrication". In: *Resistive Switching: Oxide Materials, Mechanisms, Devices and Operations* (2022), pp. 33–64.

[82] Andrea Ehrmann et al. "Recent developments in phase-change memory". In: *Applied Research* (2022), e202200024.

[83] Payal Jangra and Manoj Duhan. "A Review on Emerging Spintronic Devices: CMOS Counterparts". In: *2022 7th International Conference on Communication and Electronics Systems (ICCES)*. 2022, pp. 90–99. DOI: 10.1109/ICCES54183.2022.9835778.

[84] Shubham Jain and Anand Raghunathan. "CxDNN: Hardware-Software Compensation Methods for Deep Neural Networks on Resistive Crossbar Systems". In: *ACM Trans. Embed. Comput. Syst.* 18.6 (Nov. 2019). ISSN: 1539-9087. DOI: 10.1145/3362035. URL: https://doi.org/10.1145/3362035.

[85] Chengshuo Yu et al. "A 65-nm 8T SRAM Compute-in-Memory Macro With Column ADCs for Processing Neural Networks". In: *IEEE Journal of Solid-State Circuits* 57.11 (2022), pp. 3466–3476. DOI: 10.1109/JSSC.2022.3162602.

[86] Bo Zhang et al. "A 177 TOPS/W, Capacitor-based In-Memory Computing SRAM Macro with Stepwise-Charging/Discharging DACs and Sparsity-Optimized Bitcells for 4-Bit Deep Convolutional Neural Networks". In: *2022 IEEE Custom Integrated Circuits Conference (CICC)*. 2022, pp. 1–2. DOI: 10.1109/CICC53496.2022.9772781.

[87] Edward Choi et al. “A 133.6 TOPS/W compute-in-memory SRAM macro with fully parallel one-step multi-bit computation”. In: *2022 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE. 2022, pp. 1–2.

[88] Ping-Chun Wu et al. “A 28nm 1Mb time-domain computing-in-memory 6T-SRAM macro with a 6.6 ns latency, 1241GOPS and 37.01 TOPS/W for 8b-MAC operations for edge-AI devices”. In: *2022 IEEE International Solid-State Circuits Conference (ISSCC)*. Vol. 65. IEEE. 2022, pp. 1–3.

[89] Samuel Spetalnick and Arijit Raychowdhury. “A Practical Design-Space Analysis of Compute-in-Memory With SRAM”. In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 69.4 (2022), pp. 1466–1479. DOI: 10.1109/TCSI.2021.3138057.

[90] Joshua Klein et al. “ALPINE: Analog In-Memory Acceleration with Tight Processor Integration for Deep Learning”. In: *IEEE Transactions on Computers* (2022), pp. 1–14. DOI: 10.1109/TC.2022.3230285.

[91] Asghar Gholami et al. “A Survey of Quantization Methods for Efficient Neural Network Inference”. In: Jan. 2022, pp. 291–326. ISBN: 9781003162810. DOI: 10.1201/9781003162810-13.

[92] Jungwook Choi et al. “Pact: Parameterized clipping activation for quantized neural networks”. In: *arXiv preprint arXiv:1805.06085* (2018).

[93] Steven K Esser et al. “Learned Step Size Quantization”. In: *International Conference on Learning Representations*. 2020.

[94] Ayon Basumallik et al. “Adaptive Block Floating-Point for Analog Deep Learning Hardware”. In: *arXiv preprint arXiv:2205.06287* (2022).

[95] Jeffrey L. McKinstry et al. “Discovering Low-Precision Networks Close to Full-Precision Networks for Efficient Inference”. In: *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*. 2019, pp. 6–9. DOI: 10.1109/EMC2-NIPS53020.2019.00009.

[96] Szymon Migacz. “NVIDIA 8-bit inference width TensorRT”. In: *GPU Technology Conference*. Vol. 10. 2017.

[97] Chris Yakopcic, Md Zahangir Alom, and Tarek M Taha. “Extremely parallel memristor crossbar architecture for convolutional neural network implementation”. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 1696–1703.

[98] Peng Yao et al. “Fully hardware-implemented memristor convolutional neural network”. In: *Nature* 577.7792 (2020), pp. 641–646.

[99] Masatoshi Yamaguchi et al. “An energy-efficient time-domain analog CMOS BinaryConnect neural network processor based on a pulse-width modulation approach”. In: *IEEE Access* 9 (2020), pp. 2644–2654.

[100] Sung-Tae Lee and Jong-Ho Lee. “Neuromorphic computing using NAND flash memory architecture with pulse width modulation scheme”. In: *Frontiers in Neuroscience* 14 (2020), p. 571292.

[101] J. Lienig and J. Scheible. *Fundamentals of Layout Design for Electronic Circuits*. Springer International Publishing, 2020. ISBN: 9783030392833. URL: https://books.google.ee/books?id=qICgzAEACAAJ.

[102] Matthias Vermeer. “Interface trap density extraction from the subthreshold slope of FDSOI devices”. University of Twente, May 2019.

[103] Vikram Jain et al. "TinyVers: A Tiny Versatile System-on-Chip With State-Retentive eMRAM for ML Inference at the Extreme Edge". In: *IEEE Journal of Solid-State Circuits* (2023).

[104] Larry Alper, Kimberly M. Williams, and David N. Hyerle. *Developing Connective Leadership Successes With Thinking Maps*. Association for Supervision & Curriculum Development, 2000. ISBN: 0-87120-367-7.

[105] Xiaoying Sun et al. "A Review of Robot Control with Visual Servoing". In: *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2018, pp. 116–121. DOI: 10.1109/CYBER.2018.8688060.

[106] Dong Yu and Li Deng. "Deep Learning and Its Applications to Signal and Information Processing [Exploratory DSP]". In: *IEEE Signal Processing Magazine* 28.1 (2011), pp. 14–30. DOI: 10.1109/MSP.2010.939038.

[107] Vivienne Sze et al. "Efficient Processing of Deep Neural Networks: A Tutorial and Survey". In: *Proceedings of the IEEE* 105.12 (2017), pp. 2295–2329. DOI: 10.1109/JPROC.2017.2761740.

[108] Xinyu Dai et al. "Automatic obstacle avoidance of quadrotor UAV via CNN-based learning". In: *Neurocomputing* 402 (2020), pp. 346–358. DOI: 10.1016/j.neucom.2020.04.020.

[109] Yuhua Jiao et al. "Detection and Localization of Overlapped Fruits Application in an Apple Harvesting Robot". In: *Electronics* 9.6 (2020). ISSN: 2079-9292. DOI: 10.3390/electronics9061023. URL: https://www.mdpi.com/2079-9292/9/6/1023.

[110] Octavio Antonio Villarreal Magana et al. "Fast and Continuous Foothold Adaptation for Dynamic Locomotion Through CNNs". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 2140–2147. DOI: 10.1109/LRA.2019.2899434.

[111] Yeon Ji Choi et al. "Improved CNN-Based Path Planning for Stairs Climbing in Autonomous UAV with LiDAR Sensor". In: *2021 International Conference on Electronics, Information, and Communication (ICEIC)*. 2021, pp. 1–7. DOI: 10.1109/ICEIC51217.2021.9369805.

[112] S. Balasubramanian et al. "An Insect-Sized Robot That Uses a Custom-Built Onboard Camera and a Neural Network to Classify and Respond to Visual Input". In: (2018), pp. 1297–1302. DOI: 10.1109/BIOROB.2018.8488007.

[113] Sawyer B. Fuller. "Four Wings: An Insect-Sized Aerial Robot With Steering Ability and Payload Capacity for Autonomy". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 570–577. DOI: 10.1109/LRA.2019.2891086.

[114] Sparsh Mittal. "A survey of FPGA-based accelerators for convolutional neural networks". In: *Neural Computing and Applications* 32.4 (2020), pp. 1109–1139. DOI: 10.1007/s00521-018-3761-1.

[115] Andrea Di Mauro et al. "Always-On 674 μW @4GOP/s Error Resilient Binary Neural Networks With Aggressive SRAM Voltage Scaling on a 22-nm IoT End-Node". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 67.11 (2020), pp. 3905–3918. DOI: 10.1109/TCSI.2020.3012576.

[116] Xiangyu Zhang et al. "A multi-chip system optimized for insect-scale flapping-wing robots". In: *2015 IEEE Symposium on VLSI Circuits (VLSI Circuits)*. IEEE, 2015, pp. 1–4. DOI: 10.1109/VLSIC.2015.7231246.

[117] Jin-Hwan Kim et al. "An Ultra-Low-Power Analog-Digital Hybrid CNN Face Recognition Processor Integrated with a CIS for Always-on Mobile Devices". In: *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5. DOI: 10.1109/ISCAS.2019.8702698.

[118] Ti-Hao Hsu et al. "A 0.5-V Real-Time Computational CMOS Image Sensor With Programmable Kernel for Feature Extraction". In: *IEEE Journal of Solid-State Circuits* 56.5 (2021), pp. 1588–1596. DOI: 10.1109/JSSC.2020.3034192.

[119] Ziyang Li et al. "A 5.9μW Ultra-Low-Power Dual-Resolution CIS Chip of Sensing-with-Computing for Always-on Intelligent Visual Devices". In: *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2021, pp. 1–5. DOI: 10.1109/ISCAS51556.2021.9401338.

[120] Jongwook Choi et al. "Design of an always-on image sensor using an analog lightweight convolutional neural network". In: *Sensors (Switzerland)* 20.11 (2020), pp. 1–14. DOI: 10.3390/s20113101.

[121] Benjamin Rumberg et al. "Continuous-Time Programming of Floating-Gate Transistors for Nonvolatile Analog Memory Arrays". In: *Journal of Low Power Electronics and Applications* 11.1 (2021), p. 4. DOI: 10.3390/jlpea11010004.

[122] Arman Shafiee and Naveen Pedram. "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars". In: *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 2016, pp. 14–26. DOI: 10.1109/ISCA.2016.12.

[123] Haitao Tsai et al. "Recent progress in analog memory-based accelerators for deep learning". In: *Journal of Physics D: Applied Physics* 51.28 (2018), p. 283001. DOI: 10.1088/1361-6463/aac8a5.

[124] Jason Reuben. "Binary Addition in Resistance Switching Memory Array by Sensing Majority". In: *Micromachines* 11.5 (2020), p. 496. DOI: 10.3390/mi11050496.

[125] M. Soleimani et al. "Design of high-speed high-precision voltage-mode MAX-MIN circuits with low area and low power consumption". In: *2009 European Conference on Circuit Theory and Design*. IEEE, 2009, pp. 351–354. DOI: 10.1109/ECCTD.2009.5274998.

[126] Gary B. Huang et al. *Labeled faces in the wild: A database for studying face recognition in unconstrained environments*. Tech. rep. Technical Report 07-49. University of Massachusetts, Amherst, 2007.

[127] chetankv. *Dogs & Cats Images*. Kaggle, 2021. URL: https://www.kaggle.com/datasets/chetankv/dogs-cats-images.

[128] Udayanga De Silva et al. "RF-Rate Hybrid CNN Accelerator Based on Analog-CMOS and Xilinx RFSoC". In: *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–5. DOI: 10.1109/ISCAS45731.2020.9180556.

[129] Iman Dadras et al. "Fully-Fusible Convolutional Neural Networks for End-to-End Fused Architecture with FPGA Implementation". In: *2023*

30th IEEE International Conference on Electronics, Circuits and Systems (ICECS). 2023, pp. 1–5. DOI: 10.1109/ICECS58634.2023.10382831.

[130] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. *Learning multiple layers of features from tiny images*. Tech. rep. University of Toronto, 2009. URL: http://www.cs.toronto.edu/~kriz/cifar.html.

[131] Matthew Browne, Saeed Shiry Ghidary, and Norbert Michael Mayer. "Convolutional neural networks for image processing with applications in mobile robotics". In: *Speech, Audio, Image and Biomedical Signal Processing using Neural Networks* (2008), pp. 327–349.

[132] Kai Zhong, Zhao Song, and Inderjit S Dhillon. "Learning non-overlapping convolutional neural networks with multiple kernels". In: *arXiv preprint arXiv:1711.03440* (2017).

[133] Shuai Zhang et al. "Guaranteed convergence of training convolutional neural networks via accelerated gradient descent". In: *2020 54th Annual Conference on Information Sciences and Systems (CISS)*. IEEE. 2020, pp. 1–6.

[134] Gary B Huang et al. "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments". In: *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*. 2008.

[135] Jaihyuk Choi et al. "Design of an always-on image sensor using an analog lightweight convolutional neural network". In: *Sensors* 20.11 (2020), p. 3101.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support of many. I would like to express my deepest gratitude to:

- **My supervisors and mentors:** Jaan Raik (Taltech), Alvo Aabloo (Tartu Ülikool), Arindam Mallik and Debjyoti Bhattacharjee (IMEC), Sébastien Grondel, Éric Cattan, and Sofiane Ghenna (Univ. Polytechnique Hauts-de-France), for their invaluable guidance and mentorship.
- **My co-authors:** Mohammad Hasan Ahmadilivani (Taltech), Giuseppe M Sarda and Nathan Laubeuf (IMEC), and Sakineh Seydi (University of Tehran), for their collaboration and insights.
- **EIT Doctoral School Program**, for the scholarship, internship opportunities, and enriching winter/summer school programs. Special thanks to the organizers, Lucia Ramundo, Henri Manners, Viltarė Platzner, Martin Mareš, and Gil Manuel Gonçalves and my peers Artur, Marek, Maria, Charlos, and Alex, for their support and camaraderie.
- **The CSA team at IMEC**, for their scientific and emotional support during my internship.
- **Micro Analog System OY**, including Juha Majakulma and Illar Truumure, for their faith in me during challenging times.
- **The Iranian Community in Tartu**, for providing a sense of family and comfort during times of homesickness. Special thanks to Mozhgan, Kaveh, Bamdad, Atieh, Bahman, Mahtab, Elyad, Zahra, Rafieh, Houman, Mehrnoosh, Javad, Atefeh, Lona, Ebi, Eloy, Maryam, Nima, Sepideh, Yashar, Paria, Mona, Siim, and Novin.
- **My dear friends Faezeh, Ali, Shahla, and Karim**, for their unwavering support and kindness during the most difficult times of my life.
- **My family**, for their understanding and support, even when I could not be physically present with them during tough times due to my studies.
- **My beloved wife**, for her constant love, support, and companionship throughout these five years.

SUMMARY IN ESTONIAN

Madala võimsusega närvivõrgupõhised juhtimis- ja aktiveerimislahendused putukamõõtmeliste robotite jaoks

Putukasuurustel robotitel on tohutu potentsiaal rakendusteks mitmetes valdkondades, nagu raskesti ligipääsetavate kohtade kaugkontrolliks, keskkonnaseire andmete kogumiseks, otsingu- ja päästetöödeks loodusõnnetuste järel ning kosmose ja süvameresondluseks. Nende väiksus nõuab minimaalset materjaliresurssi, muutes need kulutõhusaks ja lihtsaks tootmiseks ja kasutamiseks parvedes. Seega sobivad nad ideaalselt ülesanneteks, mis nõuavad autonoomse tegutsemise korral suure ala katmist.

Siiski on putukasuuruste robotite võimsus- ja juhtimisautonoomia saavutamine endiselt keeruline ülesanne. Lendavate minirobotite poolt kasutatavatel piesoelektrilistel ajurmehhanismidel on vaja märkimisväärset energiakogust, mis ületab putukasuuruste robotite minimaalse kandevõime. Lisaks tarbivad juhtimisautonoomiaks vajalikud elektroonikaseadmed mitte ainult märkimisväärselt energiat, vaid suurendavad ka roboti kaalu. Seetõttu on putukasuurused robotid varustatud toite- ja juhtimisallikaga, mis piirab oluliselt nende tegevusväljundit.

Käesolev väitekiri pakub uuendusliku viisi, kuidas võimaldada putukasuurustel robotitel saavutada võimsusautonoomia, kasutades alternatiivseid ajuritüüpe ja vähem energiat tarbivaid uusi arvutuskiipe. Ioonilistel elektroaktiivsetel polümeeridel põhinevad pehmed ajurid on putukasuuruste robotite liikumise jaoks teostatav väikese energiatarbega alternatiiv, mis aitab vähendada energiatarvet, lubades robotil lendamise asemel aeglaselt roomata. Mudeldame neid ajureid roboti perspektiivist Laplace'i domeenis. Samuti kasutame hübriidset võrgu-riistvara kaasprojekteerimise lähenemist, et projekteerida uusi rakendusspetsiifilisi integraalskeeme konvolutsioonilise närvivõrgu kiirendamiseks, võimaldades miniatuursel robotil teostada visuaalset juhtimist. Need disainid hõlmavad uusi riistvararhitektuure, võrgustruktuure ja arvutuslikke tehnikaid, mis aitavad vähendada energiatarvet. Üks käesolevas väitekirjas esitatud kiirenditest sooritas klassifitseerimise energiatarbega vähem kui 1,5 nW pildi kohta. Käesoleva väitekirja tulemused sillutavad teed tulevasele uurimistööle ja arendusele autonoomsete putukasuuruste robotite alal.

PUBLICATIONS

CURRICULUM VITAE

Personal data

Name:	Iman Dadras
Date of Birth:	07.Nov.1990
Place of Birth:	Tehran
Nationality:	Iran

Education

2019–2024	Ph.D._The University of Tartu
2014–2017	MSC_Shahid Rajayi Techer Training University
2009–2014	BSC_Shahid Rajayi Techer Training University

Employment

2023–Ongoing	Analog designer at Analoogdesaini AS, Tallinn
2022–2023	Intern at IMEC, Leuven, Belgium
2014–2018	Technical Inspector at Fahameh Inc.

Scientific work

Main fields of interest:

- Analog and mixed-signal integrated circuit design
- Low noise and low power amplifiers
- Artificial Intelligence

ELULOOKIRJELDUS

Isikuandmed

Nimi:	Iman Dadras
Sünniaeg:	07.Nov.1990
Sünniaeg:	Tehran
Kodakondsus:	Iran

Haridus

2019–2024	Doktorikraad_Tartu Ülikool
2014–2017	Tehnikateaduse magister_Shahid Rajayi Techer Training University
2009–2014	Tehnikateaduse bakalaureus_Shahid Rajayi Techer Training University

Teenistuskäik

2023–jätkuv	Analoogdisainer ettevõttes Analoogdesaini AS, Tallinn
2022–2023	Praktikant IMECis, Leuvenis, Belgias
2014–2018	Tehniline inspektor ettevõttes Fahameh AS.

Teadustegevus

Peamised uurimisvaldkonnad:

- Analoog- ja segasignaalide integreeritud vooluahela disain
- Vähese müraga ja väikese võimsusega võimendid
- Tehisintelligents

DISSERTATIONES TECHNOLOGIAE UNIVERSITATIS TARTUENSIS

1. **Imre Mäger**. Characterization of cell-penetrating peptides: Assessment of cellular internalization kinetics, mechanisms and bioactivity. Tartu 2011, 132 p.
2. **Taavi Lehto**. Delivery of nucleic acids by cell-penetrating peptides: application in modulation of gene expression. Tartu 2011, 155 p.
3. **Hannes Luidalepp**. Studies on the antibiotic susceptibility of *Escherichia coli*. Tartu 2012, 111 p.
4. **Vahur Zadin**. Modelling the 3D-microbattery. Tartu 2012, 149 p.
5. **Janno Torop**. Carbide-derived carbon-based electromechanical actuators. Tartu 2012, 113 p.
6. **Julia Suhorutšenko**. Cell-penetrating peptides: cytotoxicity, immunogenicity and application for tumor targeting. Tartu 2012, 139 p.
7. **Viktoryia Shyp**. G nucleotide regulation of translational GTPases and the stringent response factor RelA. Tartu 2012, 105 p.
8. **Mardo Kõivomägi**. Studies on the substrate specificity and multisite phosphorylation mechanisms of cyclin-dependent kinase Cdk1 in *Saccharomyces cerevisiae*. Tartu, 2013, 157 p.
9. **Liis Karo-Astover**. Studies on the Semliki Forest virus replicase protein nsP1. Tartu, 2013, 113 p.
10. **Piret Arukuusk**. NickFects–novel cell-penetrating peptides. Design and uptake mechanism. Tartu, 2013, 124 p.
11. **Piret Villo**. Synthesis of acetogenin analogues. Asymmetric transfer hydrogenation coupled with dynamic kinetic resolution of α-amido-β-keto esters. Tartu, 2013, 151 p.
12. **Villu Kasari**. Bacterial toxin-antitoxin systems: transcriptional cross-activation and characterization of a novel *mqsRA* system. Tartu, 2013, 108 p.
13. **Margus Varjak**. Functional analysis of viral and host components of alphavirus replicase complexes. Tartu, 2013, 151 p.
14. **Liane Viru**. Development and analysis of novel alphavirus-based multifunctional gene therapy and expression systems. Tartu, 2013, 113 p.
15. **Kent Langel**. Cell-penetrating peptide mechanism studies: from peptides to cargo delivery. Tartu, 2014, 115 p.
16. **Rauno Temmer**. Electrochemistry and novel applications of chemically synthesized conductive polymer electrodes. Tartu, 2014, 206 p.
17. **Indrek Must**. Ionic and capacitive electroactive laminates with carbonaceous electrodes as sensors and energy harvesters. Tartu, 2014, 133 p.
18. **Veiko Voolaid**. Aquatic environment: primary reservoir, link, or sink of antibiotic resistance? Tartu, 2014, 79 p.
19. **Kristiina Laanemets**. The role of SLAC1 anion channel and its upstream regulators in stomatal opening and closure of *Arabidopsis thaliana.* Tartu, 2015, 115 p.

20. **Kalle Pärn**. Studies on inducible alphavirus-based antitumour strategy mediated by site-specific delivery with activatable cell-penetrating peptides. Tartu, 2015, 139 p.
21. **Anastasia Selyutina**. When biologist meets chemist: a search for HIV-1 inhibitors. Tartu, 2015, 172 p.
22. **Sirle Saul**. Towards understanding the neurovirulence of Semliki Forest virus. Tartu, 2015, 136 p.
23. **Marit Orav**. Study of the initial amplification of the human papillomavirus genome. Tartu, 2015, 132 p.
24. **Tormi Reinson**. Studies on the Genome Replication of Human Papillomaviruses. Tartu, 2016, 110 p.
25. **Mart Ustav Jr**. Molecular Studies of HPV-18 Genome Segregation and Stable Replication. Tartu, 2016, 152 p.
26. **Margit Mutso**. Different Approaches to Counteracting Hepatitis C Virus and Chikungunya Virus Infections. Tartu, 2016, 184 p.
27. **Jelizaveta Geimanen**. Study of the Papillomavirus Genome Replication and Segregation. Tartu, 2016, 168 p.
28. **Mart Toots**. Novel Means to Target Human Papillomavirus Infection. Tartu, 2016, 173 p.
29. **Kadi-Liis Veiman**. Development of cell-penetrating peptides for gene delivery: from transfection in cell cultures to induction of gene expression *in vivo*. Tartu, 2016, 136 p.
30. **Ly Pärnaste**. How, why, what and where: Mechanisms behind CPP/cargo nanocomplexes. Tartu, 2016, 147 p.
31. **Age Utt**. Role of alphavirus replicase in viral RNA synthesis, virus-induced cytotoxicity and recognition of viral infections in host cells. Tartu, 2016, 183 p.
32. **Veiko Vunder**. Modeling and characterization of back-relaxation of ionic electroactive polymer actuators. Tartu, 2016, 154 p.
33. **Piia Kivipõld**. Studies on the Role of Papillomavirus E2 Proteins in Virus DNA Replication. Tartu, 2016, 118 p.
34. **Liina Jakobson**. The roles of abscisic acid, CO_2, and the cuticle in the regulation of plant transpiration. Tartu, 2017, 162 p.
35. **Helen Isok-Paas**. Viral-host interactions in the life cycle of human papillomaviruses. Tartu, 2017, 158 p.
36. **Hanna Hõrak**. Identification of key regulators of stomatal CO_2 signalling via O_3-sensitivity. Tartu, 2017, 260 p.
37. **Jekaterina Jevtuševskaja**. Application of isothermal amplification methods for detection of *Chlamydia trachomatis* directly from biological samples. Tartu, 2017, 96 p.
38. **Ülar Allas**. Ribosome-targeting antibiotics and mechanisms of antibiotic resistance. Tartu, 2017, 152 p.
39. **Anton Paier**. Ribosome Degradation in Living Bacteria. Tartu, 2017, 108 p.
40. **Vallo Varik**. Stringent Response in Bacterial Growth and Survival. Tartu, 2017, 101 p.

41. **Pavel Kudrin**. In search for the inhibitors of *Escherichia coli* stringent response factor RelA. Tartu, 2017, 138 p.
42. **Liisi Henno**. Study of the human papillomavirus genome replication and oligomer generation. Tartu, 2017, 144 p.
43. **Katrin Krõlov**. Nucleic acid amplification from crude clinical samples exemplified by *Chlamydia trachomatis* detection in urine. Tartu, 2018, 118 p.
44. **Eve Sankovski**. Studies on papillomavirus transcription and regulatory protein E2. Tartu, 2018, 113 p.
45. **Morteza Daneshmand**. Realistic 3D Virtual Fitting Room. Tartu, 2018, 233 p.
46. **Fatemeh Noroozi**. Multimodal Emotion Recognition Based Human-Robot Interaction Enhancement. Tartu, 2018, 113 p.
47. **Krista Freimann**. Design of peptide-based vector for nucleic acid delivery in vivo. Tartu, 2018, 103 p.
48. **Rainis Venta**. Studies on signal processing by multisite phosphorylation pathways of the *S. cerevisiae* cyclin-dependent kinase inhibitor Sic1. Tartu, 2018, 155 p.
49. **Inga Põldsalu**. Soft actuators with ink-jet printed electrodes. Tartu, 2018, 85 p.
50. **Kadri Künnapuu**. Modification of the cell-penetrating peptide PepFect14 for targeted tumor gene delivery and reduced toxicity. Tartu, 2018, 114 p.
51. **Toomas Mets**. RNA fragmentation by MazF and MqsR toxins of *Escherichia coli.* Tartu, 2019, 119 p.
52. **Kadri Tõldsepp**. The role of mitogen-activated protein kinases MPK4 and MPK12 in CO_2-induced stomatal movements. Tartu, 2019, 259 p.
53. **Pirko Jalakas**. Unravelling signalling pathways contributing to stomatal conductance and responsiveness. Tartu, 2019, 120 p.
54. **S. Sunjai Nakshatharan**. Electromechanical modelling and control of ionic electroactive polymer actuators. Tartu, 2019, 165 p.
55. **Eva-Maria Tombak**. Molecular studies of the initial amplification of the oncogenic human papillomavirus and closely related nonhuman primate papillomavirus genomes. Tartu, 2019, 150 p.
56. **Meeri Visnapuu**. Design and physico-chemical characterization of metal-containing nanoparticles for antimicrobial coatings. Tartu, 2019, 138 p.
57. **Jelena Beljantseva**. Small fine-tuners of the bacterial stringent response – a glimpse into the working principles of Small Alarmone Synthetases. Tartu, 2020, 104 p.
58. **Egon Urgard**. Potential therapeutic approaches for modulation of inflammatory response pathways. Tartu, 2020, 120 p.
59. **Sofia Raquel Alves Oliveira**. HPLC analysis of bacterial alarmone nucleotide (p)ppGpp and its toxic analogue ppApp. Tartu, 2020, 122 p.
60. **Mihkel Örd**. Ordering the phosphorylation of cyclin-dependent kinase Cdk1 substrates in the cell cycle. Tartu, 2021, 228 p.
61. **Fred Elhi**. Biocompatible ionic electromechanically active polymer actuator based on biopolymers and non-toxic ionic liquids. Tartu, 2021, 140 p.

62. **Liisi Talas**. Reconstructing paleo-diversity, dynamics and response of eukaryotes to environmental change over the Late-Glacial and Holocene period in lake Lielais Svētiņu using sedaDNA. Tartu, 2021, 118 p.
63. **Livia Matt**. Novel isosorbide-based polymers. Tartu, 2021, 118 p.
64. **Koit Aasumets**. The dynamics of human mitochondrial nucleoids within the mitochondrial network. Tartu, 2021, 104 p.
65. **Faiza Summer**. Development and optimization of flow electrode capacitor technology. Tartu, 2022, 109 p.
66. **Olavi Reinsalu**. Cancer-testis antigen MAGE-A4 is incorporated into extracellular vesiclesand is exposed to the surface. Tartu, 2022, 130 p.
67. **Tetiana Brodiazhenko**. RelA-SpoT Homolog enzymes as effectors of Toxin-Antitoxin systems. Tartu, 2022, 132 p.
68. **Georg-Marten Lanno**. Development of novel antibacterial drug delivery systems as wound scaffolds using electrospinning technology. Tartu, 2022, 175 p.
69. **Liubov Cherkashchenko**. New insights into alphaviral nsP2 functions. Tartu, 2023, 171 p.
70. **Kristina Kiisholts**. Peptide-based drug carriers and preclinical nanomedicine applications for endometriosis treatment. Tartu, 2023, 138 p.
71. **Kai Rausalu**. Alphaviral nsP2 protease: From requirements for functionality to inhibition. Tartu, 2023, 175 p.
72. **Laura Sandra Lello**. Unraveling the intricate nature of the alphavirus RNA replicase. Tartu, 2023, 219 p.
73. **Houman Masnavi**. Visibility Aware Navigation. Tartu, 2023, 180 p.
74. **Kadir Aktas**. Cosmic Ray Tomography based Object Reconstruction and Recognition. Tartu, 2023, 104 p.
75. **Egils Avots**. Brain abnormality detection using statistical analysis of individual structural connectivity networks and EEG signals. Tartu, 2023, 223 p.
76. **Sainan Wang**. Structure-guided insights into the functions of CHIKV nsP2. Tartu, 2024, 154 p.
77. **Anneli Samel**. Unveiling the characteristics of cancer-testis antigen MAGEA10. Tartu, 2024, 136 p.
78. **Ikechukwu Ofodile**. Fault tolerant attitude control for nanosatellites: ESTCube-2 case. Tartu, 2024, 130 p.
79. **Olena Zamora**. Impacts of plant hormones on controlling stomatal conductance. Tartu, 2024, 166 p.
80. **Mariliis Hinnu**. *In vitro* methods for studying the mechanisms of ribosome-targeting antibiotics. Tartu, 2024, 143 p.
81. **Chung-Yueh Yeh**. Characterization of MPK and HT1 kinases in CO_2-induced stomatal movements. Tartu, 2024, 118 p.