

UCLA

UCLA Electronic Theses and Dissertations

Title

Synaptic Resistor Networks for Intelligent Systems with Real-Time Learning

Permalink

<https://escholarship.org/uc/item/7p15b6nw>

Author

Tudor, Andrew William

Publication Date

2017

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Synaptic Resistor Networks for Intelligent Systems with Real-Time Learning

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy
in Aerospace Engineering

by

Andrew William Tudor

2017

© Copyright by
Andrew William Tudor
2017

ABSTRACT OF THE DISSERTATION

Synaptic Resistor Networks for Intelligent Systems with Real-Time Learning

by

Andrew William Tudor

Doctor of Philosophy in Aerospace Engineering

University of California, Los Angeles, 2017

Professor Yong Chen, Chair

The human brain is a neural network capable of logically processing massively parallel information while simultaneously improving its performance via learning in dynamic environments. The parallel processing and learning functions are integrated in each synapse, the junction between neurons, but missed by electronic devices. Synaptic resistors, having parallel processing and learning functions, are arranged into a synaptic resistor network to improve an autonomous dynamical system's performance without any *a priori* knowledge, and unassisted by additional computing and memory circuits. In this way, the system can exhibit intelligent behavior in a similar way to humans. In comparison with humans learning the same task, the artificially intelligent network exhibited a similar, but slightly superior, strategy, speed, and

accuracy, in the simplified, but unknown environment. Synaptic resistor circuits could be scaled up to efficiently process and learn from massively parallel information in dynamic and unknown environments with intelligence comparable to human brains.

The dissertation of Andrew William Tudor is approved.

Tetsuya Iwasaki

Wentai Liu

Jason L Speyer

Yong Chen, Committee Chair

University of California, Los Angeles

2017

*This dissertation is dedicated to my parents,
whose never-ending love and support encouraged me
to follow my dreams.*

TABLE OF CONTENTS

LIST OF FIGURES	viii
LIST OF TABLES	xi
ACKNOWLEDGMENT.....	xii
VITA.....	xiii
PUBLICATIONS.....	xiv
1.0 Introduction.....	1
1.1 Characteristic Features and Structures of the Brain.....	1
1.2 Biological Synaptic Operations	3
1.3 Biological Style Computation in Conventional Computers.....	5
1.4 Turing Machine, Curse of Dimensionality, von Neumann Bottleneck	6
1.5 Research Goals.....	9
2.0 The Synstor, A Synaptic Transistor	10
2.1 Synstor Structure Summary	10
2.2 Synstor Operation Mechanism.....	13
2.4 Quantifiable Properties and Benchmarking	17
2.4.1 Multi-Voltage Testing.....	19
2.4.2 Multi-Pulse Testing.....	21
3.0 Spike Neuromorphic Intelligent Circuit (SNIC).....	22
3.1 Experimental Details & Set-up	22

3.1.1 Experiment Schematic	23
3.1.2 Vehicle Interface I/O.....	27
3.1.3 Pogo-Pin Adapter.....	28
3.1.4 Device Matching.....	30
3.2 Non-Linear Soma Circuit.....	33
3.3 Methods and Hardware Summary	35
3.3 Autonomous Air-Vehicle Parallel Learning and Processing Demonstration	37
4.0 System Modeling and Learning Analysis	45
4.1 Network & System Model Overview	46
4.2 Network & System Model Simplifications.....	48
4.3 Experimental Modeling Agreement.....	52
4.4 Learning Effect & System Improvement.....	54
4.4 Performance Benchmarking.....	58
4.4.1 Learning Accuracy.....	59
4.4.2 Total Learning Improvement	60
4.4.3 Learning Speed	62
4.4.4 Learning Oscillation.....	63
5.0 Conclusion	66
REFERENCES	68

LIST OF FIGURES

Figure 1: Structure and input/output relationship of a biological neuron. In humans there is a 10^3 -input dimension per neuron.	2
Figure 2: Synaptic modification due to STDP. The top pulses represent the time difference between pre-synaptic pulse at $t=0$, and post-synaptic pulses at $t=\Delta t$ [27].	5
Figure 3: Turing machine schematic. Logic and memory communicate sequentially on an infinite memory tape.	7
Figure 4: von Neumann architecture schematic. The von Neumann bottleneck is the transfer of information between the memory unit and the central processing unit.	8
Figure 5: Synstor device structure and two-terminal operation schematic.	10
Figure 6: Atomic force microscopy image of a random, sparse, and 99.9% semiconducting CNT network in the synstor channel area.	11
Figure 7: Effect of CNT density on tuning normalized tuning amplitude. Sparse CNT networks produce the largest tuning amplitude.	12
Figure 8: Synstor device in processing mode. The conductance, w , is read by applying a negative voltage on the input, x , electrode.	13
Figure 9: Schottky barrier formation at the CNT/Al contact location produces a rectifying property at low voltages.	14
Figure 10: Synstor device in learning mode modifies the charges stored in the TiO_2 layer when input, x , and learning, z , voltage intersect spatiotemporally.	16
Figure 11: Multi-voltage synstor device testing shows a strong non-linear property under coincident pulsing.	20

Figure 12: Multi-pulse synstor device testing shows a strong non-linear property under coincident pulsing.	21
Figure 13: Experiment schematic	24
Figure 14: Analog pixel sensing, P, information is converted to a pulsing frequency, F, in the I/O interface.....	28
Figure 15: 20x20 synstor crossbar network.	29
Figure 16: Pogo-pin adapter model.....	30
Figure 17: Histogram of device current.	31
Figure 18: Uniformity optimized population.....	32
Figure 19: Optimized device matching based on optimizing tuning overlap of grouped devices.	33
Figure 20: ‘Integrate and Fire’ CMOS circuit implementation	34
Figure 21: Integrate and fire neuron transfer character	35
Figure 22: Input, $x(t)$, pulsing profile	38
Figure 23: Learning Signal, $z(t)$, profile.	39
Figure 24: Amplitude of spontaneous learning operation on each synstor node.....	40
Figure 25: Actuation signal, $y(t)$	41
Figure 26: System Kinematics.....	42
Figure 27: Biological intelligent system kinematics.....	44
Figure 28: Comparison between the SNIC and human modeling and fitting error.....	53
Figure 29: Modeling and experimental agreement.	54
Figure 30: Probing Period Detail.	55
Figure 31: Effect of learning amplitude on system improvement.	57

Figure 32: SNIC and Human tuning accuracy..... 60

Figure 33: Improvement caused by learning comparison..... 62

Figure 34: Learning Time. 63

Figure 35: Oscillation induced by learning comparison. 65

LIST OF TABLES

Table 1. Comparison of physical quantities and computing methodologies between the brain and a supercomputer.	3
---	---

ACKNOWLEDGMENT

This dissertation would have been impossible without the support and guidance of my advisors, colleagues, friends, and family.

I would like to thank my advisor, Dr. Yong Chen, for his persistent guidance and support to strive to produce the best work possible. Without such an opportunity to work on an impactful research project, I would have never been able to grow and improve myself.

To my committee members, Professor Tetsuya Iwasaki, Professor Wentai Liu, and Professor Jason Speyer, for their faith and guidance towards this achievement.

To my present group members, Mr. Cameron Danesh, Mr. Chris Shaffer, Mr. Dhruva Nathan, and Mr. Rahul Shenoy, for their ongoing collaboration and support in my research project.

To my past group members, Dr. Kyunghyun Kim, Dr. Alex M. Shen, Dr. Chia-Ling Chen, Dr. Byungjin Cho, Mr. Andrew Truong, and Mr. Dongwon Lee for their effort prior to my involvement, and mentorship in my early career.

To the support given by the Air Force Office of Scientific Research (AFOSR) under the programs, “Intelligent Neuromorphic Network” (contract number: FA9550-15- 1-0056) and “Avian-Inspired Multifunctional Morphing Vehicles” (contract number: FA9550-16- 1-0087).

VITA

- 2011 B.S., Aerospace Engineering
University of Illinois at Urban-Champaign (UIUC)
- 2013 M.S., Mechanical Engineering
University of California, Los Angeles (UCLA)
- 2013-2017 Ph.D. Candidate and Graduate Student Researcher
Mechanical & Aerospace Engineering Department
University of California, Los Angeles (UCLA)

PUBLICATIONS

K Kim, **A Tudor**, CL Chen, D Lee, AM Shen, Y Chen “Bioinspired neuromorphic module based on carbon nanotube/C60/polymer composite” *Journal of Composite Materials* 49 (15), 1809-1822 (2015)

AM Shen, K Kim, **A Tudor**, D Lee, Y Chen, “Doping Modulated Carbon Nanotube Synapstors for a Spike Neuromorphic Module” *Small* 11 (13), 1571-1579 (2015)

AM Shen, CL Chen, K Kim, B Cho, **A Tudor**, Y Chen, “Analog neuromorphic module based on carbon nanotube synapses” *ACS nano* 7 (7), 6117-6122 (2013)

BT Gockel, **A Tudor**, MD Brandyberry, RC Penmetsa, EJ Tuegel, “Challenges with Structural Life Forecasting using Realistic Mission Profiles” 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials (2012)

H Lee, M Brandyberry, **A Tudor**, K Matouš, “Three-dimensional reconstruction of statistically optimal unit cells of polydisperse particulate composites from microtomography” *Physical Review E* 80 (6), 061301 (2009)

1.0 Introduction

1.1 Characteristic Features and Structures of the Brain

Biological neural networks are superior to conventional computing techniques due to the biological computers parallel signal processing, learning ability, and memory storage styles. Learning for intelligent systems are programmed by humans [1] [2], whereas the biological learning is based on a synaptic learning rule [3]. The human brain can intelligently process and learn from signals in dynamic, and unknown environments, in real-time, and without external programming, via a Hebbian learning [4]. In contrast, conventional transistor computers have signal processing, learning, and memory operating in a serial mode and occurring in different physical locations while having a fixed program strategy.

Learning functions can help autonomous systems adapt and give them the necessary intelligence to improve their performance in dynamic environments with large uncertainty given their limited energy resources [5] [6] [7]. The time and energy required for serial computation and the signal transmission between the different transistors may increase exponentially with signal dimensions [8] [9], which are referred to as the “curse of dimensionality [10]” and “von Neumann bottleneck [11]”, respectively.

Biological neural networks, for example the human brain, excel in both speed and power consumption when solving problems where a high dimensional signal must be processed quickly [12]. Pattern recognition, speech processing, artificial intelligence, and creativity are all tasks well suited for biological neural networks [13] [14] [15]. Creativity allows for biological networks to change their processing strategy, rather than being programmed by a human. Fixed

programming logic is a heavy constraint on current artificially intelligent systems, whereas biological neural computers routinely change, adapt, and learn new processing strategies.

The basic element of the brain is the synapse, the junction between two neurons. The synapse contains pulse signal processing, memory, and learning in the same physical location [16]. In this style, the biological network does not need to bus information between separate memory and logic circuits. This advantage is particularly obvious when logically processing high-dimensional sensor data. The neuron collects current from many synapses and non-linearly produces output pulsing in the soma [17]. Output pulsing may interact with other synapses in a multi-layered network.

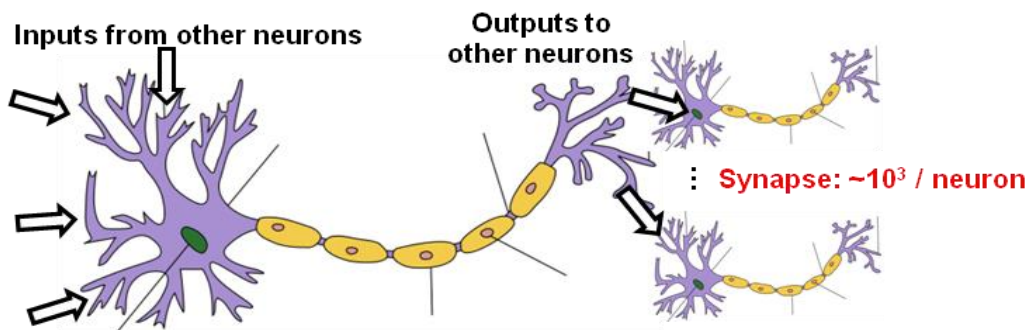


Figure 1: Structure and input/output relationship of a biological neuron. In humans there is a 10^3 -input dimension per neuron.

In a quantifiable comparison to conventional computing, the brain consumes much less power (~ 10 W) than the fastest supercomputer, as of June 2016, ($\sim 10^7$ W) [12]. Sunway TaihuLight supercomputer's power consumption is equivalent to supply electricity for 10^4 homes.

Human Brain		Supercomputer
10^1 (W)	Power Consumption	10^7 (W) [12]
10^{15} (Synapse Number) [18]	Number of Devices	10^{14} (Transistor Number) [19]
10^0 (kg)	Weight	10^5 (kg)
10^{-3} (m ³)	Size	10^3 (m ³)
10^1 (Hz) [20]	Operating Frequency	10^9 (Hz)
Parallel, Adaptive, Self-Programmed, Analog	Logic	Human Programmed, Serial Processing, Binary

Table 1: Comparison of physical quantities and computing methodologies between the brain and a supercomputer.

Supercomputers have been used to perform simulations of the neural network, but were not able to match the signal processing speed or the low power consumption of the biological equivalent [21] [22] [23]. Similarly, Silicon (Si)-based circuits have been used to emulate the neural network, but consumed considerably more energy and were unable to be scaled up to a size comparable with the biological model [24] [25].

1.2 Biological Synaptic Operations

Although the number of synapses effect the brains ability to logically process and learn new strategies, the fundamental structures, both individual synapses and neurons have properties allow for the larger networks superior intelligence abilities. Therefore, it is essential to understand the fundamental operating mechanisms.

Operating in pulsing mode, pre-synaptic, or input, spikes generate a dynamic temporal response along the neuron. The neuron can, simultaneously, collect currents from other synapses spatiotemporally to generate post-synaptic, or output, current [26]. The current is triggered on the order of several milliseconds to several minutes, depending on the time scale of the temporal

correlation. Finally, the output current is collected in the soma to produce, post-synaptic, or output pulsing. The strength of the correlation between the input spike and post-synaptic current is referred to as the synaptic weight. There are two kinds of synaptic weights, inhibitory and excitatory. Inhibitory synapses will produce negative output pulsing, while excitatory synapses will produce positive output pulsing.

The modification of the synapse is referred to as synaptic plasticity. Plasticity allows the synaptic weight to be adjusted in a non-volatile style. Without plasticity the synapse will ‘forget’ its setting quickly, making it impossible to implement learning, memory, or similar high-order functionality.

The synaptic weight can be adjusted based on the correlation between pre- and post-synaptic pulsing [27]. The modification is performed locally, without external calculation. Hebbian Learning, originally proposed by Donald Hebb in 1949, proposes an explanation for the adaptation of neurons in the brain during the learning process, including a basic mechanism for synaptic plasticity. Later, Synaptic Time Dependent Plasticity (STDP) was introduced to account for mismatch in input and output spike timing [28]. In that case, the time difference between the input and output pulsing determines the amplitude and polarity of the synaptic modification.

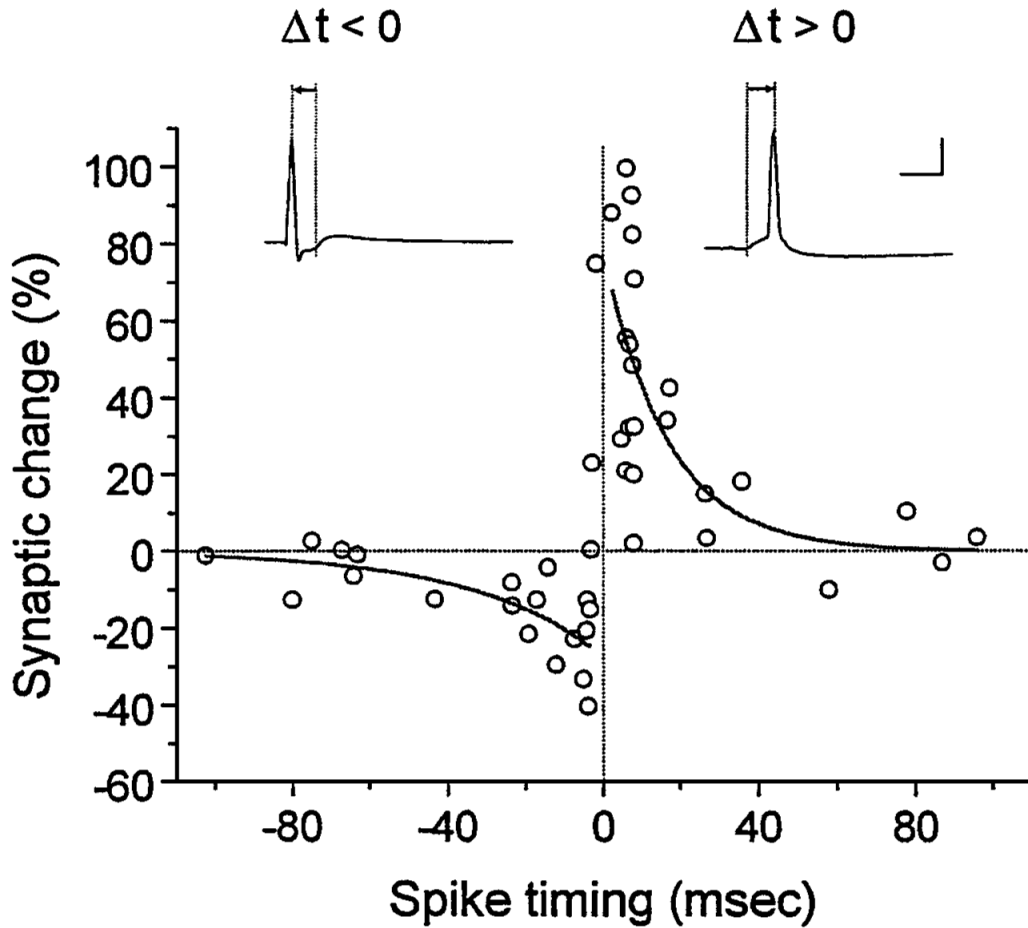


Figure 2: Synaptic modification due to STDP. The top pulses represent the time difference between pre-synaptic pulse at $t=0$, and post-synaptic pulses at $t=\Delta t$ [27].

1.3 Biological Style Computation in Conventional Computers

Biological neural networks have several distinct advantages over typical computing styles. To capture some of these advantages, artificial neural networks (ANN) can be implemented in transistor based computers to have learning progressively improve the performance of a task. ANNs have had widespread use in image recognition, speech processing, autonomous control, and others [29].

Artificially intelligent ANNs require considerable computing resources. While the biological system have hardware specialized for these complex, high-dimensional and parallel tasks, simulating even simplified and low-dimensional tasks on serial style architectures may consume vast amounts of memory, time, and power. To specify the synaptic connections will require vast amount of memory. Furthermore, the transmission of signals though the network, as well as calculation of the synaptic modification, will require enormous CPU processing power and time [29].

Biological style emulation using a CMOS style circuit directly is also inefficient when applied to artificially intelligent systems. In transistor based computers and neuromorphic circuits signals are processed by logic transistors and must communicate with other memory transistors. Learning in this style of computing is also a serial process. Although the functions of the synapse and neuron can be emulated in this way [30] [31] [32], the additional requirements of the CMOS style circuits make them ineffective for real-world problems. The power consumption, size, computing styles are closer to that of a supercomputer rather than a biological brain.

1.4 Turing Machine, Curse of Dimensionality, von Neumann Bottleneck

Modern day conventional computers are approximations of the theoretical Turing Machine, proposed by Alan Turing in 1936, describes an abstract machine which can read and write to cells on an infinite strip of tape according to a logical process. The machine head can read cells, one at a time, then perform some user-specified logic such as writing, erasing, or null to the tape. Finally, the tape is moved, and the process is repeated. Alan Turing went on to

provide a mathematical description of the process and show that the simple device is capable of arbitrary computations [1].

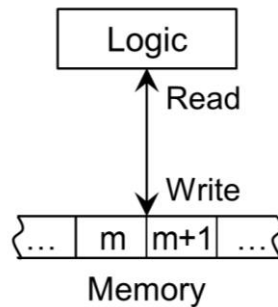


Figure 3: Turing machine schematic. Logic and memory communicate sequentially on an infinite memory tape.

The Turing machine however is impossible to implement exactly in hardware. Approximations of a Turing machine has been implemented in hardware using transistor technology. Finite memory, bussing speeds, and computation frequency are real-world limitations of Turing computer implementations.

The Von Neumann architecture [33], described in 1945 by John von Neumann, is a design architecture for a digital computer. The memory unit, containing any stored program, is interfaced to the central processing unit via a bus. The critical disadvantage of this architecture is referred to as the ‘von Neumann bottleneck’. The bus that fetches instructions from memory cannot simultaneously write to memory. In other words, writing and reading from memory are serial processes. Each time, the information must travel to or from the central processing unit in a different physical location.

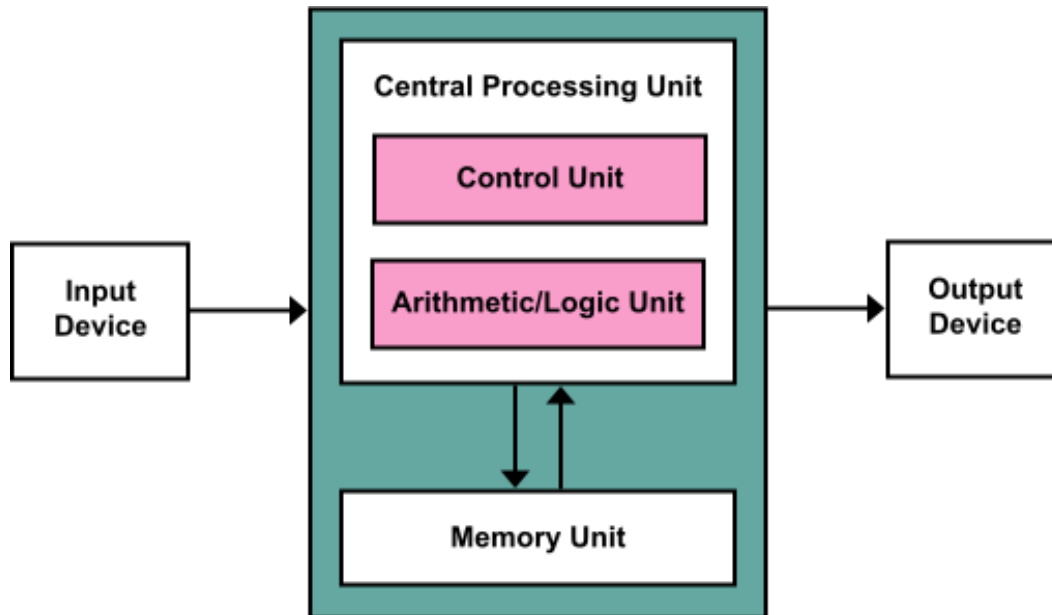


Figure 4: von Neumann architecture schematic. The von Neumann bottleneck is the transfer of information between the memory unit and the central processing unit.

Shortcomings in the von Neumann architecture are compounded when applying the style of digital computing to a large dimension. The number of unique elements in a combinatorics type problem increases exponentially with the dimension number. The amount of data need to perform learning on such a large dimensional network becomes enormous [34]. The serial central processing unit can only perform computations at a fixed rate. As a result, the time and energy required to perform a given computation greatly increases with signal dimension.

In a conventional transistor computer, signal processing, learning, and memory operate in a serial mode and occur in different physical locations. The time and energy required for serial computation and the signal transmission between the different transistors may increase exponentially with signal dimensions, which are referred to as the “curse of dimensionality” and “von Neumann bottleneck”, respectively.

1.5 Research Goals

Typical implementations of artificial intelligence are greatly limited by the usual computer and processing architecture. The goal of this research is to develop a synaptic resistor device and accompanying neuromorphic system that exhibits self-programming, artificially intelligent functions, causing improvement in unknown systems, while circumventing the ‘curse of dimensionality’ and ‘von Neumann bottleneck’.

The human brain, inspiration for the neuromorphic computing strategy, shares many of the same advantages of the synthetic neural circuits in comparison conventional computing techniques. The new style of processing and learning is heavily inspired by the biological equivalent [35] [36] [37]. The natural comparison to the neural circuits is a human performing the same task, given the same information. Both intelligent systems share a similar computing strategy they can be analyzed in the same style and compared to each other.

Specifically, the neural circuits will interact with an unknown system in a changing environment and, simultaneously, learn how to logically process the systems sensing information. It is critical that the system logically process the sensing information and intelligently modify its processing strategy in real-time, without external computing circuits or memory, and perform learning operations in parallel. Removing external computing circuits avoids the ‘von Neumann bottleneck’ while enforcing parallel learning operations eliminates the ‘curse of dimensionality’.

For the comparison, several metrics associated with the learning process can be statistically compared over several experiments. Explicitly: amplitude, accuracy, speed, oscillation of the learning modification.

2.0 The *Synstor*, A Synaptic Transistor

2.1 *Synstor* Structure Summary

The *synstor* device is a two-terminal memory element, capable of analog learning and processing functionality. Voltage signals, x_i and z_j are applied of the input and output terminals respectively to produce output current signal I_j . Under the device a reference electrode that is always grounded.

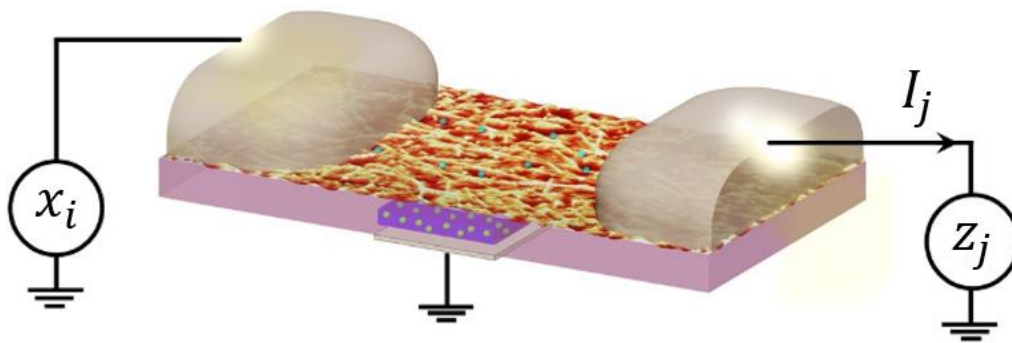


Figure 5: Synstor device structure and two-terminal operation schematic.

The device structure is essential to the operation modes advantages. Aluminum (Al) is used as the contact metal to the 99.9% semiconducting single walled carbon nanotube (CNT) layer. The random network of p-type CNTs in the channel form a percolating path between the two metal electrodes. The high percentage of semiconducting CNTs, while being sparsely distributed, is essential to form non-conducting percolating paths in the channel. The randomness of the CNT density and variation in the Al/CNT contact is reduced by making the channel area large. The channel width is 20 (μm), and the channel length is 0.4 (mm).

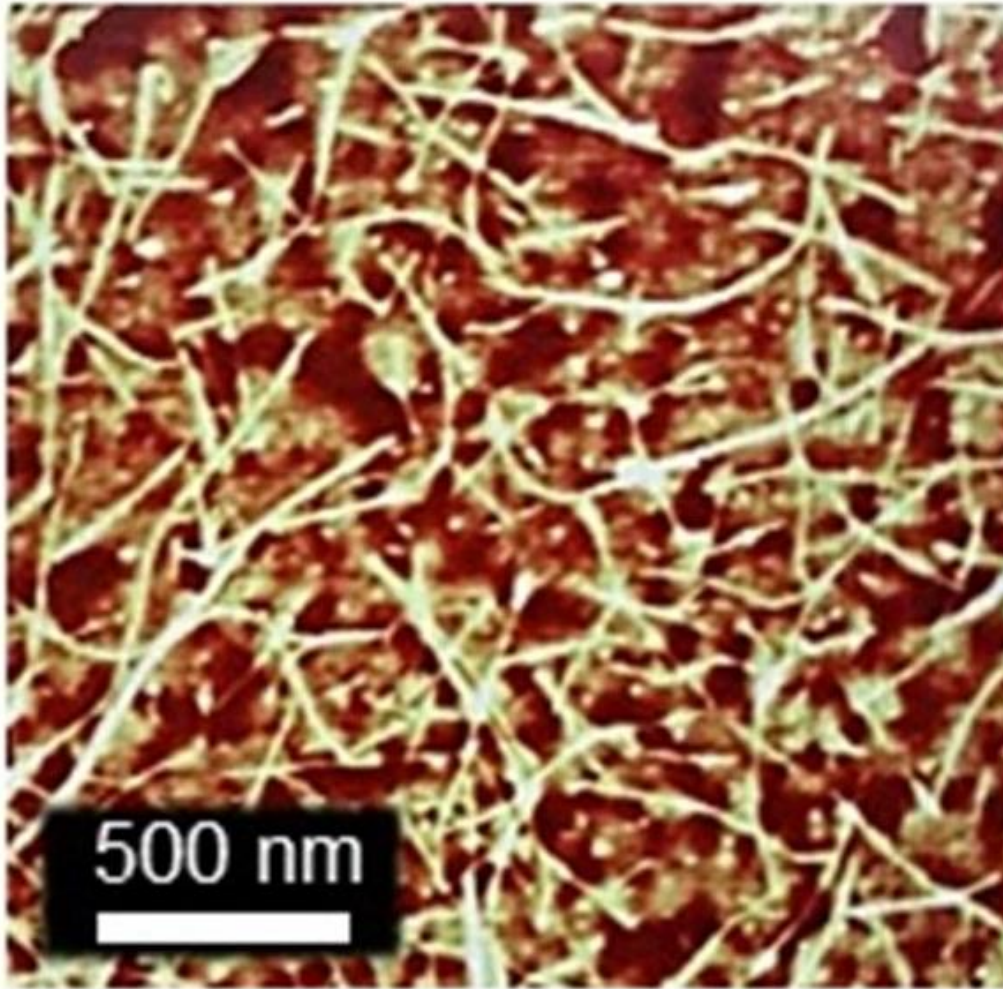


Figure 6: Atomic force microscopy image of a random, sparse, and 99.9% semiconducting CNT network in the synstor channel area.

Below the channel a $\text{HfO}_2/\text{TiO}_2/\text{HfO}_2$ structure can allow for the non-volatile storage of electrons. The HfO_2 charge injection barrier, between the CNTs and the TiO_2 is 6.5 (nm). The barrier between the gate and the TiO_2 is 22 (nm). The thickness of the TiO_2 is 2.5 (nm). The relative thicknesses in the HfO_2 layers are essential for having the smallest capacitance between the upper TiO_2 and CNT layer. Of course, once the barrier layer become too small/thick the

leakage current will be very large or the induced electric field will have negligible effect, respectively.

The proper CNT network density is critical for proper device operation. A sparse CNT network will reduce the projected area between the CNTs and the TiO₂, causing a smaller capacitance. In general, increasing CNT density will result in larger device current. The higher CNT density will make the capacitance between the TiO₂ and channel layers larger, and unfavorable. Under the same tuning condition, the overall tuning range will become decreased for a dense network. The normalized tuning amplitude, R, is plotted as a function of average device current. A sparse CNT network is essential for large tuning amplitudes. For the devices in the population with the smallest current, the largest amplitude of conductance change is observed.

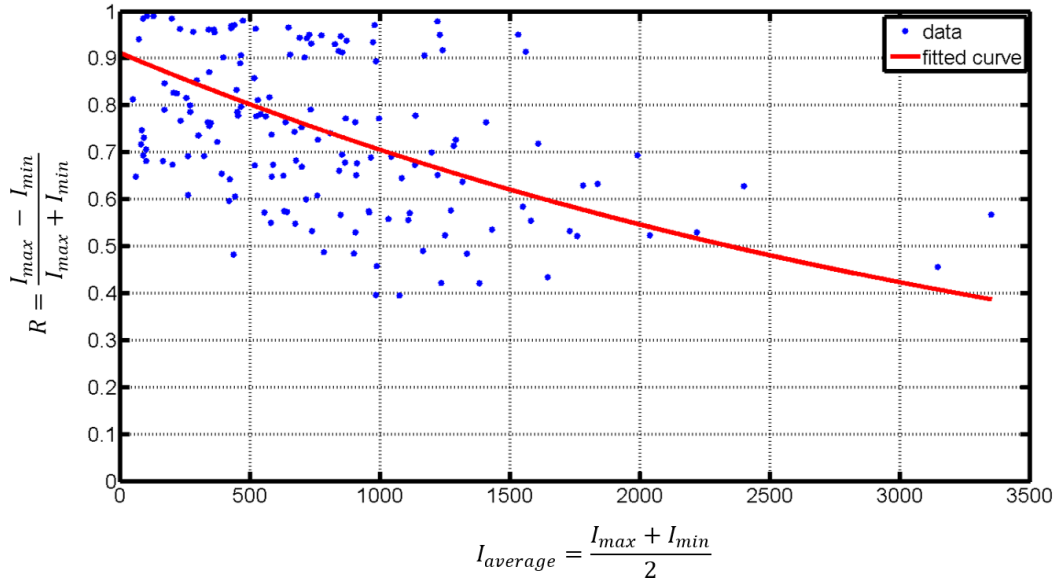


Figure 7: Effect of CNT density on tuning normalized tuning amplitude. Sparse CNT networks produce the largest tuning amplitude.

2.2 Synstor Operation Mechanism

The device has two operation modes, processing and learning, that can be switched between seamlessly. Because a single element can perform both tasks, there is no von Neumann bottleneck induced by transmitting data from memory to logic circuits. Unlike conventional circuits, the processing and learning take place in the same physical location. The operation modes are decided based on physical phenomenon rather than a programming logic.

In processing mode, the device conductance, W , is read by applying a negative voltage spike on the input, x , electrode. The negative voltage will produce output current, I , proportional to the device conductance.

$$\text{Processing: } I_j = W_{ji} x_i$$

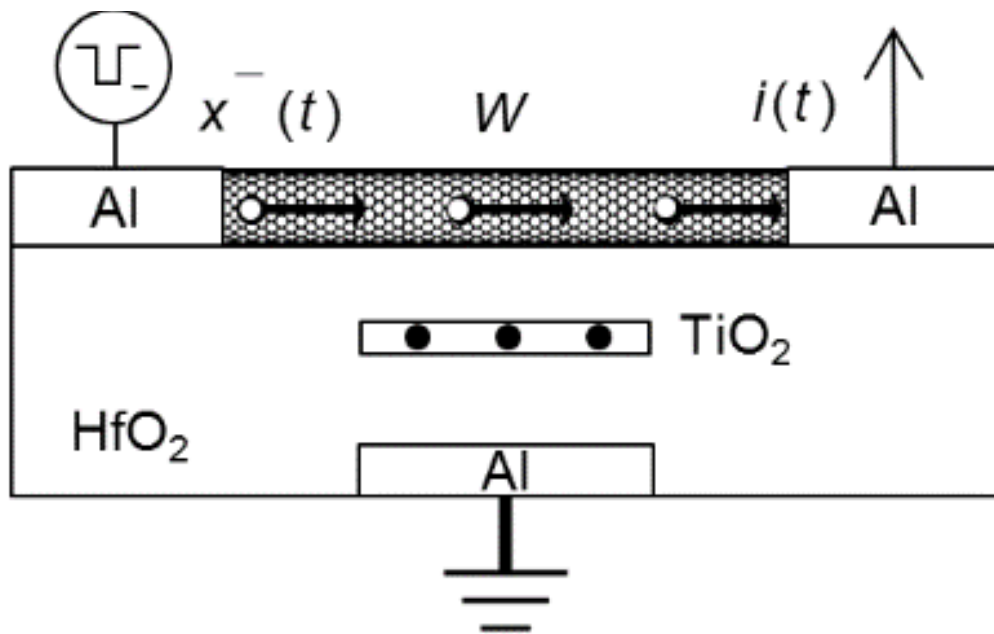


Figure 8: Synstor device in processing mode. The conductance, w , is read by applying a negative voltage on the input, x , electrode.

At the interface between the Al contact and the p-type CNT layer, a Schottky barrier is formed [38]. The Schottky barrier reduces the overall current, giving the device its non-linear operation. Furthermore, the voltage between the central parts of the CNT layer and the Al gate is small when only an input voltage is applied. The small relative voltage between the Al reference electrode and channel while in processing mode allows for the non-volatile storage of electrons in the TiO₂ layer.

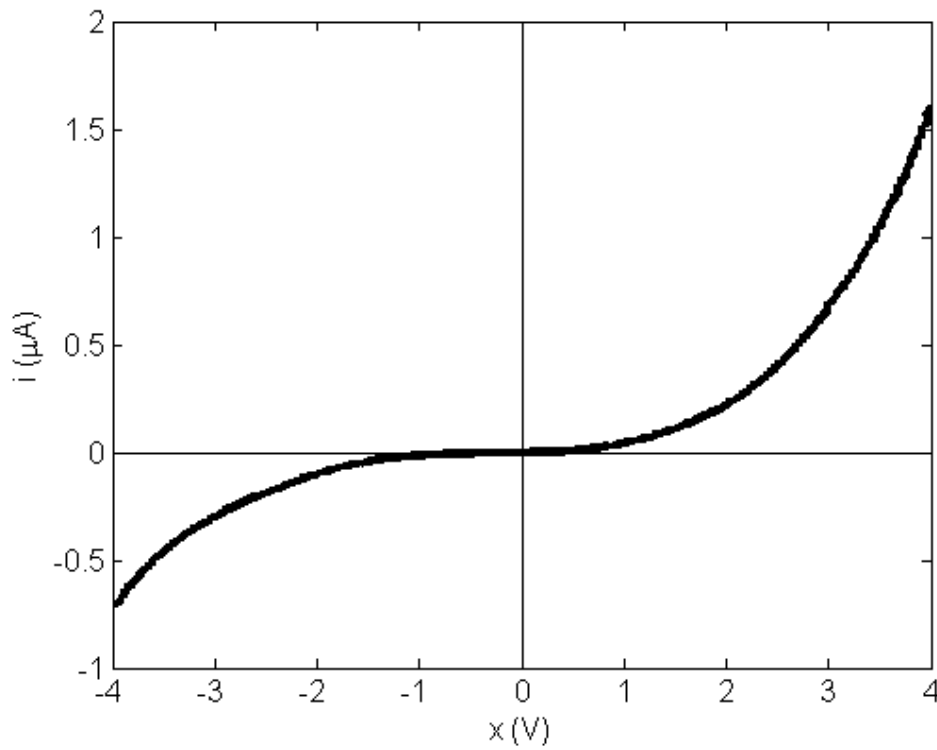


Figure 9: Schottky barrier formation at the CNT/Al contact location produces a rectifying property at low voltages.

Under small input voltage conditions, most of the voltage is reduced across the Al/CNT Schottky contact, producing small current through the CNT channel. Under these small current

conditions, the relative voltage in the CNT channel with respect to the reference electrode is small. At larger voltages the voltage reduction across the Schottky barrier does not vary strongly with input voltage, resulting in a large current increase, and higher channel voltage with respect to the reference electrode.

The conductance, W , of the CNT layer can be modified, in an analog style, by applying an electric field between the gate and the channel. Charges move through the tunneling barrier, HfO_2 , when a large voltage is applied. The non-linear electron flow through the barrier causes exponentially less charge to be transferred as the tuning voltage is reduced. The charge storage is non-volatile, causing the resistance change of the CNT layer to remain constant after tuning is applied.

The device is switched into learning mode spontaneously when a spike on the input electrode meets a spike on the output, z , electrode. Depending on the spiking polarity, the device conductance can be depressed ($\eta < 0$) or potentiated ($\eta > 0$). The relative voltage between the reference electrode and the channel will cause electron movement in the TiO_2 layer. Both the amplitude and duration of the voltage intersection can be modified to change the tuning strength. When the voltage is released on the output electrode device spontaneously returns to logic mode.

$$\text{Learning: } \dot{W}_{ji} = \eta z_j x_i^T$$

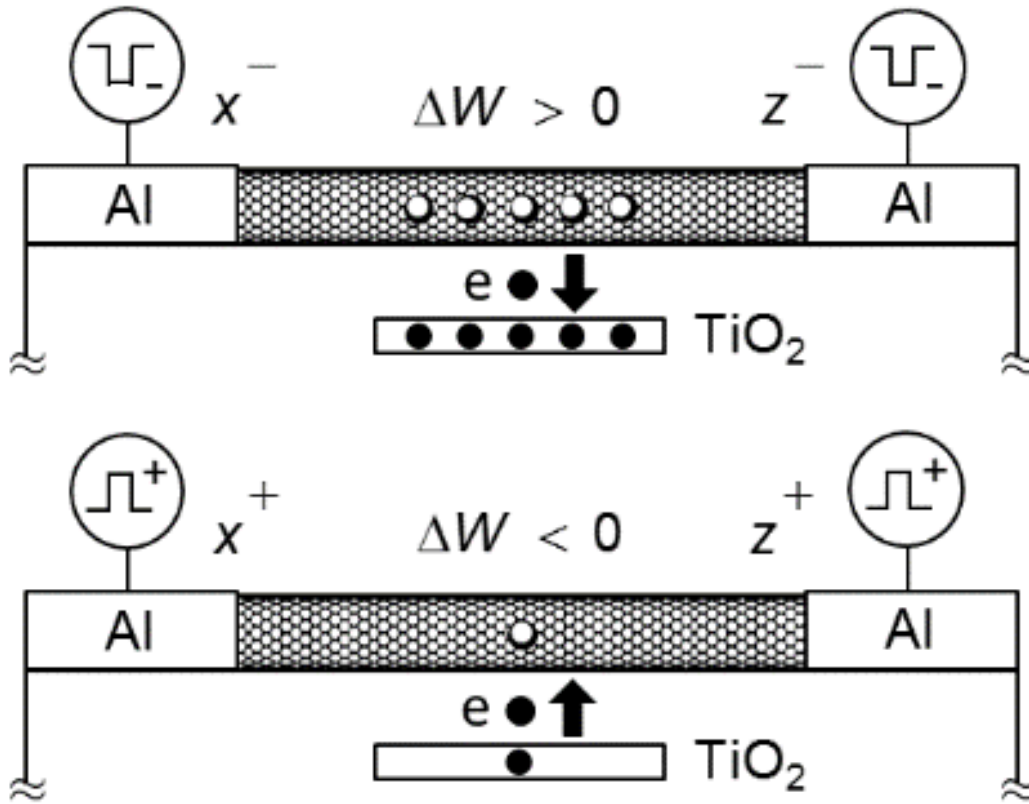


Figure 10: Synstor device in learning mode modifies the charges storied in the TiO₂ layer when input, x, and learning, z, voltage intersect spatiotemporally.

In this operating style the device and process input pulsing signals and learn based on the correlation between input and output pulsing. The device physics dictate this operation style, rather than a computer programming logic. The two processes operate simultaneously in the device, and in the same physical location. The style of operation can circumvent the von Neumann bottleneck.

2.4 Quantifiable Properties and Benchmarking

In the direction of using the synstor device for a realistic application, it is essential to measure properties that are necessary for a large-scale implementation. At the device level there are four properties that are empirically tested.

- Tuning Non-linearity

Tuning non-linearity is defined by how well the device modification agrees with the Hebbian style learning rule described in the earlier section.

$$\dot{W}_{ji} = \eta z_j x_i^T \quad (1)$$

The learning rule requires, the device conductance not be modified when either $x_i = 0, z_j \neq 0$, or, $x_i \neq 0, z_j = 0$. Although it is easily attained when implementing a neural simulation, a hardware approach will require careful choice of operating conditions.

- Tuning Smoothness

Tuning smoothness is defined by the analog tuning of the device. Smooth tuning devices will have a gradual change of conductance that is proportional to the integrated history of learning.

$$\Delta W_{ji} \propto \text{sign}(\eta) \int z_j x_i^T d\tau \quad (2)$$

In the case where the tuning is not smooth, the abrupt conductance changes will effectively erase all the learning history, making fine adjustments to the network setting nearly impossible.

- Tuning Symmetry

Tuning Symmetry requires that the modification to turn the device on and off are similar amplitude. Asymmetry introduced in the tuning ability may cause a bias in the learning process making Hebbian learning impossible.

$$\Delta W_{ji} \propto \text{sign}(\eta) \left(\int z_j^+ x_i^T d\tau - \int z_j^- x_i^T d\tau \right) \quad (3)$$

The learning bias will cause device modification in fixed direction that may not be correlated with the input signal.

- Tuning Amplitude

Tuning Amplitude refers to the total conductance modification percent caused by learning. Although at the single device level the amplitude is not a strong factor to consider, in larger networks or devices a large tuning amplitude can account for device variation.

These properties are complicated, non-linear, functions of the voltages applied and tuning history. Taking an empirical approach is particularly advantageous as it will closely mimic synstor operation in a large-scale implementation, giving a realistic, application based, understanding of the device characteristics. Furthermore, many physical phenomena governing the device behavior are difficult to quantify accurately with high enough certainty to generate a reliable model.

Two styles of testing have been developed with the goal of choosing a voltage operation condition that has favorable performance characteristics. The “Multi-Voltage Test” applies several tuning pulses and measures the change in conductance using many different voltage conditions. The “Multi-Pulse Test” applies fixed voltage pulsing but measures the change in conductance after each pulse.

2.4.1 Multi-Voltage Testing

To implement a Hebbian learning algorithm on the device, it must follow the “Learning” relation described previously.

$$\dot{W}_{ji} = \eta z_j x_i^T \quad (4)$$

Unlike neural simulations, the device does not follow an exact programming logic to implement the learning algorithm. As a result, there can be some conductance modification under the uncorrelated conditions: $x_i = 0, z_j \neq 0$, or, $x_i \neq 0, z_j = 0$. The learning induced by uncorrelated pulsing typically does not improve the system performance, a voltage condition that most closely follows the Hebbian learning rules should be chosen. The “Multi-Voltage” test was developed to quantify the relative modifications due to correlated and uncorrelated pulsing with the goal of optimizing tuning non-linearity.

The synstors were tested in the learning mode by applying multiple pairs of x_i and z_j potential pulses on their input and output electrodes simultaneously. The typical device conductance change, ΔW_{ji} , induced by the 50 pairs of pulses was measured and plotted versus the amplitudes of the x_i and z_j pulses.

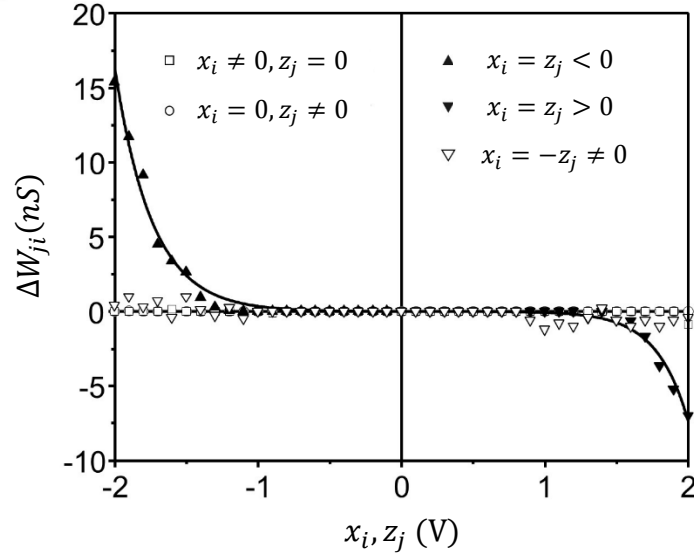


Figure 11: Multi-voltage synstor device testing shows a strong non-linear property under coincident pulsing.

At small voltage magnitudes < 1 (V), both correlated and uncorrelated tuning pulses have similar, $\Delta W_{ji} = 0$, modification. As the magnitude of the voltage is increased, the tuning caused by correlated pulsing increases while the tuning caused by uncorrelated pulsing remains small. Finally, at voltages near 2.0 (V) the uncorrelated tuning begins to increase, making the ratio between the two effects worse. The ΔW_{ji} data are fitted well by $\Delta W_{ji} = \Delta W_{ji}^0(1 - e^{\mu x_i})$ with $\Delta W_{ji}^0 = -7.8$ (pS), $\mu = -3.8$ (/V) for $x_i = z_j > 0$; and $\Delta W_{ji}^0 = 0.86$ (pS), $\mu = -4.5$ (/V) for $x_i = z_j < 0$. It was also observed that $\Delta W_{ji} \approx 0$ under $x_i = -z_j$ and $z_j \neq 0, x_i = 0$. At the condition $|x_i| = |z_i| = 1.75(V)$ the correlated pulsing was the dominant effect.

2.4.2 Multi-Pulse Testing

The multi-pulse test performs tuning on a device using a fixed voltage condition, for example $|x_i| = |z_i| = 1.75(V)$. Using these voltage amplitudes, the non-linear tuning property was shown by the multi-voltage test to have a good character. The device in this test performs sequential learning then processing operations. The conductance modification caused by each pulse is recorded relative to the initial conductance.

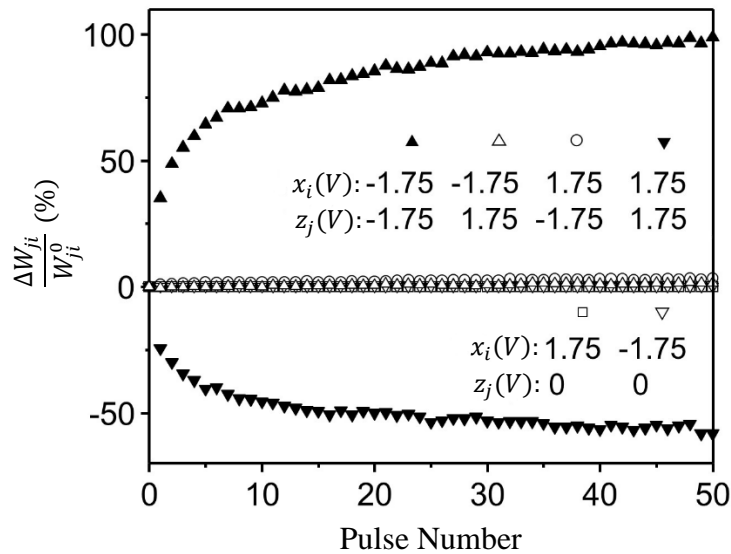


Figure 12: Multi-pulse synstor device testing shows a strong non-linear property under coincident pulsing.

For the case $|x_i| = |z_i| = 1.75(V)$, W_{ji} was gradually decreased or increased in an analog mode versus the numbers of the pulse pairs, and approached its lowest or highest saturation value after applying 50 paired pulses, which indicates η is a nonlinear function of W_{ji} . Within the first several pulses, the device shows a very smooth and gradual conductance change, with only slight non-linear effect. Within approximately 5 - 10 tuning pulses the device will have a smooth

tuning character. Although the device has some asymmetric tuning, the voltage conditions can be modified such that $|x_i| \neq |z_i|$.

In processing modes where $z_j = 0$, W_{ji} remains unchanged. This indicates that the synstor processed the x_i pulses in its logic mode with a nonvolatile memory of W_{ji} . In learning mode, the device can be modified using a Hebbian style learning rule.

3.0 Spike Neuromorphic Intelligent Circuit (SNIC)

3.1 Experimental Details & Set-up

A crossbar array of synstors, organized into neural circuits, can emulate the analog signal processing, learning, and memory functions of a biological spiking neural network. These functions occur at the same physical location in each synstor. Scaled-up neural circuits could circumvent the “curse of dimensionality” and “von Neumann bottleneck” to process and learn from sensing data with speed and energy efficiency exponentially superior to their classic counterparts.

It is reported that a 16-node, spike neuromorphic integrated circuit (SNIC), arranged into 4 neurons with in-house designed somas, can to improve the performance of an unknown system in an unknown, and dynamic environment. The crossbar network simultaneously processes the multi-dimensional sensing signals while modifying its strategy without external computation circuits or memory. Learning functions can help autonomous dynamic systems adapt and give them the necessary intelligence to improve their performance in environments with large uncertainty given their limited energy resources [4,5,6]. The performance of the autonomous system is analyzed and compared to humans performing the same task.

3.1.1 Experiment Schematic

The SNIC, arranged into a large-scale, $N \times M$ network, has combined analog signal processing, learning, and memory functions at each node. In processing mode, presynaptic potential pulse signals are sent in parallel and logically processed by the network.

$$I_j(t) = \sum_i W_{ji} x_i(t) \quad (5)$$

Where the i^{th} presynaptic potential signal, $x_i(t) \in \mathbb{R}^{N \times 1}$, will produce post synaptic current, $I_j(t) \in \mathbb{R}^{M \times 1}$, in the j^{th} neuron, by accessing its analog conductance states, $W_{ji} \in \mathbb{R}^{M \times N}$. The post synaptic current of the j^{th} channel is the sum of all the synstors contributions in that neuron.

$$\dot{W}_{ji} = \eta z_j x_i^T \quad (6)$$

The modification of the device conductance follows a Hebbian learning rule, a simplified form of spike-timing dependent plasticity (STDP), a description of the learning process in biological neural networks. The spatiotemporal intersection of x_i and $z_j \in \mathbb{R}^{M \times 1}$, the postsynaptic potential, in the network causes modification \dot{W}_{ji} . The conductance modification coefficient, η , is positive when operating in potentiation, $x_i, z_j > 0$, negative when operating in depression, $x_i, z_j < 0$.

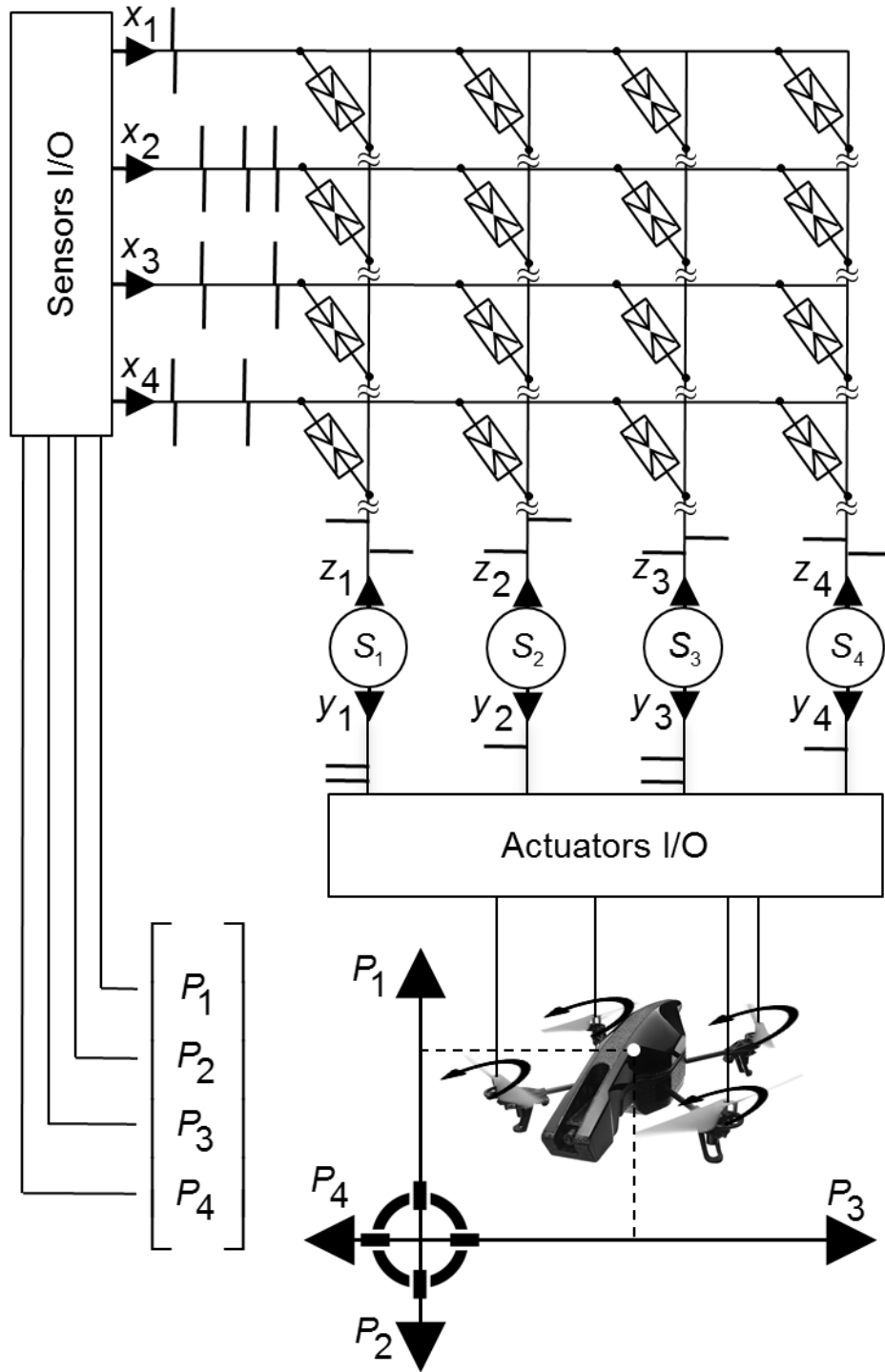


Figure 13: Experiment schematic

In processing mode, a Schottky barrier is formed between the Aluminum (Al) contact and CNT semi-conductor causing the conductance to be unchanged when $z_j = 0$. The hole density in the p-type CNT layer allows charges to flow between the two Al electrodes, the hold density is related to the device conductance. Modification of the charges stored in the Hafnium(IV) oxide (HfO₂)/Titanium Oxide (TiO₂)/ HfO₂ structure was previously used in a CNT nonvolatile memory transistor [35] [36] [37]. In learning mode, W_{ji} is modified based upon the intersection of presynaptic and postsynaptic potential signals in parallel, across the network. The voltage of the CNT channel with respect to the Al reference electrode causes charge trapping in the dielectric structure, which modifies the channel conductance.

In general, z_j pulsing can be used to implement numerous machine learning algorithms by adjustment of the pulsing profile and relative amplitude. In this learning algorithm, a zero-mean learning signal is applied on the z_j electrode, denoted \tilde{z}_j . The network spontaneously, including all devices in the j^{th} column, modifies the conductance based on the covariance between \tilde{z}_j and x_i . With increase input dimension, no additional time is required to perform the covariance based modification, avoiding the “curse of dimensionality”.

$$\Delta W_{ji}(t) \propto \int_0^t \tilde{z}_j x_i^T d\tau \quad (7)$$

The network can seamlessly switch from learning to processing modes by setting $z_j = 0$, causing $\dot{W}_{ji} = 0$, allowing devices in the j^{th} column to process input signal x_i . The unsupervised learning algorithm allows input signals to be produced in a feed-forward style, without knowledge about the networks processing/logic state. The network processes a four-dimensional input signal, in parallel, while simultaneously changing its memory state, without requiring

communication to update the processing logic, allowing the network to circumvent the “von Neumann bottleneck”.

The parallel signal processing, learning, and memory were demonstrated in a 16-node crossbar circuit with 4 soma circuits. The interaction between the synstor network and an unknown system in an unknown environment is used to demonstrate the synstor network’s ability to modify and improve the system’s dynamics via an unsupervised learning process.

The dynamic system’s performance is described by sensor vector, $P_{1,2,3,4}$. The performance is encoded as presynaptic sensing input spikes on $x_{1,2,3,4}$ inputs to the network, continuously.

The network produces postsynaptic current which is collected by the soma, $S_{1,2,3,4}$, to produce actuation pulse signals $y_{1,2,3,4}$. When learning occurs, postsynaptic potential pulses, applied on $z_{1,2,3,4}$, interact with presynaptic potential pulses to modify the network logic. The modification of the network processing logic, ΔW_{ji} , can be propagated through the system dynamics, and reflected by modification of the input spike profile.

The goal of the experiment is to improve an autonomous vehicles performance in an unknown environment. The SNIC has a randomized initial setting, with no pre-knowledge about the unmanned air vehicle or the environment. In the unknown environment, the SNIC will learn the best strategy to improve the systems performance while processing sensor data simultaneously. The advantage of the SNIC is that there is no external memory or processing circuits, circumventing the von Neumann bottleneck. The interface I/O is a simple feed forward algorithm that may be implemented in hardware. Additionally, the crossbar architecture automatically processes combination style problems automatically, in parallel, without suffering

from the curse of dimensionality. Several examples of the learning process are taken and compared statistically to a human performing the same task. The human brain, having a similar processing style, and inspiration for the SNIC, is a fair comparison to the learning circuits.

3.1.2 Vehicle Interface I/O

The dynamic system's performance is described by sensor vector, $P_{1,2,3,4}$. The performance is encoded as presynaptic sensing input spikes on $x_{1,2,3,4}$ inputs to the network, continuously. The frequency of spiking on input x_i is defined as F_{xi} .

$$\text{Frequency}(x_i(t)) = F_{xi}(t) \quad (8)$$

The Sensor I/O performs this task in a feed-forward style and may be implemented in hardware. In general, the input frequency varies monotonically with increasing sensor measurement. When P_i is small the associated input frequency, F_{xi} , is also small. As the deviation from maximum performance condition, $P_i = 0$, becomes larger the F_{xi} frequency begins increasing at a faster rate. There is an additional penalty for the system being close to the boundary of the environment.

Although in this case there is a simple mapping between the position sensor and input frequency, the sensing information can be translated into any performance metric to improve arbitrary criteria for the autonomous vehicle. For example, stability metrics, range, endurance, targeting, disturbance rejection, etc.

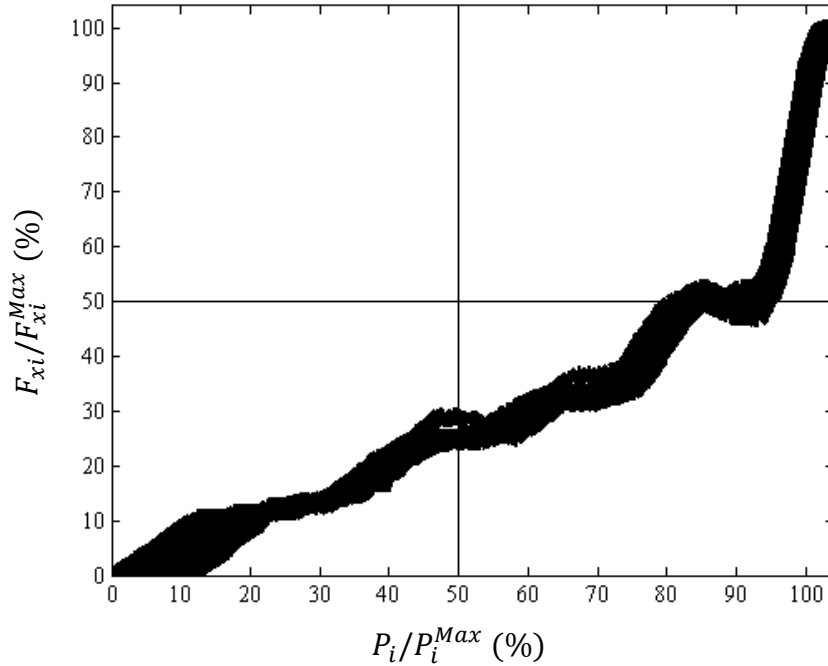


Figure 14: Analog pixel sensing, P, information is converted to a pulsing frequency, F, in the I/O interface.

At the output of the soma the y_j actuation signal must be sent to the air vehicle using a wi-fi signal. The air vehicle adjusts the rotor velocities depending on the y_j pulsing signal. The specific dynamics of the system are not a concern because the SNIC treats the vehicle as an unknown system, correlating y_j to performance changes without concern of the specific vehicle dynamics. In this way, if the system dynamics change, the SNIC will adjust its logic to account for the change.

3.1.3 Pogo-Pin Adapter

The SNIC crossbar circuit was fabricated on a 3 (cm) x 3 (cm) section of SiO₂. The 16-node, 4x4, SNIC crossbar was chosen and grouped out of a 20x20 fabricated crossbar network.

To interface the fabricated device reliably to the printed circuit board (PCB) testing system and autonomous system, an adapter contacted the metal pads on the chip surface.

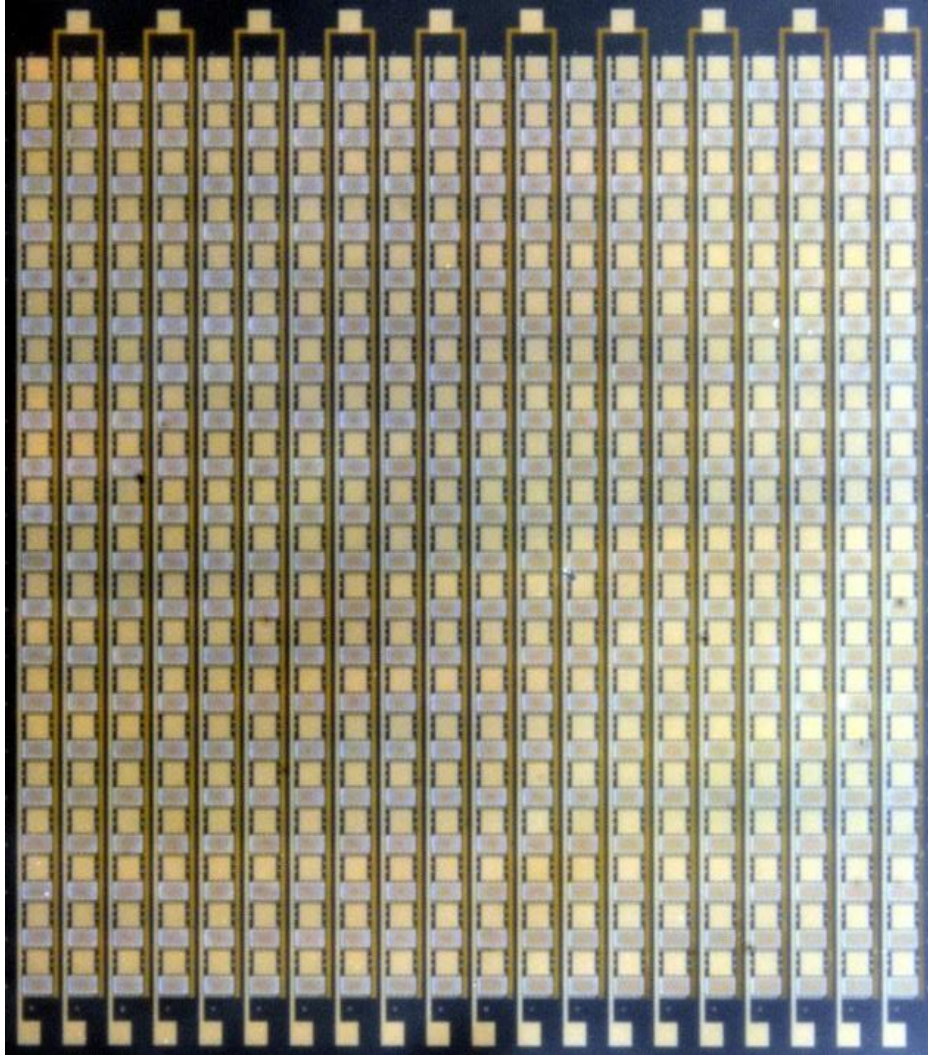


Figure 15: 20x20 synstor crossbar network.

The adapter aligned the chip to an array of pogo-pins that gently contact the Metal/SiO₂ surface. Processed sensing signals, I_j , are routed off the chip using a connecting logic to the soma

circuits in the testing PCB. Input voltage signals, x_i , are routed through the adapter and applied to all the devices in the same row. In addition, learning signals, z_j , can also automatically be applied on the same column through the pogo-pin adapter. The adapter is essentially a flexible interface that can apply voltage signals to the synstor devices.

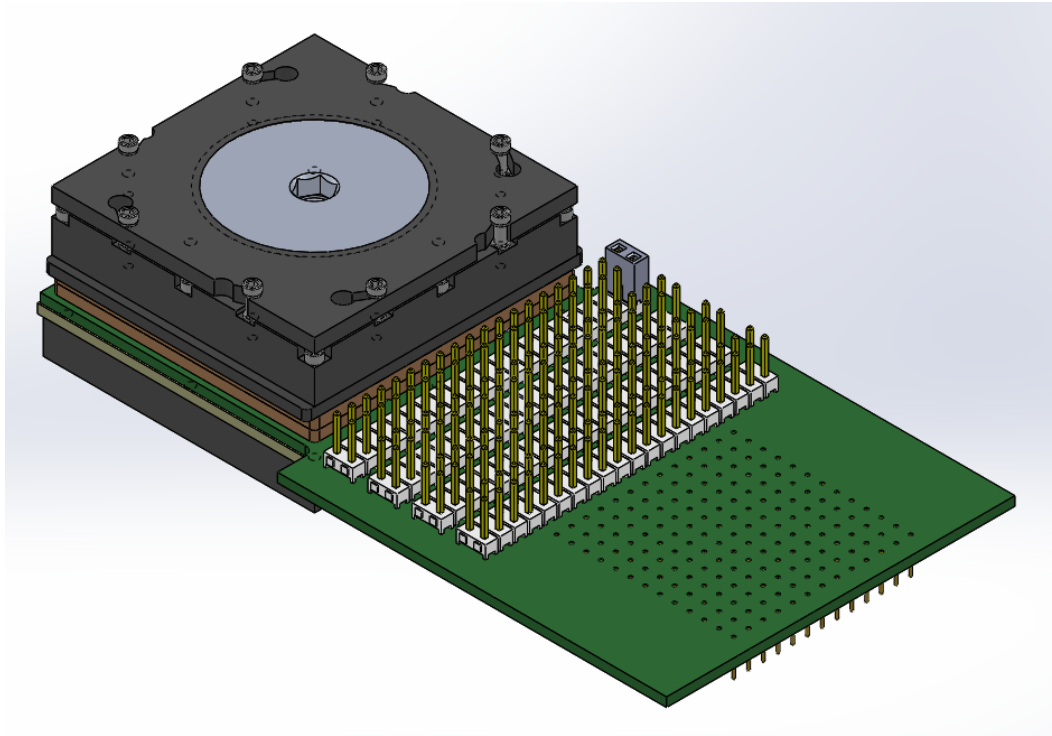


Figure 16: Pogo-pin adapter model.

3.1.4 Device Matching

To achieve good symmetry between devices, and account for variability, the devices in the crossbar were chosen to maximize the overlap between all the tuning ranges. With good matching, the operating voltage amplitudes can be smaller to have better smooth tuning characteristics while still preserving non-linearity. Because the actuation signals are signed, the actuation from neighboring columns are subtracted. Having matched tuning ranges will allow for

symmetric actuation signals. The matching between nodes was performed at the same operating conditions as the multi-voltage and multi-pulse test dictate.

The various combinations of device groupings are considered and optimized. In this case, 3 hardware inputs lines are grouped together to have the same x_i signal. Two outputs are grouped together as the same y_j signal. As a result, out of the 400 synstors in the population, 96 are chosen and grouped into 16-nodes for the demonstration. The 400 synstors population has a uniformity ratio, U , of 1.52. The uniformity ratio compares the mean device current to the standard deviation of device currents in the whole population.

$$U = \frac{\text{mean}(I_{\text{average}})}{\text{std}(I_{\text{average}})} \quad (9)$$

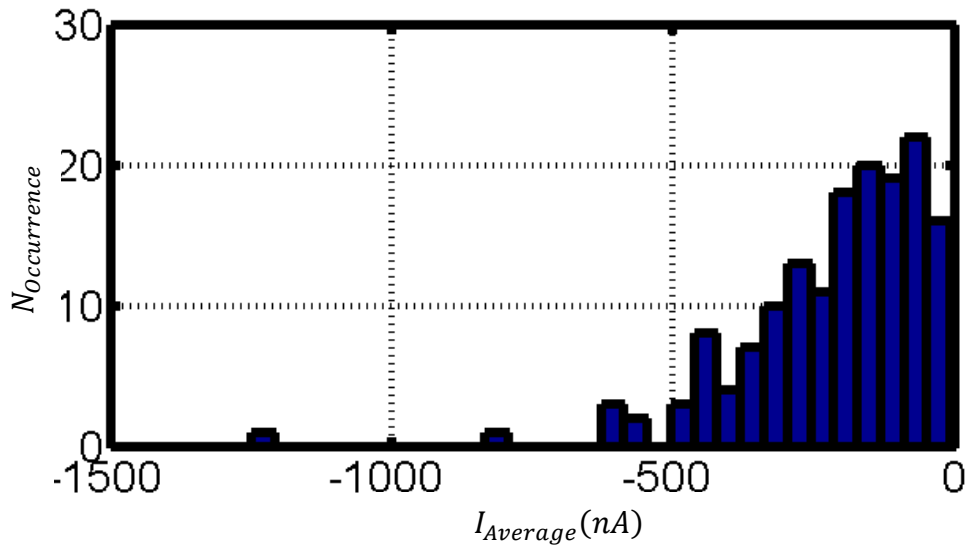


Figure 17: Histogram of device current.

With $U=1.52$ the possibility of using arbitrary devices to implement a large-scale learning implementation is unlikely. To reduce the variability a select population with a higher uniformity ratio is selected in accordance with the geometry constraints of the crossbar. After optimization, the device population has a uniformity ratio of $U=2.81$.

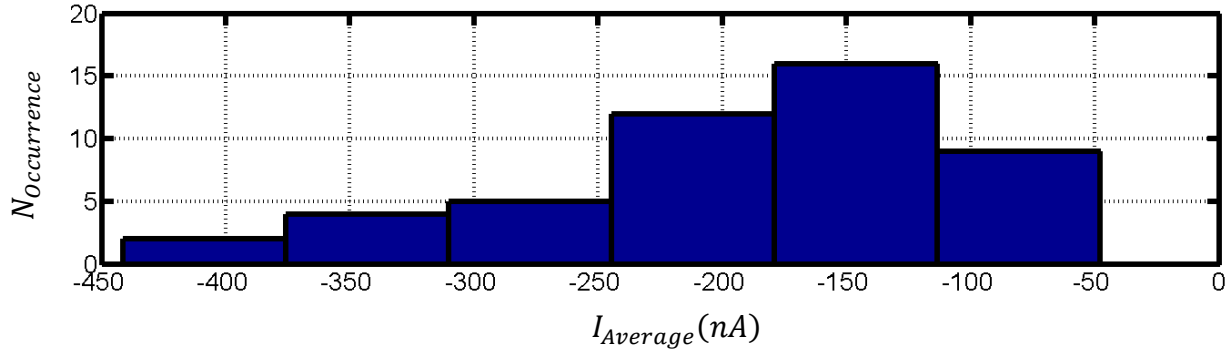


Figure 18: Uniformity optimized population.

The population is then reduced to 16 unique elements by grouping 3 input rows together for each x_i signal and matching the proper columns to maximize the overlap between the device tuning ranges. In this way the mean column uniform ratio is $U=8.47$. Using this process, the initial population uniformity is increase by a factor of ~ 6 , enabling the large-scale demonstration. The relative tuning ranges now have significant overlap.

The label of $W_{ji}x_i$ refers to the SNIC node (i,j) which is the optimized grouping of several individual synstors. The amplitude of the bars indicates the tuning range given a constant x_i voltage signal. I_j is the sum of all the individual nodal contributions in the column.

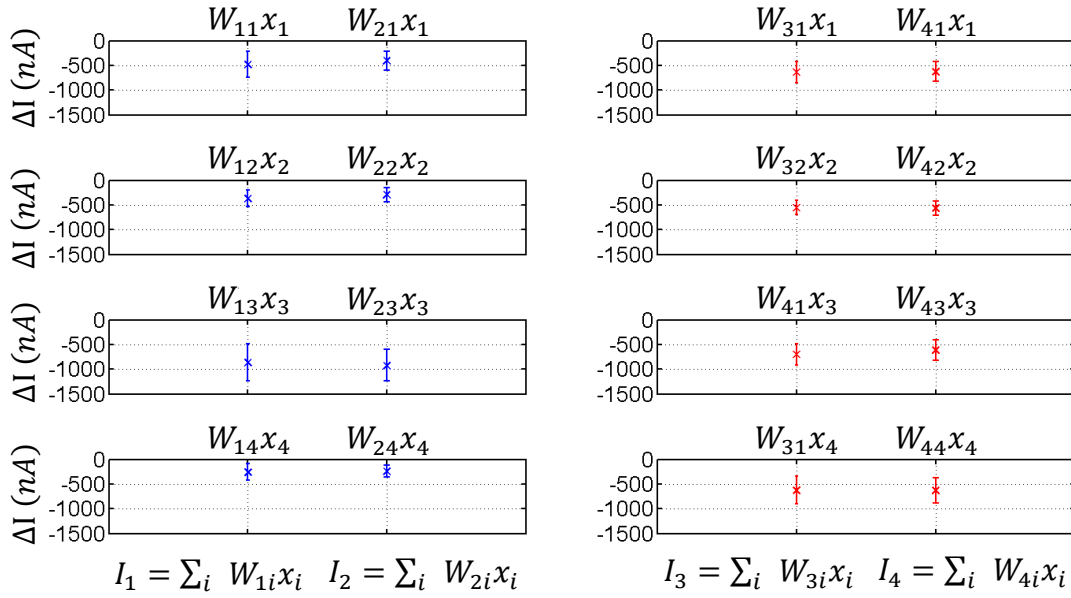


Figure 19: Optimized device matching based on optimizing tuning overlap of grouped devices.

3.2 Non-Linear Soma Circuit

We designed and fabricated an integrate-and-fire circuit with the basic functions according to the Hodgkin–Huxley neuron model [39]. The collective current, I_j , from multiple synstors flows through a diode toward a capacitor, C , increasing the potential, V_c , on the capacitor. A leakage current, I_L , flows through the resistor, R_L , decreasing V_c . V_c is proportional to the integration of $I - I_L$ with respect to time. When V_c reaches a threshold value, a Schmitt trigger composed of transistors M1-M6 is switched back and forth to generate an output pulse from the output channel, V_f . The output pulse resets V_c back to zero by switching transistors M7, M8, and M9, and the capacitor C restarts the integration of the current. The transistors in the circuit are operated in their subthreshold regions. The circuit simulation

indicated that the average power consumption of the Si “neuron” circuit is ~ 100 nW under an output pulse frequency of 1 KHz.

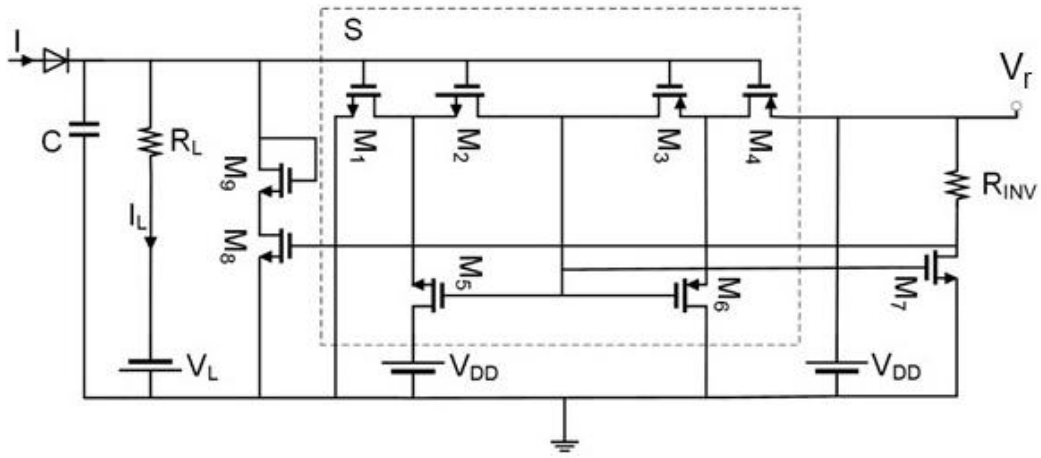


Figure 20: ‘Integrate and Fire’ CMOS circuit implementation

The typical firing rate of output pulses from a “neuron” circuit is plotted against the magnitude of the current, I . When $I < 20$ nA, no pulse is output from the “neuron” circuit. When 20 nA $< I < 90$ nA, r increases monotonically with increasing I . When $I > 90$ nA, r is saturated at 20 Hz, which is equal to the frequency of the input potential pulses.

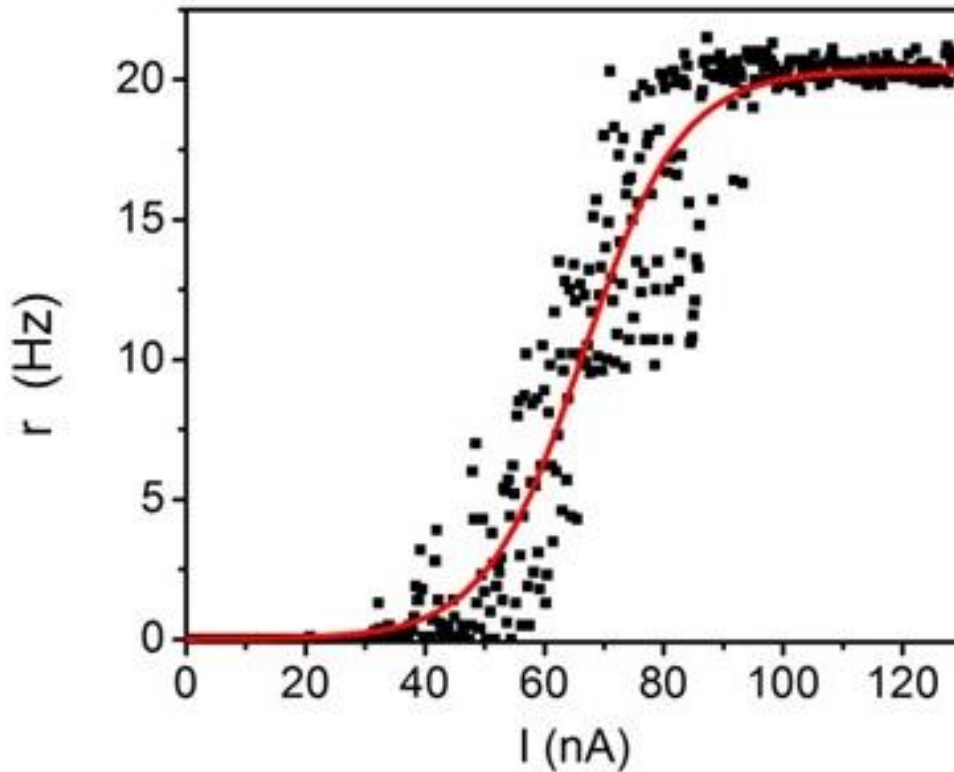


Figure 21: Integrate and fire neuron transfer character

3.3 Methods and Hardware Summary

The experiment was implemented using National Instruments (NI) LabVIEW 2015 SP1. A real-time interface between the Parrot AR Drone 2.0 and the experimental circuits was created using the NI cRIO 9063 with accompanying modules. The presynaptic and post synaptic potential pulses (x_i, z_j) were generated using the NI 9264, measured by the NI 9205, and applied to the AI input and output synstor electrodes. Currents, I_j , when $x_i < 0$ and $z_j = 0$, flowing

through the synstor, are collected by the soma to produce output pulsing non-linearly, but monotonically, related to the input current I_j , are sensed by the NI 9403.

The Parrot AR Drone 2.0, equipped with a 2500 mAh lithium polymer battery, is flown in an indoor environment while communicating with the host computer. A third-party add-on “AR Drone Toolkit for LabVIEW – LVH” was implemented to communicate actuation commands to the drone using a WiFi connection.

The position of the drone was captured using the Genius WideCam F100 camera. A reference image of the environment is compared to an image of the environment including the drone, in real-time. The pixel location of the drone is converted to a presynaptic pulsing frequency monotonically.

The currents from synstor circuits flowed to in-house built integrate-and-fire “soma” circuits to trigger output pulses and postsynaptic pulses. The output pulses are counted until reaching a threshold which triggers an actuation pulse, y_j , to be sent to the drone.

Feedback pulsing is triggered when the drone performance is below a certain threshold. Zero mean noise is injected into the actuator, which then propagates through the drone system. Simultaneously, zero mean feedback, z_j , is applied to interact with the noise injected into the plant, causing learning, ΔW_{ij} , in the network.

For comparison, the human performs the same task, using the same information, the circuit has available. Behind a curtain, with audio isolation, the human has 4 unlabeled, paired, buttons that activate the drone actuation y_j . The human can only process 4 unlabeled sensing signals, F_{xi} , and is instructed to reduce the frequencies to a minimum. Through a biological learning method,

the human modifies their processing logic to optimize the performance of the drone in the same environment.

3.3 Autonomous Air-Vehicle Parallel Learning and Processing Demonstration

The synstor network was tested using in a fixed voltage, dynamic frequency, pulsing mode. The goal of the network operation is to reduce the sum of the input sensing frequencies, Π , through an unsupervised learning process. Π represents the energy of the system.

$$\Pi(t) = \sum_{i=1}^4 F_{xi}(t) \quad (10)$$

Input sensing signals are a square wave with $x^+ = 1.5(V)$, $x^- = -1.75(V)$ and pulse period of 30 (ms) operating at frequencies up to 20 (Hz). The pulsing profile, shown below, is the input to the network. There are four input channels that are used to encode the sensing information to the intelligent network. Over time, the frequency of pulsing is reduced, as the system reached a favorable configuration. The input pulsing generation does not change depending if learning is occurring. In this style, the network and modify itself and process sensing signals in the same mode.

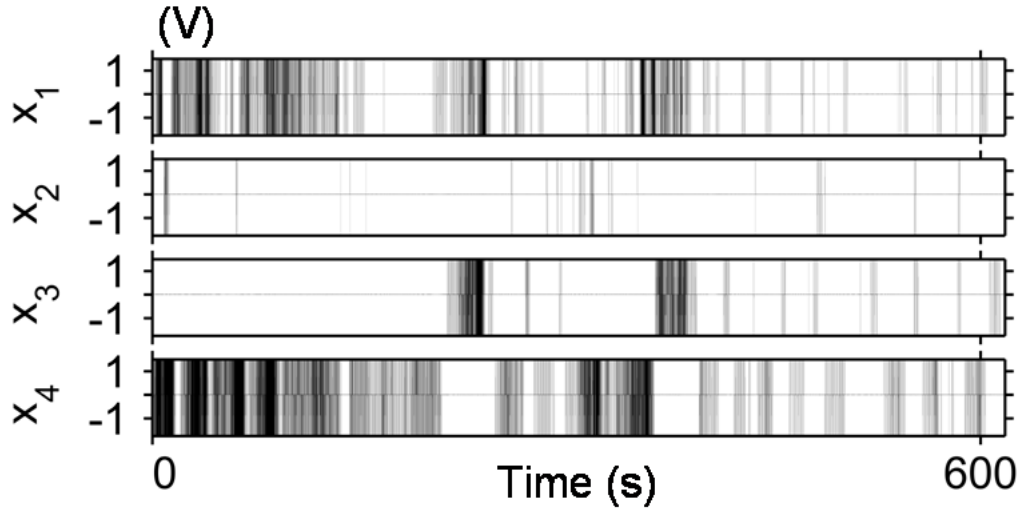


Figure 22: Input, $x(t)$, pulsing profile

Simultaneously, with input pulsing, learning pulsing interacts with the synstor devices. Learning pulses are a square wave with $z^+ = 1.5(V)$, $z^- = 0(V)$ for a positive pulse and $z^+ = 0(V)$, $z^- = -2.0(V)$ for a negative pulse. The pulses are clustered in 5-pulse groupings throughout a 1.5 (sec) duration. After a 2 second pause another 5-pulse cluster is applied in the opposite polarity. The time difference between the first cluster and the second allows the perturbation to be propagated through the plant, effecting the number of coincident pulses during the second cluster. A learning period is defined as a pair of presynaptic pulses.

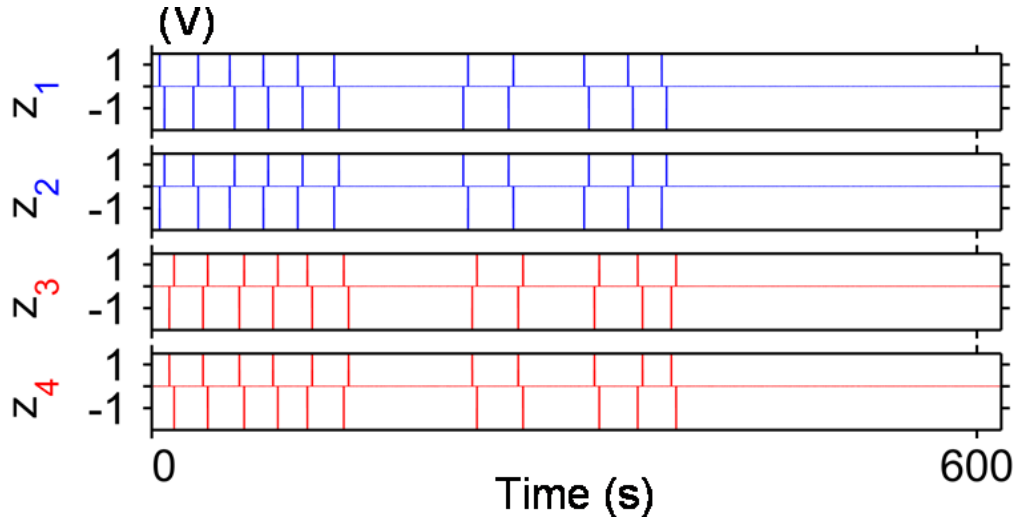


Figure 23: Learning Signal, $z(t)$, profile.

The total change in the nodal conductance in the n^{th} learning period is related to the number of intersections applied on that device, taking place $t_n < \tau < t_{n+1}$, which is also related to the correlation between the postsynaptic learning signal and the input pulsing. The amplitude of the covariance between pluses on the j^{th} neuron and i^{th} input is defined as v_{ij} .

$$\Delta W_{ji}(n) \propto v_{ji}(n) = \int_{t_n}^{t_{n+1}} \tilde{z}_j x_i^T d\tau \quad (11)$$

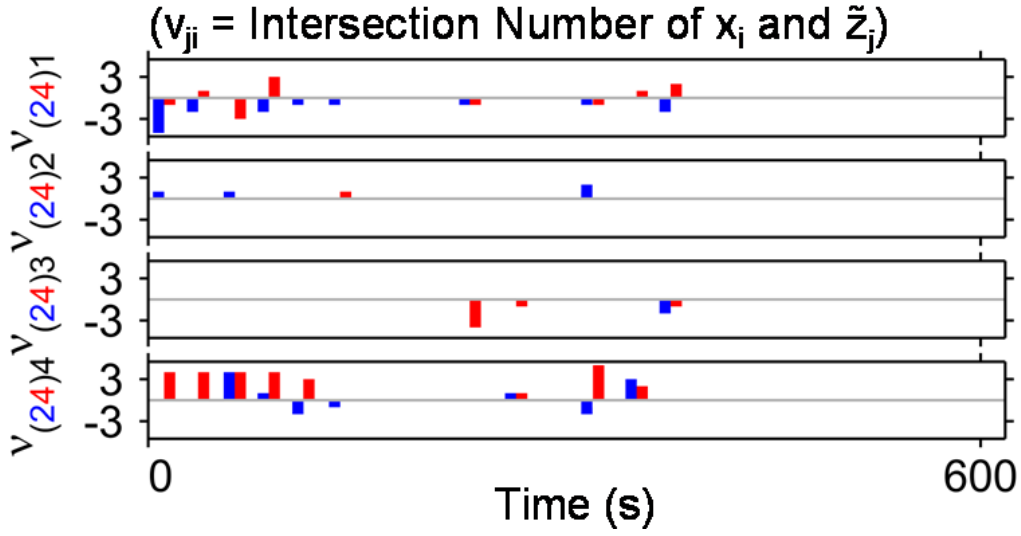


Figure 24: Amplitude of spontaneous learning operation on each synstor node.

The covariance is shown, roughly, to be $v_{21,22,43,44} \neq 0, v_{41,41,23,24} \sim 0$. Due to symmetry in the learning pulse, $z_2 = -z_1$, the intersection profile is also symmetric $v_{1i} = -v_{2i}$ and omitted for this figure. As the system reaches a favorable configuration the amplitude, and frequency, of the learning is decreased.

The actuation pulsing, y_j , is a command sent to the dynamic system, adjusting its performance in the environment. The actuation frequency of signal y_j , F_{y_j} , is a non-linear function, S_j , of the postsynaptic current collected by the soma. The soma calibration is fixed during the experiment and the same for all neurons.

$$F_{y_j} = S_j(I_j) \quad (12)$$

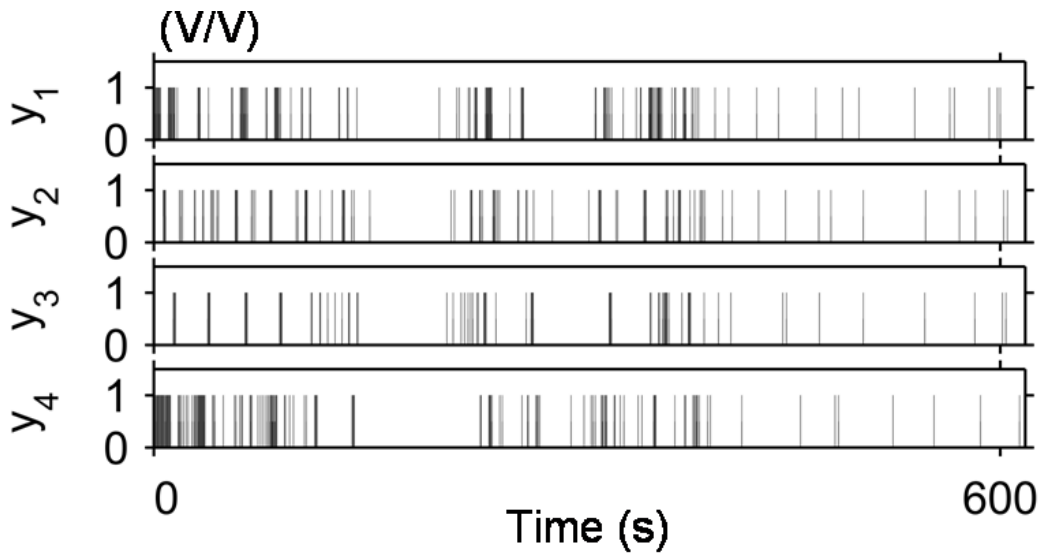


Figure 25: Actuation signal, $y(t)$.

The energy of the system is determined by the sum of the sensing frequencies, where F_{x_i} represents the frequency of pulsing on presynaptic input x_i , $F_{x_i} \geq 0$. At first, the network logic is not well suited for the unknown dynamic system it is interfaced with. Over time, the unsupervised correlative learning algorithm modifies the processing logic resulting in the flight becoming less erratic, and approaching $\Pi \sim 0$. The system is operating in a dynamic environment, the trendline (magenta) shows the average movement, ignoring zero-mean noise.

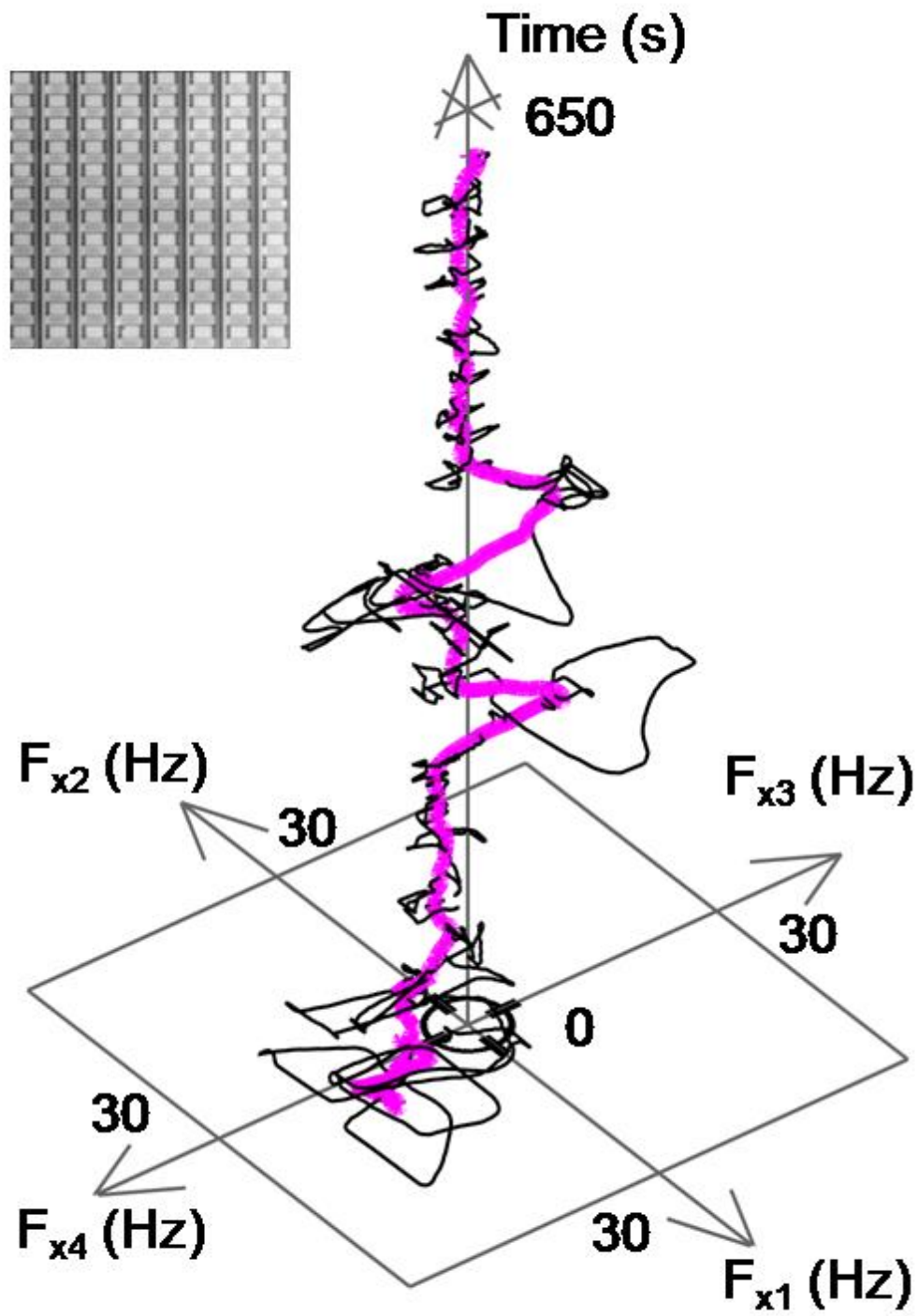


Figure 26: System Kinematics.

Biological neural networks can process parallel signals, while optimizing its logic in real-time in a similar strategy to the synthetic neural networks described. In comparison to the artificial intelligence exhibited by the SNIC, the biological equivalent is used for comparison. In a similar way, both styles of network can circumvent the “curse of dimensionality” and the “von Neumann bottleneck” while exhibiting intelligent operation. The input pulsing frequency, the only sensing information available to the human, is plotted. The human is isolated from the environment and can only use the unlabeled input pulsing frequency as sensing information. The human is given two paired actuators which they can press and use to correlate the sensing and actuation signals. The human is given the same information that is available to the SNIC.

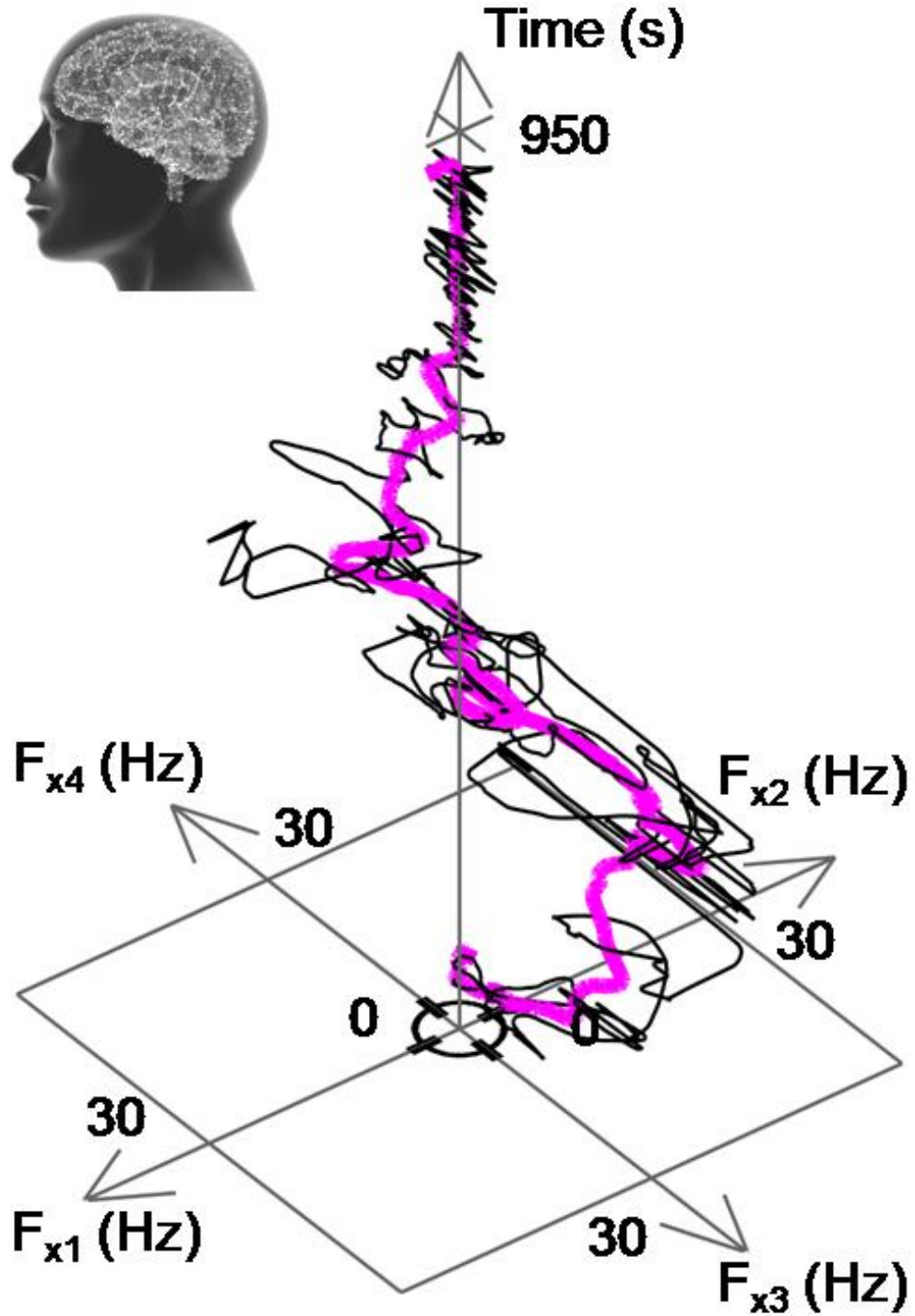


Figure 27: Biological intelligent system kinematics.

Although improvement of the systems performance is obvious, a way to quantify the behavior of the learning process, and its interaction with the circuit can help to understand why the system improved.

4.0 System Modeling and Learning Analysis

A simple model for the interaction between the synstor circuit, learning process, and the system will help to understand and quantify the properties of the intelligent system. There are four quantifiable properties that can be compared.

- Learning Modification Error
- Total System Improvement
- Learning Speed
- Oscillation Induced by Learning

In this way, the human and the SNIC experiments can be analyzed. The model is robust enough to be used to quantify further experiments as well as, simply developing a methodology to quantify an experiment result. Although there can be many models and analysis techniques, the goal is to provide a simple analysis that is able to quantify the above performance metrics and to give an understanding of how the learning process behaves. Developing the most complicated model to include every possible effect is not the goal of the analysis. Some assumptions are made in the direction of simplicity and ease of understandability. To support the modeling choices, an error quantification is performed to demonstrate a strong agreement between the model and the system behavior.

4.1 Network & System Model Overview

The network and the system interact to adjust the energy function, Π , the sum of the input frequencies.

$$\Pi(t) = \sum_i F_{xi}(t) \quad (13)$$

$$F_{xi}(t), \Pi(t) \geq 0, \forall t \quad (14)$$

Many factors can influence the systems energy function, the environment, damage, learning, sensing signals, and fitting parameters. The goal is to generate a model that relates the measured variables to the energy of the system. The measured variables are, $V_{ji'}(t)$, the accumulated number of learning pulses in the device at location (i', j') . In general, the total conductance change is increasing with increasing $V_{ji'}(t)$. $F_{xi}(t)$ is the frequency of the sensing signal on the i^{th} input. The performance can also be directly a function of time, t . There may also be a time delay, τ , between the measured signals and the resulting modification of the energy. Finally, a noisy term representing parts of the model that are uncorrelated with the measured variables, $\varepsilon(t)$.

$$\Pi(t) = f(V_{ji'}(t), F_{xi}(t), t, \tau) + \varepsilon(t) \quad (15)$$

Although measuring the conductance of the network directly would make the problem much simpler, it is impossible to measure directly. The devices are fabricated into a crossbar network, automatically summing the currents in the column. There is no way to measure each device individual contribution without separating the device from the rest of the network. Based on the multi-pulse test data it is reasonable to assume a correlation between the number of learning pulses, $V_{ji'}(t)$, and the total conductance modification, $\Delta W_{ji'}(t)$.

$$\frac{E[\widetilde{V}_{ji'} \Delta W_{ji'}]}{E[\widetilde{V}_{ji'}^2] E[\Delta W_{ji'}^2]} \gg 0 \quad (16)$$

Due to this strong correlation coefficient we can assume that the measured variable $V_{ji'}(t)$ is representative of the learning modification, $\Delta W_{ji'}$. The dynamic between the energy change of the system and the measured variables can be written exactly as follows.

$$\frac{d\Pi}{dt} = \sum_i \frac{\partial F_{xi}}{\partial t} + \sum_i \frac{\partial F_{xi}}{\partial V_{ji'}} \frac{dV_{ji'}}{d\Delta W_{ji'}} \frac{d\Delta W_{ji'}}{dt} + \sum_i \frac{\partial F_{xi}}{\partial F_{xi'}} \frac{dF_{xi'}}{dt} + \sum_i \dot{\epsilon}_i \quad (17)$$

There are four main contributions to the change in energy function. $\sum_i \frac{\partial F_{xi}}{\partial t}$ represents the change in the system energy that is only a function of time, excluding all the measured variables. Intelligent systems can modify their logic in time varying environments or under if the system behavior changes as a function of time. For example, the battery may produce less power in the later parts of the experiment or the downwash induced by the rotor can perturb the movement.

The second contribution, $\sum_i \frac{\partial F_{xi}}{\partial V_{ji'}} \frac{dV_{ji'}}{d\Delta W_{ji'}} \frac{d\Delta W_{ji'}}{dt}$ represents the change energy due to the modification of the conductance. This term is of specific importance to show the advantage of the learning process. Additionally, $\frac{dV_{ji'}}{d\Delta W_{ji'}}$ is assumed to be a non-zero number based upon the results of the multi-pulse test. In that style, it is possible to model how the conductance modification is reflected in the system energy.

The system, absent learning, damage, and disturbance is represented by $\sum_i \frac{\partial F_{xi}}{\partial F_{xi'}} \frac{dF_{xi'}}{dt}$. The term reflects the initial setting of the network. In the case $\Delta W_{ji'} = 0$, this term will dominate the energy function behavior. In the experiment, $W_{ji'}(t = 0)$ is initialized randomly, without any

prior knowledge about the behavior of the system. In that way, this term may help or hurt the systems energy function. In any event, the learning process is tasked to overcome poor initial settings and guide the device conductance to a critical value.

Finally, $\sum_i \dot{\epsilon}_i$, is the parts of the energy function that are uncorrelated with the influence of the measured variables. In general, the power of this term is minimized in the fitting process.

Re-writing the equation into its constituent parts,

$$\dot{\Pi}^{Environment} = \sum_i \frac{\partial F_{xi}}{\partial t} \quad (18)$$

$$\dot{\Pi}^{Learn} = \sum_i \frac{\partial F_{xi}}{\partial V_{ji'}} \frac{dV_{ji'}}{d\Delta W_{ji'}} \frac{d\Delta W_{ji'}}{dt} \quad (19)$$

$$\dot{\Pi}^{W^0} = \sum_i \frac{\partial F_{xi}}{\partial F_{xi'}} \frac{dF_{xi'}}{dt} \quad (20)$$

$$\dot{\Pi}^{\epsilon} = \sum_i \dot{\epsilon}_i \quad (21)$$

In general, the whole system can be molded as the sum of its constituent parts. To be exact, all are non-linear functions of time.

$$\dot{\Pi}(t) = \dot{\Pi}^{Environment}(t) + \dot{\Pi}^{Learn}(t) + \dot{\Pi}^{W^0}(t) + \dot{\Pi}^{\epsilon}(t) \quad (22)$$

4.2 Network & System Model Simplifications

Beginning with a quadratic model for each sensor measurements influence the sensor measurement change, it is possible to begin to understand the energy function behavior. $g^{(0,1,2)}$

are modeling parameters and are assumed to be constant. There can be some time delay between F_{xi} and the rate change of F_{xi} but it not explicitly written here.

$$\frac{d\Pi}{dt} = \sum_i \frac{dF_{xi}}{dt} = \sum_i (g_i^{(0)} + \sum_{ji'} g_{ij}^{(1)} (W_{ji'} F_{xi'}) + \sum_{ji'} g_{ij}^{(2)} (W_{ji'} F_{xi'})^2 + n_i) \quad (23)$$

Without knowing W directly it is possible to split the effect into an initial setting and a variable that correlated well with the measured variables, $W_{ji'}(t = 0) = W_{ji'}^0$.

$$\frac{d\Pi}{dt} = \sum_i \frac{dF_{xi}}{dt} = \sum_i \left(g_i^{(0)} + \sum_{ji'} g_{ij}^{(1)} (W_{ji'}^0 + \Delta W_{ji'}) F_{xi'} + \sum_{ji'} g_{ij}^{(2)} (W_{ji'}^0 + \Delta W_{ji'})^2 F_{xi'}^2 + n_i \right) \quad (24)$$

Expanding the equation, and grouping the terms into the various effects, and generating fitting parameters ' σ ', which are constant over the experiment, the final relation has several parts. The time dependent terms, including any time delay are written out explicitly.

- Effects from the environment and time varying system dynamics:

$$\dot{\Pi}^{Environment} = f(\sigma_i^{Env}) = \sum_i \sigma_i^{Env} \quad (25)$$

- Effects of the learning operation:

$$\begin{aligned} \dot{\Pi}^{Learn}(t) &= f(\sigma_{iji'}^{F^2V}, \sigma_{iji'}^{F^2V^2}, \sigma_{iji'}^{FV}) \\ &= \sum_{iji'} \sigma_{iji'}^{F^2V} V_{ji'}(t - \tau) F_{xi'}^2(t - \tau) + \sigma_{iji'}^{F^2V^2} V_{ji'}^2(t - \tau) F_{xi'}^2(t - \tau) \\ &\quad + \sigma_{iji'}^{FV} V_{ji'}(t - \tau) F_{xi'}(t - \tau) \end{aligned} \quad (26)$$

- Effect of the initial setting:

$$\dot{\Pi}^{W^0}(t) = f\left(\sigma_{iji'}^{F^2}, \sigma_{iji'}^F\right) = \sum_{iji'} \sigma_{iji'}^{F^2} F_{xi'}^2(t - \tau) + \sigma_{iji'}^F F_{xi'}(t - \tau) \quad (27)$$

- Noisy, uncorrelated effects:

$$\dot{\Pi}^\varepsilon(t) = \sum_i n_i(t) \quad (28)$$

The relation can be re-written in matrix form, allowing for the pseudoinverse to be taken and solve the system of equations. $\dot{\Pi}^\varepsilon(t)$ is the part of the model that is uncorrelated with the measured variables. Matrix A is a function of the measured variables, $(F_{xi}, V_{ji'})$, and time delay, τ .

$$\begin{aligned} & A(F_{xi}, V_{ji'}, t, \tau) \\ &= [V_{ji'}(t - \tau)F_{xi'}^2(t - \tau) \quad V_{ji'}^2(t - \tau)F_{xi'}^2(t - \tau) \quad V_{ji'}(t - \tau)F_{xi'}(t - \tau) \quad F_{xi'}^2(t - \tau) \quad F_{xi'}(t - \tau) \quad 1] \end{aligned} \quad (29)$$

The fitting variables can be consolidated into a column vector denoted σ . The vector is not a function of time. The fitting variables represent the system model.

$$\sigma = \begin{bmatrix} \sigma_{iji'}^{F^2V} \\ \sigma_{iji'}^{F^2V^2} \\ \sigma_{iji'}^{FV} \\ \sigma_{iji'}^{F^2} \\ \sigma_{iji'}^F \\ \sigma_i^{Env} \end{bmatrix} \quad (30)$$

The resulting relation becomes,

$$\frac{d\Pi}{dt}(t) = A(t - \tau)\sigma + \dot{\Pi}^\varepsilon(t) \quad (31)$$

The experimental data is used to fit the model in a least squares sense. Each F_{xi} is fitted individually then summed together to get the energy function. The energy function is the linear superposition of the sensing frequencies. The time delay variable, τ , is adjusted to minimize the overall error in the fitting compared to the experiment. Quantifiably, the fitting accuracy of F_{xi} is denoted E_{xi} . F_{xi}^{Model} is defined as the cumulative integration of the fitted \dot{F}_{xi} with an offset applied due to the integration. RMS is defined as the conventional ‘‘Root Mean Square’’ function.

$$E_{xi}(\tau) = \frac{RMS \left(F_{xi}^{Experiment}(t) - F_{xi}^{Model}(t, \tau) \right)^2}{RMS \left(F_{xi}^{Experiment}(t) \right)^2} \quad (32)$$

The value of E_{xi} is produced at several different τ values between 0.5 (s) and 15 (s). Each E_{xi} is considered when choosing the best τ selection. Typically, a value of roughly 8 (s) is chosen to be the optimum fit. The optimum is determined by the sum of the fitting errors.

$$\tau^* = \min_{\tau > 0} \frac{1}{4} \sum_{i=1}^4 E_{xi}(\tau) \quad (33)$$

In the biological case, the human measurement is only F_{xi} , the learning done by the human is not able to be recorded. Based upon an equivalent biological learning model, it is possible to estimate the learning amplitude between two fictional learning intervals $t_i < t < t_{i+1}$. The intervals are bounded to have a fixed amount actuation signal, Y , in a learning period. The value of Y is fixed for the analysis.

$$Y = \int_{t_k}^{t_{k+1}} \sum_j y_j(\tau) d\tau \quad (34)$$

The value of Y to be taken for all the analysis is 1 (s). On either side of the fictional learning interval a pulsing couple is applied to interact with F_{xi} at that time, similar to the SNIC operation. The conductance modification is applied at the center of the interval.

$$\Delta W_{ji} \left(\frac{t_{k+1} + t_k}{2} \right) \propto V_{ji} \left(\frac{t_{k+1} + t_k}{2} \right) = \int_{t_k}^{t_{k+1}} F_{xi} \tilde{z}_j d\tau \quad (35)$$

The polarity and amplitude of the \tilde{z}_j pulsing is proportional to the average actuation signal for each j^{th} column.

$$\Delta W_{ji} \left(\frac{t_{k+1} + t_k}{2} \right) \propto V_{ji} \left(\frac{t_{k+1} + t_k}{2} \right) = \frac{\Delta F_{xi}}{t_{k+1} - t_k} \int_{t_k}^{t_{k+1}} y_j(\tau) d\tau \quad (36)$$

4.3 Experimental Modeling Agreement

The human brain is an example of an intelligent system and may be compared to other artificially intelligent systems as a benchmark. The non-linear model is used to fit the experimental data with the goal of benchmarking and comparing the performance.

$$\frac{d\Pi}{dt}(t) = A\sigma + \dot{\Pi}^\varepsilon(t) \quad (37)$$

The σ fitting variables are constant over the experiment and eventually will be used to extrapolate benchmarking parameters. A satisfactory degree of agreement between the model and the experiment can show that the model is a reasonable representation of the system dynamics. A comparison between $F_{xi}^{Experiment}(t)$ and $F_{xi}^{Model}(t, \tau^*)$ are shown, F_{xi}^0 is an offset introduced by the integration.

$$F_{xi}^{Model}(t, \tau^*) = \int_0^t \dot{F}_{xi}^{Model} d\tau + F_{xi}^0 = \Delta F_{xi}^{Model}(t) + F_{xi}^0 \quad (38)$$

The SNIC and human experiment are fit the same way, using the same model. The agreement between the accuracy of the model or predict the kinematics of each intelligent system is comparable, with low relative error. The SNIC result is in the left column, the human result is in the right column.

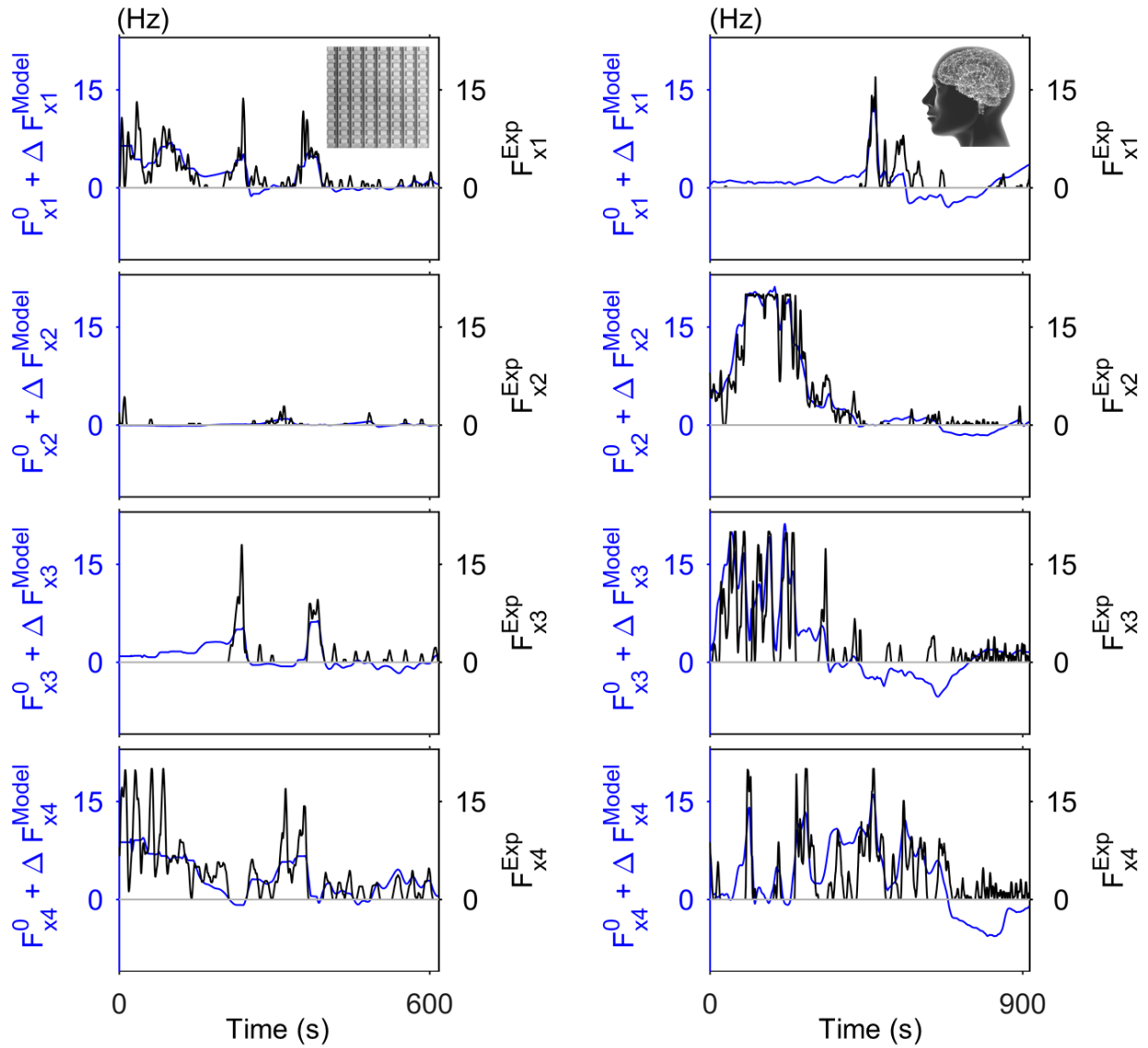


Figure 28: Comparison between the SNIC and human modeling and fitting error.

The model agrees well with the experimental result. Some high frequency components are missed in the model but could be a result of sensor noise. The assumptions made to the human data to calculate the synaptic modification, V_{ji} , can also introduce fitting error, causing the relative error to be larger in comparison to the circuit fitting. The model was fit to 10 datasets total, 5 for each of the intelligent systems. The fitting result was similar in quality. The fitting error, E_{xi} , is quantified for each fit and documented. The strong correlation between the modeling assumptions and the actual system provide evidence for a strong foundation for further analysis. The effects of only the correlated terms are considered, $\dot{\Pi}^\varepsilon = 0$, in this comparison.

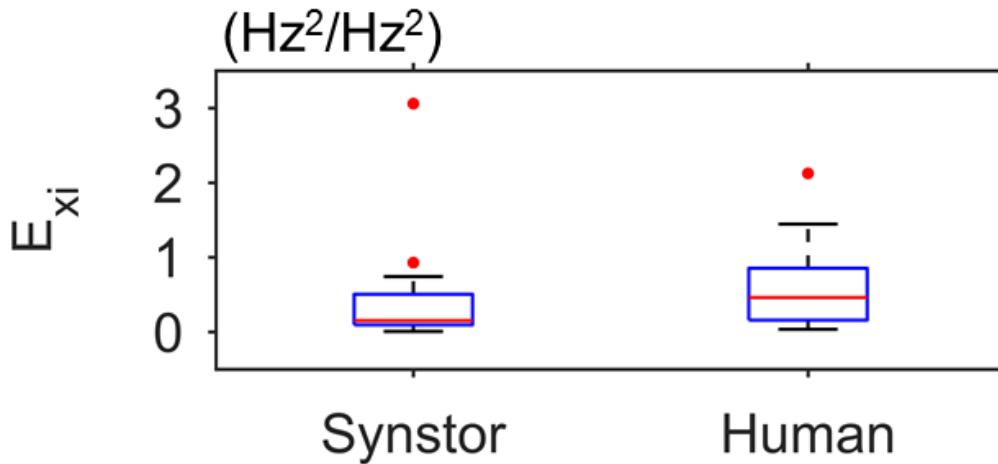


Figure 29: Modeling and experimental agreement.

4.4 Learning Effect & System Improvement

To gradually improve the performance of the system, the learning process will perform modification of the devices, in parallel. In that way, the time for modification of the parallel sensing signal logic is not increased with input signal dimension, a key advantage of the circuit. During the modification process, the system follows the same model as before,

$$\frac{d\Pi}{dt} = \sum_i \frac{dF_{xi}}{dt} = \sum_i (g_i^{(0)} + \sum_{j_i'} g_{ij}^{(1)}(W_{j_i'} F_{xi'}) + \sum_{j_i'} g_{ij}^{(2)}(W_{j_i'} F_{xi'})^2 + n_i) \quad (39)$$

To perform the exploration operation and sample the system behavior, a zero-mean signal replaces the $W_{j_i'} F_{xi'}$ on a single column, the signal is denoted $\tilde{y}(t)$. The exploration signal is synchronized with the leaning signal, $\tilde{z}(t)$. The two signals have amplitude $|\tilde{y}|$ and $|\tilde{z}|$ respectively.

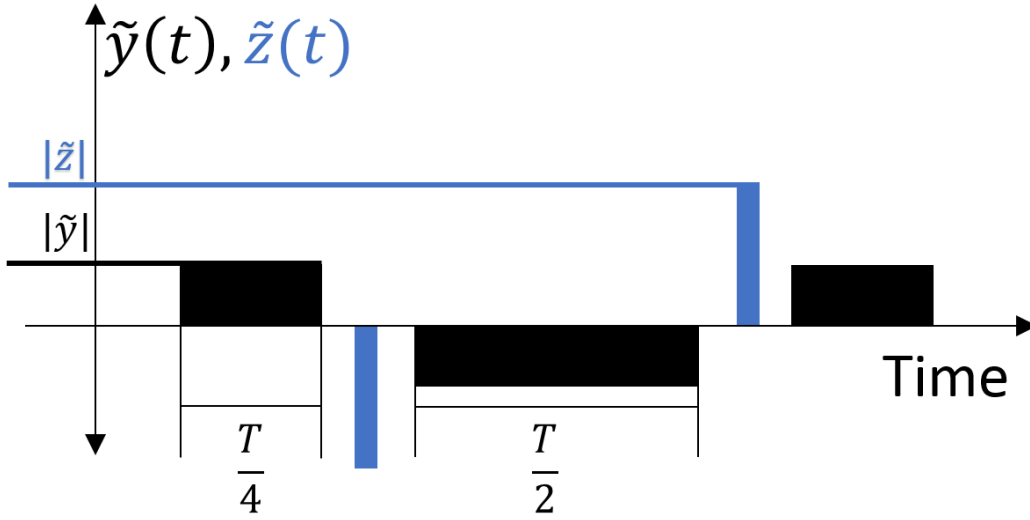


Figure 30: Probing Period Detail.

The probing signal is propagated through the systems causing change in F_{xi} , the changing F_{xi} interacts with the learning signal. The correlation between the probing signal and the changing F_{xi} updates the W_{ji} setting. Using the earlier non-linear approximation, the conductance modified due to learning becomes simplified to the following.

$$\Delta W_{j_i'} \alpha \frac{T|\tilde{z}|}{2} (g_{i'}^0 - g_{i'j}^{(1)}|\tilde{y}| + g_{i'j}^{(2)}|\tilde{y}|^2) \quad (40)$$

The change in the system dynamic can be quantified by comparing the change in $\frac{dF_{xi}}{dt}$ before and after the learning period. The change caused by applying the ΔW_{ji} amount is denoted by $\Delta \dot{F}_i$.

$$\Delta \dot{F}_{xi} = \dot{F}_{xi}(W_{ji'}^0 + \Delta W_{ji'}, F_{xi}^*) - \dot{F}_{xi}(W_{ji'}^0, F_{xi}^*) = \sum_{ji'} g_{ij}^{(1)} \Delta W_{ji'} F_{xi'}^* + O(\Delta W_{ji'}^2) \quad (41)$$

The dominant effect of the conductance modification on \dot{F}_{xi} occurs on the device that processes the F_{xi} signal. In other words, the covariance between \dot{F}_{xi} and $F_{xi'}$ is largest when $i = i'$. Although it does not have to be necessarily true, the assumption offers a tangible explanation of the learning process.

$$\Delta \dot{F}_i \alpha \frac{TF_{xi}|\tilde{z}|}{2} \left(g_{ji}^{(2)} g_{ji}^{(1)} |\tilde{y}|^2 - g_{ji}^{(1)2} |\tilde{y}| + g_i^{(0)} g_{ji}^{(1)} \right) + O(\Delta W_{ji}^2 (|\tilde{z}|^2, |\tilde{y}|^4, T^2, \dots)) \quad (42)$$

Referring to the modification of $\Delta \dot{F}_i$ as a function of $|\tilde{y}|$ can give insight into the effect of the learning operating on the system. In general, increasing $|\tilde{y}|$, $|\tilde{z}|$, or T will not always result in favorable effects. Modification of these parameters in an active sense is outside the scope of this report but can produce favorable results. However, following a paradigm of a slow learning rate will most likely produce improvement, but not necessarily at the fastest rate.

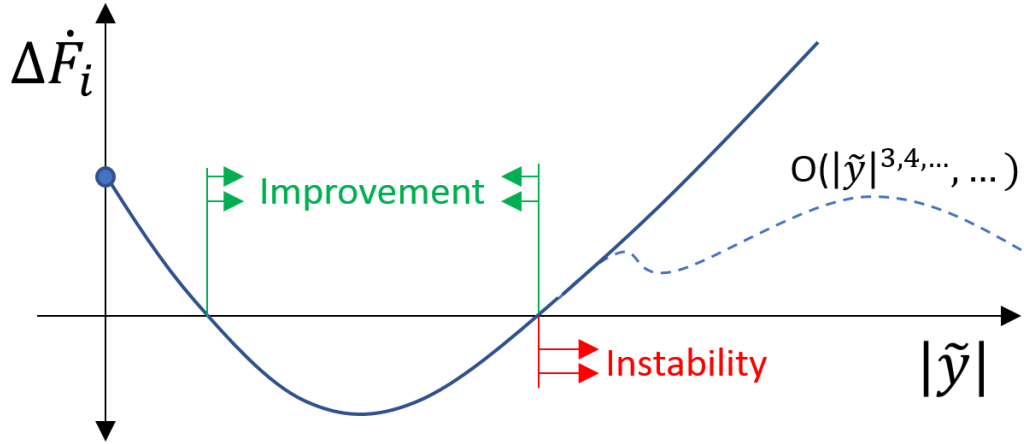


Figure 31: Effect of learning amplitude on system improvement.

There are several major considerations that this relation uncovers. First, the modification amplitude of $\Delta\dot{F}_i$ is automatically showed as F_{xi} is reduced. In that way, as the system approached the minimum energy condition, the learning amplitude will be reduced. Although the learning amplitude $|\tilde{y}|$ and $|\tilde{z}|$ are constant, higher order modification of the feedback amplitude is not required for a modification of the learning rate.

The second observation is that the zero order zero, under null actuation, $g_i^{(0)}$, can have an influence on the learning effect. With strong enough $g_i^{(0)}$ effect the learning can have negligible, favorable, or adverse effect. In general, the influence of the actuation signal can outweigh the environment effect. In the situation where the environment effect dominates, the actuation will be ineffective at modifying the system performance. In realistic situations, $|\tilde{y}|$, can be increased to overcome environment effects. Perhaps, in an active style.

The third observation is that increasing the exploration signal $|\tilde{y}|$ does not necessarily improve the learning speed. Higher order considerations, captured by the non-linear terms show

that as $|\tilde{y}|$ the $\Delta\dot{F}_i$ response may become unfavorable or have negligible effect when applying large amplitudes of learning. The learning amplitude may be modified in an active style to increase the improvement rate in a higher order effect.

The fourth observation is only apparent when including $O\left(\Delta W_{ji}^2 (|\tilde{z}|^2, |\tilde{y}|^4, \dots)\right)$. In the current model, it appears that increasing $|\tilde{z}|$ or T will only amplify the modification amount. However, due to the higher order terms, increasing the effect without bound will have adverse effects, similarly to modification of $|\tilde{y}|$. These terms may also be modified in an active style.

4.4 Performance Benchmarking

The human is an example of an intelligent system, with many similarities to the SNIC. Due to the heavy similarity it is fair to compare and benchmark the performance of the SNIC to human. The performance of the learning process of the SNIC can be analyzed to show comparable, but slightly better, performance with respect to humans performing the same task. To be specific, the systems are analyzed and modeled the same way, under the same experimental conditions. The humans were not given any prior knowledge about the system dynamics. The actuation and sensor information supplied to both systems are the same. There are four metrics to compare the relative performance of the two intelligent systems.

- Learning Accuracy
- Total Learning Improvement
- Learning Speed
- Learning Oscillation

The average parameters, and variation can be compared over comparing 5 experiments by the human and SNIC each. Variation in initial setting of the SNIC, drone, and human experience, comparing a single experiment is not appropriate.

4.4.1 Learning Accuracy

Learning accuracy refers to how close to the optimum logic setting the intelligent system's learning achieves. The optimum learning setting is found based upon the earlier modeling and fitting results. A set of fitting parameters, σ , can be used to calculate an optimum learning modification, \hat{V}_{ji} . The optimal configuration can be compared to the final system modification, V_{ji}^{Final} . The format of the comparison is equivalent to the modelling error quantification.

$$E_V = \frac{RMS(V_{ji}^{Final} - \hat{V}_{ji})^2}{RMS(\hat{V}_{ji})^2} \quad (43)$$

\hat{V}_{ji} , the optimal modification the device, can be calculated by minimizing the non-linear model on average. This represents the total modification resulting in $\frac{d\Pi}{dt}$ to be at a minimum configuration. Effectively, this condition will have optimum logic to drive the system it its best configuration the fastest possible.

$$\left. \frac{d}{dV_{ji'}} \frac{d\Pi}{dt} \right|_{V_{ji'} = \hat{V}_{ji'}} = \left. \frac{d}{dV_{ji'}} A\sigma \right|_{V_{ji'} = \hat{V}_{ji'}} = 0 \quad (44)$$

When the system is performing optimally, $\Pi = 0$, the condition $\frac{d}{dV_{ji'}} \frac{d\Pi}{dt} = 0$ is satisfied.

At the $\Pi = 0$ setting, adjusting the tuning, $V_{ji'}$, logic will not result in any change in performance

because $\sum_{ij} W_{ji} F_{xi} = 0$ and $F_{xi} = 0$. The optimal setting, \hat{V}_{ji} , can be found using the following relation.

$$\hat{V}_{ji} = \frac{\sum_i \sigma_{ij}^{F^2 V}}{-2 \sum_i \sigma_{ij}^{F^2 V^2}} \quad (45)$$

For each experiment there are 8 total \hat{V}_{ji} values. The total error is taken using the E_V relation to quantify the total modification, rather than each individual node. When comparing the average value for both the SNIC and human the SNIC is roughly twice as accurate, on average. However, both systems have large variation but will always result in a modification that improves the network setting, $E_V < 1$.

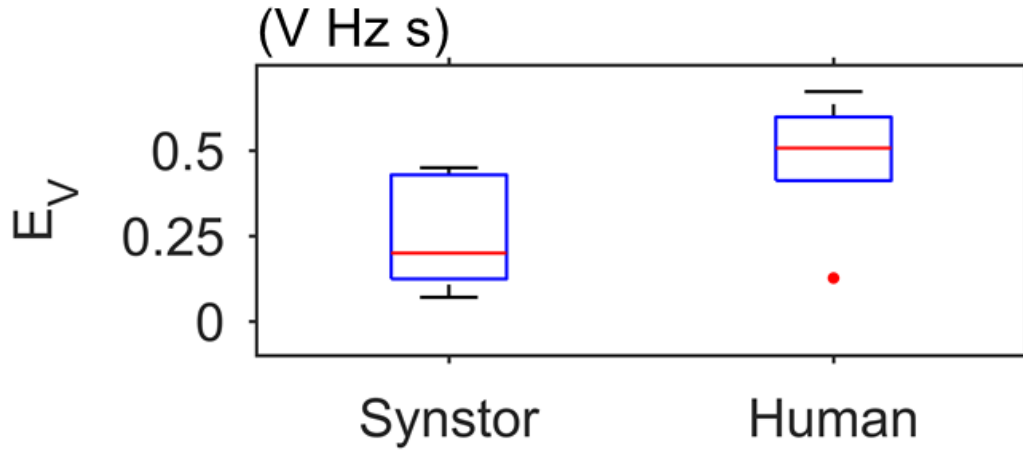


Figure 32: SNIC and Human tuning accuracy.

4.4.2 Total Learning Improvement

The modification of the network processing logic, in biological and synthetic sense, modify the system F_{xi} behavior. The effect of learning, represented by $\dot{\Pi}^{Learn}(t)$ can be

integrated over the experiment to quantify the learning improvement effect on the overall behavior. $\dot{\Pi}^{Learn}$ can be directly calculated by, using both experimental and fitting data.

$$\begin{aligned} \dot{\Pi}^{Learn} = & ij i' \sigma_{iji'}^{F^2 V} V_{ji'}(t - \tau) F_{xi'}^2(t - \tau) + \sigma_{iji'}^{F^2 V^2} V_{ji'}^2(t - \tau) F_{xi'}^2(t - \tau) \\ & + \sigma_{iji'}^{FV} V_{ji'}(t - \tau) F_{xi'}(t - \tau) \end{aligned} \quad (46)$$

The total improvement due to learning $\Delta \Pi^{Learn}$ is normalized by the average $F_{xi} = \bar{F}_x$ over the whole experimental time. The normalized amount is defined as $\Delta \Pi^{Learn, ND}$. The normalization is to make every experiment comparable, relative to the average energy.

$$\Delta \Pi^{Learn, ND} = \frac{\Delta \Pi^{Learn}}{\bar{F}_x} = \frac{\int_0^{t^{final}} \dot{\Pi}^{Learn} d\tau}{\frac{1}{4 t^{final}} \int_0^{t^{final}} \Pi d\tau} \propto \frac{\int_0^{t^{final}} \dot{\Pi}^{Learn} d\tau}{\int_0^{t^{final}} \Pi d\tau} \quad (47)$$

Both systems exhibit negative change in energy due to learning. This agrees with the analytical results showing that the learning process should improve the system. The experimental result verifies that the learning process was effective at improving the system, on average. Both intelligent systems we able to learn to improve the system in every experiment, although with different amplitudes. The improvement induced by the circuit leaning was of higher amplitude as compared to the human. However, both exhibit wide variation.

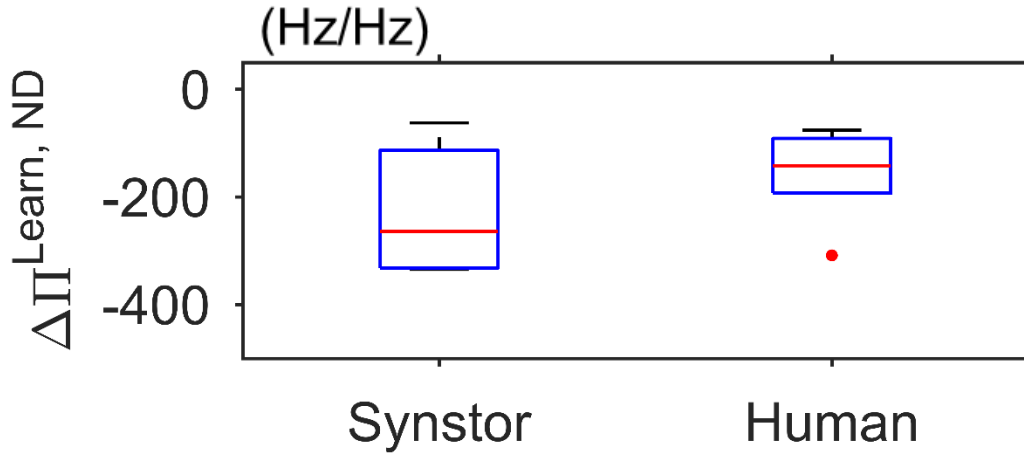


Figure 33: Improvement caused by learning comparison.

4.4.3 Learning Speed

The learning speed for both systems are similar with the SNIC performing slightly better.

The learning speed refers to the time it takes for the system to reach a steady, $\frac{dV_{ji}}{dt} \sim 0$. The

learning time, T_{ji}^{Learn} , can be estimated by using the V_{ji} data directly.

$$\frac{dV_{ji}}{dt} = \frac{-1}{T_{ji}^{Learn}} (V_{ji} - V_{ji}^*) \quad (48)$$

Both variables, T_{ji}^{Learn} and V_{ji}^* are fitted in a least square sense from the experimental data. V_{ji}^* is not necessarily equal to V_{ji}^{Final} because this is only a linear approximation of the change of V_{ji} without consideration of the effects due to F_{xi}, W_{ji}^0 , time delays, For each (i,j) node in the network a T_{ji}^{Learn} time can be calculated. As convention, only devices that have a high enough amplitude of V_{ji} while having a correlation coefficient large enough will be counted.

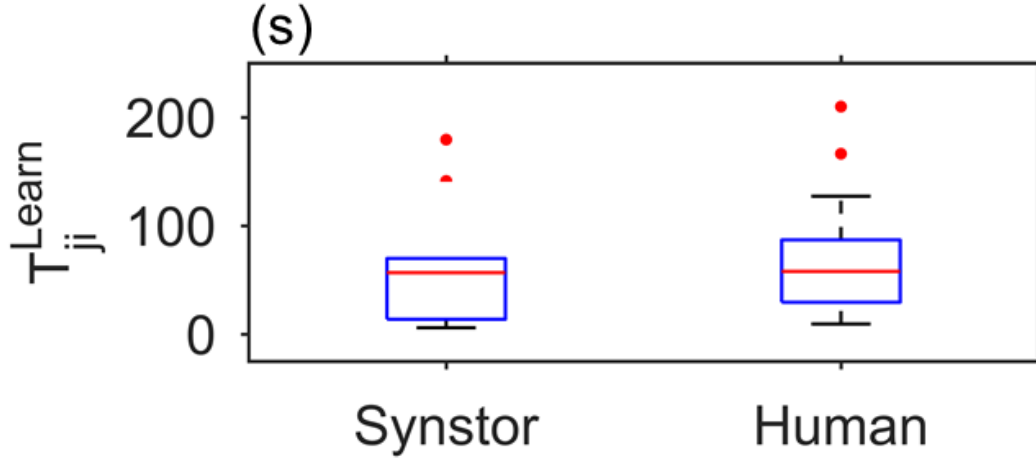


Figure 34: Learning Time.

The learning times for the human and SNIC have a similar median value but the human exhibits larger variation. The large human variation is probably due to the skill levels of the user at similar tasks. Humans that have very little experience in similar tasks will take longer to achieve a good learning setting. Users that have better native intuition about the system tend to perform at similar levels to the SNIC.

4.4.4 Learning Oscillation

Once the system has reached a state where there is only zero-mean noisy change in V_{ji} the noise injected into F_{xi} can be quantified.

$$\dot{F}_{xi}^{Osc} = \dot{F}_{xi}(V_{ji} = \hat{V}_{ji} + \tilde{V}_{ji}, \bar{F}_x) - \dot{F}_{xi}(V_{ji} = \hat{V}_{ji}, \bar{F}_x) \quad (49)$$

Comparing the change in \dot{F}_{xi} with and without the contribution of \tilde{V}_{ji} variation can quantify the effect of a continuous learning process on the system. The total amplitude of the

learning oscillation on \dot{F}_x is taken by quantifying the overall RMS amplitude of the oscillation on all the sensing signals.

$$\dot{F}_x^{Osc} = RMS(\bar{F}_x^2 \sum_{j'i'} E[\tilde{V}_{ji}^2] \sigma_{ij'i'}^{F^2 V^2}) \quad (50)$$

The oscillation of the learning signal, $E[\tilde{V}_{ji}^2]$, is found by synthesizing the V_{ji} directly.

The oscillation amplitude is found by finding the later parts of the experiment times from $t = t^*$ to $t = t^{Final}$ that have a high standard deviation of V_{ji} relative to the mean signal over the same interval. This signal is maximized when the detonator is approaching 0.

$$t^* = \max_{t'} \frac{std(V_{ji}(t' \rightarrow t^{Final}))}{|mean(V_{ji}(t' \rightarrow t^{Final}) - V_{ji}(t'))|} \quad (51)$$

$$E[\tilde{V}_{ji}^2] \approx std(V_{ji}(t' \rightarrow t^{Final})) \quad (52)$$

The oscillation is compared for both intelligent systems. In this case, the SNIC network has a higher oscillation amplitude, on average. The human tends to have higher learning frequency with a small amplitude of modification due to each learning operation. In comparison, the SNIC have a slower learning rate at a higher amplitude. The larger amplitude results in a larger $E[\tilde{V}_{ji}^2]$ amount.

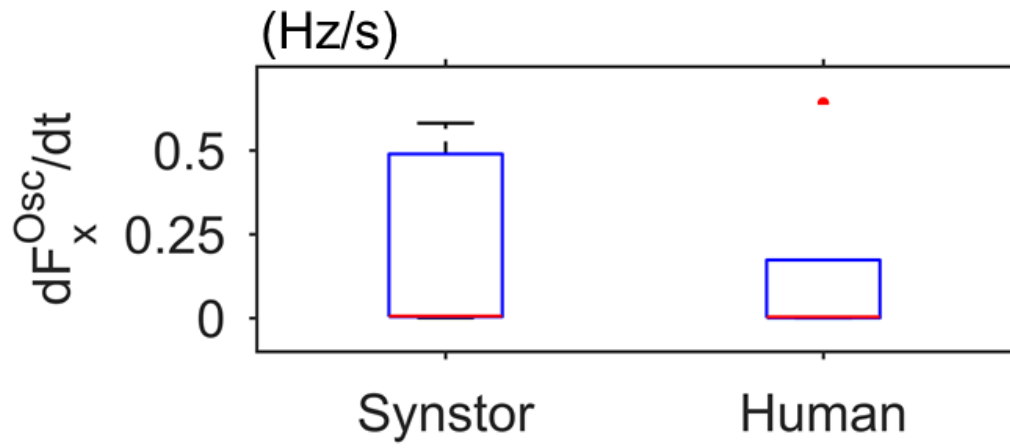


Figure 35: Oscillation induced by learning comparison.

5.0 Conclusion

The brain is a powerful computer with the ability to process large numbers of signals in real-time, while simultaneously modifying its logic via a learning process. The human brain, is unlike conventional computers, which often are larger, less efficient, and operate in a serial style. Conventional computers, often has trouble in regimes where biological computers excel. For example, tasks that have high dimensional signals, require real-time processing, and operate in unknown and changing environments are all tasks where biological computers has a strong advantage. The ‘curse of dimensionality’ and the ‘von Neumann bottleneck’ are core problems transistor based computers are unable to escape.

The biological neural computers can exhibit intelligence. Intelligence is defined as the ability to vary its state or action in response to varying situations, varying requirements, and experience. [5] The type of modification is referred to as ‘learning’, transistor-based computing has difficulty when tasked with problems that require intelligence. At the fundamental level, the synapse is much different from a transistor, these differences enable intelligence in the larger system. Taking biological inspiration from the synapse and using it to developing a neural computer exhibiting intelligent function is certainly possible.

The goal of the project is to develop a synaptic resistor device and accompanying neuromorphic system that exhibits self-programming intelligent functions, causing improvement in unknown systems, while circumventing the ‘curse of dimensionality’ and ‘von Neumann bottleneck’.

For demonstration, the synaptic device, a *synstor*, was interfaced with an unmanned air vehicle in an unknown and changing environment. There was no pre-training of the *synstor* or

system. Initially, the system behaved poor, over time, the performance improved due to a parallel, real-time learning process. The network exhibited intelligent functions by modifying its or action in response to varying system and environment. The elimination of external computation and memory circuits is a central aspect of the demonstration. It may be possible for conventional computer to improve the system performance, but the computing style will never be able to circumvent the ‘curse of dimensionality’ and ‘von Neumann bottleneck’.

The resulting experiments were analyzed to quantify and understand the learning process. The learning process can be understood as a gradient descent algorithm. The modification to each device is related to the correlation between its actuation signal and the systems improvement. Although the devices do not numerically process signal to calculate the correlation directly. Instead, the physical properties of the device and operating mode spontaneously implement the modification. the that way, the learning process can cause improvement without the need of external circuits and memory. The learning process in this experiment is of fixed pulsing amplitude, however there may be several advantages to higher-order modification. Even in the fixed amplitude, ‘simple’, case the improvement is quite evident.

The total improvement, learning rate, oscillation, and accuracy were documented. The ability of the SNIC was compared to humans, another intelligent system. The benchmarks for each system, are compared statistically over several experiments. In general, the SNIC performance comparable to a human, but slightly superior on average.

REFERENCES

- [1] A. M. Turing, "Computing machinery and intelligence," *Mind*, p. 28, 1950.
- [2] R. Reddy, "Foundations and Grand Challenges of Artificial-Intelligence," *Ai Mag* , pp. 9-21, 1988.
- [3] D. O. Hebb, "The organization of behavior; a neuropsychological theory.," *A Wiley book in clinical psychology*, p. 335, 1949.
- [4] N. a. D. Y. Caporale, "Spike Timing–Dependent Plasticity: A Hebbian Learning Rule," *Annual Review of Neuroscience*, vol. 31, pp. 25-46 , 2008.
- [5] D. a. W. R. J. Floreano, "Science, technology and the future of small autonomous drones," *Nature*, vol. 521, no. 7553, pp. 460-466, 2015.
- [6] W. Zhao, B. H. Kim, A. Larson and R. Voyles, "FPGA implementation of closed-loop control system for small-scale robot," in *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, Seattle, WA, USA, 2005.
- [7] H. a. S. T. Hagaras, "Intelligent Learning and Control of Autonomous Robotic Agents," *Information Sciences*, vol. 145, no. 1-2, pp. 1-12, 2002.

- [8] D. e. a. Silver, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, pp. 484-489, 2016.
- [9] J. e. a. Biamonte, "Quantum machine learning," *Nature*, vol. 549, pp. 195-202, 2017.
- [10] R. & R. C. Bellman, *Dynamic Programming*, Princeton University Press, 1957.
- [11] J. Backus, "Can Programming Be Liberated from Von Neumann Style - Functional Style and Its Algebra of Programs," *Commun Acm*, vol. 21, pp. 613-641, 1978.
- [12] J. D. H. S. M. M. Erich Strohmaier, "Top 500," Top 500, 1 June 2016. [Online]. Available: <https://www.top500.org/lists/2016/06/>. [Accessed 1 October 2017].
- [13] D. Hebb, *The organization of behavior, a neuropsychological theory*, Wiley, 1949.
- [14] Z. H. S. E. J. J. & B. S. Chen, "Correlative Learning: A Basis for Brain and Adaptive Systems.," *Correlative Learning: A Basis for Brain and Adaptive Systems* , no. doi:10.1002/9780470171455, pp. 1-448, 2007.
- [15] S. Zeki, "A massively asynchronous, parallel brain," *Philos T R Soc B* , vol. 370, pp. 103-116, 2015.
- [16] R. S. M. & S. S. M. Llinas, "Transmission by presynaptic spike-like depolarization in the squid giant synapse," *Proc. Natl. Acad. Sci*, vol. USA 79, pp. 2415-2419, 1982.

- [17] S. S. J. P. & W. R. S. T. Kumar, "Chaotic dynamics in nanoscale NbO₂ Mott memristors for analogue computing," *Nature*, vol. 548, pp. 318-321, 2017.
- [18] D. D, "Do we have brain to spare?," *Neurology*, vol. 64, no. 12, 2005.
- [19] A. L. Shimpi, "Anand Tech," 31 October 2012. [Online]. Available: <https://www.anandtech.com/show/6421/inside-the-titan-supercomputer-299k-amd-x86-cores-and-186k-nvidia-gpu-cores>. [Accessed 1 October 2017].
- [20] "Psychology," New York, Worth Publishers, 1995, p. 115.
- [21] C. Koch, "Computation and the Single Neuron," *Nature*, vol. 385, pp. 207-210, 1997.
- [22] M. Fischetti, "Graphic Science: Computers vs. Brains," *Scientific American Magazine*, vol. 305, pp. 104-104, 2011.
- [23] A. Watson, "Why Can't a Computer be More Like a Brain?," *Science*, vol. 277, pp. 1934-1936, 1997.
- [24] G. Indiveri, E. Chicca and R. Douglas, "A VLSI Array of Low-Power Spiking Neurons and Bistable Synapses with Spike-Timing Dependent Plasticity," *IEEE Trans. Neural Netw*, vol. 17, pp. 211-221, 2006.
- [25] M. Mahowald and R. Douglas, "A Silicon Neuron," *Nature*, vol. 354, pp. 515-518, 1991.

- [26] R. S. M. & S. S. M. Llinas, "Transmission by presynaptic spike-like depolarization in the squid giant synapse," *Proc. Natl. Acad. Sci.* , vol. USA 79, pp. 2415-2419, (1982).
- [27] G. & P. M. Bi, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annu. Rev. of Neurosci*, vol. 24, pp. 139-166, 2001.
- [28] K. D. M. & L. F. A. Sen Song, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticityNature," *Nature Neuroscience*, vol. 3, no. doi:10.1038/78829, p. 919–926, 2000.
- [29] C. Edwards, "Growing pains for deep learning," *Communications of the ACM*, vol. 58 (7), no. 7, p. 14–16, 25 June 2015.
- [30] B. e. a. Benjamin, "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations.," *P IEEE*, vol. 102, pp. 699-716, 2014.
- [31] P. A. e. a. Merolla, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science* , vol. 345, pp. 668-673, 2014.
- [32] S. B. e. a. Eryilmaz, "Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array," *Front Neurosci-Switz* , vol. 8 , 2014.
- [33] J. v. Neumann, "First draft of a report on the EDVAC," *IEEE Annals of the History of Computing*, vol. 15, no. 4, pp. 27 - 75, 1993.

- [34] G. Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE Transactions on Information Theory*, vol. 14, no. 1, p. 55–63, 1968.
- [35] C. C. K. K. B. C. A. T. Y. C. AM Shen, "Analog neuromorphic module based on carbon nanotube synapses," *ACS nano*, vol. 7, no. 7, pp. 6117-6122, 2013.
- [36] K. K. A. T. D. L. Y. C. AM Shen, "Doping Modulated Carbon Nanotube Synapstors for a Spike Neuromorphic Module," *Small*, vol. 11, no. 13, pp. 1571-1579, 2015.
- [37] A. T. C. C. D. L. A. S. Y. C. K Kim, "Bioinspired neuromorphic module based on carbon nanotube/C60/polymer composite," *Journal of Composite Materials*, vol. 49, no. 15, pp. 1809-1822, 2015.
- [38] J. S. a. E. E. B. Campbell, "Schottky barriers in carbon nanotube-metal contacts," *Journal of Applied Physics*, vol. 111101, p. 110, 2011.
- [39] B. Hassard, "Bifurcation of periodic solutions of the Hodgkin-Huxley model for the squid giant axon," *Journal of Theoretical Biology*, vol. 71, no. 3, pp. 401-420, 1978.