Proceedings Paper:
Liu, Junxiu, Harkin, Jim, McDaid, Liam et al. (2016) Self-repairing mobile robotic car using astrocyte-neuron networks. In: 2016 International Joint Conference on Neural Networks, IJCNN 2016. 2016 International Joint Conference on Neural Networks, IJCNN 2016, 24-29 Jul 2016 Proceedings of International Joint Conference on Neural Networks. IEEE, CAN, pp. 1379-1386.

https://doi.org/10.1109/IJCNN.2016.7727359

Self-Repairing Mobile Robotic Car using Astrocyte-Neuron Networks

Junxiu Liu, Jim Harkin, Liam McDaid
School of Computing and Intelligent Systems,
University of Ulster, Magee Campus,
Derry, Northern Ireland, UK, BT48 7JL
{j.liu1, jg.harkin, lj.mcdaid}@ulster.ac.uk

David M. Halliday, Andy M. Tyrrell, Jon Timmis
Department of Electronics,
University of York, Heslington,
York, UK, YO10 5DD
{david.halliday, andy.tyrrell, jon.timmis}@york.ac.uk

***Abstract*—A self-repairing robot utilising a spiking astrocyte-neuron network is presented in this paper. It uses the output spike frequency of neurons to control the motor speed and robot activation. A software model of the astrocyte-neuron network previously demonstrated self-detection of faults and its self-repairing capability. In this paper the application demonstrator of mobile robotics is employed to evaluate the fault-tolerant capabilities of the astrocyte-neuron network when implemented in a hardware-based robotic car system. Results demonstrated that when 20% or less synapses associated with a neuron are faulty, the robot car can maintain system performance and complete the task of forward motion correctly. If 80% synapses are faulty, the system performance shows a marginal degradation, however this degradation is much smaller than that of conventional fault-tolerant techniques under the same levels of faults. This is the first time that astrocyte cells merged within spiking neurons demonstrates a self-repairing capabilities in the hardware system for a real application.**

Keywords—astrocyte; fault-tolerant; robot car; spiking neural networks; repair; self-adaptive.

I. INTRODUCTION

As the hardware device geometries scale down, the reliability of devices become lower, exhibiting higher risks of faults occurring post-manufacturing. Faults in electronic systems include permanent faults from a wear-out effect, rupture in a manufacturing process and temporary faults caused by power supply fluctuations and radiation effects. Therefore engineers must aim to design reliable systems on unreliable fabrics. Traditional fault-tolerant approaches include redundancy models [1], [2], error correction techniques [3], and reconfiguration computing [4], [5] etc. These approaches only provide limited levels of reliability due to the inherent architectural constraints. Recent approaches start to look to biology for inspiration in exploiting the efficient and reliable computing properties. Bio-inspired systems provide high levels of parallel computing which model varied levels of computationally efficient biological systems. Various approaches have been proposed, e.g. using VLSI technique to implement neuro-biological architectures [6] (i.e. neuromorphic chips); bio-inspired systems for learning and information processing (e.g. visual object recognition and tracking [7]–[9]) and neuro-inspired controller [10] and robots [11]. These systems are designed and implemented based on large-scale high density chips, therefore the system reliability is an increasing design challenge [12]–[14], which requires the system can be adaptive to faulty conditions post-manufacturing. The bio-inspired system has shown the potential to mimic the fault-tolerant computing capability in the brain [5], [15]–[17]. For example in our previous work [18], a bio-inspired fault detection and self-correction mechanism for the neuro-inspired PID controller [10] is proposed. It used the synapse models with excitatory and inhibitory responses to detect the faults of spike-based PID controller. This approach and related approaches (e.g. [15]) underpin that the future electronic system can harness similar mechanisms that are found in biology to fully realise the fault-tolerant and self-repairing capabilities for the hardware systems. Recent research have highlighted that one type of glial cells, astrocyte, continually couples with synapses and neurons, and plays a curial rule in the brain re-wiring [19]. The authors have previously developed a computational model for the spiking astrocyte-neuron networks (SANN) which captures this behaviour [20], [21] and have demonstrated how astrocytes cells merged within spiking neurons can perform distributed and fine grained self-repair under the presence of faults. In this paper, we go further and propose a mobile robot car hardware architecture, which demonstrates the self-repairing mechanism of the SANN in hardware and in particular, evaluates the system performance under various fault conditions in this real application. This is the first time that astrocyte cells merged within spiking neurons demonstrate self-repairing capabilities in the hardware system for a real application. Section II outlines the authors' current self-repairing SANN software model. Section III introduces the hardware architecture of the self-repairing robot car and section IV provides the experimental results under various faulty conditions to demonstrate the repair capability. Section V provides a summary and conclusion.

II. SPIKING ASTROCYTE-NEURON NETWORKS

This section provides a brief background on how astrocytes interact with spiking neurons. Astrocytes enwrap many synapses which are connected to a neuron, and it is this process by which bi-directional communication is performed. This type of synapse is known as a tripartite synapse. In a tripartite synapse, the arrival of an action potential axon, releases glutamate across the cleft and binds to receptors on the post-synaptic dendrite causing a depolarization of the post-synaptic neuron; then voltage gated channels on the dendrite allow the influx of calcium (Ca^{2+}) into the dendrite causing endocannabinoids to

be synthesized and subsequently released from the dendrite. The 2-arachidonyl glycerol (2-AG), a type of endocannabinoid and retrograde messenger, is known to feed back to the pre-synaptic terminal in two ways: (a) *Directly*. 2-AG binds directly to type 1 Cannabinoid Receptors (CB1Rs) on the pre-synaptic terminal. This results in a decrease in transmission probability (PR) and is termed Depolarization-induced Suppression of Excitation (DSE); and (b) *Indirectly*. 2-AG binds to CB1Rs on an astrocyte which enwraps the synapse increasing IP_3 levels within the astrocyte and triggering the intracellular release of Ca^{2+}. This results in the astrocytic release of glutamate which binds to pre-synaptic group I metabotropic Glutamate Receptors (mGluRs). Such signalling results in an increase of synaptic transmission PR termed e-SP. The associated PR of each synapse is affected by the DSE and e-SP (1):

$$PR(t) = (\frac{PR(t_0)}{100} \times DSE(t)) + (\frac{PR(t_0)}{100} \times eSP(t)) \quad (1)$$

where $PR(t_0)$ is the initial PR. The signal exchange and chemical transmission in the tripartite synapse are briefly described as follows: when a post synaptic neuron fires, 2-AG is released and described in (2) by:

$$\frac{d(AG)}{dt} = \frac{-AG}{\tau_{AG}} + r_{AG}\delta(t - t_{sp}) \quad (2)$$

where AG is the quantity of the released 2-AG; τ_{AG} and r_{AG} is the decay and production rate of 2-AG, respectively; t_{sp} is the time of the post-synaptic spike. When the 2-AG binds to CB1Rs on the astrocyte, IP_3 is generated which is dependent on the amount of released 2-AG and is given by (3).

$$\frac{d(IP_3)}{dt} = \frac{IP_3^* - IP_3}{\tau_{ip_3}} + r_{ip_3}AG \quad (3)$$

where IP_3 is the quantity within the cytoplasm, IP_3^* is the baseline of IP_3 when the cell is in a steady state and receiving no input, τ_{ip_3} and r_{ip_3} is the decay and production rate of IP_3, respectively. The Ca^{2+} dynamics within the cell is described by (4).

$$\frac{d(Ca^{2+})}{dt} = J_{chan}(Ca^{2+}, h, IP_3) + J_{leak}(Ca^{2+}) - J_{pump}(Ca^{2+}) \quad (4)$$

where J_{chan} is the IP_3 and Ca^{2+} -dependent Ca^{2+} release, J_{pump} is the amount of Ca^{2+} pumped from the cytoplasm into the Endoplasmic Reticulum (ER), J_{leak} is the Ca^{2+} which leaks out of the ER. In this approach, a linear correspondence is assumed between the released 2-AG and the DSE, and given by:

$$DSE = AG \times K_{AG} \quad (5)$$

where AG is the amount of released 2-AG, K_{AG} is a scaling factor used to convert the level of 2-AG into the desired negative range. The intracellular astrocytic calcium dynamic are used to regulate the release of glutamate from the astrocyte which drives e-SP. The quantity of released glutamate targeting group I mGluRs is given by:

$$\frac{d(Glu)}{dt} = \frac{-Glu}{\tau_{Glu}} + r_{Glu}\delta(t - t_{Ca}) \quad (6)$$

where Glu is the quantity of glutamate, τ_{Glu} and r_{Glu} are the decay and production rate of glutamate respectively, and t_{Ca} is the time of the Ca^{2+} threshold crossing. The level of e-SP is dependent on the quantity of glutamate, which is modelled by (7) where τ_{eSP} is the decay rate of e-SP, m_{eSP} is a weighting constant.

$$\tau_{eSP}\frac{d(eSP)}{dt} = -eSP + m_{eSP}Glu(t) \quad (7)$$

For the neuron model, the Leaky Integrate and Fire (LIF) [22] is used due to its simplistic nature, as given by:

$$\tau_m \frac{dv}{dt} = -v(t) + R_m \sum_{i=1}^{m} I_{syn}^i(t) \quad (8)$$

where τ_m and v is the time constant and membrane potential respectively, R_m is the membrane resistance, $I_{syn}^i(t)$ is the current injected to the neural membrane at synapse i. The firing threshold voltage is 9mv. The neuron model also includes a refractory period of 2ms.

The synapse model is a probabilistic-based model. A uniformly distributed pseudorandom number generator is used to generate a random number which is denoted by $rand$. If it is less than or equal to the release PR, a fixed current I_{inj} is injected into the LIF neuron shown by:

$$I_{syn}^i(t) = \begin{cases} I_{inj}, rand \leq PR \\ 0, rand > PR \end{cases} \quad (9)$$

The signal pathways of DSE in (5) and e-SP in (7) modulate the PR of the synapse as shown by (1). More details on the self-repair mechanism in the SANN can be found in our previous works [20], [21].

III. Self-Repairing Robot Car

In this section, the self-repairing robot car hardware architecture is presented. The concept and principle of spiking astrocyte-neuron networks are given firstly; then the self-repairing robot car hardware architecture and implementation are discussed in detail.

A. Self-repairing strategy of the spiking astrocyte-neuron networks

An example of a spiking astrocyte-neuron network fragment is given in Fig. 1 which illustrates the proposed self-repair paradigm. Astrocyte, *A*, enwraps the synapses connected to neurons *N1* and *N2*. The 2-AG (DSE) is a local signal associated with each synapse connected to N1 or N2. The e-SP is a global signal associated with all synapses connected to astrocyte *A*. To illustrate the self-repair concept, we first consider the healthy network in Fig. 1(a). In this case, the direct and indirect signals from DSE and e-SP compete at each synapse and PR reaches a stable state. In Fig. 1(b), a fault occurs at the synapses associated with N2, hence both direct and indirect retrograde feedback from N2 cease. This creates an imbalance in PR at all synapses associated with N2; however the PR of the healthy synapses associated with N2 are enhanced to restore the firing activity of N2. This is due to the indirect retro-

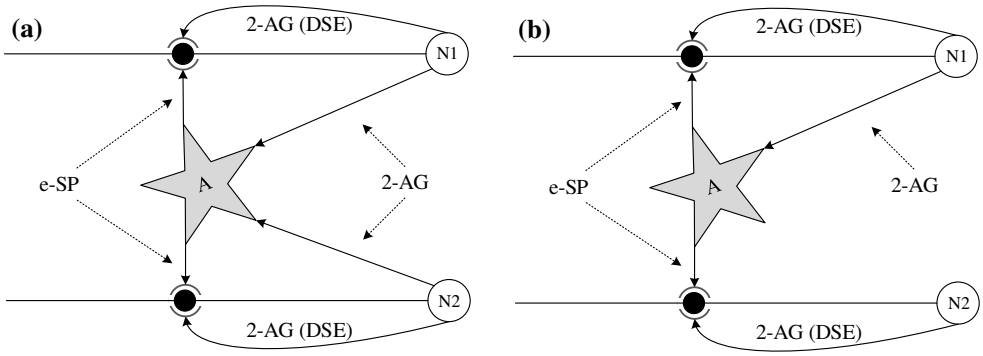

Fig. 1. Spiking astrocyte-neuron networks which illustrates endocannabinoid self-repair.

grade feedback from N1. This is the principle of self-repair which allows the detection of fault(s) in synapses and the catalyst for repair whereby PR of healthy synapses is increased. This process is regulated by the astrocyte. The astrocyte-neuron coupling system and self-repair mechanism were briefly discussed in this subsection however more details can be found in our previous work [20], [21].

B. Self-repairing mechanism of the robot car

The robot car hardware implementation is shown in Fig. 2, which consists of three parts – (a). The spiking astrocyte-neuron network FPGA implementation which is the controlling module for the robot car; (b). A robot car wheel mobile hardware module which includes the voltage level converter, motor driver circuit and motors; and (c). An FPGA hardware module which reads signal data and presents to monitoring software on a PC (this is used for aiding the design and performance analysis of the FPGA-based astrocyte-neuron network running on the robot). The self-repairing mechanism for the robot car is presented in detail in the following text.

Fig. 2(a) illustrates the FPGA hardware architecture of the spiking astrocyte-neuron network (SANN). It includes two neurons and one astrocyte. For each neuron, several synapses are associated. This small network fragment is used to control the robot car and allows demonstration of the repair principle in a real hardware application. A key component in the SANN is the *neuron facility* (e.g. Neuron #1 and #2 in Fig. 2(a)). The *neuron facility* receives the signals from synapses (shown by eqn. (9) and (8)) and outputs the spike signals, e.g. 2-AG and DSE. In the *neuron facility*, the LIF neuron model of (8) is used. When a post synaptic neuron fires, 2-AG is released by

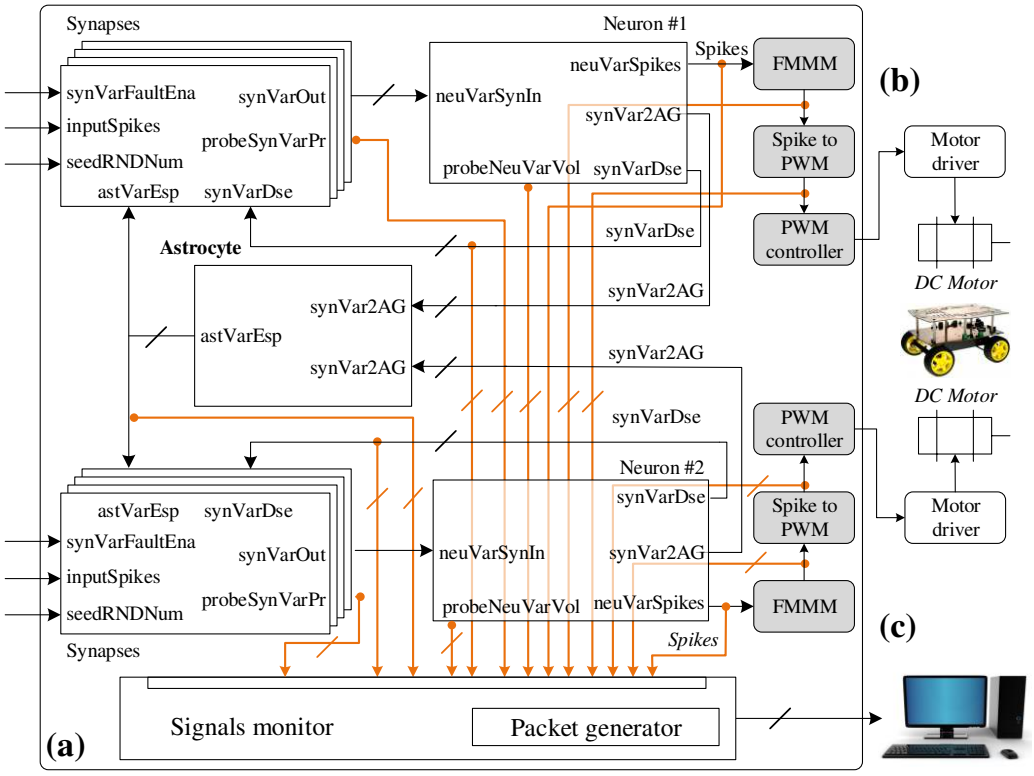

Fig. 2. Robot car hardware architecture. (a). Spiking astrocyte-neuron network FPGA implementation where FMMM denotes the frequency moving mean module. (b). Robot wheel car. (c). Signals collection and monitoring software.

(2) and DSE is generated by (5), which are the core process in the neuron facilities. The released 2-AG binds to the CB1Rs on the astrocyte process, which results in the indirect increase of synaptic transmission PR. However, the DSE has a direct effect on the pre-synaptic terminal, i.e. decreasing the synaptic transmission PR. The *astrocyte facility* in the SANN receives the signal of 2-AG from the *neuron facilities*, and outputs the signal of e-SP to the synapses to modulate the synaptic transmission PR. This process is described by the equations of (3), (4), (6), (7) and (1), which can be summarized as follows. After the 2-AG binds to CB1Rs of the astrocyte, the IP_3 is generated in the *astrocyte facility* (see (3)); Due to the IP_3 generation, the Ca^{2+} in the astrocyte has a dynamic change (see (4)), which is described by three channels - J_{chan}, J_{leak} and J_{pump}; then the intracellular astrocytic calcium dynamic regulates the release of glutamate (see (6)), which drives the e-SP (see (7)). The e-SP is the output signal of the *astrocyte facility*, which connects to *synapse facilities* and has the effects on the pre-synaptic terminals, i.e. increasing the synaptic transmission PR. The *synapse facility* is designed using a probabilistic-based synapse model in (9). It receives the input train spikes, a seed signal for the random number generator, the DSE/e-SP signals from neuron/astrocyte facilities; and has one output signal (i.e. synVarOut in Fig. 2(a)) which is connected to the *neuron facility*. As discussed previously, the synaptic transmission PR is regulated by the DSE and e-SP. Two PR adjustors are designed for the regulation process where the former (for DSE) is used to decrease the PR value and the latter (for e-SP) increases the PR value. Besides these, another input fault enable signal (i.e. synVarFaultEna) is designed to simulate a fault. If it is high, the PR value is set to be very low (e.g. 0.1 in this approach); otherwise the PR has an initial value of 0.5 and only effected by the DSE and e-SP.

The output signals of the SANN are the spikes from the *neuron facilities*. The spike outputs from the neurons are used to control the motors of the robot car. As the synapses use the probabilistic-based model, the frequency of the neuron output spike is not fixed and has fluctuations. In this approach, a moving mean algorithm is employed to smooth out the short-term frequency fluctuations and highlight the long-term trends, which is given by:

$$f' = \frac{1}{n}\sum_{i=0}^{n-1} f_{m-i} \tag{10}$$

where f' is the average frequency which is used to control the robot motor, m is the current time point, n is the size of the neuron frequency subset, and f_{m-i} is the neuron frequency at the time point of $(m-i)$. This calculation process creates a series of average frequency of different subset of the full neuron frequency data set. The frequency moving mean module (i.e. FMMM in the Fig. 2(a)) is designed to calculate the f'. It's hardware architecture is shown by Fig. 3.

The FMMM includes a FIFO with 2^k depth where $2^k = n$. When a new neuron frequency data enters the FIFO, it is added to the frequency summary (f_{sum}); if the FIFO is full and a neuron frequency data is removed from FIFO, the f_{sum} is subtracted by this value. Therefore, the FIFO component stores the neuron frequency data with the subset size of n, and the f_{sum} is the summary of all the data in this subset. In order to calculate

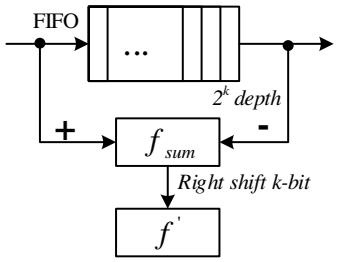

Fig. 3. FMMM architecture.

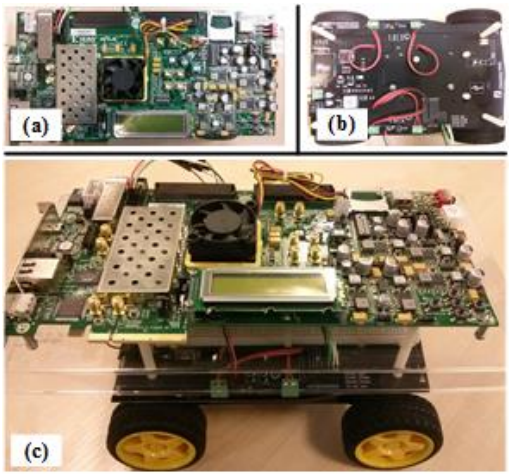

Fig. 4. The robot car with SANN system based on the FPGA device. (a). FPGA device. (b). Motor drive circuits. (c). Mobile robot car.

f', f_{sum} is shifted by k-bit to implement the division and the result is equal to the average frequency f'. Comparing to the direct implementation of the division operation in hardware, using the bit-shift operation saves the hardware resource cost and achieves a high computation speed, as the division operation in FPGA occupies lots of area and takes several clock cycles for completion. The size of neuron frequency subset and performance analysis is given in section IV.

The output of the FMMM, i.e. the average spike frequency, f', is connected to the *spike to PWM* module. This module converts the f' to the duty cycle value for the PWM controller, using the function of (11),

$$D = F(f') \tag{11}$$

where F is the mapping function, D is the duty cycle value. The duty cycle value is connected to the *PWM controller* which controls the motor of the robot car. The mapping function F and PWM controller design (e.g. the PWM frequency etc) depend on the application. The next section outlines the detailed function and parameters for the application, and provides a range of experimental results to demonstrate the self-repairing capabilities of the proposed robot car hardware system.

IV. Experimental Results

This section presents the robot car hardware setup and the experimental results on the proposed self-repairing hardware architecture. The SANN and motor controlling modules in Fig. 2(a) were implemented on a Xilinx Virtex-7 XC7VX485T-

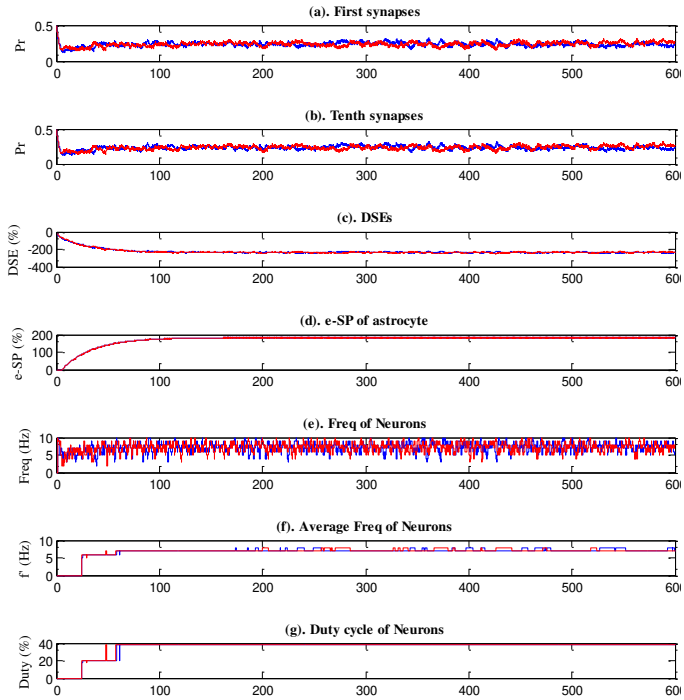

Fig. 5. Neuron #1 (blue) and #2 (red) and correponding outputs in the robot car under no fault.

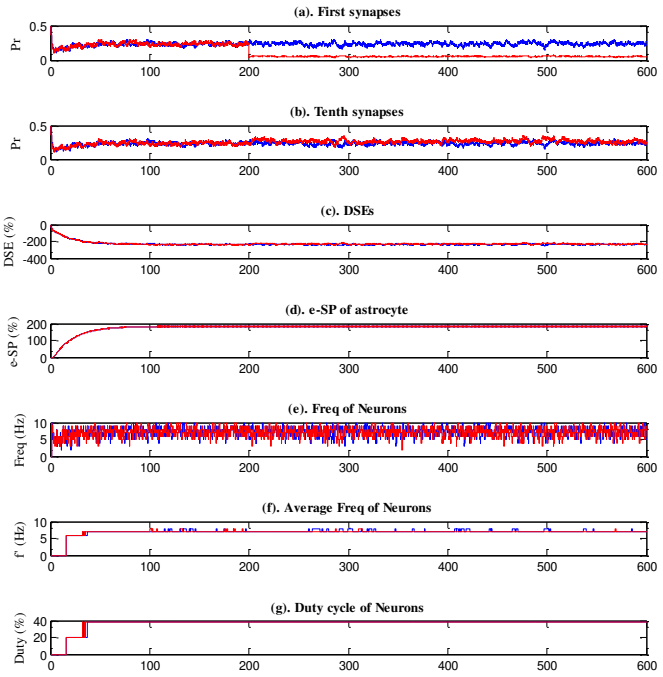

Fig. 6. Neuron #1 (blue) and #2 (red) and correponding outputs in the robot car under 20% fault rate.

2FFG1761C FPGA device, which includes the astrocyte-neuron network of two neurons and one astrocyte. Each neuron is associated with ten synapses and the frequency of the input spike train is set at ~10Hz in these experiments. This value sets the speed of the robot car and is selected based on the size of the robot. The initial transmission synaptic PR is 0.5. Faults are injected in the astrocyte-neuron network hardware by reducing the PR of selected synapses to a value of 0.1. This method enables permanent hardware faults such as open circuits to be emulated. In addition, temporary faults can also be emulated by temporally reducing PR. The fault rate denotes the percentage of faulty synapses associated with specific neuron, e.g. fault rate 40% means 4 of 10 synapses connected to a neuron are set to have PR=0.1, e.g. faulty. The output of the PWM controller is connected to an H-bridge motor driver circuits using L298 ICs, see Fig. 2(b) and Fig. 4(b). The speed and direction of the robot car are controlled by the two PWM controllers implemented in the FPGA device. The robot car is shown in the Fig. 4(c) with the FPGA platform (Fig. 4(a)) mounted. A signal monitoring framework in our previous work [23] is employed to probe hardware signals and collect data from circuits running on the FPGA device. The signal data is uploaded to a PC for real-time analysis. The signals of interested are highlighted (coloured brown) in Fig. 2.

A straight line moving task is set for the robot car. The output spike of neuron #1 and #2 are designed to drive the left and right wheels of the robot car, as illustrated by Fig. 2. In order to implement this task, for the FMMM, a 2^{14} depth FIFO is used to calculate the average frequency f', and the mapping function in (11) is given by a piecewise function where a duty cycle is given based on the value of average frequency f'. In this approach, the PWM frequency is 500Hz, resolution is 256 (i.e. $r = 8$), f'_{max} is 10Hz. Based on (11), given any average frequency f', the duty cycle (i.e. D) can be calculated. The value of D is then used to control the robot's left and right wheel motors.

For the first experiment, the robot car is tested under no faults, i.e. the fault rate is 0%. Neuron #1 and #2 receive the pre-synaptic stimuli, also coupled with the astrocyte via the 2-AG signal. When the membrane potential is larger than the threshold value, they output spikes. Fig. 5 shows the data for the variables of both neurons. The first and tenth synapses are selected to give the captured data for the synaptic transmission PR values, as two examples only (this is to provide clarity in viewing the PR). The synapses have an initial PR value of ~0.5, and change to an average value of ~0.25 after coupling with DSE and e-SP, i.e. the astrocyte-neuron network regulates the steady state value of PR when no faults are present. For the DSE and e-SP, they achieve a stable value after ~100s, i.e. -201 and 190 respectively. Fig. 5(e) gives the output frequencies of neuron #1 and #2. It can be seen that although the data have some fluctuations, but both neurons achieve very similar profiles. The average frequencies are almost same (i.e. ~7Hz), see Fig. 5(f). This calculation process from frequency to the average frequency is completed by the FMMM using (10). The average frequencies of the neuron output spikes are used to control the duty cycle of the PWM controller, i.e. to control the speed the robot car wheels. Fig. 5(g) gives the duty cycles for both neurons. The neuron #1 and #2 have the same duty cycles, i.e. ~40% duty cycle for the PWM controller; therefore the motors of the left and right wheels have the same speed, and the robot car moves straight. This experiment confirms that the proposed hardware SANN, under non faulty conditions, can control the robot car to complete the straight line moving task.

In the next experiment, the fault rate of neuron #2 is set to be 20%, i.e. two of the ten synapses associated with neuron #2

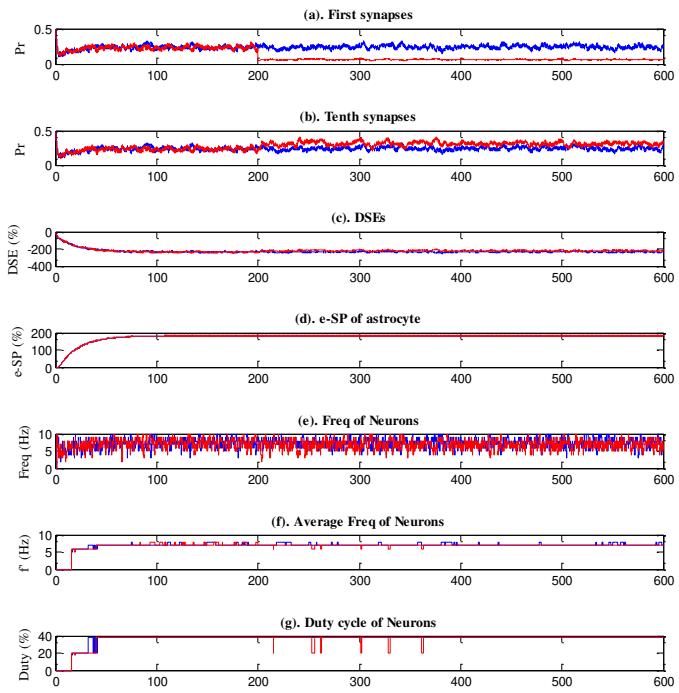

Fig. 7. Neuron #1 (blue) and #2 (red) and correponding outputs in the robot car under 40% fault rate.

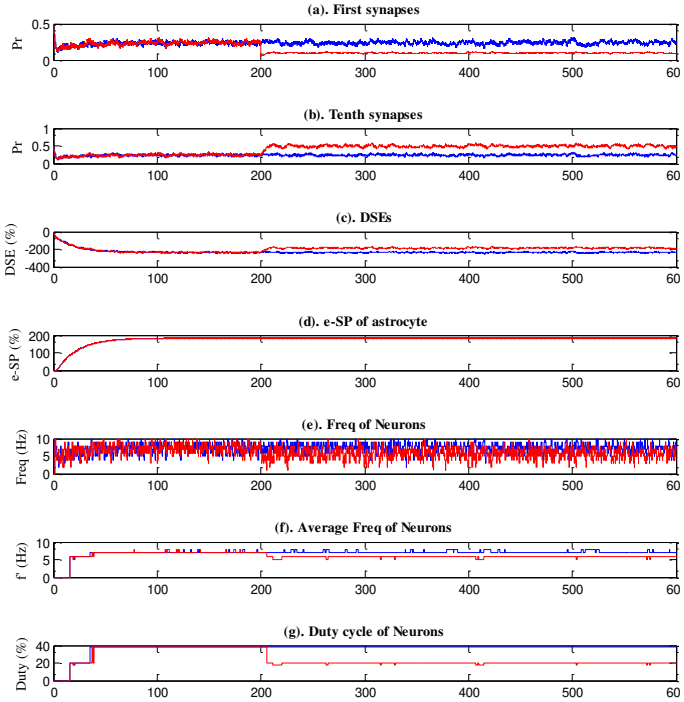

Fig. 8. Neuron #1 (blue) and #2 (red) and correponding outputs in the robot car under 80% fault rate.

are injected with faults therefore their PR value is reduced. Fig. 6 gives the results of variables related to both neurons. The first and second synapses are damaged for the neuron #2 in this experiment. Fig. 6(a) shows the synaptic transmission PR value for the damaged synapse of neuron #2 in coloured red. After 200s, the first synapse of neuron #2 is injected by a fault, thus the PR value is changed to 0.1 (i.e. the same value in the software model approach [20]). It is much lower than the normal initial healthy value (i.e. 0.5), as shown by the Fig. 6(a). After the fault occurs, the healthy synapses are enhanced by the feedback from the astrocyte facility in the SANN (i.e. they have a larger synaptic transmission PR values), as shown by the Fig. 6(b) where the healthy synapse of neuron #2 in red has a slightly larger PR value than the neuron #1 in blue. This is the repair process which aims to maintain the neuron output frequency to the target value. Fig. 6(e) and (f) give the neuron frequencies and average frequencies for both neurons. It can be seen that the neuron #2 (with 20% faulty synapses) has the same output frequencies as neuron #1 (with all healthy synapses). As a result, the duty cycles for both neurons are the same, which are shown by Fig. 6(g). Note that they also achieve the same values as the first experiment (i.e. no fault in Fig. 5(g)). In this experiment, even though the neuron #2 has 20% faulty synapses, the robot car can still complete the straight line moving task, due to the self-repairing capability of the astrocyte facility in the SANN system.

In the third experiment, the fault rate of neuron #2 is increased to 40% where the associated first to fourth synapses are damaged. Fig. 7 shows the experiment results. After injecting the faults at 200s, for neuron #2 the synaptic transmission PR value of the first synapse is 0.1 and the PR value of the tenth synapse (i.e. the healthy synapse) is enhanced by the astrocyte, as discussed previously. Compared to the 20% fault rate, the PR enhancement of the healthy synapses under 40% fault rate is larger, e.g. the PR value increases from average 0.25 to 0.35-0.4 for the 40% fault rate, and only from average 0.25 to 0.3 for the 20% fault rate. For the output, the neuron #2 has almost the same spike frequency as the neuron #1, as shown by Fig. 7(e) and (f). At some specific time point, the neuron #2 has a little lower frequency, e.g. at time steps 220s, 250s, 300s etc. For these time points, the duty cycles of neuron #2 (i.e. 20%) is smaller than neuron #1 (i.e. 40%), which implies that a slight difference in speed is driven to each wheel; although this is minimal given the system is under a 40% fault rate. Fig. 7 shows that the neuron #2 outputs the same speed as neuron #1 for most of time points, i.e. it maintains the target performance. For some time points, it outputs a slightly lower speed for a short time period (e.g. <5s). However, considering the synapses associated with neuron #2 are significantly damaged (e.g. 40% in this experiment), this performance degradation is acceptable; the principle is that the system can adapt and repair to maintain a level of graceful degradation in performance. In particular, this performance is much better than the traditional fault-tolerant approaches, e.g. the performance degradation is ~9% for the fault rate of 20% in the approach of [24].

In the fourth experiment, the fault rate of neuron #2 is increases to 80% (severe level). The results are given by Fig. 8. It can be seen that after faults occur (i.e. at time step 200s), the healthy synapse (e.g. the tenth synapse) of the neuron #2 is enhanced to have a much higher transmission PR value of ~0.5 due to the feedback from the astrocyte facility seeking significant repair of the remaining healthy synapses in the system. The DSE of neuron #2 has a slight decrease which is an expected behaviour as defined in the software model evaluation in [20]. The neuron #2 aims to maintain the same output (e.g.

~7Hz frequency) as close as possible. It has a marginal decrease for the spike frequencies and the average frequencies, as shown by Fig. 8(e) and (f). This causes the duty cycle of the PWM controller associated with neuron #2 to decrease from 40% to 20%. Therefore the motor speed of the right wheel of the robot car is slower than the right wheel. The robot car cannot complete the straight line moving task under the fault rate of 80%. Note that the fault rate of 80% is a severe level of damage. Even under this high fault rate, the SANN system still self-adapts and self-adjusts with the aim to maintaining the target frequency as close as possible, e.g. the actual average frequency is ~5Hz and the target frequency is ~7Hz. These experiments under different fault rates demonstrate the dynamic behaviours of the SANN system and its self-repairing capability under various faulty conditions.

The occupied hardware resource of the FPGA-based robot car is outlined in Table I. It can be seen that in the SANN system, the astrocyte facility occupies the largest area, the synapse & neuron module is less, and DSE generator exhibits the smallest area overhead of the three; i.e. three key astrocyte-neural network implementation blocks. For the FMMM, although a FIFO is used to calculate the average frequency, its area overhead is not large as current high performance FPGA devices provide extensive memory resource. The PWM controller exhibits the smallest area overhead due to its standard functional operation. In total, all combined components only consumes ~20% resource of the Xilinx Virtex-7 XC7VX485T-2FFG1761C FPGA. The computing latency for the SANN system is less than 2,500 clock cycles in this approach running at 200MHz, which is used for one time step calculation of SANN system. This level of latency permits real-time processing of input sensor data and more importantly the capability to repair quickly.

TABLE I. HARDWARE RESOURCE

	Slice LUTs	Slice Registers	Slice	Block RAM Tile	DSPs
Astrocyte	11,394	11,666	3,552	5	42
Synapse & Neuron	9,865	10,383	3,120	0.5	45
DSE generator	4,353	4,688	1,394	0	14
FMMM	65	80	40	4	0
PWM Controller	21	20	6	0	0

V. CONCLUSION

A self-repairing robot car is presented in this paper. It used a spiking astrocyte-neuron network to control the motor speed. A real application task is used to evaluate the fault-tolerant and self-repairing capabilities of the robot car. This is the first time astrocyte-neuron network have demonstrated this capability in an applied application. Analysis of the robot car demonstrated that it can maintain system performance under a 20% fault rate, and has marginal performance degradation at 40% fault rates and greater. It achieves a more fine-grained repair capability compared to the traditional fault-tolerant approaches, e.g. the approaches of [24], [25] have a performance degradation of >9% under the fault rate of 20% while our work demonstrated 0% degradation. Future work will explore the use of sensory data (e.g. infrared, sonic) as input to the FPGA system and also optimisation of the hardware implementation for large-scale astrocyte-neuron networks.

ACKNOWLEDGEMENTS

The work is part of the SPANNER project and is funded by EPSRC (EP/N00714X/1). The authors also acknowledge the support of the Intelligent Systems Research Centre at University of Ulster.

REFERENCES

[1] K. Zhang, G. Bedette, and R. F. Demara, "Triple Modular Redundancy with Standby (TMRSB) Supporting Dynamic Resource Reconfiguration," in IEEE Autotestcon, 2006, pp. 690–696.

[2] I. Pomeranz and S. M. Reddy, "Robust Fault Models Where Undetectable Faults Imply Logic Redundancy," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 18, no. 8, pp. 1230–1234, 2010.

[3] T. Lehtonen, P. Liljeberg, and J. Plosila, "Analysis of Forward Error Correction Methods for Nanoscale Networks-on-Chip," in International Conference on Nano-Networks, 2007, pp. 1–5.

[4] Z. Zhang and Y. Wang, "Method to Self-repairing Reconfiguration Strategy Selection of Embryonic Cellular Array on Reliability Analysis," in NASA/ESA Conference on Adaptive Hardware and Systems (AHS), 2014, pp. 225–232.

[5] J. A. Walker, M. a. Trefzer, S. J. Bale, and A. M. Tyrrell, "PAnDA: A Reconfigurable Architecture that Adapts to Physical Substrate Variations," IEEE Transactions on Computers, vol. 62, no. 8, pp. 1584–1596, 2013.

[6] S. Carrillo, J. Harkin, L. McDaid, S. Pande, S. Cawley, B. McGinley, and F. Morgan, "Advancing Interconnect Density for Spiking Neural Network Hardware Implementations using Traffic-aware Adaptive Network-on-Chip Routers," Neural Networks, vol. 33, no. 9, pp. 42–57, 2012.

[7] R. Serrano-Gotarredona, M. Oster, P. Lichtsteiner, A. Linares-Barranco, R. Paz-Vicente, F. Gomez-Rodriguez, L. Camunas-Mesa, R. Berner, M. Rivas-Perez, T. Delbruck, S.-C. Liu, R. Douglas, P. Hafliger, G. Jimenez-Moreno, A. Civit Ballcels, T. Serrano-Gotarredona, A. J. Acosta-Jimenez, and B. Linares-Barranco, "CAVIAR: A 45k Neuron, 5M Synapse, 12G Connects/s AER Hardware Sensory-Processing-Learning-Actuating System for High-Speed Visual Object Recognition and Tracking," IEEE Transactions on Neural networks, vol. 20, no. 9, pp. 1417–1438, Sep. 2009.

[8] C. Zamarreño-ramos, A. Linares-barranco, and T. Serrano-gotarredona, "Multicasting Mesh AER: A Scalable Assembly Approach for Reconfigurable Neuromorphic Structured AER Systems . Application to ConvNets," IEEE Transactions on Biomedical Circuits and Systems, vol. 7, no. 1, pp. 82–102, 2013.

[9] H. Naganuma, K. Kiyoyama, and T. Tanaka, "A 37x37 Pixels Artificial Retina Chip with Edge Enhancement Function for 3-D Stacked Fully Implantable Retinal Prosthesis," in IEEE Biomedical Circuits and Systems Conference, 2012, pp. 212–215.

[10] A. Jimenez-Fernandez, G. Jimenez-Moreno, A. Linares-Barranco, M. J. Dominguez-Morales, R. Paz-Vicente, and A. Civit-Balcells, "A Neuro-Inspired Spike-Based PID Motor Controller for Multi-Motor Robots with Low Cost FPGAs," Sensors, vol. 12, no. 4, pp. 3831–3856, Jan. 2012.

[11] A. Linares-Barranco, F. Gomez-Rodriguez, A. Jimenez-Fernandez, T. Delbruck, and P. Lichtensteiner, "Using FPGA for Visuo-motor Control with A Silicon Retina and A Humanoid Robot," in IEEE International Symposium on Circuits and Systems, 2007, pp. 1192–1195.

[12] S. Carrillo, J. Harkin, L. J. McDaid, F. Morgan, S. Pande, S. Cawley, and B. McGinley, "Scalable Hierarchical Network-on-Chip Architecture for Spiking Neural Network Hardware Implementations," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 12, pp. 2451–2461, 2013.

[13] J. Liu, J. Harkin, Y. Li, and L. Maguire, "Online Traffic-Aware Fault Detection for Networks-on-Chip," Journal of Parallel and Distributed Computing, vol. 74, no. 1, pp. 1984–1993, 2014.

[14] K. Chakrabarty, "Towards Fault-Tolerant Digital Microfluidic Lab-on-Chip: Defects, Fault Modeling, Testing, and Reconfiguration," in IEEE Biomedical Circuits and Systems Conference, 2008, pp. 329–332.

[15] M. McElholm, J. Harkin, L. McDaid, and S. Carrillo, "Bio-Inspired Online Fault Detection in NoC Interconnect," in Energy-Efficient Fault-Tolerant Systems, 2014, pp. 241–267.

[16] T. Horita, T. Murata, and I. Takanami, "A Multiple-Weight-and-Neuron-Fault Tolerant Digital Multilayer Neural Network," in 21st IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, 2006, pp. 554–562.

[17] M. R. Boesen, D. Keymeulen, J. Madsen, T. Lu, and T. H. Chao, "Integration of the Reconfigurable Self-Healing eDNA Architecture in an Embedded System," in IEEE Aerospace Conference Proceedings, 2011, pp. 1–11.

[18] J. Liu, J. Harkin, M. McElholm, L. McDaid, A. Jimenez-Fernandez, and A. Linares-Barranco, "Case Study: Bio-inspired Self-adaptive Strategy for Spike-based PID Controller," in IEEE International Symposium on Circuits and Systems (ISCAS), 2015, pp. 2700–2703.

[19] M. De Pittà, N. Brunel, and A. Volterra, "Astrocytes: Orchestrating Synaptic Plasticity?," Neuroscience, pp. 1–19, 2015.

[20] J. Wade, L. McDaid, J. Harkin, V. Crunelli, and S. Kelso, "Self-Repair in a Bidirectionally Coupled Astrocyte-Neuron (AN) System based on Retrograde Signaling," Frontiers in Computational Neuroscience, vol. 6, no. 76, pp. 1–12, Jan. 2012.

[21] M. Naeem, L. J. McDaid, J. Harkin, J. J. Wade, and J. Marsland, "On the Role of Astroglial Syncytia in Self-Repairing Spiking Neural Networks," IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 10, pp. 2370–2380, 2015.

[22] W. Gerstner and W. M. Kistler, Spiking Neuron Models: Single Neurons, Populations, Plasticity. Cambridge University Press, 2002.

[23] J. Liu, J. Harkin, Y. Li, L. Maguire, and A. Linares-Barranco, "Low Overhead Monitor Mechanism for Fault-tolerant Analysis of NoC," in IEEE 8th International Symposium on Embedded Multicore/Many-core Systems-on-Chip, 2014, pp. 189–196.

[24] J. Liu, J. Harkin, Y. Li, and L. P. Maguire, "Fault Tolerant Networks-on-Chip Routing with Coarse and Fine-Grained Look-ahead," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 2, pp. 260–273, 2016.

[25] J. Liu, J. Harkin, Y. Li, and L. Maguire, "Low Cost Fault-tolerant Routing Algorithm for Networks-on-Chip," Microprocessors and Microsystems, vol. 39, no. 6, pp. 358–372, 2015.