A Deep Reinforcement Learning based Homeostatic System for Unmanned Position Control

Priyanthi M. Dassanayake
p.m.dassanayake@derby.ac.uk
University of Derby
UK

Ashiq Anjum
a.anjum@derby.ac.uk
University of Derby
UK

Warren Manning
w.manning@derby.ac.uk
University of Derby
UK

Craig Bower
c.bower@derby.ac.uk
University of Derby
UK

ABSTRACT

Deep Reinforcement Learning (DRL) has been proven to be capable of designing an optimal control theory by minimising the error in dynamic systems. However, in many of the real-world operations, the exact behaviour of the environment is unknown. In such environments, random changes cause the system to reach different states for the same action. Hence, application of DRL for unpredictable environments is difficult as the states of the world cannot be known for non-stationary transition and reward functions.

In this paper, a mechanism to encapsulate the randomness of the environment is suggested using a novel bio-inspired homeostatic approach based on a hybrid of Receptor Density Algorithm (an artificial immune system based anomaly detection application) and a Plastic Spiking Neuronal model. DRL is then introduced to run in conjunction with the above hybrid model. The system is tested on a vehicle to autonomously re-position in an unpredictable environment. Our results show that the DRL based process control raised the accuracy of the hybrid model by 32%.

CCS CONCEPTS

• Computing methodologies → Online learning settings; **Motion path planning**; **• Computer systems organization → Real-time system architecture**; **• Software and its engineering → Real-time systems software**.

KEYWORDS

Deep Reinforcement Learning; Artificial Immune System; Receptor Density Algorithm; Plastic Spiking Neuron; Deep Neural Network; Bio-inspired; Homoeostasis-inspired

ACM Reference Format:
Priyanthi M. Dassanayake, Ashiq Anjum, Warren Manning, and Craig Bower. 2019. A Deep Reinforcement Learning based Homeostatic System for Unmanned Position Control. In *Proceedings of the IEEE/ACM 6th International Conference on Big Data Computing, Applications and Technologies (BDCAT '19), December 2–5, 2019, Auckland, New Zealand.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3365109.3368780

1 INTRODUCTION

During the last decade, the unmanned vehicles industry has experienced exponential growth. It is starting to play a major role in different types of missions including search and rescue, environmental monitoring, navigation, security surveillance, transportation and inspection[20]. Many of these operations involve functioning in unknown territories.

The computer vision-based navigational systems, the state of art (SOA) for unmanned navigation, demand massive computing power due to I/O operations and 3D map building. They require constant access to powerful hardware either onboard or remotely to cater to the processing demands. Consequently, they are prone to latency issues due to substantial processing requirements, which make them unsuitable for real-time operations. These limitations heighten the urgency for a light-weight autonomous system for unmanned navigation.

Deep Reinforcement Learning (DRL), on the other hand, can evolve and achieve autonomy and robustness with minimal human intervention. DRL has proven to be capable of designing an optimal control theory by minimising the error in dynamic systems [27]. In some applications such as Atari games and Go competitions, DRL has even been able to achieve superhuman performances autonomously [26]. DRL can improve dynamic performance and computational efficiency.

Since DRL is model-free, the learning is performed via the measurements of rewards obtained through the environmental response. For the learning to be consistent, the rewards generated has to be reliable for the corresponding environmental conditions. This requires that the environment remains stationary and the underlying probability distribution of the environment is held constant which makes the cause and effect directly observable. For example, in board games, the rules can predict the next state of an action performed by the agent. In robotic control, there are rules of physics that govern the motion of the limbs of the robot.

This establishes the notion that the reward for a certain action remains unique due to the predictability of the next state. However, for unpredictable environments where the processes become non-stationary, the validity of the reward function for a certain action

BDCAT '19, December 2–5, 2019, Auckland, New Zealand
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7016-5/19/12…$15.00
https://doi.org/10.1145/3365109.3368780

does not remain consistent. Therefore, in unstable environments, the application of reinforcement learning may not seem feasible.

DRL has been applied even in chaotic and turbulent environments, providing states can be predicted using laws of physical relationships [5] or approximated [4]. To use DRL in unstable environments, several attempts to predict the states can be found in recent literature. For example, a homeostatic, instinctive concept presented in [7] is an viable alternative that maintains critical parameters of the agent for the reliable state prediction. However, these methods require the environment to be at least partially observable.

An unobservable environment can be illustrated in a vessel in the deep sea. The state changes of the vessel can be entirely random and unpredictable due to the environmental forces. These forces can be originated by fluid dynamics, wind, ocean currents, wave dynamics and volcanic eruptions in the sea bed etc. In such environments, DRL cannot be used because neither the actions nor the states can be defined due to numerous factors affecting the motion.

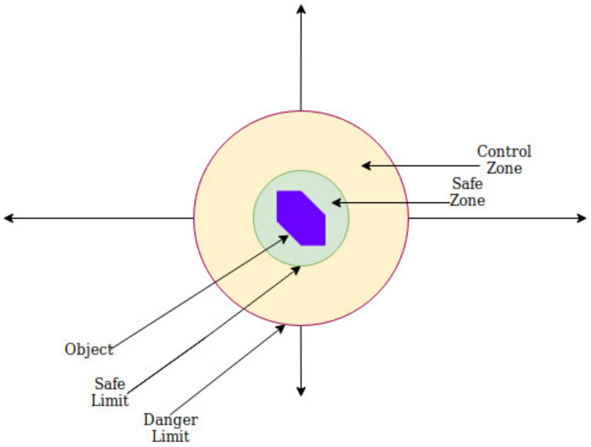


Figure 1: Control Limits

In a parallel branch of research, bio-inspired, model-based systems have also endeavoured to achieve autonomy by taking human homoeostasis as an example. Several bio-inspired algorithms have been developed to emulate autonomy under unsupervised conditions. They have been successfully applied in parts of control process functions such as anomaly detection and fault-tolerant applications [23],[14]. However, they are yet to adapt the whole control process due to abstractions preventing them to evolve the same way as their biological counterparts, impeding the ability to auto-correct.

In this paper, a novel bio-inspired homeostatic algorithm is extended to providing a platform to implement DRL by managing the unpredictability of environmental conditions. A specific application of an unmanned vehicle that maintains its position in an unobservable environment is used as the motivational use case. The unmanned vehicle generates an opposing force against the resultant environment force acting on the vehicle. The objective is to maintain the position within pre-defined boundaries by applying the opposing force mitigating the environmental effect. Figure 1 shows control limits in a two-dimensional space. The vehicle requires to maintain the position at most at the safe zone or at least within the control zone autonomously; to summarise, this paper includes,

- A True Homeostatic autonomous system which evolves, corrects and stabilises extreme environmental conditions with the aid of DRL and
- A Deep Reinforcement Learning (DRL) system operates in an unpredictable environment with the aid of a bio-inspired homeostatic system

The paper is organised as follows. Section 2 presents the existing approaches justifying their inability to achieving autonomy under unpredictable conditions. Section 3 describes the approach taken by this paper. Section 4 presents the application of the above approach in the use case. Section 5 present experiments and simulation results along with performance, design considerations and scalability.

2 RELATED WORK

The state of the art (SOA) for operating in unknown territories is computer-vision based navigation. The performance of computer vision-based navigation depends on the hardware and Input/Output (I/O) operations such as wide field-of-view cameras for feature tracking and camera-IMU (Inertial Measuring Units) for extrinsic calibration. These systems require massive software resources for reconstructing a 3D dense environment of the actual environmental conditions, trajectory planning and feedback control. For these reasons, the highly complex mechanisms of computer vision systems are not feasible for real-time operations as the required hardware proficiency is not sufficient enough to cater to the processing demands [3].

The SOA as well as other alternative solutions developed for unpredictable environments such as Simultaneous Localisation And Mapping (SLAM)[6], Fuzzy logic based tracking [32] etc. involve map building relative to the surrounding objects such as landmarks to identify the position. This requires sensors reflecting signals from the landmarks. However, for the areas with no distinguishable variations in the landscapes such as oceans and deserts, it may be impossible for the sensors such as sonar to capture a perception about the environment [19].

Several real-world applications, operating in unpredictable environments employ a model-based approach. They can function effectively via data assimilation techniques such as Kalman Filters [31]. However, these methods do not facilitate autonomy, due to the dependence of stationary models rather than evolving models. In contemporary research, the prominent categories that could potentially expedite evolving models are applications of DRL and homeostatic algorithms. As identified in the previous section, both categories are unable to support non-stationary real-time processes with unknown states. In short, DRL suffers the inability to function in unpredictable environments and homeostatic algorithms which are still in their infancy, are unable to evolve as same as their biological counterparts.

There have been several attempts to adopt reinforcement learning for non-stationary processes. One of those is learning the approximate distribution of possible values [4]. Alternatively, a homeostatic, instinctive concept has been used to maintain critical parameters for the state prediction [7]. However, these require the identification of available states of a Markov process which may not be available for an unpredictable environment.

Another approach is modelling the unstable environment as an adversarial agent, whose goal is to destabilise the system by damaging the system agent's action. To understand the environment's behaviour, the adversarial agent's disturbances are modelled in training and test scenarios. This knowledge is later applied to accomplish the system goal of stability by solving for equilibrium for the system agent. This equilibrium establishes stationary policies which enable the system to outperform the environment. This field of study is known as Robust Adversarial reinforcement learning (RARL) [25]. However, to withstand the adversary, the system agent needs to be trained under similar adversarial conditions, which may not be observable in an unpredictable environment.

Despite being inspired by biology, the best autonomous system there is, homoeostasis inspired algorithms cannot achieve autonomy as they do not represent the homeostatic behaviour as same as their biological counterpart. Hence, their functions remain mediocre with limited applications such as masking faulty conditions. Developed by abstracting the brain cells' ability to compensate for an injury, the Plastic Spiking Neuron (PSN), is so far, the only algorithm which has come close to mimic homoeostasis at least partially [14]. A PSN functions by detecting the reduction of the spiking activity due to faults and altering the voltage to establish constant spiking activity. By utilising this behaviour, a PSN network has been able to establish a fault-free behaviour in the presence of faults of various degrees. Since there is no fault correction mechanism in place emulating the healing process of the biological homeostatic counterparts, the system is unable to re-establish normal conditions. Therefore, under persistent faulty conditions, the plasticity of the neuron is destined to cease as it does not emulate true homoeostasis (see Figure 2b).

Artificial Immune Systems are one of the interesting fields to investigate when working with unpredictable environments. The Receptor Density Algorithm (RDA), one of the versatile artificial immune system inspired algorithms, has been applied in a vast array of real-world applications such chemical detection [12], monitoring industrial processes[8],safety and security applications [16], error detection in swarm robotics [17], wireless sensor networks [18] , online fraud detection [13] etc. The secret behind the popularity of RDA is its ability to operate unsupervised in real-time conditions. A notable feature of RDA is the ability to maintain an equilibrium state. This aspect is an attractive feature for stabilisation of the vehicle in our use case. However, this condition is only true if the receptor is influenced by a constant input, as steady negative feedback generated against the direction of the anomalous condition, only applies if the input remains constant (see Figure 3). An adequate extension of RDA for an unpredictable input may facilitate our objective of autonomous stabilisation.

3 APPROACH

The objective of the motivational use case depicted in Figure 1 is to stabilise the unmanned vehicle within the safe zone under unpredictable environmental conditions. RDA's ability to model the unpredictable conditions enables the anticipation of unobservable states. This supports the utilisation of DRL as the unobservable states have now become observable. RDA is designed for anomaly detection and only reaches equilibrium under constant forces. By setting the target of DRL as the equilibrium state, RDA can be extended to reach equilibrium under all conditions. Hence, this paper contributes to RDA to expand the spectrum of its applications from anomaly detection to preventive maintenance. Also, the utilisation of DRL for unobservable states is a novelty which open doors for many real-world applications to benefit from DRL.

RDA is a T-Cell receptor communication inspired algorithm where the receptor is denoted as a tuple (p, n, β, l) with position $p \geq 0$, negative feedback n, safe limit $\beta > 0$, and the danger limit $l > \beta$. u_t refers to the distance caused by the current unpredictable force (see Figure 3). ($u_t \epsilon \mathbb{R}$ and $u_t \geq 0$.) at time t = 0, 1, 2,. . . .

The p_t, n_t can be mapped to p_{t+1}, n_{t+1},

$$p_{t+1} = b * p_t + u_t - a * n_t \tag{1}$$

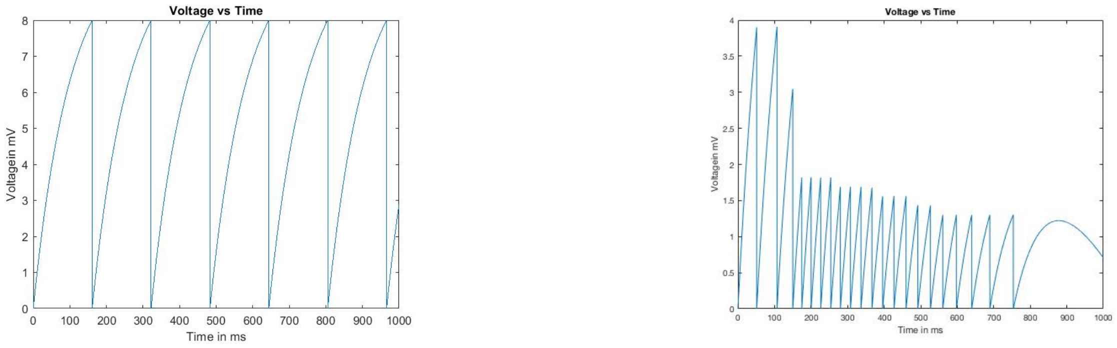
$$n_{t+1} = \begin{cases} d * n_t & if\, p_{t+1} < \beta \\ d * n_t + g & if\, p_{t+1} > \beta \end{cases} \tag{2}$$

The parameters $0 < b < d < 1$ refer to decay rate, $a > 0$ refers to negative feedback influence and $g > 0$ refers to the growth rate. These parameters are derived from the actual biological function T-cell receptor and are constant values. Having constant parameters debilitates RDAs capacity to adapt for different conditions. With the constant parameters, the equilibrium can only be achieved under constant input, which requires the environment to be constant.

Since relying on constant parameters will not generate an evolving model, a DRL is applied to directly generate n_t. With DRL n_t is generated as a response to the environmental conditions which allows the RDA to be tuned to maintain equilibrium under varying forces. Since the position has to be maintained within the safe limit ($p_{t+1} \leq \beta$), the target position for DRL can be assigned to β. A Deep Neural Network (DNN) can be inferred real-time with this information to generate n_t according to the conditions and maintain equilibrium.

Transient operation of a DNN can be expensive resource-wise. By using RDA, this can be managed, as running DNN is only required if the position exceeds the safe limits ($p_{t+1} > \beta$) (see Equation 2). The negative feedback n_t is decayed automatically once the vehicle is within the safe limits ($p_{t+1} \leq \beta$) causing the system to release the resources.

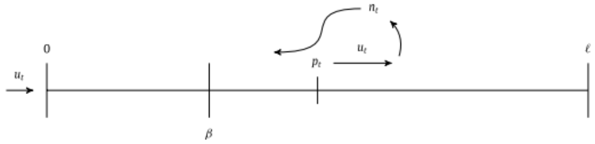
The behaviour of the above described RDA -DRL based homeostatic system cannot be monitored just by the position itself. For example, in a 3D space, the information about the position would not provide an interpretation of the dynamics of the autonomous system. The system requires real-time visualisation to observe the autonomous behaviour of the system. A Plastic Spiking Neuron (PSN) is proposed to accomplish the above purpose. Figure 4 depicts the relationship between RDA, PSN and DRL. As a PSN receives an input signal as an electrical pulse, it accumulates the neuron potential until it reaches the threshold voltage (V) and as soon as



(a) No motion

(b) The plasticity is lost at 750ms

Figure 2: Plastic Spiking Neuron.



Under a constant input u_t, the receptor reaches an equilibrium state as the negative feedback n_t becomes constant

Figure 3: The Receptor [23]

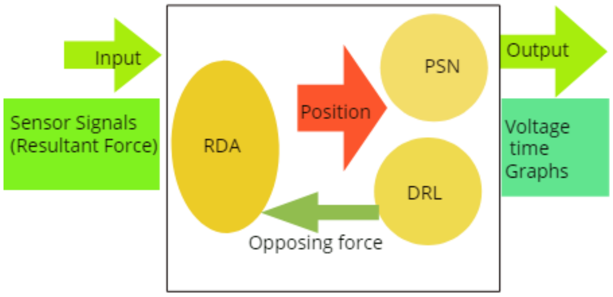


Figure 4: Approach

the PSN surpasses V the neuron fires and resets to resting potential (in our experiment to zero) and starts to accumulate voltage depending on the input current.

By assuming the resting potential is equal to zero the voltage ($v(t)$) of a spiking neuron (modelled as a leaky capacitor with membrane resistance R_m under a constant input current I_t) is calculated using the following equation [9].

$$v(t) = -\tau_m \frac{dv}{dt} + I_t R_m \tag{3}$$

In above Equation, t is current time and the time constant is denoted by $\tau_m = R_m C_m$ (where C_m is the membrane capacitance).

By solving the Equation 3 assuming the input current is constant, the voltage update during the time step Δt can be written as:

$$v_{t+\Delta t} = I_t * R_m + (v_t - I_t * R_m)e^{-\frac{\Delta t}{\tau m}}$$

By marking the starting position of RDA equivalent to the threshold voltage (V), the signal supplied to the PSN can be set to the maximum allowed step current. As long as the object has not moved from the initial position, the PSN may fire consistently at V (See Figure 2a). As the object moves away from the initial position, the input signal is set to decline as the object moves towards the danger limit.

As a result, the PSN halts firing and settles at a lower voltage. In this occasion, if V of PSN is tuned to a lower threshold, the PSN can be set to elicit fire again. This enables the PSN to display homeostatic behaviour providing visual evidence for the closeness to the initial position. The threshold voltage multipliers for different ranges of the input current can be derived from the experimental data in [14]. Firing thresholds adjusted according to these multipliers and tested by decreasing the input current incrementally towards zero (See Figure 2b). In this experiment, under continued reduced input current, the neuron lost its plasticity when the input current approached zero. Since there is no fault correction mechanism in place (emulating the healing process of the biological homeostatic counterparts), the system is unable to re-establish normal conditions.

With the aid of RDA and DRL hybrid, the PSN can be tuned to display healing properties. The auto-correction feature of RDA aided by DRL enables the PSN to maintain its plasticity by gradually increasing the firing threshold displaying true homoeostasis.

4 THE AUTONOMOUS SYSTEM

Autonomous self-correction is set to perform by the RDA and DRL conjunction. As depicted in Figure 5, as the vehicle acts against the environment, it is moved towards the safe limit through learning mechanism of DRL. Reactions of the DRL are captured as an increasing input signal and the voltage of PSN is tuned accordingly.

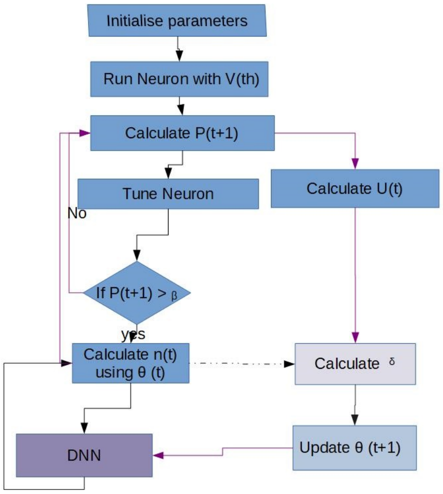


Figure 5: The Overview

4.1 Learning Mechanism of the DRL

Reinforcement learning seeks to identify the best policy $\pi(s)$ for a particular state s which maximises the return at that state under that policy. The return function is also known as the action-value function $Q(s, a)$ or simply the Q function. Selecting the best policy for a certain state is interchangeably defined as the application of action maximises the return for that state $Q^*(s, a)$.

$$\pi(s) = arg_a max[Q^*(s, a)]$$

Since the return $Q^*(s, a)$ is typically unknown, a function is approximated as a multiplication of hypothetical weights θ and available features at that particular state after the application of action a denoted by the function $\phi_t(s, a)$. Therefore, a general description for a certain state $Q(s, a, \theta)$ can be defined as [30],

$$Q(s, a, \theta) = \theta_t^\intercal \phi_t(s, a) \tag{4}$$

Since the above approximation should reflect the success of the operation, a function can be defined as the update target U_t. The update target has to be the expected outcome of performing a certain action. For example, in a game, winning the game can be defined as the update target. For this particular case, winning the game is equivalent to maintaining the position within the safe zone for all environmental conditions.

Equation 1 of RDA for our specific application, specifies that the only reason for p_t of the vehicle to be altered is the resultant environmental force. By assuming that negative feedback does not influence by a specific condition, negative feedback influence a can be ignored. Therefore, both the opposing decay rates b and negative feedback influence a can be omitted from the equation (i.e. $a = b = 1$). By anticipating the negative feedback to generate the minimum force which is large enough to bring the vehicle into the safe limit ($p_{t+1} = \beta$), the update target can be defined as,

$$U_t = n_t = |p_t| + u_t - \beta$$

Due to operational constraints, above U_t may not be feasible to be applied by the vehicle. Therefore, in such situations, the expected negative feedback can be defined as,

$$U_t = n_t = n_{max} - u_t$$

where n_{max} is the maximum force that can be applied by the vehicle.

Since U_t is the expected negative feedback, the feature vector $\phi(s, a)$ can be equated to the current negative feedback n_t, which the new negative feedback has to build upon. Therefore, the action value function from Equation 4 can be rewritten as,

$$Q(s, a, \theta) = \theta_t^\intercal n_t$$

The action value function $Q(s, a, \theta)$ should approach desired outcome expected U_t. By comparing U_t with $Q(s, a, \theta)$, the hypothetical weights of θ can be adjusted via function approximation of DNN. Therefore, in a DNN with n hidden layers and m nodes at each layer, θ can be defined as,

$$\theta_t = \begin{bmatrix} W_{11} & \dots & W_{1m} \\ W_{21} & \dots & W_{2m} \\ \dots & \dots & \dots \\ W_{n1} & \dots & W_{nm} \end{bmatrix}$$

where W_{ij} is the weight assigned on the synapse which transitions from i^{th} node to j^{th} node. Function approximation can be defined as the minimisation the cost function between the target U_t and the output $Q(s, a\theta)$.

In DNN the weights are generally adjusted to approximate the expected value by using stochastic gradient descent (SGD). In SGD, the random values initially assigned for θ are optimised to reinforce the optimal value of n_t by measuring the rate at which the δ_t changes in respect to the rate at which each W_{ij} of synapse changes.

Figure 6, shows forward propagation of a DNN structure which consists of single input layer of feature vector equivalent to the current negative feedback n_t, n fully connected hidden layers with m nodes at each layer and a single output layer equivalent to the generated next negative feedback n_{t+1} which will be compared against the U_t to optimise the synaptic weights. Arrows represent the synaptic weights W_{ij} and Σ represents the summation of the outcome. f represents the application of the activation function (a differentiable function, the sigmoid function in this case). V_{ij} represents the value generated after the application of the sigmoid function at each node. This value is then multiplied with the next set of synaptic weights $W_{1j}, \dots W_{mj}$ and so forth.

The differentiable activation functions not only standardise the values but also enable the application of backpropagation through chain rule [35] to optimise the parameters θ_t. The DNN is inferred with SGD using backpropagation as a gradient computing technique. Algorithm 1 summarises the forward propagation and the backpropagation of the DRL system.

4.2 Experimental Environment

The motion caused by the resultant force is captured as a position change of the receptor of RDA. The safe zone β is seized as a sphere surrounding the initial position of $p_0 = 0$. Once the vehicle leaves the safe zone, the decision-making process of DRL begins. In DRL, the DNN is inferred against the update target U_t. The weights are adjusted at each synapse to generate the optimal n_{t+1} to navigate

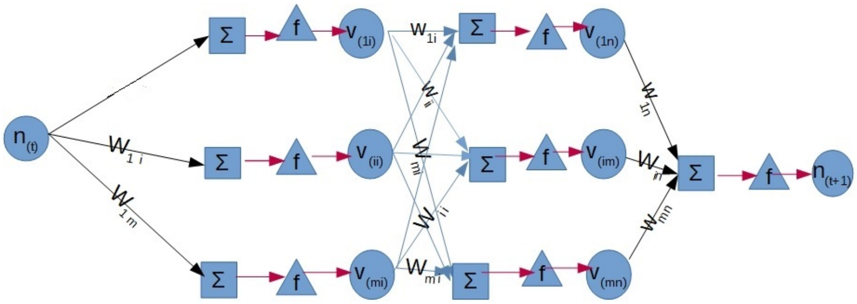


Figure 6: Layers of the DNN Architecture for optimising the parameters

Algorithm 1 Inferring DRL

```
$U_{t+1} \leftarrow p_t + u_t - \beta$
if $U_{t+1} \geq n_{max}$ then
    $U_{t+1} \leftarrow n_{max}$
end if
# forward propagation
$a^{l1} \leftarrow \sigma(\theta_t^{(l1)\intercal} n_t)$ # input layer ($l1 \equiv layer1$)
$a^{lk} \leftarrow \sigma(\sum_i \theta_{t(ji)}^{(lk)\intercal} a_i^{(l(k-1))})$
$n_{t+1} \leftarrow a^{ln} \leftarrow \sigma(\sum_i \theta_{t(ji)}^{(ln)\intercal} a_i^{(l(n-1))})$# output layer ($ln$)
#back propagation
$\delta_{t(ln)} \leftarrow (U_{t+1} - n_{t+1})\sigma'(a^{ln})$ #minimising the cost function
$\delta_{t(lk)} \leftarrow \delta_{t(lk+1)}\theta_t^{(lk+1)\intercal}\sigma'(a^{lk})$
$\delta_{t(l1)} \leftarrow \delta_{t(l2)}\theta_t^{(l2)\intercal}\sigma'(a^{l1})$
$\theta_{t+1}^{(lk)} \leftarrow \theta_t^{lk} - \alpha * \delta_{t(lk)} * \theta_t^{(lk)}$ # update parameters
```

the vehicle towards the safe limit. This brings new data n_{t+1} for RDA corresponding to the current conditions.

Algorithm 2 PSN for Visualisation

```
# update the input current:
$I_t \leftarrow I_{max} - \frac{p_{t+1}}{l}$
# voltage:
$v_{t+1} \leftarrow I_t R_m + (v_t - I_t R_m) * exp\frac{-[(t+1)-t]}{\tau_m}$
# look up multiplier according to the error
$V \leftarrow (multiplier_{\leftarrow \frac{p_{t+1}}{l}}) * V^{th}$ # derive new threshold voltage
if $v_{t+1} \geq V$ then
    $v_{t+1} \leftarrow 0$ # reset
end if
% Display PSN graph %
Plot($v_{t+1}, t+1$)
```

The conditions are visualised using a PSN. The position of the vehicle p_{t+1} is reflected on PSN as a weakening input signal (from 1mA to 0mA) by subtracting the displacement ($\frac{p_{t+1}}{l}$)). To maintain the constant behaviour of the PSN, the threshold voltage is multiplied by the initial firing threshold (V = 8mV) with a constant reflecting the percentage displacement $\frac{p_t}{l}$%. Algorithm 2 summarises the voltage adjustment process of PSN.

As soon as the vehicle is brought within the safe zone, the inferencing DNN halts and the application of Equation 2 is resumed with residual negative feedback to navigate the vehicle towards the initial position. The system prioritises on maintaining the position within the safe zone by keeping the vehicle within an assigned range of $0 \leq p_{t+1} \leq \beta$.

This feature is utilised for adverse environmental conditions, where the vehicle has the option of altering the β to l to maintain the vehicle at most at the danger limit. Algorithm 3 summarises the DRL-enabled homeostatic system. The program was implemented in the MATLAB environment. A DNN was implemented with $n = 1$ hidden layers of $m = 7$ hidden nodes to simulate DRL. The RDA consisted of a danger limit l of 10 meters and a safe limit β of 2 meters. The maximum distance the vehicle can be moved against the motion was defined as n_{max}=0.9m.

Algorithm 3 The Homeostatic DRL system

```
repeat
    $p_{t+1} \leftarrow p_t + u_t - n_t$
    Run PSN (Algorithm 2)
    if $p_t \leq \beta$ then
        $n_{t+1} \leftarrow d * n_t$
    else
        if $p_t \geq l$ then
            $\beta \leftarrow l$ # maintenance mode
            $l \leftarrow l * K$
        end if
        Run DRL (Algorithm 1)
    end if
until $t = t_\infty$
```

Note that the experimented system is implemented only as a proof of concept and does not have the required scalability nor the complexity for real-world deployment. In actual deployment, implementing the DNN using Python-based TensorFlow or PyTorch environment is recommended. For deployments, depending on the specific environments the historical data may be used to structure the DNN. The complexity of the program may vary depending on the structure of DNN. By using parallel programming the performance and the latency of the program can be significantly improved.

5 RESULTS AND DISCUSSION

The performance of the system is monitored via the voltage vs time graphs generated by the PSN. The safe limit of Figure 1 is represented by a red line across the graphs which is generated using the threshold voltage equivalent to β. An overall spiking threshold above this line indicates that the algorithm functions according to the system requirements. The algorithm is not up to the standards if the majority of spikes are below this line.

Time-variant motion is initially used to assess the impact of DRL. This is evaluated by using the hybrid of RDA and PSN as the benchmark. The hybrid system gracefully handled sinusoidal input even with added noise (See Figure 7a) and also excelled in handling milder harmonic motion, as well as a motion under low impact constant forces(See Figure 7d). This concludes that the hybrid of RDA and PSN (the benchmark) can function without the aid of DRL in milder environments.

When the sinusoidal force with added noise gets 3 times large (See Figure 7b) or the impact of the constant force gets 2 times large (See Figure 7e), the system fails to maintain its position within the safe zone, almost in all occasions. Hence, when the environment gets rough, the hybrid system fails to function appropriately.

As shown in Figure 7c and Figure 7f, the overall spiking threshold can be seen above the safe limit mark. Hence, for rough environments, a significant improvement in the performance can be observed when DRL was introduced. In unpredictable environments, the magnitude of the forces cannot be predicted. Therefore, using DRL in conjunction with the hybrid system is essential for such environments.

The unpredictable environments were represented using, samples from normal distributions since many of the real-world phenomena follow the normal distribution. The data sets were randomly generated and normalised using z-score, where 68% of data lied between [-1,1] and 95% of data within [-2,2].

The spikes in Figure 8a show the behaviour of the benchmark system in an unpredictable environment. Except for the initial few seconds, the benchmark failed to maintain the vehicle within the safe zone. However, as shown in Figure 8b, the system performed more consistently with higher reliability with DRL. Despite the simplistic structure of the DNN, the hybrid system with DRL performs impressively in random environments. This indicates that the DRL can predict actions which are large enough to cancel the unpredictable environmental forces.

To quantify the performance of the system, 31 datasets were produced for various time intervals. In those, the instances that the vehicle remained within the safe were counted for both DRL enabled hybrid and the hybrid benchmark. As shown in Table 1, the central limit theorem is used to statistical estimation of the population mean. Our calculations indicate that with 95% confidence, the population mean of DRL-enabled hybrid system lies between 64.67% and 68.58% and the population mean of the benchmark lies between 29.26% and 40.99. A performance indicator was defined by obtaining the average of the upper bound (68.58 − 29.26%) and the lower bound (64.67 − 40.99%). According to the performance indicator, the system has reached approximately 32% accuracy with DRL. For the actual deployments, optimising the DNN structure is recommended for significant performance increment.

Table 1: Statistical Estimation of the performance

	With 95% confidence interval		
	Sample Mean	***Std dev.***	***Poputation Mean***
DRL	66.63%	5.55%	64.67% - 68.58%
Benchmark	35.13%	16.66%	29.26% - 40.99%
Maximum Performance		39.322% -Upperbound	
Minimum Performance		23.68%-Lower bound	
Avg. Performance		**31.50%**	

5.1 Evolution of the DRL

The performance of the system can be measured by calculating the mean instances the vehicle remained within the safe zone. The latency of the system can be estimated by calculating the average time taken to generate the opposing force. As shown in See Figure 9, plotting the performance and the latency for various time scales for both the DRL-enabled system and the benchmark enables the visualisation of their efficiency. Linear trend lines enable the comparison of the differences between the DRL system and the benchmark.

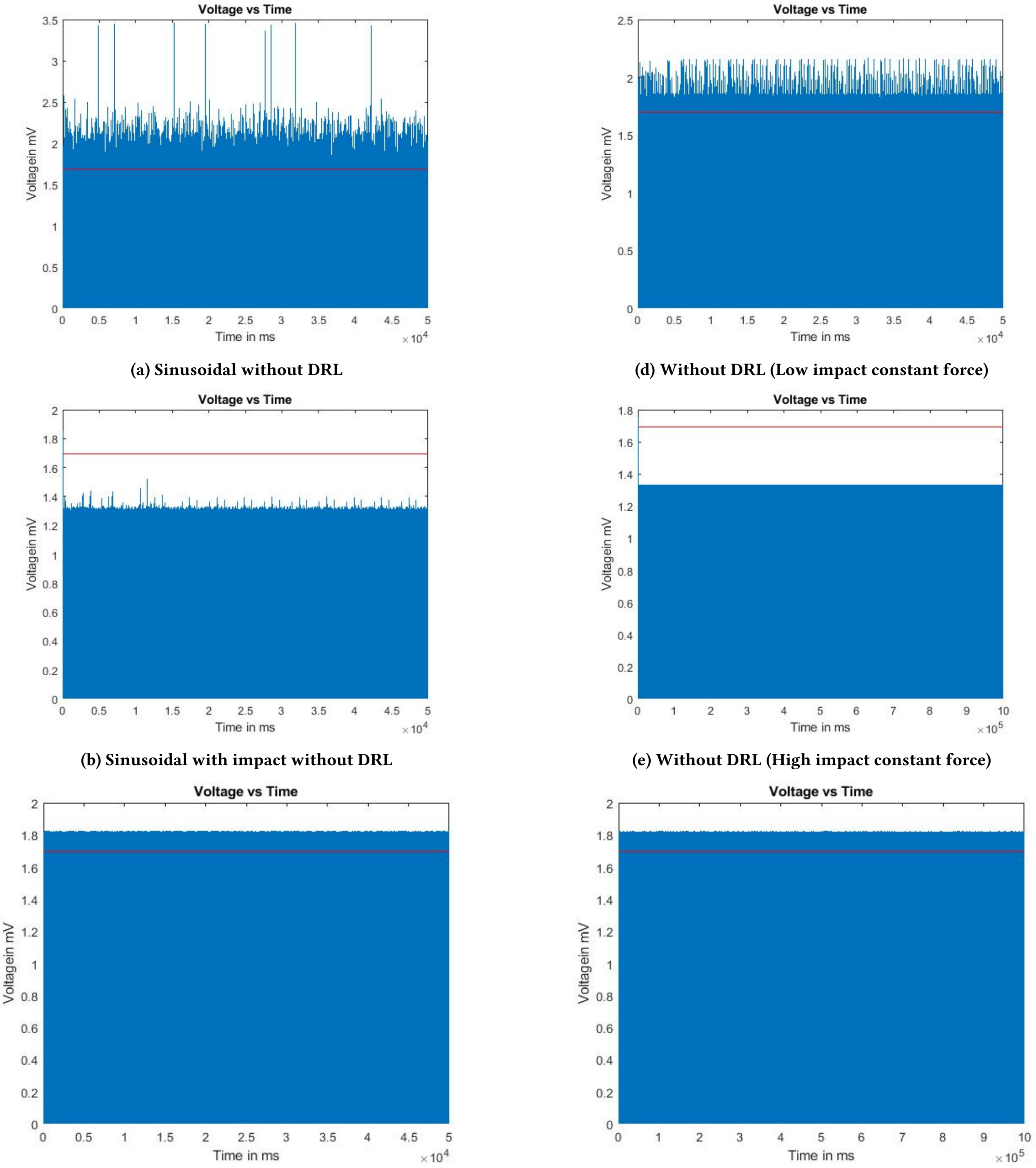
As revealed in Figure 9a a comparison can be made between the percentage performance of the algorithms over time. The DRL based algorithm displays stable performance and predictable accuracy in the long run. Displaying stability in an unsupervised algorithm is a promising aspect. By optimising the structure of DNN using historical data may enable achieving optimal performance for unseen conditions.

Figure 9b indicates that the latency of the DRL system surpasses the normal execution time of the benchmark. This is promising as DRL generates forces which allow the vehicle to remain within the safe limit. This indicates that the DNN parameters have been optimised for the random environment. This aspect is a superhuman ability that artificial intelligence can offer since anticipating a random environment is mathematically improbable.

The scientific reasoning behind the decision-making process of DRL can not be seen as implicit operations of DRL are not observable. The main reason behind this black box functionality is that DRL does not have a model to quantify its decision-making process. In this system, however, the model-based RDA may provide the reasoning behind the superhuman abilities of DRL. Understanding the decision-making process of applications of deep neural networks is a newly emerging field of research known as eXplainable Artificial Intelligence (XAI). Henceforth, extensions of this research may contribute to the field of XAI.

The novelty of the system can be summarised as follows.

- Applying DRL in unobservable environments with the aid of a bio-inspired hybrid model.
- Utilising an anomaly detection algorithm (RDA) for position control and improving it for an anomaly avoidance algorithm with the aid of DRL.



(a) Sinusoidal without DRL

(d) Without DRL (Low impact constant force)

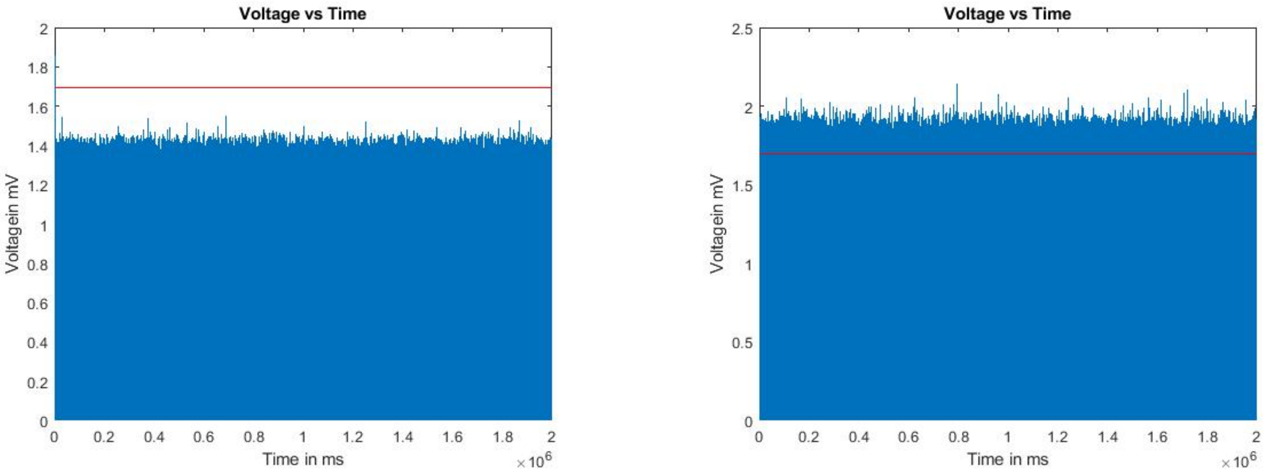
(b) Sinusoidal with impact without DRL

(e) Without DRL (High impact constant force)

(c) Sinusoidal with impact with DRL

(f) With DRL (High impact constant force)

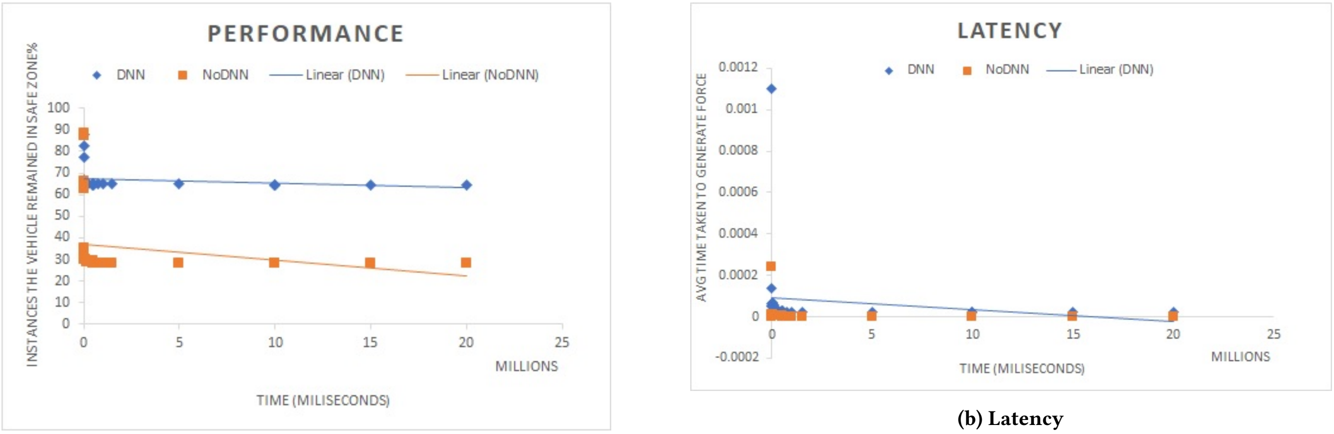
Figure 7: Time varient forces



(a) Without DRL

(b) With DRL

Figure 8: Application of DRL on Unpredictable Environments



(b) Latency

(a) Performance

Figure 9: Evolution of DRL

- Utilising plastic spiking neuronal model (PSN) to visualise and improving it to maintain its plasticity with the aid of RDA and DRL.
- Utilising the voltage-time graphs of model-based RDA-PSN combination to describe the decision-making mechanism of DNN

6 CONCLUSIONS AND FUTURE DEVELOPMENTS

This paper addressed the issue of the validity of action generated by a reinforcement learning algorithm in unpredictable environments. In our approach, we proposed a bio-inspired homeostatic model to introduce stability to the unpredictable environment. The model includes RDA, an artificial immune system based, vehicle-centric, homeostatic, control system and PSN, a homeostatic, position monitoring, spiking neuronal model. The hybrid of RDA-PSN model operates by performing auto-corrections (RDA) and visualisation (PSN).

The hybrid of RDA-PSN model operated satisfactorily in milder environments. However, when the environmental forces get significantly large, the performance of the hybrid model is hindered.

However, the model provided critical state prediction functionality that the DRL requires. With the hybrid model, the DRL was successfully applied in unpredictable environments.

In the DRL system, as the DNN learns to generate the optimal action for the unpredictable conditions, the vehicle may remain within the safe zone longer. As the subsequent environmental changes may not cause malpositions (as they become manageable by the RDA to maintain the vehicle within the safe zone), the DNN is accessed less frequently. Hence, ultimately, the system may consume fewer computing resources compared to the SOA. The system is lightweight and robust and ideal for real-time operations which surpasses the benefits of the SOA.

By observing the voltage-time graphs of PSN, the behaviour of DRL and the underlying decision-making process of the DNN becomes explainable. For example, our results showed that DRL can learn the unpredictable environment and the underlying decision-making process for this knowledge is observable from the PSN graphs. Therefore, the system may open doors for investigating new territories of eXplainable Artificial Intelligence (XAI) applications.

REFERENCES

[1] Ashiq Anjum, Richard McClatchey, Arshad Ali, and Ian Willers. 2006. Bulk scheduling with the DIANA scheduler. *IEEE Transactions on Nuclear Science* 53, 6 (2006), 3818–3829.

[2] Charlie Baker, Ashiq Anjum, Richard Hill, Nik Bessis, and Saad Liaquat Kiani. 2012. Improving cloud datacentre scalability, agility and performance using OpenFlow. In *2012 Fourth International Conference on Intelligent Networking and Collaborative Systems*. IEEE, 20–27.

[3] Andrew J Barry, Peter R Florence, and Russ Tedrake. 2018. High-speed autonomous obstacle avoidance with pushbroom stereo. *Journal of Field Robotics* 35, 1 (2018), 52–68.

[4] Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, PMLR, International Convention Centre, Sydney, Australia, 449–458.

[5] Michele Alessandro Bucci, Onofrio Semeraro, Alexandre Allauzen, Guillaume Wisniewski, Laurent Cordier, and Lionel Mathelin. 2019. Control of chaotic systems by Deep Reinforcement Learning. (2019). arXiv:arXiv preprint arXiv:1906.07672

[6] Sagarnil Das. 2018. Simultaneous Localization And Mapping (SLAM) using RTAB-Map. (2018). arXiv:arXiv:1809.02989

[7] Ildefons Magrans de Abril and Ryota Kanai. 2018. Curiosity-driven reinforcement learning with homeostatic regulation. In *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, IEEE, Piscataway, NJ, 1–6.

[8] Wenli Du, Ying Tian, and Feng Qian. 2013. Monitoring for nonlinear multiple modes process based on LL-SVDD-MRDA. *IEEE Transactions on Automation Science and Engineering* 11, 4 (2013), 1133–1148.

[9] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. 2014. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, Cambridge, UK.

[10] Irfan Habib, Ashiq Anjum, Richard Mcclatchey, and Omer Rana. 2013. Adapting scientific workflow structures using multi-objective optimization strategies. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)* 8, 1 (2013), 4.

[11] Khawar Hasham, Antonio Delgado Peris, Ashiq Anjum, Dave Evans, Stephen Gowdy, José M Hernandez, Eduardo Huedo, Dirk Hufnagel, Frank van Lingen, Richard McClatchey, et al. 2011. CMS workflow execution using intelligent job scheduling and data access strategies. *IEEE Transactions on Nuclear Science* 58, 3 (2011), 1221–1232.

[12] James A Hilder, Nick DL Owens, Mark J Neal, Peter J Hickey, Stuart N Cairns, David PA Kilgour, Jon Timmis, and Andy M Tyrrell. 2012. Chemical detection using the receptor density algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42, 6 (2012), 1730–1741.

[13] Rentian Huang, Hissam Tawfik, and Atulya Nagar. 2011. Towards an artificial immune system for online fraud detection. In *International Conference on Artificial Immune Systems*. Springer, 383–394.

[14] Anju P Johnson, Junxiu Liu, Alan G Millard, Shvan Karim, Andy M Tyrrell, Jim Harkin, Jon Timmis, Liam J McDaid, and David M Halliday. 2017. Homeostatic fault tolerance in spiking neural networks: A dynamic hardware perspective. *IEEE Transactions on Circuits and Systems I: Regular Papers* 65, 2 (2017), 687–699.

[15] Saad Liaquat Kiani, Ashiq Anjum, Michael Knappmeyer, Nik Bessis, and Nikolaos Antonopoulos. 2013. Federated broker system for pervasive context provisioning. *Journal of Systems and Software* 86, 4 (2013), 1107–1123.

[16] Anna Ladi, Jon Timmis, Andy Tyrrell, and Richard G Smith. 2012. An Automated Sniffer Dog: Real-time, Adaptive Molecular Signature Detection. *Sponsoring Institutions* (2012), 64.

[17] HuiKeng Lau, Iain Bate, Paul Cairns, and Jon Timmis. 2011. Adaptive data-driven error detection in swarm robotics with statistical classifiers. *Robotics and Autonomous Systems* 59, 12 (2011), 1021–1035.

[18] TiongHoo Lim, HuiKeng Lau, Jon Timmis, and Iain Bate. 2012. Immune-inspired self healing in wireless sensor networks. In *International Conference on Artificial Immune Systems*. Springer, 42–56.

[19] Yi Lin, Fei Gao, Tong Qin, Wenliang Gao, Tianbo Liu, William Wu, Zhenfei Yang, and Shaojie Shen. 2018. Autonomous aerial navigation using monocular visual-inertial fusion. *Journal of Field Robotics* 35, 1 (2018), 23–51.

[20] Giuseppe Loianno, Davide Scaramuzza, and Vijay Kumar. 2018. Special issue on high-speed vision-based autonomous navigation of uavs. *Journal of Field Robotics* 1, 1 (2018), 1–3.

[21] Richard McClatchey, Andrew Branson, Ashiq Anjum, Peter Bloodsworth, Irfan Habib, Kamran Munir, Jetendr Shamdasani, Kamran Soomro, neuGRID Consortium, et al. 2013. Providing traceability for neuroimaging analyses. *International journal of medical informatics* 82, 9 (2013), 882–894.

[22] Richard McClatchey, Irfan Habib, Ashiq Anjum, Kamran Munir, Andrew Branson, Peter Bloodsworth, Saad Liaquat Kiani, neuGRID Consortium, et al. 2013. Intelligent grid enabled services for neuroimaging analysis. *Neurocomputing* 122 (2013), 88–99.

[23] Nick DL Owens, Andrew Greensted, Jon Timmis, and Andy Tyrrell. 2013. The receptor density algorithm. *Theoretical Computer Science* 481 (2013), 51–73.

[24] Bas J Pijnacker Hordijk, Kirk YW Scheper, and Guido CHE De Croon. 2018. Vertical landing for micro air vehicles using event-based optical flow. *Journal of Field Robotics* 35, 1 (2018), 69–90.

[25] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. 2017. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, PMLR, International Convention Centre, Sydney, Australia, 2817–2826.

[26] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354.

[27] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press, 1 Rogers Street, Cambridge, MA 02142.

[28] Frank van Lingen, Conrad Steenberg, Michael Thomas, Ashiq Anjum, Tahir Azim, Faisal Khan, Harvey Newman, Arshad Ali, Julian Bunn, and Iosif Legrand. 2005. The Clarens Web service framework for distributed scientific analysis in grid projects. In *2005 International Conference on Parallel Processing Workshops (ICPPW'05)*. IEEE, 45–52.

[29] Frank Van Lingen, M Thomas, T Azim, I Chitnis, A Anjum, D Bourilkov, M Kulkarni, C Steenberg, RJ Cavanaugh, J Bunn, et al. 2005. Grid enabled analysis: architecture, prototype and status. (2005).

[30] Harm Van Seijen, A Rupam Mahmood, Patrick M Pilarski, Marlos C Machado, and Richard S Sutton. 2016. True online temporal-difference learning. *The Journal of Machine Learning Research* 17, 1 (2016), 5057–5096.

[31] Gref Welch and Gary Bishop. 2006. *An introduction to the Kalman Filter*. Technical Report. Department of Computer Science, Chapel Hill, NC, USA.

[32] Xianbo Xiang, Caoyang Yu, Lionel Lapierre, Jialei Zhang, and Qin Zhang. 2018. Survey on fuzzy-logic-based guidance and control of marine surface vehicles and underwater vehicles. *International Journal of Fuzzy Systems* 20, 2 (2018), 572–586.

[33] Muhammad Usman Yaseen, Ashiq Anjum, Omer Rana, and Richard Hill. 2018. Cloud-based scalable object detection and classification in video streams. *Future Generation Computer Systems* 80 (2018), 286–298.

[34] Ali Reza Zamani, Mengsong Zou, Javier Diaz-Montes, Ioan Petri, Omer Rana, Ashiq Anjum, and Manish Parashar. 2017. Deadline constrained video analysis via in-transit computational environments. *IEEE Transactions on Services Computing* (2017).

[35] Jacek M Zurada. 1992. *Introduction to artificial neural systems*. Vol. 8. West publishing company St. Paul, Eagan, Minnesota.