Models for reinforcement learning and design of a soft robot inspired by *Drosophila* larvae

Tianqi Wei

Doctor of Philosophy
Institute of Perception, Action and Behaviour
School of Informatics
University of Edinburgh
2019

Abstract

Designs for robots are often inspired by animals, as they are designed mimicking animals' mechanics, motions, behaviours and learning. The *Drosophila*, known as the fruit fly, is a well-studied model animal. In this thesis, the *Drosophila* larva is studied and the results are applied to robots. More specifically: a part of the *Drosophila* larva's neural circuit for operant learning is modelled, based on which a synaptic plasticity model and a neural circuit model for operant learning, as well as a dynamic neural network for robot reinforcement learning, are developed; then *Drosophila* larva's motor system for locomotion is studied, and based on it a soft robot system is designed.

Operant learning is a concept similar to reinforcement learning in computer science, i.e. learning by reward or punishment for behaviour. Experiments have shown that a wide range of animals is capable of operant learning, including animal with only a few neurons, such as *Drosophila*. The fact implies that operant learning can establish without a large number of neurons. With it as an assumption, the structure and dynamics of synapses are investigated, and a synaptic plasticity model is proposed. The model includes nonlinear dynamics of synapses, especially receptor trafficking which affects synaptic strength. Tests of this model show it can enable operant learning at the neuron level and apply to a broad range of NNs, including feedforward, recurrent and spiking NNs.

The mushroom body is a learning centre of the insect brain known and modelled for associative learning, but not yet for operant learning. To investigate whether it participates in operant learning, *Drosophila* larvae are studied with a transgenic tool by my collaborators. Based on the experiment and the results, a mushroom body model capable of operant learning is modelled. The proposed neural circuit model can reproduce the operant learning of the turning behaviour of *Drosophila* larvae.

Then the synaptic plasticity model is simplified for robot learning. With the simplified model, a recurrent neural network with internal neural dynamics can learn to control a planar bipedal robot in a benchmark reinforcement learning task which is called bipedal walker by OpenAI. Benefiting efficiency in parameter space exploration instead of action space exploration, it is the first known solution to the task with reinforcement learning approaches.

Although existing pneumatic soft robots can have multiple muscles embedded in a component, it is far less than the muscles in the *Drosophila* larva, which are well-organised in a tiny space. A soft robot system is developed based on the muscle pattern of the *Drosophila* larva, to explore the possibility to embed a high density of muscles in a limited space. Three versions of the body wall with pneumatic muscles mimicking the muscle pattern are designed. A pneumatic control system and embedded control system are also developed for controlling the robot. With a bioinspired body wall will a large number of muscles, the robot performs lifelike motions in experiments.

Lay Summary

Understanding of animals can provide inspiration to the design of robots. Even tiny animals can have impressive learning capability and motion capability that a robot does not have. For example, a fruit fly larva can learn by trial and error with a tiny brain as well as move flexibly without limbs in complex environments. The learning by trial and error and the flexible motions are essential capabilities for a robot to work in realistic situations.

Study of the brain of the fruitfly suggests that oscillation of synapses, which are the connections between neurons, can help a fruit fly larva learn by trial and error. This finding is applied to the training of neural networks and teaching a robot to run in a 2D-video-game-like environment. Study of the layout of muscles in the fruit fly larva suggests how this contributes to its motion ability. A soft robot is designed with an effort to reproduce the layout. With a simplified layout of muscles, the robot is capable of life-like motions

Acknowledgements

Many thanks to my principal supervisor Barbara Webb. She gives me the chance to enter the interdisciplinary field of computational neuroscience and robotics, and has huge patience in reviewing my manuscripts.

Also thanks to my assistant supervisor Adam Stokes, who helped me a lot in designing and building the soft robot.

Thanks to my colleagues Jane Loveless, Konstantinos Lagogiannis, Antoin Wystrach and David Wood. We cooperated in Project MINIMAL (Miniature Insect Model for Active Learning), which supported my PhD study.

I would also give thanks to my parents, who provide me with a firm base to study aboard.

Special thanks to my wife's companionship and support during my PhD study.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Tianqi Wei*)

魏天骐

To my wife.

Table of Contents

List of Figures

Chapter 1

Introduction

Animals provide abundant inspiration for robot design, from the structure, such as humanoid robots and four-legged robots, to the approach of perceptions, such as vision and tactile. This work studies *Drosophila* larvae (fruit fly maggots) for the development of robot reinforcement learning algorithms and design of soft robots. They are two fields in robot research that can contribute to each other. On the one hand, reinforcement learning requires action explorations which is dangerous for traditional rigid-body robots, while soft robots are safer on in uncharted contacts. On the other hand, soft robots are hard to control with traditional robot control approaches, while robot reinforcement learning can provide potential alternatives. *Drosophila* larvae are capable of operant learning and flexible in controlling their soft bodies. Hence they are studied as reference models in this work for building reinforcement learning models and designing a robot. The studies in this thesis cross multiple topics include synapse modelling, neural circuit modelling, robot reinforcement learning and soft robotics. For the convenience of understanding, this chapter is only an overview of the work. Detailed literature reviews of the topics in more depth will be presented in later chapters.

Robot learning can enhance the adaptability of robots to various tasks and reduce the routines of users in deploying the robots (Sigaud and Peters, 2012; Arulkumaran et al., 2017; Kober et al., 2015). Robot reinforcement, which is that robots learn by trying, is a subset of robot learning. By reinforcement learning, robots can perform tasks without prior knowledge. However, there are two significant challenges in robot reinforcement learning: low effectiveness of action exploration, such as action exploration in continuous action space using discrete noise ; and dangerous in action exploration during learning, such as interference among bodies and collision with objects.

Existing mainstream robot learning models are introduced from machine learning. However, the tasks of machine learning are usually different from the tasks of robot learning. The former are time-invariant learning tasks, such as curve fitting and vector classification, while the latter are dynamic tasks such as motion control and decision making in varying situations(Sigaud and Peters, 2012). The difference impedes the effectiveness of existing robot learning models.

Another trend in building robot learning models is to draw inspiration from animals and base theory on neurophysiology, ethology or psychology, such as computational neuroscience models (Chiel and Beer, 1997), evolutionary models (Weng, 2004; Bongard, 2013), imitation learning (Hussein et al., 2017). As existing examples in nature solve the problems we are facing in robot learning, by referencing them, robot learning models can be more promising to achieve the targets.

Soft robots are a type of robots that are significantly safer than conventional rigid robots(Lee et al., 2017). For conventional rigid robots in traditional robot application scenarios and with conventional control approaches, safety to the environment or human is guaranteed by predefined working space and physical isolation, and safety to objects and robot itself is by predefined functions of perception, path/motion planning, and compliant control. However, in some scenarios, such as robots working in human environments, robots cannot be isolated from environment and humans; during learning, especially reinforcement learning, those functions are not predefined, and motion explorations are usually unpredictable, so collision is highly possible to happen. Properties of soft materials can avoid injury to robots or the environment during collision (Lee et al., 2017), which makes soft robot ideal for robot reinforcement learning, even in the same environment with humans.

This work is an effort to provide some solutions to meet the challenges. Computational neuroscience models are adopted and developed for reinforcement learning, and a soft pneumatic robot inspired by larvae is developed as a soft robot with high degree-of-freedoms and bionic muscle patterns. As *Drosophila melanogaster* is well researched in both the neural system and the motor system, it is selected as the biological prototype.

1.1 The model animal

Drosophila melanogaster is a model animal in biology and has been widely studied. Of great convenience in studying *Drosophila melanogaster* is that people have developed abundant transgenic toolkits for studying it (Olsen and Wilson, 2008). For example, there are various genetic lines of *Drosophila melanogaster* with the GAL4/UAS system, which is a biochemical method for studying gene expression and function in organisms developed by Brand and Perrimon (1993). There are genetic lines with fluorescent muscles that show the muscle patterns and motions in live larvae (Heckscher et al., 2012), and genetic lines in which their reward neurons can be activated by light of specific wavelength(Hige et al., 2015). This convenience facilitates both studies of their motor systems to design the soft robot and their learning behaviours to evaluate neural circuit models. Another convenience in studying *Drosophila melanogaster* is that they have a well-balanced complexity of behaviours and number of neurons. The neural system of a *Drosophila melanogaster* larva only has about 10^5 neurons (Chiang et al., 2011). For comparisons, a rat has about 2×10^8 neurons, and a human has about 8.6×10^{10} neurons (Herculano-Houzel, 2009). However, *Drosophila melanogaster* can have complex behaviours. *Drosophila melanogaster* adults have behaviours such as courtship and navigation, and abilities to learn such as associative learning (Aso et al., 2014a) and operant learning (Brembs, 2009), and motor controls such as walking and flying. Although a larva has less ability and a smaller brain than an adult, they have a similar architecture of neural circuits and the ability to learn. Some findings indicate that *Drosophila* larvae are capable of associative learning (Gerber and Stocker, 2007) and operant learning (Eschbach, 2011).

Although the size of a *Drosophila* larva brain is small, it has a complex structure, which is similar to other insect brains. An essential component of their central neural system is the mushroom body, which is known to be a ”learning centre” and has an architecture similar to mammals’ cerebellum (Farris, 2011). With recently developed technologies of observation, such as Two-photon excitation microscopy, the detailed connections at the synapse level can be observed. Recent research has shown that the architecture of the mushroom body provides a recurrent multilayer circuit for associative learning (Aso et al., 2014a). There are also some existing models of the mushroom body, such as the work by Wessnitzer et al. (2012), showing the possible mechanics and dynamics for associative learning, but not yet the potential role of the mushroom body in operant learning.

Experiments have shown that *Drosophila* is capable of operant learning, such as heat box experiment (Putz and Heisenberg, 2002) and fly torque experiment (Brembs and Heisenberg, 2000). In the fly torque experiment, turning behaviour in one direction is punished during training, and a significant decrease in the proportion of turns in that direction is observed. Because of its architecture and that it is known as a learning centre, the mushroom body is a potential candidate for the basis of operant learning. To prove it, a biological experiment was conducted by my colleagues using transgenic *Drosophila* larvae, for whom the light of specific wavelength causes a reward signal to be released in the mushroom body when the larva turns/runs. Operant learning behaviours were observed, and the modelling of the learning capability is thus one target of the work in the thesis.

1.2 Robot learning

Robots are successfully applied in various tasks, where robots are controlled with approaches based on classical control theories. However, these control approaches are only suitable for process-specific tasks in invariant environments, such as work on assembly lines, but not flexible tasks in dynamic environments, such as work at home. This limitation constrains applications of robots.

Robot learning and robot artificial intelligence are the approaches people try to break the limitations (Pierson and Gashler, 2017; Argall et al., 2009; Deisenroth et al., 2011). There are efforts to introduce recent developments in machine learning to robot control, such as supervised learning of image recognition and reinforcement learning of motion control (Arulkumaran et al., 2017; Kober et al., 2014). However, the basis of algorithms used in the efforts, such artificial neural networks, are designed for time-invariant learning tasks such as curve fitting and vector classification, but not for dynamic tasks robots can encounter, such as motion control and decision making in changing situations. Hence, lots of the effort to introduce these approaches to robot control are spent on building adapters between static learning algorithms to dynamic tasks, and efficiency is lost. It is partly because those models were developed to describe the results of higher-level intelligence activity (static abstract concepts) but are applied to lower level dynamic activity (motor control). However, for animals, responses for lower level dynamic activities are developed first to form the basis of higher-level intelligence activities. Hence, to enable robots with animal-like intelligence that have high efficiency in dynamic tasks, we need new approaches that are directly for dynamic tasks based on lower level activities, then build the learning system up suited to the level that able to process higher-level intelligence.

There are two ways that we can draw inspiration from animals to improve robot learning: neural circuit architectures and learning rules.

The feedforward neural networks in computer science, which form the majority of applied neural networks, are static. They do not have computational abilities that dynamic systems have, such as differentiation and integration. As a comparison, classical control systems and biological neural circuits have these abilities. Missing the abilities is potentially a key reason why neural networks are less efficient in dynamic tasks than in static tasks. If we can introduce those computational abilities to neural networks, the efficiency can be significantly improved.

However, dynamics alone are not sufficient to make a neural architecture suitable for dynamic tasks. Whether a type of architecture is practical in a specific task is also constrained by the learning rules that can be applied. The most popular learning rule for neural networks is gradient descent (GD) with error backpropagation (BP). These require networks to be time-invariant, so the neural network architectures in which they can be applied are constrained. For example, most recurrent neural networks (except some special cases such as Long short-term memory) suffer from the gradient vanish/explosion problem in which the tricks for deep feedforward neural networks are not applicable, thus cannot been efficiently trained with GD and BP; and neural networks with dynamical neuron models, such as the models in computational neuroscience, are time-variant, thus cannot been efficiently trained with GD and BP either. There are some other biologically plausible or inspired learning rules, such as the Hedonistic Synapse (Seung, 2003) and modulated spike-timing-dependent plasticity (MSTDP) (for a review, see Fremaux et al. (2010)). However, these models only apply to spiking neural networks, which need more computational resource than firing-rate neural networks, and have to introduce some arbitrary mechanism, such as a random number generator, to explore action space (i.e. generate random number sequences for joint angles). Hence, if a new learning rule can remove those constraints and support the training of neural networks with various architectures, proposing new neural network architectures can be more straight forward and easier. Moreover, reinforcement learning with existing learning rules is based on action exploration and uses the exploration information to calculate the parameter updates of neural networks/circuits. If a new learning rule can directly explore the parameters and evaluate the explored parameters according to the resulted actions, the learning process could be simplified. The works in this thesis show that parameter exploration is possible based on the known biophysical properties of synapses, which is detailed in chapter 2.

1.3 Soft Robot

The soft robot is a type of robot being developed recently (Lee et al., 2017; Trivedi et al., 2008). As the name implies, the most apparent difference between soft robots and traditional robots is that they have soft bodies. An advantage of soft robots is safety. They have less possibility to hurt the people or damage the objects they are interacting with, and are resistant to mechanical Damage (Martinez et al., 2014). There are various types of soft robots with different shapes or configurations(Marchese et al., 2015), such as starfish-like grapes (Stokes et al., 2014), starfish-like walker (Shixin Mao et al., 2013), finger with serial air chambers (Galloway et al., 2016) .

These types of soft robots usually have an external system for power supplies and control. Compared with them, a tubular robot has more potential to provide internal space for containing the external system, or for operating objects, such as convey and grasp. Although the integration of a soft robot body and the external system requires the external system to be generally soft which still needs further fundamental research, exploration of soft tubular robots for the design, manufacture and control is still an essential step.

Drosophila larvae can be a reference for the design of soft robots. The *Drosophila* larva motor system has a large number of muscles distributed on a layer of their body wall with patterns, harmonically driving the body for subtle and complex motions. For example, the crawling forward motion is not merely a wave of contraction and extension, but also includes sequentially lifting up and placing down of body segments, hooking to ground with the mouth hooks, as well as the piston-like motion of viscera inside the body. The motions need fine coordination of multiple muscles among several body segments. There is no existing soft robot replicate the muscle patterns of a larva. A robot designed based on the *Drosophila* larva is help for more accurate replication of the motions and understanding of the motor system.

1.4 Contributions

The work presented in this thesis is inter-disciplinary, as are the hypotheses and contributions. The related disciplines include computational neuroscience, operant learning, robot learning and soft robotics.

Hypotheses:

1. the dynamics in synapses are non-linear so that chaos can exist;

2. the chaotic dynamics support and facilitate operant learning, which is learning by trial and error;
3. with synaptic dynamics, a mushroom body model is capable of operant learning;
4. a synapse model with the abstracted dynamics can train neural networks that are unable to be trained with backpropagation, for robot reinforcement learning;
5. a soft robot with larval *Drosophila* muscle patterns can have the ability of life-like motion.

Results:

1. Chaos appears in the mathematical model built upon known biological findings of synapses, which supports the hypothesis that chaotic dynamics can exist in synapses;
2. with the presence of the neural modulator, synapses with the chaotic dynamics successfully enabled operant learning of a feedforward neuron, a central pattern generator, and a spiking neural network;
3. with the synapse model, a mushroom body model is capable of operant learning;
4. with the simplified version of the synapse model a neural network outperformed existing models in a bipedal locomotion reinforcement learning task;
5. applying the muscle patterns of *Drosophila* larvae to a soft robot without modification is not practical, while some adjustments, such as merging of adjacent muscles that have similar functions and removal of materials that hinder the relative sliding between muscles, can improve the performance of the robot.

Highlights:

1. the potential chaos in synapses was ignored in previous models, and the proposed model in this thesis is the first synapse model with chaos;
2. the learning rule with the synapse model is compatible with existing artificial neural networks as well as biologically plausible neural architectures;
3. the mushroom body model agrees with a biological and behaviour experiment of Drosophila larva operant learning, which was not captured by existing mushroom body models;
4. the learning process with the simplified synapse model is a type of parameter exploration, thus the action exploration of the agent is correlated with sensory input, and the learning is more efficient and effective than existing algorithms;
5. simplified and abstracted from the *Drsophila* larva muscle pattern, the soft robot has 24 compact muscles in a single layer.

Outcomes:

1. A journal paper *A model of operant learning based on chaotically varying synaptic strength* was published in the *Neural Networks* (Wei and Webb, 2018a). It is about hypotheses 1 and 2, results 1 and 2, as well as highlights 1 and 2. Barbara Webb is the co-author of the paper, who advised on the work and the writing of the paper. This paper is included in chapter 2.
2. A paper is in preparation for hypothesis 3, result 3, and highlight 3.
3. A conference paper *A Bio-inspired Reinforcement Learning Rule to Optimise Dynamical Neural Networks for Robot Control* was published in *the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Wei and Webb, 2018b). It is about hypothesis 4, result 4,and highlight 4. Barbara Webb is the co-author of the paper, who advised on the work and the writing of the paper. This paper is included in chapter 4.
4. A conference paper *A soft pneumatic maggot robot* was published in *The 5th International Conference on Biomimetic and Biohybrid Systems* (Wei et al., 2016). It is about hypotheses 5, result 5,and highlight 5. Adam Stokes and Barbara Webb are the co-authors of the paper, who advised on the work and the writing of the paper. This paper is included in chapter 5. A soft robot is designed and built.

1.5 Structure of this thesis

For the convenience of understanding, this chapter is only an overview of the work. More in-depth and detailed literature reviews of the topics will be presented in later chapters.

Chapter 2 details the biologically plausible synapse model with chaotic dynamics and the learning rule based on the model.

Chapter 3 shows the mushroom body model for operant learning with the synapse model and the learning rule.

Chapter 4 provides an example for the application of the simplified model in a reinforcement learning task about robot dynamic control.

Chapter 5 is about the soft pneumatic maggot robot system.

Chapter 6 discusses the work as a whole and future work.

Chapter 2

The “Dynamic Synapse” Learning Model

2.1 Background

A synapse is a structure that conveys signals between neurons. It usually consists of an axon terminal, a synaptic cleft and a dendrite spine. In the neural networks in the field of computer science [1], a synapse is simplified as a static weight that can be adjusted by learning. However, in fact, a synapse is very dynamic and complex, in aspects such as its microscopic architectures maintaining its functions and its macroscopic effect on the neural circuits.

For synapses that pass chemical signals between neurons, the processes that relate to neurotransmitter release and reception contribute the majority of its dynamics. For example, (1) because neurotransmitter is stored in synaptic vesicles at axon terminals and the speed of producing and recycling is slower than releasing, the conduction of the synapse can decrease after neurotransmitter is exhausted; (2) because neurotransmitter moves between neurons by diffusion in the synaptic cleft, there is a delay during signal transmission; and (3) because neurotransmitter receptors can move between the post-synaptic regions where neurotransmitter are reachable or unreachable, the sensitivity of the dendrite is not constant. The research presented in this chapter exploits the third type of dynamics to form the basis of the plasticity of synapses, introducing a new learning rule for neural circuits or networks.

[1] In this thesis, I use ’neural network’ to refer to the computational architecture in computer science that is inspired by biological neural circuits, while ’neural circuit’ the actual biological structures.

2.2 A model of operant learning based on chaotically varying synaptic strength

The paper shown in the following pages is the journal paper *A model of operant learning based on chaotically varying synaptic strength* published on the *Neural Networks* (Wei and Webb, 2018a). It is about Hypotheses 1 and 2, Results 1 and 2, as well as Highlights 1 and 2 in Chapter 1. Barbara Webb is the co-author of the paper, who advised on the work and writing of the paper.

The paper reviews related work and findings, including operant learning, existing synaptic models, chaos, learning with chaos, the chaos in biological systems, the receptor dynamics and the effect of the neuromodulator on the receptor dynamics.

Based on the findings, the “dynamic synapse” model is built and presented. A simulation result without neuromodulator shows that chaotic fluctuation emerged from the model. Three toy experiments show that the “dynamic synapse model” is capable of operant learning with a neuromodulator. The experiments are, respectively, a linear summation neuron maximising its output with limited neurotransmitter receptors, a central pattern generator approximating its output frequency to target frequency, and a spiking neuron network learning to control an agent for foraging and avoiding a predator.

The relation of the “dynamic synapse” model and existing models, as well as the potential variations of the model, are discussed.

The mathematical equations are detailed in the method section. Some formation problems of the equations are caused by typos and corrected in the following corrigendum.

Neural Networks 108 (2018) 114–127

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

A model of operant learning based on chaotically varying synaptic strength[☆]

Tianqi Wei[a,b,*], Barbara Webb[a]

[a] School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, EH8 9AB, United Kingdom
[b] School of Engineering, University of Edinburgh, King's Buildings, Alexander Crum Brown Road, Edinburgh, EH9 3FF, United Kingdom

ARTICLE INFO

Article history:
Received 26 April 2018
Received in revised form 12 July 2018
Accepted 2 August 2018
Available online 11 August 2018

Keywords:
Dynamic Synapse
Operant learning
Chaos
Receptor Trafficking

ABSTRACT

Operant learning is learning based on reinforcement of behaviours. We propose a new hypothesis for operant learning at the single neuron level based on spontaneous fluctuations of synaptic strength caused by receptor dynamics. These fluctuations allow the neural system to explore a space of outputs. If the receptor dynamics are altered by a reinforcement signal the neural system settles to better states, i.e., to match the environmental dynamics that determine reward. Simulations show that this mechanism can support operant learning in a feed-forward neural circuit, a recurrent neural circuit, and a spiking neural circuit controlling an agent learning in a dynamic reward and punishment situation. We discuss how the new principle relates to existing learning rules and observed phenomena of short and long-term potentiation.

1. Introduction

Operant learning (also called operant conditioning or instrumental conditioning) is a type of learning in which a new behaviour is increased, or an existing behaviour is suppressed, by pairing it with reward or punishment. For example: (a) In a Skinner box, when a rat occasionally presses a lever, it gets some food. After a while, it increases the rate of lever pressing (Jensen, 1963). (b) In a flight simulator, a fruit fly is heated when it generates yaw torque to one side and released from heat when it generates yaw torque to the other side. In minutes the fly learns to maintain its torque in the range that is without punishment (Wolf & Heisenberg, 1991). (c) When an *Aplysia* produces a bite, the esophageal nerve can be stimulated in vivo to mimic the food signal. After training, it produces more bites than a yoked control that has received the same stimulation without the coupling to its own actions (Brembs, 2003; Cash & Carew, 1989).

Some of this research, e.g. in *Aplysia* (see review in Nargeot & Simmers, 2011), implies that mechanisms at the single neuron level can play important roles in operant learning. There are some existing single neuron or synapse models intended to account for operant learning. For example, the 'Hedonistic Synapse' is a spike-based synapse model with stochastic synaptic transmissions, where the probability of transmitter release (the synaptic strength) is updated continuously according to the correlation between the transmitter fluctuation and a reward signal (Seung, 2003). Learning models based on modulated spike-timing-dependent plasticity (MSTDP) have also been applied to operant learning, using a reward signal to alter the weight of synapses that have been tagged by STDP as contributing to the output that produced the reward (for a review, see Frémaux, Sprekeler, & Gerstner, 2010). These models only apply to spiking neural networks, and moreover, they have to introduce some arbitrary mechanism, such as a random number generator, to explore output space (i.e. generate different actions). Use of random number generators leads to the exploration of discrete output spaces with ever-present unpredictability.

An alternative option for generating exploration of the output space is chaos. Chaotic motion, which is a type of irregular motion that can exist in simple systems, has very complex, unpredictable and ergodic solutions (Eckmann & Ruelle, 1985; Tél, Gruiz, & Kulacsy, 2006). Chaos is widely found in biological systems (for a review, see Cavalieri & Koçak, 1994), including neurons and neural circuits. In a neuron, the dynamics of membrane potential and ion flows can be chaotic, as has been verified in several models, such as Canavier, Clark, and Byrne (1990), Nobukawa, Nishimura, Yamanishi, and Liu (2014) and Storace, Linaro, and De Lange (2008), and observed in the Nitella intermodal cell (Hayashi, Nakao, & Hirakawa, 1983). Simulations of neural circuits also show chaos can exist at the circuit level, e.g. Angulo-Garcia and Torcini (2014) and Sussillo (2014). A chaotic system can be a source to generate

[☆] The work is funded by European Commission under FP7-ICT (Project ID: 618045).

[*] Corresponding author at: School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, EH8 9AB, United Kingdom.
E-mail address: Tianqi-Wei@outlook.com (T. Wei).

https://doi.org/10.1016/j.neunet.2018.08.006
0893-6080/

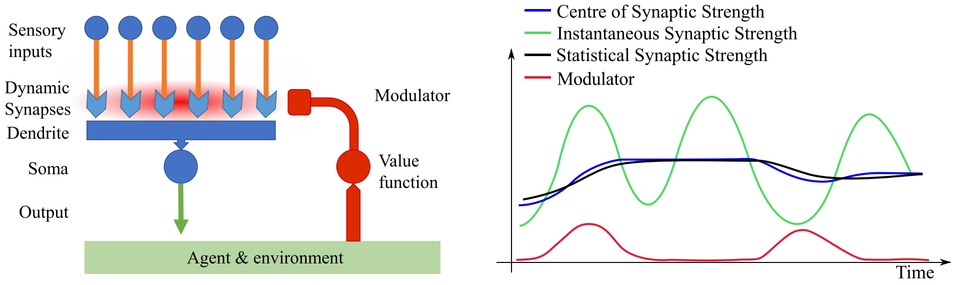


Fig. 1. Basic concept of how operant learning works with a Dynamic Synapse. (Left): A neuron has multiple inputs, and its output is the sum of the inputs multiplied by the synaptic strengths, passed through a non-linear function. Because the synapses are dynamic, their values continuously change, and thus the output will explore a space of possible outputs. A value function on the output controls the release of a modulator which alters the synaptic strengths. (Right): Illustrating the dynamic synaptic strength of one synapse. During learning, the centre of synaptic strength oscillation is shifted towards the instantaneous synaptic strength that coincides with increased modulator, e.g., as illustrated, the modulator (red) is high when the instantaneous strength (green) is high, so the centre of synaptic strength is gradually increased (blue). The modulator also affects the damping of the oscillation, so the amplitude of oscillation decreases, and the learning can converge. An observer can infer the 'effective' synaptic strength by low-pass filtering on the instantaneous synaptic strength (black) but note this is only an approximation of the actual centre of oscillation which cannot be directly observed.

unpredictable, continuous and ergodic actions for operant learning or reinforcement learning. This idea has been applied to algorithms for robot learning, such as a Fish-Catching Robot that uses a chaotic generator for unpredictable motion planning to avoid fishes adapting to repetitive motions (Inukai, Minami, & Yanou, 2015) and a hexapod robot with a chaotic Central Pattern Generator (CPG) that produces chaotic signals for exploration of new motions to free its leg from a hole in the floor (Steingrube, Timme, Woergoetter, & Manoonpong, 2011). The signals generated by a chaotic process are more continuous and more suitable for controlling a robot's (or animal's) interaction with the physical world than the signals generated by a random number generator, which are usually discrete white noise. Chaos in a physical system usually results in a more continuous and smooth variation of states than a random system. This property allows a transient delay of reward and modulator, which is common in learning in the real world. In principle, continuous and smooth trajectories can be obtained from a random number generator using interpolation, but, unlike chaos, the system will be predictable during the interpolation.

Although chaos is widely found in biological systems, the potential for chaos in synaptic dynamics and how this could support learning has not been previously considered. Here, we hypothesise that the following 'Dynamic Synapse' mechanism could underlie operant learning (Fig. 1). A neuron (Fig. 1(left)) has multiple input synapses, for which the synaptic strengths spontaneously fluctuate with uncorrelated phases (Fig. 1(right) green curve) around the centre of oscillation (Fig. 1(right) blue curve). We argue in more detail below that this could be caused by receptor trafficking. The neuron receives inputs (e.g. from sensors or other neurons), and the inputs are multiplied by the synaptic strengths, summed up and passed through a non-linear function to determine the output. The output of the neuron causes some outcome (e.g. for an agent in an environment) which results in release of a neuromodulator according to a value function (Fig. 1(right) red curve). The modulator acts to bias the centre of the synaptic strength oscillation towards the instantaneous synaptic strength, and to decrease the amplitude of oscillation. Thus the synaptic strengths will converge to match the input–output properties of the neuron to the value function.

Is there a plausible biological mechanism that could produce the hypothesised synaptic strength fluctuation? The number of neurotransmitter receptors (from now on we will refer simply to receptors) embedded in the membrane of a post-synaptic dendritic spine is a key factor in synaptic strength (Sheng & Hoogenraad, 2007). Enlargement of a dendritic spine increases its capacity for anchoring structure, including scaffold proteins and cytoskeleton, and thus the number of neurotransmitter receptors it can accommodate (Allison, Gelfand, Spector, & Craig, 1998). However, the size and the capacity are not closely coupled (Cingolani & Goda, 2008). As shown in Fig. 2, under certain conditions, synaptic strength can change without changes in spine size, and spine size can change without changes in synaptic strength.

The number of receptors in the membrane of a spine is also affected by two broad types of movement between synaptic and non-synaptic pools: lateral movement, which is mainly passive diffusion on the cell membrane; and endosomal trafficking, which is active transportation (Lau & Zukin, 2007). The lateral movement is affected by the cytoskeleton, which restricts or guides the diffusion (Jaqaman et al., 2011). In particular, the actin cytoskeleton has an active contribution to the regulation of postsynaptic receptor mobility both in and out of synapses (Cingolani & Goda, 2008). The endosomal trafficking includes endocytosis of receptors from cell membrane to endosome, intracellular transportation of endosome, and exocytosis of receptors from endosome to the cell membrane (Roth, Zhang, & Huganir, 2017). Endosomal trafficking can recycle receptors, transporting them between different regions (Petrini et al., 2009). There are also ongoing processes of receptor synthesis and degradation (Triller & Choquet, 2005).

The timescale of these receptor dynamics can be relatively fast. Receptors move from synaptic to extrasynaptic regions and vice versa usually with periods of up to a few minutes (Triller & Choquet, 2005). The size of a post-synaptic dendrite spine and the amount of actins in it oscillate in a time scale from tens of seconds (in immature dendrite spine) to a half hour (in a mature synapse) (Honkura, Matsuzaki, Noguchi, Ellis-Davies, & Kasai, 2008; Koskinen & Hotulainen, 2014). Receptors anchored to the actin cytoskeleton (Hausrat et al., 2015) can move with the actin flow (Sergé, Fourgeaud, Hémar, & Choquet, 2003). Post-synaptic receptor dynamics have been modelled at a mesoscopic level treating the regulation of numbers of the receptors and scaffold proteins as quasi-equilibrium based on thermodynamic theory (Sekimoto & Triller, 2009). The model proposed in Haselwandter, Calamai, Kardar, Triller, and Azeredo Da Silveira (2011) describes formation and stability of synaptic receptor domains as a reaction–diffusion system. We note these models are dynamic, but not chaotic. We propose (i) that the complexity of post-synaptic dynamics (Choquet & Triller, 2013), especially receptor trafficking (Triller & Choquet, 2005) can support chaos and (ii) that this can provide a mechanism for operant learning as described in Fig. 1.

It is notable that dopamine has been shown to affect the same receptor trafficking dynamics (Sun, Milovanovic, Zhao, & Wolf, 2008). This supports the possibility that, in an operant learning paradigm, the relationship between the current synaptic strength (changing chaotically due to receptor trafficking) and a reward (signalled by neurotransmitter release) is a basis for learning. The

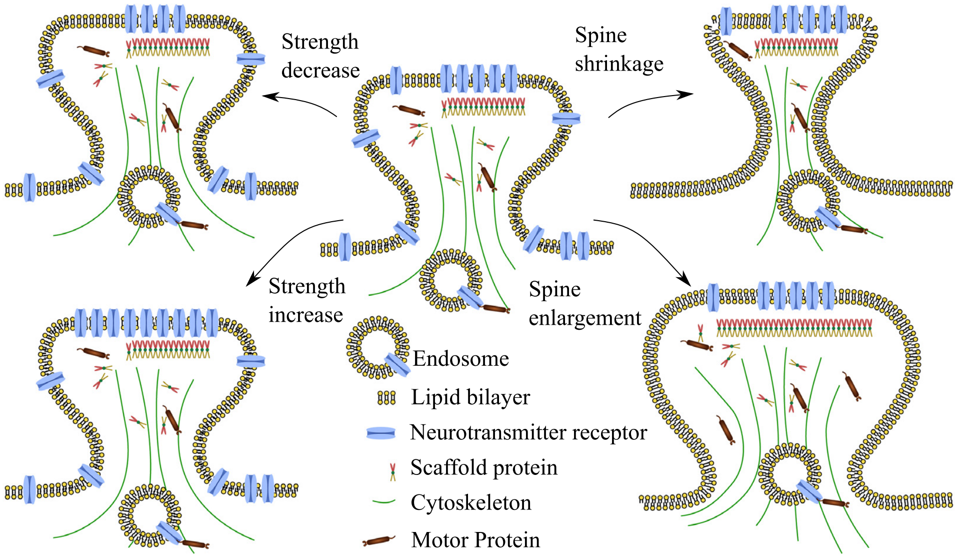


Fig. 2. Decoupling between changes in spine size and synaptic strength under certain conditions. The membrane is formed mainly by the lipid bilayer and proteins. Cytoskeleton supports the shape of the dendrite spine. There are two forms of receptor trafficking. Lateral movement of receptors is observed as Brownian motion on the membrane. Endosomal trafficking carries receptors driven by motor protein along the cytoskeleton. Scaffold proteins can help receptors to anchor, increasing the capacity of the dendrite spine to hold the receptors. On the left, the size of neural spine stays the same, but the synaptic strength (number of receptors) varies. On the right, the size of dendrite spine varies, but the synaptic strength stays the same.
Source: Modified from Cingolani and Goda (2008).

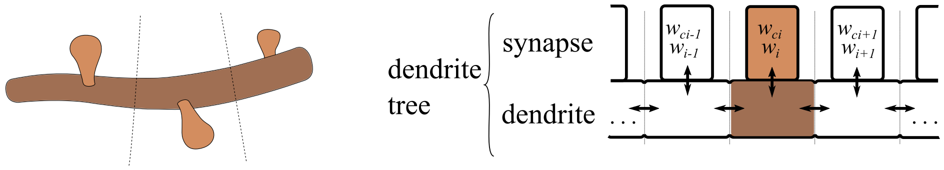


Fig. 3. (Left) A dendrite tree consists of a dendrite (in dark brown) and multiple synapses (in light brown). (Right) A schematic diagram of the dendrite tree. Receptors can move between dendrite and synapse to dynamically modify the synapse strength w_i around some centre W_{ci}.

possible role of alteration in postsynaptic receptor distribution and size of dendritic spines in learning (particularly in short-term and long-term potentiation (STP & LTP) protocols) is well established (Isaac, Nicoll, & Malenka, 1995; Kauer, Malenka, & Nicoll, 1988; Shepherd & Huganir, 2007). In Shouval, Castellani, Blais, Yeung, and Cooper (2002), Shouval et al. proposed a thermodynamic model of AMPA receptor endosomal trafficking to explain bi-directional synaptic strength variation during LTP and long-term depression (LTD). Xie, Liaw, Baudry, and Berger (1997) proposed a synapse level model in which AMPA receptors are attracted towards NMDA receptors during STP, and some of the AMPA receptors become anchored near the NMDA receptors while others diffuse again during LTP. The plausibility that such changes in receptor distribution could alter synaptic efficiency has also been demonstrated (Allam et al., 2015).

In the learning model presented here, we do not include any Hebbian process (see discussion). Instead, we allow chaotic synapses in a neuron to explore possible synaptic strengths; the neuron thus becomes a function on its inputs with chaotic coefficients, generating unpredictable output signals to explore action spaces. If the consequences of the action are reflected in a reinforcement signal delivered to the synapses, the parameters of the chaos can be altered to centre around synaptic strengths that optimise the output. We show through simulation the learning functionality of such a system in several different scenarios.

2. Result

Our model simplifies the structure of a neuron to consist of multiple input synapses and a dendrite, which together comprise the dendritic tree (Fig. 3). We do not model the soma and axon of the neuron but simply calculate the soma's input as the sum (across the dendritic tree) of the synaptic inputs multiplied by their respective synaptic strengths, then calculate the soma's output by passing the input through a non-linear function. The number of receptors in a synapse represents the synaptic strength of the synapse. Receptors in the dendrite do not contribute any synaptic strength. Because of the receptor trafficking dynamics, the synaptic strength fluctuates spontaneously. In the methods we provide an abstracted mathematical model for receptor trafficking, but summarise here the key properties needed to support learning:

1. Spontaneously and smoothly varying synaptic strength w_i around an oscillation centre w_{ci};
2. The phases of the oscillations are not locked
3. The oscillation centre w_{ci} and amplitude depend on properties of the dendrite tree that can be altered by a learning signal.

When a neuron or network of neurons with such synapses produces output in a way that meets a specific requirement (given by a value function), modulator representing reward is released.

The modulator affects the centre of synaptic strength oscillation, which shifts towards the instantaneous synaptic strength at the time of the modulator release. The simplest way to implement this is as a learning rule depends only on the current centre of synaptic strength oscillation, the instantaneous synaptic strength and amount of the modulator:

$$\dot{w}_{ci} = \mathrm{k}_w(w_i - w_{ci})n_\mathrm{M} \tag{1}$$

where n_M is amount of the modulator, and k_w is a coefficient controlling the learning rate. By this learning rule, a circuit with dynamic synapses can conduct operant learning, as the instantaneous synaptic strength is near or in the range that satisfy a criterion when modulator is released (note in the experiments that follow we use a slightly altered rule (Eq. (23) in Methods) to compensate for a biased drift in synaptic strength). To allow learning to converge, the learning rule should also reduce the oscillation amplitude (Eq. (24)). Conceptually, we relate the centre of oscillation to the capacity of a dendritic spine to hold receptors (Fig. 2; and the amplitude of oscillation to the damping of the receptor movement dynamics. We assume these can result from changes in spine size or to the scaffold cyto-skeleton complex, but do not model these explicitly.

2.1. Simulation of a dendrite tree

In Fig. 4, we show in simulation that our receptor trafficking model produces apparently chaotic and unpredictable oscillation of the synaptic weights. The simulated dynamic synapse system has six synapses, and the trajectory of the first three is plotted: it can be seen that it samples relatively evenly in the space of synaptic weight values. Fig. 4(right) shows how the range of exploration can be controlled. If the damping factor of a synapse increases, oscillation in the corresponding dimension of the plot will be narrower. If the capacity of a synapse changes, the centre of oscillation of the corresponding dimension in the plot will translate. These properties are the basis of the principle by which the system can learn and converge. In this example, the periods of the oscillations are from 10 s to 20 s. With different parameters, the periods can be in a different range, such as in tens of minutes or hours, and the oscillations still appear chaotic after the equivalent of several days of simulated time. It is important for learning in our model that the synaptic dynamic timescale matches the causal dynamics of the learning situation. That is, when the reward is delivered, the state of the synapse should still be near the state that caused the action that resulted in reward. However, the timescale cannot be too long or else the generation of new actions will be limited, and the learning might converge to a local minimum. We note there may be other factors that produce unpredictable synaptic strengths, such as Brownian movement of receptors due to thermal noise, but suggest that these may be subsumed within the higher level dynamics described above, and it is not necessary to include them as a source of noise to support learning.

2.2. Applying learning in a simple linear example

In this experiment we test learning in a single neuron with reward provided when the output is higher than a threshold and increasing. The neuron is a linear neuron, i.e. its output is the sum of the product of input values and their synaptic strengths. During the simulation, the input values of the neuron are constants ranging from 0 to 5 as shown in Fig. 5. The reward function is:

$$n_\mathrm{m} = \begin{cases} \mathrm{k}_{\mathrm{m}_1}\dot{y}(y - y_0) & \text{if } \dot{y} > 0 \wedge y - y_0 > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where n_m is the amount of modulator, k_{m_1} a coefficient, y the output of the neuron, and y_0 a threshold of y to trigger the release of modulator.

Fig. 6(a) shows the instantaneous synaptic strengths, and the labels of lines show the constant input value of corresponding synapses. The equilibrium synaptic strengths, which are also average synaptic strengths, are shown in Fig. 6(b). Note that the later equilibrium synaptic strengths have the same ordering from highest to lowest as input strengths. The neuron has a fixed total of receptors, for which it finds an efficient distribution across the synapses to maximise. Fig. 6(c) shows the output of the neuron. In the first half of the learning process, the output decreased a little because the initial value is high but not stable. In the second half, the output gradually increased. Fig. 6(d) shows the trajectory of first three synaptic strengths. The trajectory starts by exploring a large volume then gradually converges.

2.3. Tuning the period of a central pattern generator

A Central Pattern Generator (CPG) is a type of Recurrent Neural Network (RNN) which exists in many animals to control rhythmic motions, such as walking and heartbeat. It is also applied in legged robot control as an alternative to explicit motion planning (Ijspeert, 2008; Xia et al., 2017). However, online training of a CPG is difficult. People often have to tune it by hand or by offline parameter optimisation, such as brute force search or Genetic Algorithms. Our approach has a potential advantage in tuning or training a CPG because it can train a CPG online. This experiment shows an example of tuning a CPG to change its period. The CPG model is modified from the model described in Mori, Nakamura, Sato, and Ishii (2004). The CPG is symmetric, and the synapses are replaced by Dynamic Synapses (as shown in Fig. 7). The initial values of dynamic synaptic strengths were set to be the original synaptic strengths, and the initial amplitude of oscillation of synaptic strengths are scaled by an exponential function to be in the nearby order of magnitude of the original synaptic strengths.

$$w_{i_{cpg}} = w_{i_0}\beta^{w_i - 0.5} \tag{3}$$

where $w_{i_{CPG}}$ is CPG synapses' weights, w_{i_0} the ith initial synaptic weight of the CPG, β is a base of exponentiation that scales the weights. As the CPG is symmetric, in the model, the state of dynamic synapses of one neuron is a mirror of the other one. When the output of the CPG crosses zero, the error between the target period and the actual period is calculated, and the modulator is released at a speed that is proportional to the decline of the error compared with the previous error. If the error increased, no modulator is released:

$$\epsilon_i = \omega_i - \omega_{obj} \tag{4}$$

$$n_{\mathrm{m_i}} = \begin{cases} \mathrm{k}_{\mathrm{m}_2}(|\epsilon_{i-1}| - |\epsilon|) & \text{if } |\epsilon_{i-1}| - |\epsilon| > 0 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where ω_i is the period of the CPG from ith to $i + 1$th zero crossing, ω_{obj} the target period, ϵ_i is the error between them, n_{m_l} the amount of modulator released.

The CPG originally had a period of about 0.5 s. The target of training is to alter the period to be 2 s by tuning the synaptic strengths. The results are shown in Fig. 8. Using the same operant learning rule as before, the period of the CPG converges to the target period. The period of the output of CPG and the synaptic strength is nonlinear and dynamic synapses have no prior knowledge of the CPG, but the simple neural circuit still finds and learns the parameters of the target effectively. The experiment shows that the Dynamic Synapse can be applied to an RNN without requiring any specific analysis of the properties of the network.

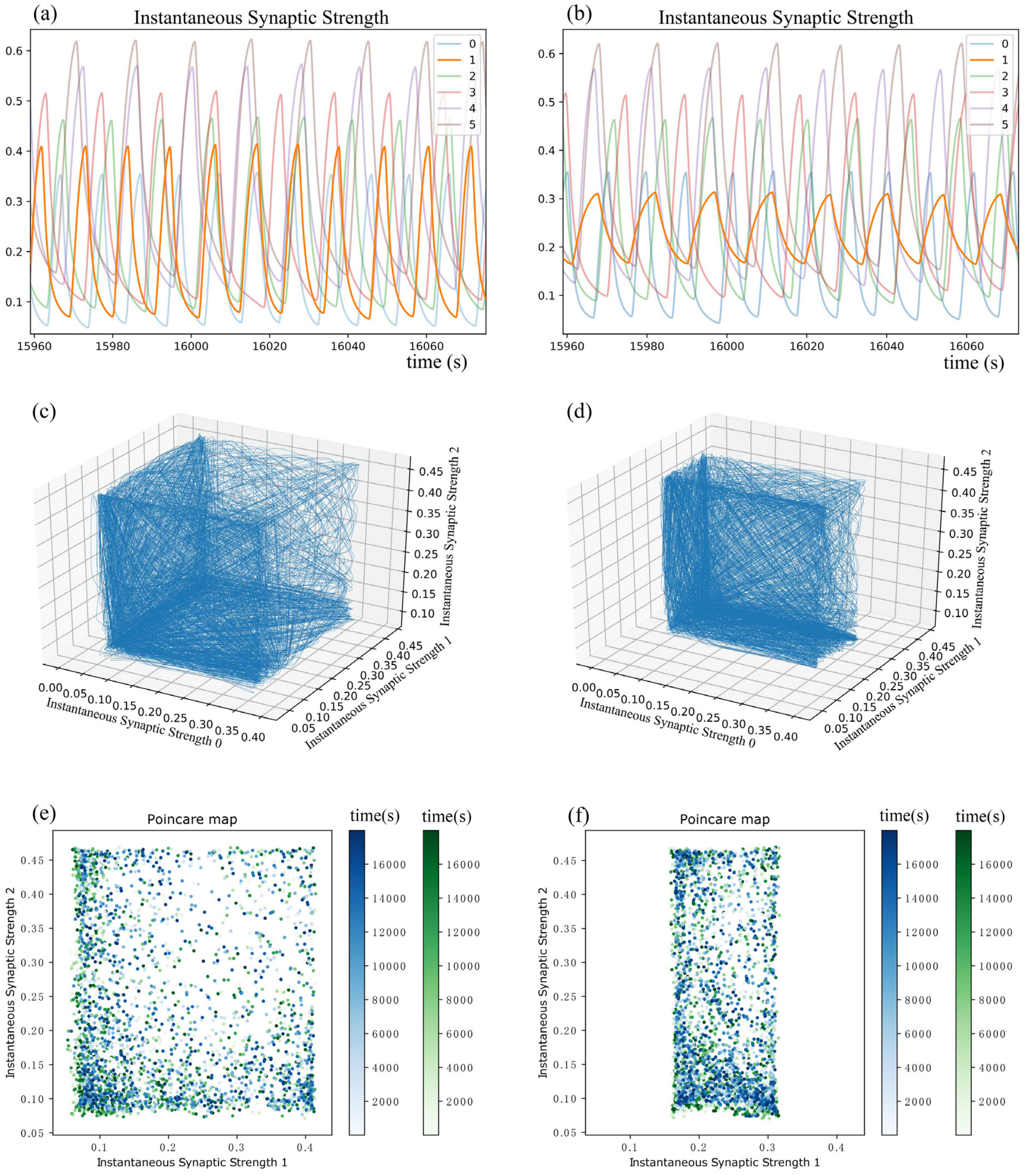


Fig. 4. Trajectories of synaptic strengths. (Left): all synapses have the same damping factors. (Right): synapse one has a higher damping factor than others. (a) & (b) show the change over time of the synaptic strengths (the proportional number of receptors in each synapse); (c) & (d) plot the trajectory formed by the first three synapses (for (d) the synapse on the X-axis has higher damping); (e) & (f) are Poincaré maps, i.e., sections of (c) and (d) when the instantaneous synaptic strength passes the plane defined by the centre of oscillation for one synapse (blue and green are for two different directions, and time of intersection is indicated by the intensity). It can be seen that synaptic strength oscillates chaotically and unpredictably, tracing out a search space. With higher damping factors, the amplitude of the oscillation for that synapse is decreased, reducing the search space. The periods of the oscillations can be different with different parameters.

2.4. Reinforcement learning in Puckworld

The Dynamic Synapse model was tested in a game named PuckWorld, available as part of the Python Learning Environment. The game has a planar environment with three agents (Fig. 9): a player that is controlled by a reinforcement learning algorithm, a reward source that changes its location after a specific period, and a punishment source that chases the player and decreases the reward if the player is within a specific range of the punishment source.

In the game, the player can move in 4 directions: left, right, down and up. The states of the player and the environment can be observed (Fig. 10). The states are the velocity of the player, the locations of the player, the position of the reward source and the position of the punishment source. The states are pre-processed then used as sensor input. In this instance, the sensory inputs are the velocity of the player, the distance to the reward source, and

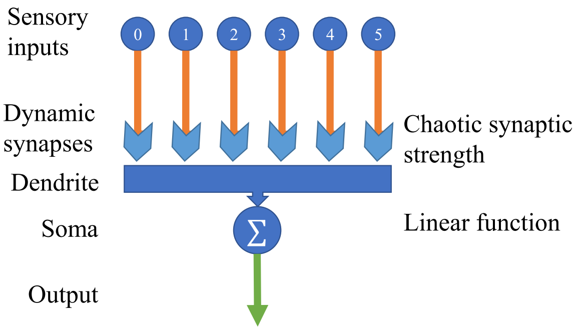


Fig. 5. A linear neuron with dynamic synapses and several constant inputs. Its output is the sum of the inputs, each weighted by the respective synaptic strength.

the shortest distance the player is from the edge of the range of the punishment source (the distance to escape). As the game codes the states using an absolute coordinate system, the player does not have orientation. To transform the potentially negative values and direction of distance information in absolute coordinates into positive sensor values, the player is assumed to have sensors in 4 directions that correspond to the positive and negative directions of the x- and y-axis of the coordinate system, and the sensor on the side of the agent information coming from is positive, while the other side is zero (Fig. 10). As the player has a symmetric structure, the neural circuits are designed in a symmetric structure: four integrate-and-fire motor neurons control the motion in the four directions, respectively. Each neuron gets three types of sensory inputs (as outlined above) in the four directions. Each sensory input feeds into the neuron through a dynamic synapse. Also because of the symmetry of the structures and motions, to simplify and accelerate the training, the dynamic synapses of each motor neuron from sensors in the same direction relative to that motor neuron are treated as the same (have the same dynamics and parameters during the learning).

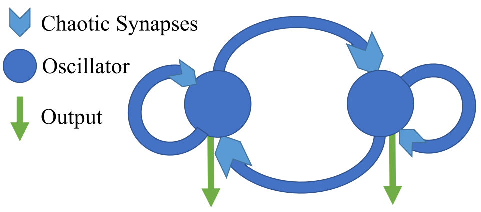


Fig. 7. A CPG with the learning rule. Two neurons with spontaneous firing inhibit each other's firing alternately. The simulation aims to tune the period of oscillation, using the same operant learning rule to alter the synaptic strengths.

The function of the motor neurons is:

$$\dot{v} = \sum_{i=1}^{n} w_i s_i \tag{6}$$

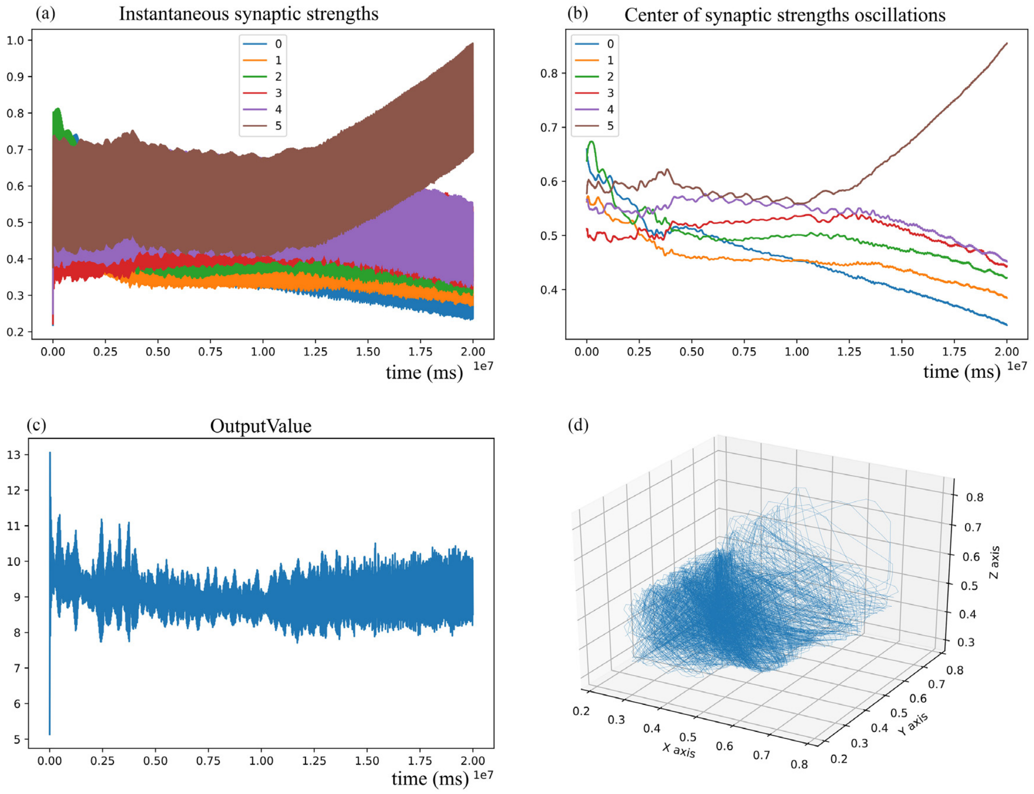


Fig. 6. Simulation results of the simple linear example. The value function determining modulator release is that the output is higher than a threshold and increasing. (a) The instantaneous synaptic strengths, the labels of lines show the input value of corresponding synapses (b) the central synaptic strengths (c) the output value of the neuron (d) trajectory of the first three synaptic strengths. Note that the statistical output value starts to increase after unstable initial fluctuation. At the end of the learning, the centre of the oscillation of the synaptic strength shifts so that the order of strengths is the same as the order of the input values, and the synaptic strength of the synapse with highest input value increased while the others declined, which is the most efficient way to get higher output with conservation of the total number of receptors.

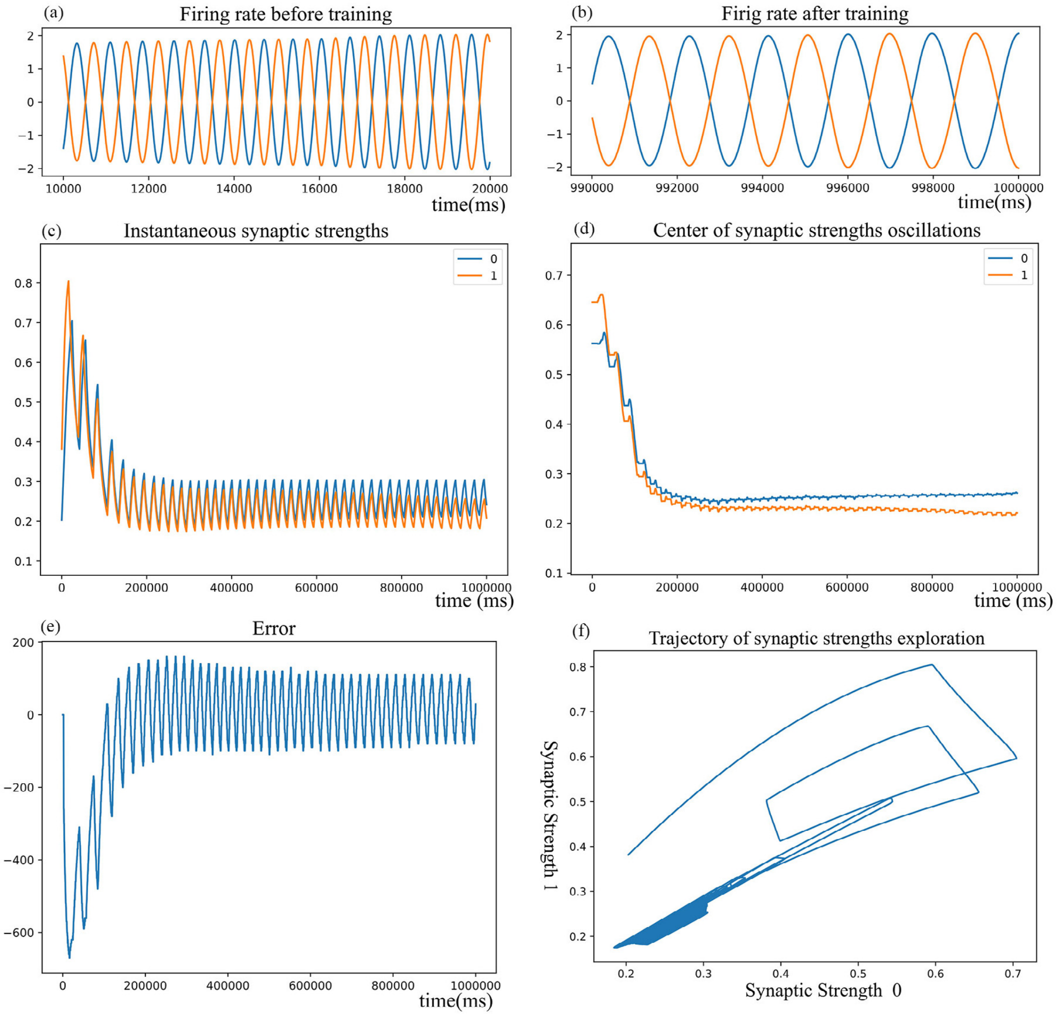


Fig. 8. Results of tuning CPG with Dynamic Synapse. (a) Before learning the period of oscillation is about 500 ms. (b) After learning the period of oscillation is about 2000 ms. (c) The instantaneous synaptic strengths before scaling by the exponential function. As the model is symmetric, the two neurons share same states of dynamic synapses. Hence, only two synapses are plotted. Same in (d) and (e). (d) The centre of synaptic strength oscillation before scaling by the exponential function. (e) The error between the period of the output of the CPG and the target period during simulation. (f) The trajectory of chaotic exploration of the synaptic strength, which converged on the bottom left.

$$\text{if } \mathrm{v} > \mathrm{v}_{\text{threshold}} \quad \mathrm{v} = \mathrm{v}_{\text{rest}} \tag{7}$$

where v is membrane potential, s_i the ith sensory input, v_{rest} the rest membrane potential and $v_{threshold}$ the threshold of firing.

The reward of the game is the weighted sum of the normalised distance to the reward source and the normalised distance into the range of the punishment source:

$$R = \begin{cases} -(d_r + 2d_e) & \text{if player is in punishment range} \\ -d_r & \text{otherwise} \end{cases} \tag{8}$$

where R is reward, d_r the distance between player and reward source, d_e the distance between player and the edge of punishment range.

The reward is fed into a firing rate neuron with an adaptive current, which releases the modulator. With the adaptive current, the neuron is sensitive to the change of the reward but insensitive to the value of the reward. The adaptation speed factor from low to high is higher than the adaption speed factor from high to low, thus the neuron has a trend to increase the expectation of the reward:

$$\dot{I}_{adapt} = \begin{cases} \left(k_r R + I_{adapt}\right) k_{adapt_1} & \text{if } R > I_{adapt} \\ \left(k_r R + I_{adapt}\right) k_{adapt_2} & \text{if } R < I_{adapt} \end{cases} \tag{9}$$

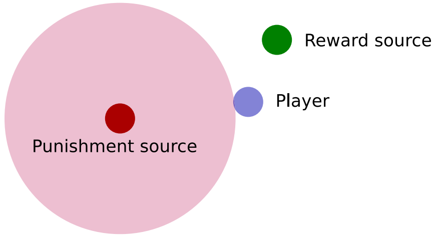


Fig. 9. The environment of PuckWorld. The green point is the reward source, the blue point is the player, the red point is the punishment source, and the dark magenta circle is the range the punishment source effects.

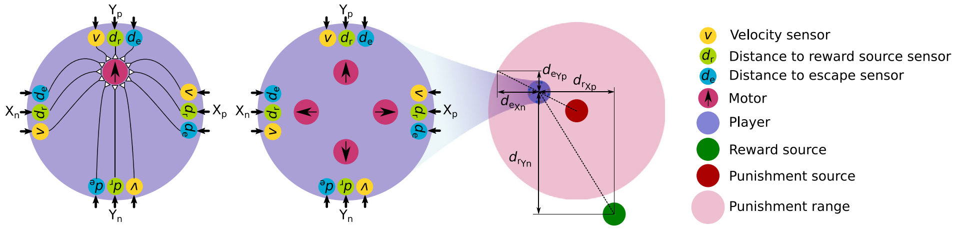


Fig. 10. Sensors and neural circuits model for PuckWorld. (a) Velocity (v) sensors, distance to reward source (d_r) sensors and distance to escape (d_e) sensors get input from four directions; a motor neuron gets all of the sensory inputs by Dynamic Synapses. (b) There are four sets of neural circuits in the player; because the neural circuits, agents and the environment are symmetric, all homologous synapses are assumed to share the same dynamics and synaptic strengths to accelerate the learning. (c) The sensors indicate distances by orthogonal decomposition; when a measured object is in the direction that can be projected to the positive direction of a sensor, the sensory value is positive, otherwise 0.

where I_{adapt} is the current intensity, k_R a factor from reward to current intensity, k_{adapt_1} and k_{adapt_2} are factors of adaption speed. Thus modulator amount n_m is given by:

$$n_m = 2/(1 + e^{-k_{mI}(k_R R - I_{adapt})}) - 1 \qquad (10)$$

where k_{mI} is a factor to map the current after adaption to an appropriate range.

As this is a single layer circuit, the ability of a player controlled by the circuit is simple and limited. Hence, we can analyse the possible best solution of the synaptic strengths and compare it with the solution obtained by operant training with dynamic synapses. Treating the single layer circuit as a linear function, the whole system can be interpreted as a second-order system. For an appropriate solution, the interactions of the elements in the system should work as though (1) there is an extension spring connecting the player and reward source; (2) the punishment range is an elastic ball that pushes the player away; and (3) the elastic coefficient of the elastic ball is higher than the elastic coefficient of the spring so the player will avoid punishment even when the reward is inside the punishment range. Because of (1), the synaptic strengths of positive y distance to reward input should be higher than the synaptic strengths of negative y distance to reward input; because of (2), the synaptic strengths of positive y distance to escape input should be higher than the synaptic strengths of negative y distance to escape input; and because of (3) the synaptic strengths of positive escape input should be higher than the synaptic strengths of positive reward input.

The simulation results are shown in Fig. 11. The simulation result was largely consistent with the analysis above, as shown in Fig. 11(a) and (c). However, surprisingly the highest synaptic strength is for negative x distance to reward input (line 4 in Fig. 11(a)) are higher than other lines, which means the agent would go forward when the reward source is on its left side. The positive y velocity (line 3) is also higher than negative y velocity (line 2), which means the agent tends to accelerate. These appear to be two strategies to avoid chasing by the punishment source.

In addition, Fig. 11(b) shows the exploration of 3 instantaneous synaptic strengths. Fig. 11(d) shows the damping factor of the oscillation of the instantaneous synaptic strengths. Fig. 11(e) is a Poincare map of the Dynamic synapse, i.e. the section of (b) when the instantaneous synaptic strengths 0 passed the centre of synaptic strength oscillation. It shows that the exploration is chaotic and unpredictable, and that the region of sampling shrinks during learning and the density of sampling increases during learning. (f) The line labelled Reward is the value R returned by the simulation environment by the reward function; The line labelled Filtered Reward is the low-pass-filtered R which shows the overall trend; the line labelled Reward Adaption is the adaption current I_{adapt}; the line labelled Reward after Adaption is the value of $k_R R - I_{adapt}$, which determines the modulator release and is more sensitive to variations of the reward than to the absolute value of the reward.

The source code for simulations of the model and experiments is available online https://github.com/InsectRobotics/DynamicSynapsePublic.

3. Discussion

We have proposed a model of operant learning based on continuous unpredictable synaptic strength fluctuations, with dynamics that are altered in response to a reinforcement signal. We illustrate the application of this principle to optimise the output, for given inputs, first in a simple linear neuron model, then to tune a recurrent CPG network to a target period, and finally to enable a spiking neural circuit embedded in an agent to improve performance in a continuous environment with dynamic reward and punishment.

An important property of our approach is that the source of variation that supports operant learning is continuous, unlike reinforcement learning algorithms that are based on random number generators, which have either discrete random outputs, or are partially predictable because of interpolation. By defining a system that has chaotic dynamics we can generate continuous motion without interpolation, so the unpredictability is continuous on any scale. An additional advantage over alternative synapse-level models for operant learning, such as the Hedonistic Synapse (Seung, 2003), is that the applications are not limited to a specific type of neural circuit or neural network. We have shown we can use our Dynamic synapse in both spiking and firing rate neural circuits, and the method can also be suitable for general online parameter optimisation, as it acts to scale the synaptic strength value to the suitable ranges. It can also be applied to discrete systems by adjusting the time step to an appropriate range or by sampling. We plan to further explore the application of this model to a range of problems in robot learning and reinforcement learning.

A key difference between our model and previous models is that our model learns in parameter space but not action space. Previous models usually alter the synaptic strength based on the pattern of synapse activities (i.e. those conveying signals that led to reward), but our model directly learns the synaptic strengths that led to reward. As the synapse dynamics reflect recent states of the synapse, exploring parameter space enables our model to solve the credit assignment problem without an eligibility trace, which is necessary for some previous models, such as extended STDP models by Gurney, Humphries, and Redgrave (2015) and Izhikevich (2007). As the time scale of synaptic strength fluctuations is longer than synapse activity dynamics, the model can function with temporally distant reward. Exploring parameter space means that the learning concerns the overall function instead of the specific outputs of the neural circuits, so our model allows remodelling of synaptic connections independently from action potentials of neurons, which is a potentially powerful tool for neural computation.

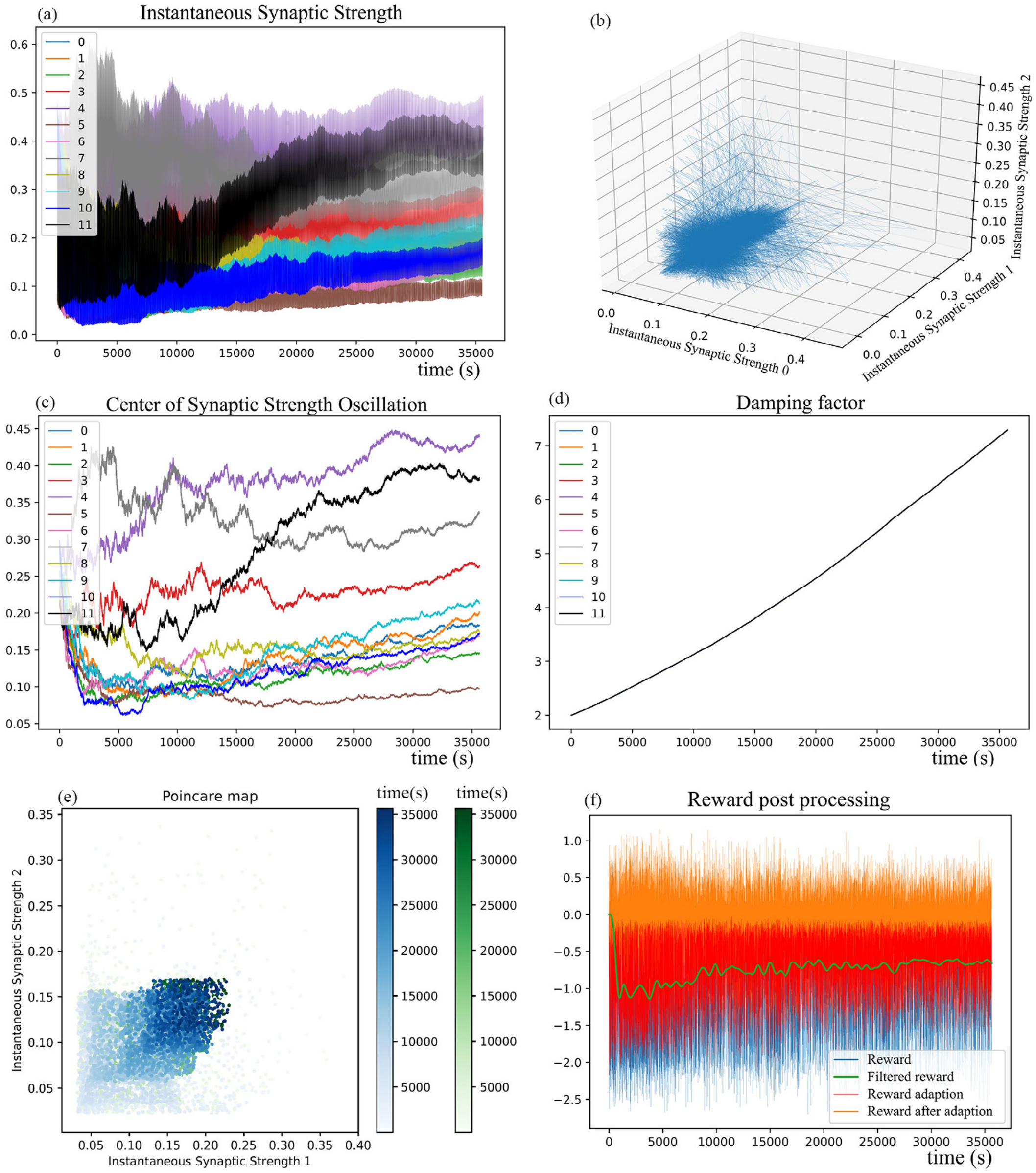


Fig. 11. The simulation results of Dynamic Synapse in PuckWorld. The relationships between the labelled number of synapses and the sensor a synapse connects to are: 0,1: x-velocity; 2,3 y-velocity; 4,5 d_r in x; 6,7 d_r in y; 8,9 d_e in x; 10,11 $d_e$in y; in each case odd numbers are the inputs in the positive direction as explained in the text. (a) Instantaneous synaptic strength of 12 synapses. (b) The trajectory of the first 3 synaptic weights; the explored range gradually converges. (c) The centres of synaptic strength oscillations; (d) The damping factors of instantaneous synaptic strength oscillation. All lines overlap. (e) A Poincaré map of Dynamic Synapse. It is a section of (b) when instantaneous synaptic strength passes its centre of oscillation. Each point is an intersection of the trajectory and the plane defined by the centre of oscillation. The blue and green points show the intersections from two different directions. The intensity of colour indicates the time of intersections. (f) shows the reward R, adaption current I_{adapt} and Reward after adaption.

We have proposed a possible grounding for the chaotic dynamics in the phenomena of receptor movement in dendritic spines. The model is inspired by recent evidence concerning the extent and mechanisms of these dynamics, but abstracted from the level of individual proteins to the level of the receptor flows between a dendrite and synapses as an integrated system. By focusing on postsynaptic receptor dynamics, our model can be related to synaptic mechanisms of short and long-term potentiation and depression (STP/LTP, STD/LTD). For example, the relations between STP and LTP as well as STD and LTD are similar to the relation in our model between the instantaneous synaptic strength and the centre of synaptic strength oscillation. The model can be expanded to explicitly explain some phenomena during STP, LTP, STD or LTD. For example, in STP–LTP model proposed in Xie et al. (1997), AMPA receptors are attracted towards the activated NMDA receptors when neurotransmitter is released, then a proportion of AMPA receptors diffuse again. This learning rule can be implemented by adding $k_{w1}n_T$ into the function describing the change of the amount of receptors in a synapse:

$$\dot{w}_i = \begin{cases} (v_i + k_{w1} n_T)\, c_d & \text{if } v_i > 0 \\ (v_i + k_{w1} n_T)\, \dfrac{w_i}{V_i} & \text{if } v_i < 0 \end{cases} \tag{11}$$

where n_T is amount of the synaptic transmitter, k_{w1} is a coefficient. In this extended model, when neurotransmitter is released, the instantaneous synaptic strength (the number of receptors) will tend to increase, resulting in STP. When the instantaneous synaptic strength is higher than the centre of the oscillation, if modulator is released, the capacity of the synapse to contain receptors will increase. Because of the oscillation of the amount of receptors in the synapse, some of the receptors diffuse again. Because the capacity is increased, more receptors are held in the synapse, resulting in LTP.

The model in this paper represents postsynaptic dynamics in a simplified form, at the statistical level of receptor trafficking, allowing it to emulate some features of receptor flow dynamics and synapse dynamics. Modelling individual receptors is out of the scope of this study because it would not be relevant at the level of learning. However, the mathematical functions for the receptor dynamics in our model are not exclusive. As long as the receptor dynamics have the features of chaotic oscillation, and the centre of oscillation is controllable by our learning rule, our learning rule could work for alternative formulations. The model could be extended to include more detail. For example, the receptor trafficking within the dendrite is assumed to be fast enough (compared to dendrite to synapse trafficking) to ignore its time constant. In reality, variations of AMPA receptor numbers on neighbouring dendrite spines are usually in the same direction (Zhang, Cudmore, Lin, Linden, & Huganir, 2015). This phenomenon could be modelled by taking account of the speed of receptor trafficking in the dendrite, which would have the consequence that neighbouring synapses would tend to have a similar concentration of receptors in the dendrite. Hence the receptor oscillation in neighbouring synapses would have a higher probability to be in similar phases than in distant synapses. Our model depends on several hypothetical assumptions, such as the form of the dynamics of receptor trafficking, dynamics of capacity to contain receptors, and the equilibrium point of receptor oscillation, which are not yet directly supportable from biological research. To understand the dynamics of receptor trafficking requires continuous observation of the collective motion of receptors and concentration change of receptors in dendrites and synapses on timescales from seconds to hours. Similarly, understanding the dynamics of capacity to contain receptors requires continuous observation of actin flow between synapses and dendrites, size change of synapses and size change of postsynaptic density on similar timescales. Both types of observations are difficult but becoming experimentally more plausible, e.g. approaches of video microscopy in Esteves da Silva et al. (2015) and Zhang et al. (2015) continuously recorded the motions of proteins that can be observed as a group enabling the concentrations and flows to be understood. Observation of the phase relations between the oscillation of the receptors or structural components would be helpful for validating our model. In our model, we assume that the instantaneous weight leads the change of equilibrium point of receptor oscillation when the modulator is present. This could be tested by transplanting receptors to or from a synapse and giving modulator treatment, then observing if the synapse size or postsynaptic density changes. Thus several predictions arise from our model which we hope may be tested in future experiments.

However, the key concept presented here is not crucially dependent on the details of receptor trafficking. Other models of chaotic neurons or neural circuits suggest chaos exists in the membrane potential, and alternative chaotic processes in an animal could also possibly contribute to the generation of actions and learning with the same desirable properties of continuous unpredictability. Rather, the key properties are that the learning mechanism is entirely local to the synapse, and does not require an explicit 'tag' for the Hebbian correlation of pre- and post-synaptic activity but rather allows this property to emerge from the behavioural or output consequences caused by the recent state of the circuit. That is, synapses that contribute to obtaining reward are strengthened; but this does not depend on the firing of either the pre- or post-synaptic neuron, except insofar as this is necessary to cause behavioural outputs that result in reward.

It is nevertheless interesting to consider a simple variation on the learning rule we have used to make synapses with active presynaptic neurons (neurons that have released neurotransmitter, indicating they have fired) learn actively (cf. Eqs. (1) and (24)):

$$\dot{w}_{ci} = \mathrm{k}_{w2}\,(w_i - w_{ci})\, n_M n_T \tag{12}$$

$$\dot{b} = \mathrm{k}_b b n_M n_T \tag{13}$$

where n_T is amount of the synaptic transmitter. With n_T, variation of synaptic strength of a synapse is proportional to the presynaptic neuron activity, which can help to improve the pertinence of learning to the inputs. For example, a neuron gets multiple inputs but only a small set of them is activated by a specific stimulus, and with this rule, the synaptic plasticity only applies between the neuron and these activated inputs. Note this is a 3-factor learning rule, depending on the correlation between the amount of the synaptic transmitter, the amount of modulator, and the difference between instantaneous synaptic strength and the centre of the oscillation. When the absolute value of the correlation is higher, the variation of the centre of the oscillation is more significant.

However, another possible learning rule could use the weighted average, rather than the product, of the synaptic transmitter and instantaneous synaptic strength:

$$\dot{w}_{ci} = \mathrm{k}_{w3}\,(q(\mathrm{k}_{w4} n_T - w_{ci} + \alpha) + (1-q)\,(w_i - w_{ci}))\, n_M \tag{14}$$

where k_{w4} is a coefficient to fit the amount of transmitter to synaptic strength, q a proportion representing the relative weighting of these two factors, and α a constant. Notably, this rule can potentially account for Pavlovian classical conditioning, where the stimulus and reinforcer (neuromodulator) are presented together irrespective of the output. When $q = 1$, the learning rule is Pavlovian learning; when $q = 0$, the learning rule is operant learning. When q is close to 1, the learning process might look like classical conditioning with noise. Thus, classical and operant learning may coexist in the same neuron and even in the same synapse.

4. Methods

4.1. Overview

We first present a verbal description of how our model represents the alteration of synaptic strength in terms of the dynamic movement of receptors, and then provide a precise mathematical formulation of the principle.

Two forms of receptor trafficking can move receptors between the synapses and the dendrite. Lateral diffusion creates a passive flow along a gradient from a high concentration region to lower concentration region. Endosomal trafficking acts as an active flow that can move receptors against the gradient. The active flow is formed by endosome transportation which carries numbers of receptors. Our model has a minimal form to capture the key phenomena. Endosomal trafficking is active transportation and is modelled with a positive feedback term which provides motive force, and

Table 1
Symbols in the equations.

Symbol	Explanation	Typical value
N	Number of synapses on a dendrite tree	An integer, > 3
$\mathrm{V_d}$	Capacity of a dendrite	$\mathrm{NV_s}$
$\mathrm{V_s}$	Average capacity of a dendrite per synapse	1
V_i	Capacity of the ith synapse	
$\mathrm{w_{total}}$	Total amount of receptors in the dendritic tree	
D_i	Occupation of a receptor in *i*th synapse	0 to 1
p	The constant coefficient for dimension conversion of the amount of receptors	
w_i	Instantaneous Synaptic strength of *i*th synapse	Usually from 0.01 to 1
w_{ci}	Balance point of *i*th synapse	Usually from 0.01 to 1
c_{d_i}	Concentration of the receptors in *i*th dendrite region	
$\frac{w_i}{V_i}$	Concentration of the receptors of the *i*th synapse	
v_i	Bidirectional movement rate from dendrite to synapse	
r	Movement rate inertia	3.5×10^6 to 2.5×10^7
a	The positive feedback coefficient of movement rate	170 to 850
b	The damping factor of movement rate	14 000 to 2.6×10^7
q_d	The coefficient from concentration difference between neighbouring dendrite regions to receptor diffusion flux	
n_{M}	Amount of the modulator	Usually from 0 to 1.5
k_w	A coefficient of balance point update speed	Usually from 0.0003 to 0.002
k_{wc}	A constant factor to compensate the bias	0.4
k_b	A coefficient of damping factor update speed	Usually from 10^{-7} to 10^{-8}

two negative feedback terms which limit the speed of transportation. The negative feedback are the receptor concentration gradient, which is proportional to the concentration difference between a synapse and dendrite, and 'friction' of endosome transportation, which is proportional to the endosome transportation speed. These properties together produce dynamic oscillation of the number of receptors in each synapse. Because of the concentration gradient, the equilibrium point of the dynamics of endosome transportation of a single synapse is when the concentration of receptor in the synapse is same as the concentration in the dendrite. It is also the equilibrium point of lateral diffusion. Note that because effects of receptor synthesis and degradation on receptor concentration are slower than receptor trafficking, they are assumed to have a negligible contribution to the dynamics. The proportion of receptors in endosomes is also ignored. Hence, in our model the total number of receptors in a dendritic tree is constant.

There are two factors in addition to receptor trafficking that could affect the concentration of receptors in each synapse: the size of the synapse and the number of receptors per unit area the synapse can accommodate. The size of the synapse is affected by the activity of actin. The number of receptors per unit area a synapse can accommodate is affected by scaffold–cytoskeleton complex. The two factors are not distinguished in the model but are jointly represented as the 'capacity' of the region to hold receptors. Thus, the equilibrium point of receptor motion can be altered by altering the capacity. The mechanism of learning in our model is to alter the capacity according to the following rule: whenever a neuromodulator signalling reinforcement is present, the instantaneous number of receptors in a synapse determines a change in its effective capacity, establishing a new equilibrium point nearer to that instantaneous value.

4.2. Mathematical model

When the number of receptors per synapse is sufficiently large, their dynamics can be modelled statistically using differential equations (Holcman & Triller, 2006), e.g. like gas, which consists of free-moving molecules and uncertain intermolecular distance. However, even for a smaller number of receptors per synapse, we note their contribution to synaptic strength can be proportional to their distance from the centre of the synaptic cleft, due to diffusion of neurotransmitter (Fig. 13). Thus, rather than explicitly represent discrete receptors and their positions, we represent the number of receptors in a synapse that currently contribute to its synaptic strength as a continuous 'amount'.

In the following equations, constants are represented by normal font and variables by italics (except v for membrane potential of integrate-and-fire neurons). The meanings of the symbols are shown in Table 1. The unit of time is millisecond.

The model assumes that the capacity of the dendrite to contain receptors is proportional to the number of synapses:

$$\mathrm{V_d} = \mathrm{N\,V_s} \tag{15}$$

where $\mathrm{V_d}$ is the capacity of a dendrite, N the number of synapses, and $\mathrm{V_s}$ a constant factor, which is the average capacity of a dendrite per synapse.

The concentration of receptors in the dendrite, c_d, is given by:

$$C_d = \mathrm{W_{total}} - \sum_{i=1}^{n} w_i/\mathrm{V_d} \tag{16}$$

where $\mathrm{w_{total}}$ is the (fixed) total amount of receptors in the dendrite tree; w_i is the amount of the receptors in the *i*th synapse; and $\mathrm{V_d}$ is the capacity of the dendrite.

We model the continuous flow of receptors between synapses and dendrite as a movement rate times the concentration of receptors on the source side:

$$\dot{w}_i = \begin{cases} v_i c_d & \text{if } v_i > 0 \\ v_i \dfrac{w_i}{V_i} & \text{if } v_i < 0 \end{cases} \tag{17}$$

where w_i is the amount of receptors of the *i*th synapse, w_i/V_i is concentration of receptors of the *i*th synapse, c_d the concentration of receptors in the dendrite, and v_i is the bidirectional movement rate, which is affected by lateral diffusion, endosomal trafficking and friction as described in the overview:

$$\dot{v}_i = 1/r \left(c_d - w_i/V_i + \mathrm{a}\,\mathrm{sign}(Vi) \times \sqrt[2]{|V_i|} - b v_i \right) \tag{18}$$

where v_i is bidirectional movement rate from dendrite to synapse (the direction from dendrite to synapse is positive); r is movement rate inertia, which represents factors (e.g. properties of actin) that drive receptors to keep their direction of flow; V_i is the capacity of ith synapse, which is affected by w_{ci}; $c_d - w_i/V_i$ is a term that represents the concentration difference between synapse and dendrite, which causes motion of receptors by diffusion; a $\mathrm{sign}(Vi) \times \sqrt[2]{|V_i|}$ is

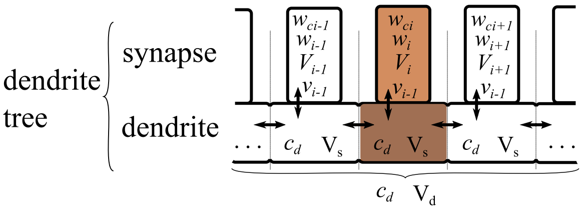

dendrite tree
synapse
dendrite
w_{ci-1} w_{i-1} V_{i-1} v_{i-1}
w_{ci} w_i V_i v_{i-1}
w_{ci+1} w_{i+1} V_{i+1} v_{i-1}
. . . c_d V_s c_d V_s c_d V_s . . .
c_d V_d

Fig. 12. Schematic Diagram and Symbols of Dynamic Synapse. A schematic diagram of the dendrite tree; the main variables and parameters of the model are indicated. For the meaning of the symbols, see Table 1.

positive feedback term of the movement, with positive feedback coefficient a; $-bv_i$ is a damping term with represents friction during the motion, with damping factor b.

As shown in Fig. 12, the receptors also move between neighbouring dendrite regions by diffusion:

$$\dot{c}_{d_i} = q_d \left(c_{d_{i-1}} + c_{d_{i+1}} - 2c_{d_{i+1}} \right) \tag{19}$$

where q_d is a coefficient from concentration difference to concentration variation rate. In practice, we found that when the number of synapses is less than 33, modelling this diffusive process has little effect. Hence, in the simulations in this paper, the diffusion is treated as instantaneous. For larger numbers of synapses, neglecting the dendritic diffusion can result in collapse of the chaotic dynamics, but these can be recovered if we run simulations with limited diffusion (results not included here).

As receptors diffuse in the dendrite tree, there is an equilibrium point when the concentration of receptors in a synapse and its neighbouring dendrite region are same. The equilibrium point forms the centre of synaptic strength oscillation, while the instantaneous synaptic strength oscillates around this point. We consider the effective strength of the synapse to be its equilibrium point, which can be established as follows. We assume that the receptors take a shorter time to diffuse between a synapse and its neighbouring region of the dendrite than to diffuse to regions in the neighbourhood of other synapses. Thus, in a short time interval, there is conservation of the amount of receptors in a synapse and its neighbourhood, and the equilibrium point is given by:

$$c_{ci}V_i / c_{ci}V_i + c_{ci}\mathrm{V}_s = w_{ci} / w_i + c_{d_i}\mathrm{V}_s \tag{20}$$

where c_{ci} is the equilibrium concentration of receptors in ith synapse, w_{ci} is the equilibrium amount of receptors in ith synapse, w_i is the instantaneous amount of receptors in ith synapse, V_i is capacity of the ith synapse, c_{d_i} is concentration of the receptors in ith dendrite region and V_s is average dendrite capacity per synapse.

To set or alter the strength of a synapse, we alter w_{ci}. Solving the above equation for V_i, we get:

$$V_i = \mathrm{V}_s w_{ci} / c_d \mathrm{V}_s + w_i - w_{ci} \tag{21}$$

By updating V_i according to this function, the amount of receptors will converge to the given equilibrium value. Thus, we can define (or alter) the centre of synaptic strength oscillation. We can also alter the amplitude of oscillation around this centre by changing the damping factor b in (18).

These equations describe a system which contains multiple coupled second-order systems. A second-order system, such as a spring–mass–damper system, usually has the property of oscillation. When coupled together, they usually end in phase-locked oscillations, which means they have a fixed trajectory of oscillation. However, when the second-order systems include appropriate nonlinear functions, the system oscillates chaotically. In the model, the receptor trafficking between a synapse and dendrite is a second-order system. Multiple synapses are coupled by a dendrite, and updating of V_i is a nonlinear function. As we illustrate, the resulting oscillation appears to be chaotic. Because chaotic motion has a very complex, unpredictable and ergodic solution, the chaotic changes in synaptic strength can explore an output space for a neuron or neural circuit. Simulations are shown in the Results section.

As described in the Results section, a simple learning rule for this system is:

$$\dot{w}_{ci} = \mathrm{k}_w (w_i - w_{ci}) n_{\mathrm{M}} \tag{22}$$

where n_M is amount of a neuromodulator that represents reward, and k_w is a coefficient controlling the learning rate. In practice we need to slightly modify this rule to compensate for a biased drift in synaptic strength. If, during an oscillation period, the integrated values of the differences between instantaneous synaptic strength and the centre of oscillation on each side are not equal (as shown in Fig. 14, the sizes of adjacent yellow and blue coloured areas), uncorrelated modulator release (e.g. the release experienced by a synapse that is not making any useful contribution to satisfying the value function) can cause the centre of oscillation to become biased during long training times. During learning, if the centre of oscillation changes in a small range, the rate of bias can be approximated as a constant. To compensate it, a learning rule with compensation can be applied:

$$\dot{w}_{ci} = \begin{cases} \mathrm{k}_w \, (w_i - w_{ci}) \, n_M \, (1 + \mathrm{k}_{wc}) & \text{if } w_i > w_{ci} \\ \mathrm{k}_w \, (w_i - w_{ci}) \, n_M & \text{else} \end{cases} \tag{23}$$

where k_{wc} is a constant factor to compensate the bias. However, if the centre of oscillation changes in a larger range, the bias is variable, and cannot be compensated using the above rule. In our model, this bias is towards positive values for a centre of oscillation above 0.5, and negative values below 0.5. As a consequence there can be a positive feedback effect that accelerates learning.

To allow learning to converge, the learning rule should also reduce the oscillation amplitude. When the modulator is present, damping factors also increase:

$$\dot{b} = \mathrm{k}_b b n_M \tag{24}$$

where b is the damping factors, k_b a coefficient.

Acknowledgements

This work was supported by FP7 FET-Open project "Minimal". We thank Matthieu Louis for discussions of earlier versions of this work.

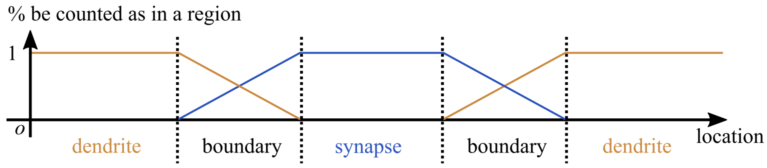


Fig. 13. Justification for a continuous representation of the effects of receptor location between dendrite and synapse. The boundary between a synapse and dendrite can be considered wide and smooth, and as a receptor approaches the synapse, it can receive more neurotransmitters and contribute more to the synaptic strength. Rather than model the boundary area explicitly, we associate synaptic strength with the 'amount' of receptors a synapse contains, treated as a continuous variable.

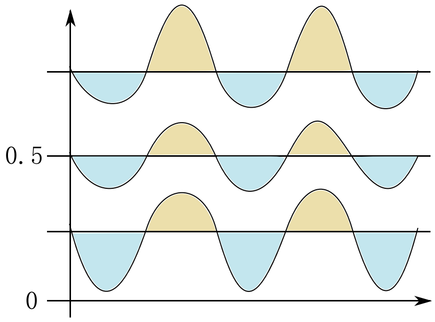


Fig. 14. The bias of oscillation at different centre of oscillation. The curves are instantaneous synaptic strength, which oscillate around centres of synaptic strength oscillation (shown as straight lines). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

References

Allam, S. L., Bouteiller, J. M. C., Hu, E. Y., Ambert, N., Greget, R., Bischoff, S., et al. (2015). Synaptic efficacy as a function of ionotropic receptor distribution: A computational study. *PLoS One*, *10*(10), 1–20.

Allison, D. W., Gelfand, V. I., Spector, I., & Craig, A. M. (1998). Role of actin in anchoring postsynaptic receptors in cultured hippocampal neurons: Differential attachment of NMDA versus AMPA receptors. *Journal of Neuroscience*, *18*(7), 2423–2436. URL http://www.jneurosci.org/content/18/7/2423.short.

Angulo-Garcia, D., & Torcini, A. (2014). Stable chaos in fluctuation driven neural circuits. *Chaos, Solitons & Fractals*, *69*, 233–245. URL http://dx.doi.org/10.1016/j.chaos.2014.10.009.

Brembs, B. (2003). Operant conditioning in invertebrates. *Current Opinion in Neurobiology*, *13*(6), 710–717.

Canavier, C. C., Clark, J. W., & Byrne, J. H. (1990). Routes to chaos in a model of a bursting neuron. *Biophysical Journal*, *57*(6), 1245–1251. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1280834/.

Cash, D., & Carew, T. J. (1989). A quantitative analysis of the development of the central nervous system in juvenileAplysia californica. *Journal of Neurobiology*, *20*(1), 25–47. URL http://doi.wiley.com/10.1002/neu.480200104.

Cavalieri, L., & Koçak, H. (1994). Chaos in biological systems. *Journal Theorical Biology*, *169*(1985), 179–187.

Choquet, D., & Triller, A. (2013). The dynamic synapse. *Neuron*, *80*(3), 691–703. URL http://dx.doi.org/10.1016/j.neuron.2013.10.013.

Cingolani, L. a., & Goda, Y. (2008). Actin in action: the interplay between the actin cytoskeleton and synaptic efficacy. *Nature Reviews Neuroscience*, *9*(5), 344–356. URL http://www.nature.com/doifinder/10.1038/nrn2373.

Eckmann, J. P., & Ruelle, D. (1985). Ergodic theory of chaos and strange attractors. *Reviews of Modern Physics*, *57*(3), 617–656.

Esteves da Silva, M., Adrian, M., Schätzle, P., Lipka, J., Watanabe, T., Cho, S., et al. (2015). Positioning of AMPA receptor-containing endosomes regulates synapse architecture. *Cell Reports*, *13*(5), 933–943.

Frémaux, N., Sprekeler, H., & Gerstner, W. (2010). Functional requirements for reward-modulated spike-timing-dependent plasticity. *Journal of Neuroscience*, *30*(40), 13326–13337. URL http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.6249-09.2010.

Gurney, K. N., Humphries, M. D., & Redgrave, P. (2015). A new framework for Cortico-Striatal plasticity: behavioural theory meets in vitro data at the reinforcement-action interface. *PLoS Biology*, *13*(1), e1002034. URL http://dx.plos.org/10.1371/journal.pbio.1002034.

Haselwandter, C. A., Calamai, M., Kardar, M., Triller, A., & Azeredo Da Silveira, R. (2011). Formation and stability of synaptic receptor domains. *Physical Review Letters*, *106*(23), 1–4.

Hausrat, T. J., Muhia, M., Gerrow, K., Thomas, P., Hirdes, W., Tsukita, S., et al. (2015). Radixin regulates synaptic GABAA receptor density and is essential for reversal learning and short-term memory. *Nature Communications*, *6*, 6872. URL http://www.nature.com/doifinder/10.1038/ncomms7872.

Hayashi, H., Nakao, M., & Hirakawa, K. (1983). Entrained, harmonic, quasiperiodic and chaotic responses of the self-sustained oscillation of Nitella to Sinusoidal stimulation. *Journal of the Physical Society of Japan*, *52*(1), 344–351.

Holcman, D., & Triller, A. (2006). Modeling synaptic dynamics driven by receptor lateral diffusion. *Biophysical Journal*, *91*(7), 2405–2415. URL http://linkinghub.elsevier.com/retrieve/pii/S0006349506719567.

Honkura, N., Matsuzaki, M., Noguchi, J., Ellis-Davies, G. C., & Kasai, H. (2008). The subspine organization of actin fibers regulates the structure and plasticity of Dendritic spines. *Neuron*, *57*(5), 719–729.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, *21*(4), 642–653.

Inukai, H., Minami, M., & Yanou, A. (2015). Generating chaos with neural-network-differential-equation for intelligent fish-catching robot. In *2015 10th Asian control conference: Emerging control techniques for a sustainable World, ASCC 2015*.

Isaac, J. T., Nicoll, R. A., & Malenka, R. C. (1995). Evidence for silent synapses: Implications for the expression of LTP. *Neuron*, *15*(2), 427–434.

Izhikevich, E. M. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral Cortex*, *17*(10), 2443–2452. URL https://link.springer.com/article/10.1186/1471-2202-8-S2-S15.

Jaqaman, K., Kuwata, H., Touret, N., Collins, R., Trimble, W. S., Danuser, G., et al. (2011). Cytoskeletal control of CD36 diffusion promotes its receptor and signaling function. *Cell*, *146*(4), 593–606. URL http://dx.doi.org/10.1016/j.cell.2011.06.049.

Jensen, G. D. (1963). Preference for bar pressing over "freeloading" as a function of number of rewarded presses. *Journal of Experimental Psychology*, *65*(5), 451–454. URL http://content.apa.org/journals/xge/65/5/451.

Kauer, J. A., Malenka, R. C., & Nicoll, R. A. (1988). A persistent postsynaptic modification mediates long-term potentiation in the hippocampus. *Neuron*, *1*(10), 911–917.

Koskinen, M., & Hotulainen, P. (2014). Measuring F-actin properties in dendritic spines. *Frontiers in Neuroanatomy*, *8*(August), 1–14. URL https://doi.org/10.3389/fnana.2014.00074.

Lau, C. G., & Zukin, R. S. (2007). NMDA receptor trafficking in synaptic plasticity and neuropsychiatric disorders. *Nature Reviews Neuroscience*, *8*(6), 413–426.

Mori, T., Nakamura, Y., Sato, M.-a., & Ishii, S. (2004). Reinforcement learning for a CPG-driven biped robot. In *Aaai 2004* (pp. 623–630).

Nargeot, R., & Simmers, J. (2011). Neural mechanisms of operant conditioning and learning-induced behavioral plasticity in Aplysia. *Cellular and Molecular Life Sciences*, *68*(5), 803–816.

Nobukawa, S., Nishimura, H., Yamanishi, T., & Liu, J. Q. (2014). Analysis of routes to chaos in Izhikevich neuron model with resetting process. In *2014 joint 7th international conference on soft computing and intelligent systems, SCIS 2014 and 15th international symposium on advanced intelligent systems, ISIS 2014* (pp. 813–818).

Petrini, E. M., Lu, J., Cognet, L., Lounis, B., Ehlers, M. D., & Choquet, D. (2009). Endocytic trafficking and recycling maintain a pool of mobile surface AMPA receptors required for synaptic potentiation. *Neuron*, *63*(1), 92–105. URL https://dx.doi.org/10.1016%2Fj.neuron.2009.05.025.

Roth, R. H., Zhang, Y., & Huganir, R. L. (2017). Dynamic imaging of AMPA receptor trafficking in vitro and in vivo. *Current Opinion in Neurobiology*, *45*, 51–58. URL http://dx.doi.org/10.1016/j.conb.2017.03.008.

Sekimoto, K., & Triller, A. (2009). Compatibility between itinerant synaptic receptors and stable postsynaptic structure. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, *79*(3), 1–13.

Sergé, A., Fourgeaud, L., Hémar, A., & Choquet, D. (2003). Active surface transport of metabotropic glutamate receptors through binding to microtubules and actin flow. *Journal of Cell Science*, *116*(Pt 24), 5015–5022.

Seung, S. (2003). Learning in spiking neural networks by reinforcement of stochastics transmission. *Neuron*, *40*, 1063–1073. URL https://doi.org/10.1016/S0896-6273(03)00761-X.

Sheng, M., & Hoogenraad, C. C. (2007). The postsynaptic architecture of excitatory synapses: a more quantitative view. *Annual Review of Biochemistry*, *76*, 823–847.

Shepherd, J. D., & Huganir, R. L. (2007). The cell biology of synaptic plasticity: AMPA receptor trafficking. *Annual Review of Cell and Developmental Biology*, *23*(1), 613–643. URL http://www.annualreviews.org/doi/10.1146/annurev.cellbio.23.090506.123516.

Shouval, H. Z., Castellani, G. C., Blais, B. S., Yeung, L. C., & Cooper, L. N. (2002). Converging evidence for a simplified biophysical model of synaptic plasticity. *Biological Cybernetics*, *87*(5–6), 383–391.

Steingrube, S., Timme, M., Woergoetter, F., & Manoonpong, P. (2011). Self-organized adaptation of a simple neural circuit enables complex robot behaviour. *Nature Physics*, *6*(3), 16. URL https://www.nature.com/articles/nphys1508.

Storace, M., Linaro, D., & De Lange, E. (2008). The Hindmarsh-Rose neuron model: Bifurcation analysis and piecewise-linear approximations. *Chaos*, *18*(3), 1–11.

Sun, X., Milovanovic, M., Zhao, Y., & Wolf, M. E. (2008). Acute and chronic dopamine receptor stimulation modulates AMPA receptor trafficking in nucleus Accumbens neurons cocultured with prefrontal cortex neurons. *Journal of Neuroscience*, *28*(16), 4216–4230. URL http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.0258-08.2008.

Sussillo, D. (2014). Neural circuits as computational dynamical systems. *Current Opinion in Neurobiology*, *25*, 156–163. URL http://dx.doi.org/10.1016/j.conb.2014.01.008.

Tél, T., Gruiz, M., & Kulacsy, K. (2006). *Chaotic dynamics : an introduction based on classical mechanics*. (p. 393). Cambridge University Press.

Triller, A., & Choquet, D. (2005). Surface trafficking of receptors between synaptic and extrasynaptic membranes: And yet they do move!. *Trends in Neurosciences*, *28*(3), 133–139.

Wolf, R., & Heisenberg, M. (1991). Basic organization of operant behavior as revealed in Drosophila flight orientation. *Journal of Comparative Physiology A*, *169*(6), 699–705.

Xia, Z., Deng, H., Zhang, X., Weng, S., Gan, Y., & Xiong, J. (2017). A central pattern generator approach to footstep transition for biped navigation. *International Journal of Advanced Robotic Systems*, *14*(1), 1–9.

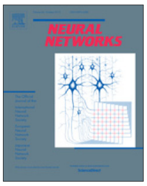
Xie, X., Liaw, J. S., Baudry, M., & Berger, T. W. (1997). Novel expression mechanism for synaptic potentiation: alignment of presynaptic release site and postsynaptic receptor. *Proceedings of the National Academy of Sciences of the United States of America*, *94*(June), 6983–6988.

Zhang, Y., Cudmore, R. H., Lin, D. T., Linden, D. J., & Huganir, R. L. (2015). Visualization of NMDA receptordependent AMPA receptor synaptic plasticity in vivo. *Nature Neuroscience*, *18*(3) URL http://www.nature.com/doifinder/10.1038/nn.3936.

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Corrigendum

Corrigendum to "A model of operant learning based on chaotically varying synaptic strength" [Neural Netw. 108 (2018) 114–127]

Tianqi Wei [a,b,*,1], Barbara Webb [b]

[a] School of Informatics, University of Edinburgh, 10 Crichton Street, Edinburgh, EH8 9AB, United Kingdom
[b] School of Engineering, University of Edinburgh, King's Buildings, Alexander Crum Brown Road, Edinburgh, EH9 3FF, United Kingdom

ARTICLE INFO

Article history:
Available online xxxx

The authors regret that there are typos in the equations 9, 16, 18, 19, 20 and 21. The experiments and results are not affected.

In equation 9, the plus signs should be minuses.

Equation 9 reads as:

$$\dot{I}_{adapt} = \begin{cases} \left(k_R R + I_{adapt}\right) k_{adapt_1} & \text{if } R > I_{adapt} \\ \left(k_R R + I_{adapt}\right) k_{adapt_2} & \text{if } R < I_{adapt} \end{cases}$$

It should be:

$$\dot{I}_{adapt} = \begin{cases} \left(k_R R - I_{adapt}\right) k_{adapt_1} & \text{if } R > I_{adapt} \\ \left(k_R R - I_{adapt}\right) k_{adapt_2} & \text{if } R < I_{adapt} \end{cases}$$

In equations 16, 20 and 21, the fraction signs are inline while numerators and denominators are not bracketed.

Equation 16 reads as:

$$c_d = \mathrm{w}_{\mathrm{total}} - \sum_{i=1}^{\mathrm{n}} w_i / \mathrm{V_d}$$

It should be:

$$c_d = \frac{\mathrm{w}_{\mathrm{total}} - \sum_{i=1}^{\mathrm{n}} w_i}{\mathrm{V_d}}$$

Equation 20 reads as:

$$c_{ci} V_i / c_{ci} V_i + c_{ci} \mathrm{V_s} = w_{ci} / w_i + c_d \mathrm{V_s}$$

It should be:

$$\frac{c_{ci} V_i}{c_{ci} V_i + c_{ci} \mathrm{V_s}} = \frac{w_{ci}}{w_i + c_d \mathrm{V_s}}$$

Equation 21 reads as:

$$V_i = \mathrm{V_s} w_{ci} / c_d \mathrm{V_s} + w_i - w_{ci}$$

It should be:

$$V_i = \frac{\mathrm{V_s} w_{ci}}{c_d \mathrm{V_s} + w_i - w_{ci}}$$

In equation 18, the second and third V_i should be v_i.

Equation 18 reads as:

$$\dot{v}_i \tau = \frac{1}{\mathrm{r}} \left(c_d - \frac{w_i}{V_i} + \mathrm{a}\ \mathrm{sign}\,(v_i) \times \sqrt[2]{|V_i|} - bV_i \right)$$

The correct equation 18 with a better format should be:

$$\dot{v}_i \tau = \frac{1}{\mathrm{r}} \left(c_d - \frac{w_i}{V_i} + \ \mathrm{sign}\,(v_i)\ \mathrm{a} \sqrt[2]{|v_i|} - bv_i \right)$$

In equation 19, the second $c_{d_{i+1}}$ should be c_{d_i}.

Equation 19 reads as:

$$\dot{c}_{d_i} = q_d \left(c_{d_{i-1}} + c_{d_{i+1}} - 2c_{d_{i+1}} \right)$$

It should be:

$$\dot{c}_{d_i} = q_d \left(c_{d_{i-1}} + c_{d_{i+1}} - 2c_{d_i} \right)$$

The authors would like to apologise for any inconvenience caused.

DOI of original article: https://doi.org/10.1016/j.neunet.2018.08.006.

[*] Correspondence to: Department of Computer Science, The University of Sheffield, 211 Portobello, Sheffield S1 4DP, United Kingdom.

E-mail addresses: Tianqi-Wei@outlook.com, chitianqilin@163.com (T. Wei).

[1] Ph.D. candidate at School of Informatics, University of Edinburgh, research associate at Department of Computer Science, University of Sheffield.

https://doi.org/10.1016/j.neunet.2019.02.003

Please cite this article as: T. Wei and B. Webb, Corrigendum to "A model of operant learning based on chaotically varying synaptic strength" [Neural Netw. 108 (2018) 114–127]. Neural Networks (2019), https://doi.org/10.1016/j.neunet.2019.02.003.

2.3 Discussion

In the paper, a synaptic plasticity model based on potential chaotic dynamics in the dendritic tree is proposed. The chaotic dynamics causes the chaotic synaptic strength exploration, searching the parameter space of synaptic strength for operant learning of a neural circuit. In the tests, the model and the learning rule based on the model trained three different types of neural networks in three different tasks, and the chaotic explorations converge to the points in the parameter space with the higher possibility to obtain rewards. The training of neural networks/circuits with large scale is yet to be tested, in which case the “curse of dimensionality” could impact the learning efficiency. However, as the learning rule is a local learning rule, a neural network/circuit with large scale can be divided into smaller networks/circuits to attenuate the impact of dimensionality.

2.3.1 Chaotic exploration and stochastic exploration

As detailed in the above paper, chaos emerges from the “dynamic synapse” model and contributes to the exploration of synaptic strength, which is a type of parameter exploration for the neural circuits. It is different from the reinforcement learning models in computer science, which uses stochastic processes for action explorations. Chaotic exploration and stochastic exploration have their pros and cons.

Chaos can emerge from a smooth non-linear process. Hence the exploration based on the chaos can be perfectly smooth. The smoothness is conducive to real-world tasks, such as reinforcement learning with a physical robot. The exploration based on random number generators (RNGs), however, introduces stepped number sequences which affect the smoothness, although there are some tricks to improve the smoothness, such as integrating the sequences.

Chaos can have arbitrarily small steps without changing the dynamics of generated sequence with respect to time, while an RNG with smaller step moves the spectrum of the generated sequence to higher frequency respect to time. Hence, the sample sequence of chaotic exploration can conveniently adjust for matching the sample sequences in computation and control.

Chaotic exploration needs stricter conditions for exploration than RNGs. The trajectory of a chaotic exploration evolves toward the corresponding strange attractor, which could be with limited range and limited volume, even zero volume. The chaotic exploration should be well conditioned so the attractor is (1) in an adequate range, which covers the maximum and minimum, (2) with reasonable distribution, which should not introduce unnecessary bias, and (3) with sufficient decency sampling, by which the adjacent samples result in undistinguishable behaviours. In practice, a chaotic exploration is usually hard to tune, and the difficulty varies with different dynamics. Without the limitation of biological plausibility, a chaotic exploration process is ideal to be built based on chaotic dynamics with a wide window of parameters that chaos can exist.

In existing practices, RNGs are widely used, because of their accessibility and adjustability. With APIs provided in various programming libraries, random numbers with user-specified distribution can be generated. The random numbers can also be easily adjusted by passing them to mathematical functions. However, there are no libraries with similar functions for generating chaos, so it is not straight forward in applications. In fact, the mainstream RNGs are pseudo random number generators. They are functions either with very long periods or with chaotic processes. In the latter case, it supports that chaos can be used in exploration. If a well-conditioned chaotic process can be implemented and packaged in a library, chaotic exploration can be more convenient in practice.

2.3.2 Comparison with Rescorla-Wagner model

The mathematical expression of the dynamic synapse learning rule (DSL) proposed in this chapter shares some similarities with the mathematical expression of the Rescorla-Wagner model (RWM). RWM is a theory for explaining a variety of phenomena involving associative learning (Rescorla et al., 1972). The mathematical expression of RWM is as follows (please note the notation in this subsection is not same with the notation in the rest of this thesis):

When AX, which is a compound of conditional stimulations (CS), A and X, is followed by an unconditional stimulation (US), such as US_1, the changes in associative strength of the respective components may be represented as:

$$\Delta V_A = \alpha_A \beta_1 (\lambda_1 - V_{AX}) \tag{2.1}$$

and

$$\Delta V_X = \alpha_X \beta_1 (\lambda_1 - V_{AX}) \tag{2.2}$$

where ΔV_X is the change in the strength of the association between CS X and the US_1, α_A the salience of X, β_1 the learning rate parameter for the US_1, λ_1 the maximum conditioning possible for the US_1, V_{AX} the associative strength of the compound A and X.

If AX is followed by a differently valued US, such as US_2, the changes in associative strength of the respective components may be represented as:

$$\Delta V_A = \alpha_A \beta_2 (\lambda_2 - V_{AX}) \tag{2.3}$$

and

$$\Delta V_X = \alpha_X \beta_2 (\lambda_2 - V_{AX}) \tag{2.4}$$

The similarities are that, (a) the update of strength is the product of two scalars and a result of the subtraction, (b) one of the scalars is the factor for learning rate, (c) the other scalar is the strength of the US, (d) the overall weights are bounded.

However, DSL and RWM are very different in various aspects.

The target for the weights (synaptic strength) to approximate is different. The first term in the subtraction of DSL is the instantaneous strength of a synapse. Thus the second term, the balance point of a single synaptic strength, approaches the instantaneous strength with the presence of rewards (US). While the first term in the RWM is the maximum conditioning possible for the US, shared by all of the CSs. Thus the second term, the total associate strength of all CSs, approaches the maximum possible conditioning.

The overall weights (synaptic strength) are bounded with a different mechanism. In the DSL, the synaptic strength of all the synapses on the same dendritic tree is limited by the amount of synaptic receptors; while in the RWM, the overall weights associating with the US, which are not on the same dendrite, are controlled by the parameter λ. Hence, for DSL, there is competition between synapses in the same dendrite; for RWM, there is competition between different neurons.

The two models are for different types of conditioning. The DSL is for operant learning, while RWM is for classical conditioning.

2.3.3 Chaotic pattern generator and chaotic function generator

Reinforcement learning with chaos can be classified into two types according to the role of chaotic sources. The first type is the chaotic pattern generator, which utilises chaos for exploring actions and a learner for associative learning of the emerged actions and sensory inputs. The second type is the chaotic function generator, which utilises chaos to explore functions from sensory inputs to actions directly.

2.3.3.1 Chaotic pattern generator

Some existing works of robot learning that are based on neural networks utilise chaos for robotic motion exploration by using chaotic pattern generators (ChaoticPG) as the chaos sources. ChaoticPGs are similar to central pattern generators (CentralPG), but their outputs can be chaotic with some specific parameters. This approach is straightforward in applications, as it can replace random number generators in previous reinforcement learning approaches. For example, as reviewed in the paper above, Steingrube et al. (2011) utilise neuron networks with ChaoticPG to generate new motions for a robot to escape from a hole which trapped one of its legs. The ChaoticPG can switch between chaotic state and several periodic states. However, the ChaoticPG are usually not controllable or tunable during learning. The learning happens in neural networks that can work independently from the ChaoticPG. General functions of this type of networks can be:

$$\mathbb{X}_G = G(t) \tag{2.5}$$

$$\mathbb{X}_\pi = \pi(s;\theta) \tag{2.6}$$

$$\mathbb{C} = \begin{cases} B(\mathbb{X}_G) & \text{if exploring} \\ B(\mathbb{X}_\pi) & \text{if not exploring} \end{cases} \tag{2.7}$$

$$\mathbb{M} = H(\mathbb{C}) \tag{2.8}$$

$$r = V(\mathbb{M}, s) \tag{2.9}$$

$$\theta \leftarrow \theta + A(\theta, \mathbb{X}_G, \mathbb{X}_\pi, s, r) \tag{2.10}$$

Where G is a chaotic pattern generator, t the time, $\mathbb{X}_G$ the output of G, π the sensory processing function, s the sensory input, θ the parameters of π, $\mathbb{X}_\pi$ the output of π, B the motor controller, $\mathbb{C}$ the motor control signal, H the physical system dynamics, $\mathbb{M}$ the motion of robot or animal, V the function to calculate rewards, r the rewards of decisions or motions, A the function to update parameters θ.

2.3.3.2 Chaotic function generator

A Chaotic Function Generator (ChaoticFG) combines the chaotic process and learning process in the same neural network. In this case, the chaotic process provides chaotic parameters to the neural network, thus the neural network explores the parameter space. The learning system generates new functions and tests these functions continuously and chaotically. The learning rule proposed in this chapter belongs to this category. General functions of this type of networks can be:

$$\mathbb{X}_G = G(t;\theta_G) \tag{2.11}$$

$$\mathbb{X}_\pi = \pi(\mathbb{X}_G, s, t;\theta_\pi) \tag{2.12}$$

$$\mathbb{C} = B(\mathbb{X}_\pi) \tag{2.13}$$

$$\mathbb{M} = H(\mathbb{C}) \tag{2.14}$$

$$r = V(\mathbb{M}, s) \tag{2.15}$$

$$\theta_\pi \leftarrow \theta_\pi + A_\pi(\theta_\pi, \mathbb{X}_G, \mathbb{X}_\pi, s, r) \tag{2.16}$$

$$\theta_G \leftarrow \theta_G + A_G(\theta_G, \mathbb{X}_G, \mathbb{X}_\pi, s, r) \tag{2.17}$$

Where G is chaotic pattern generator, t the time, $\mathbb{X}_G$ the output of G, θ_G the parameters of π_G, π the sensory processing function, s the sensory input, θ_π the parameters of π, $\mathbb{X}_\pi$ the output of π, B the motor controller, $\mathbb{C}$ the motor control signal, H the physical system dynamics, $\mathbb{M}$ the motion of robot or animal, V the function to calculate rewards, r the reward of decisions or motions, A_π the function to update parameters θ_π, A_G the function to update parameters θ_G.

The main differences between these two type of approaches are in Equations 2.6 and 2.12, 2.7 and 2.13, and 2.17. In chaoticPG approaches, the outputs of a chaoticPG do not feed into the learning network, whereas in chaoticFG approaches, they do. In chaoticPG approaches, the input source of the motor controller switches between the chaoticPG and learning network, whereas in chaoticFG approaches the input source is the learning network. In chaoticPG approaches, the parameters of a chaoticPG usually are not be tuned, whereas in chaoticFG approaches they are.

Chapter 3

Maggot Operant Learning

3.1 Introduction

Based on the dynamic synapse model proposed in chapter 2, a model for the mushroom body, which is a learning centre in insects, is built to reproduce an operant learning behaviour of *Drosophila* larvae.

Drosophila adults are capable of operant learning, as observed in the heat box experiment(Brembs, 2003) and fly torque experiment(Brembs and Heisenberg, 2000). *Drosophila* larvae have immature brains, but the essential structures are similar to those of adults. Hence, *Drosophila* larvae are also believed to be capable of operant learning. There are also experiments support that *Drosophila* larvae are capable of operant learning (Eschbach, 2011).

To further investigate the operant learning capacity of *Drosophila* larvae, especially the role of the mushroom body during operant learning, an experiment about operant learning of turning behaviour has been conducted by my collaborators. The line of *Drosophila* larva is transgenic that a dopaminergic neuron in the mushroom body, which represent reward, can be activated with a specific wavelength of light. In this experiment, when a *Drosophila* larva bends its body exceeding the threshold of body bending angle, the light is shined over it as a reward. The data is recorded and analysed. Base on the experiment, a mushroom body model with dynamic synapse for operant learning of *Drosophila* larval turning behaviour, is proposed and tested based on existing research on the mushroom body. The simulation result agrees with the biological experiments.

3.1.1 Mushroom body

A mushroom body is a part of a *Drosophila* brain as a learning and memory centre (Takemura et al., 2017) with neural circuit architecture similar to the cerebellum. The mushroom body receives signals from multiple primary sensory centres and processes multiple sensory modalities, such as chemosensory, hygrosensory, or thermosensory(Yagi et al., 2016). The outputs of the mushroom body can guide memory-based action selection (Aso et al., 2014b). The Kenyon cells are neurons intrinsic in the mushroom body and receive signals from other regions of the brain. Kenyon cells represent the sensory signals with sparse coding, which enhances the discrimination difference between sensory signals (Lin et al., 2014). An adult *Drosophila* has about 2200 Kenyon cells (Aso et al., 2009). Kenyon cells (KCs) have long parallel axons across the mushroom body, along with which are multiple compartments. Each of the compartments has at least a mushroom body output neuron (MBON) and a Dopaminergic neuron (DAN). There are mutual connections among the MBON, the DAN and KCs (Eichler et al., 2017). DANs and MBONs from different compartment nerves to different regions, suggesting different compartments have different functions. DANs can be activated by unconditioned stimuli, such as sugar or quinine, which can be reward or punishment for the learning of conditioned stimulus. That is, when a DAN spikes, it release Dopamine to the compartment where the DAN locates, modulating the synaptic plasticity between the KCs and the MBONs. These characteristics of the mushroom body provide a basis for operant learning of turns.

3.2 Experiments and observation

The set-up of the experiments is shown in Figure 3.1. The base is a platform on which a *Drosophila* larva can crawl freely. Above that, a Cartesian robot holds a camera and light for observing and rewarding a larva. A software system controls the robot to track a moving maggot, calculates the body bending angle of the larva, and controls the Light-emitting Diode (LED) to reward the larva with light when the larva is turning.

The larvae in the experiments are of the following genotype: R58E02-Gal4xUAS-CsChrimson (R58E02>CsChrimson). For more information about the GAL4/UAS system in *Drosophila*, please see the review by Duffy (2002). On the *Drosophila* with the genotype, the red-shifted light-gated ion channels CsChrimson are expressed in dopaminergic neurons innervating the mushroom body. Activation of the neuron has previously been shown to substitute for reward in classical conditioning experiments.

The body bending angle of the maggot is defined by taking a line from the body middle point to the tail and a line from head point to body middle point (Figure 3.2).

There are three groups of experiments, with different protocols to control the light. The first protocol is to activate the light if the maximum turn (bending angle) within a time period exceeds a threshold; the second protocol is similar to the first one but only reward every second turn; the third protocol is also similar to the first one but only turning to one side (either left or right) activates the light. The body bending angle data of these experiments are recorded and analysed. Other information of the locomotion, such as the speed, is observed and recorded, but not used in the analysing and modelling.

For each group of experiments, there are 200 larvae observed successively. Among them, the first 100 larvae are the experimental group, who get light as the reward according to the protocols. The second 100 larvae are the yoked control group, each of which gets the same sequence of light with the corresponding one in the experiment group. The yoked control group is to eliminate possible direct effects on turning behaviour from experiencing light or reward, by providing the same experience but uncoupled from the animals' behaviour. For each larva, there are five phases of observation. The light is switched on according to the body bending angle in phase 2 and 4. The time length of each phase is 45, 210, 45, 210 and 90 seconds, respectively.

A key observation object is the proportion of time spent in turns for each phase. A turn motion is defined as the motions with a maximum angle larger than ±30 °, and the beginning and end of a turn are when the maggot body is straight. Figure 3.3 shows four turns in green.

3.3 Models and Methods

The objective of our model is to qualitatively reproduce the statistical results of *Drosophila* larvae's turning behaviours before and after learning under the control of the light. As the study is to investigate the learning with the mushroom body, the change of the behaviours is more important than the behaviours themselves. As the learning of motions is in a behavioural level instead of an action level, our model does not intend to reproduce the motions accurately, such as the turning trajectory. The motions are reproduced qualitatively on a simplified body model.

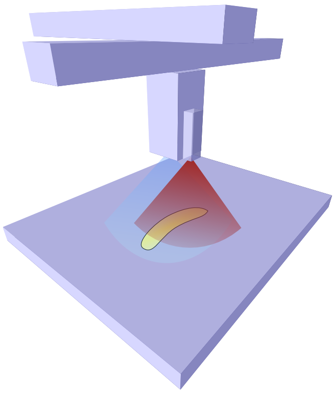


Figure 3.1: The experiment set-up for operant learning of turning behaviour. The base is a platform on which a *Drosophila* larva can crawl freely. Above that a Cartesian robot holds a camera and light for observing and rewarding the larva.

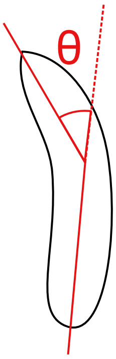


Figure 3.2: The measurement of a *Drosophila* larva's body bending angle.

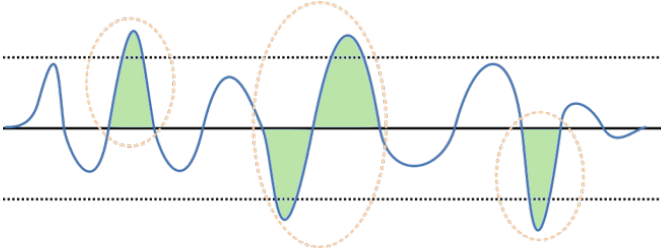


Figure 3.3: The illustration of turns. The horizontal direction is time. The The dash lines are the body bending angle threshold defining turns. The curve is body bending angle. The green regions represent four turns.

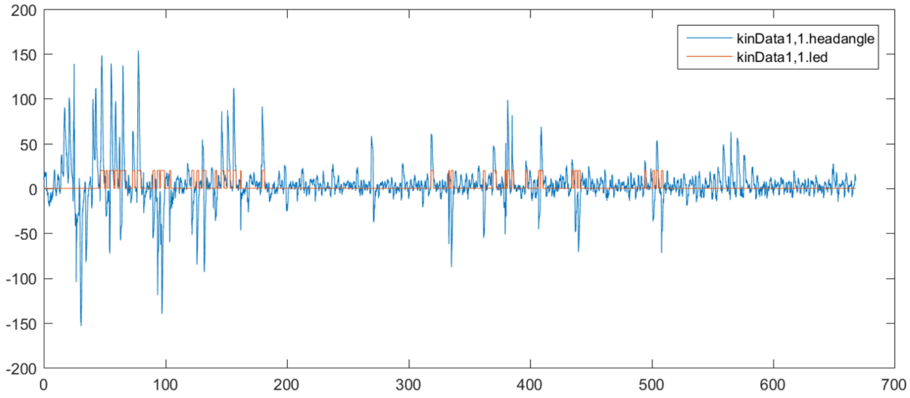


Figure 3.4: Trace of a *Drosophila* larva's body bending angle (blue) and the state of light for reward (red). When the body bending angle exceeds the threshold, the LED is turned on. When the angle smaller than the threshold, the LED is turned off. The LED is not functional during the phase 1,3 and 5 of every experiment.

3.3.1 Analysis of relations between observed behaviours and mushroom body

The traces of larvae body bending angle has multiple randomnesses. For the model, we considered the time when a turn starts, the maximum body bending angle and the time length of a turn. Fig 3.4 shows a recording trace of a larva body bending angle. The larva body bending angle has a range from -150°to 150°, and turning time range from about 1 second to about 10 seconds.

In the experiments, there was no typical olfactory stimulus to the larvae. Because of the low-level olfactory stimulus, the sensitivity of the olfactory system increased, and signal-to-noise ratio (SNR) of odour signal decreased, hence their KCs activities could interpret as random signals. Therefore the strengthen synaptic connections from KCs to MBONs increases the possibility of MBON's firing.

If the MBON, which is in the same compartment with the light controlled DAN, modulates turning, firings of the MBON should be able to modulate its downstream neural circuit for turning control. The time when a turn starts could be caused by when MBON fires. As MBONs integrates outputs of KCs, and the noisy KCs activities cause random outputs, hence when MBONs fires are uncertain.

The direction of turning could be influenced by the coupled dynamics of the MBONs, the downstream neural circuit for motor control, and the larval body. If the coupled dynamics could be disturbed asymmetrically, larval head during locomotion can result

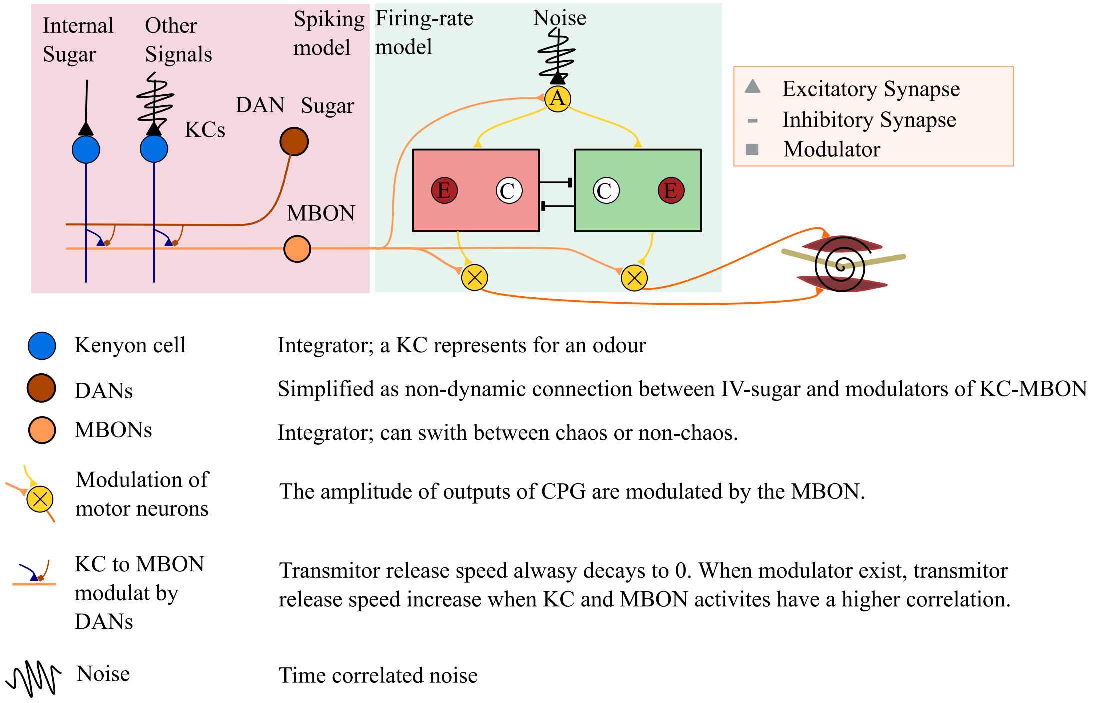

Internal Sugar
Other Signals
Spiking model
Firing-rate model
Noise
KCs
DAN
Sugar
MBON
A
E
C
C
E
Excitatory Synapse
Inhibitory Synapse
Modulator
Kenyon cell
Integrator; a KC represents for an odour
DANs
Simplified as non-dynamic connection between IV-sugar and modulators of KC-MBON
MBONs
Integrator; can swith between chaos or non-chaos.
Modulation of motor neurons
The amplitude of outputs of CPG are modulated by the MBON.
KC to MBON modulat by DANs
Transmitor release speed alwasy decays to 0. When modulator exist, transmitor release speed increase when KC and MBON activites have a higher correlation.
Noise
Time correlated noise

Figure 3.5: Schematic diagram of the model

in asymmetric oscillation. The coupled dynamics could also determine the maximum body bending angle and the time length of a turn.

In the experiment, we also notice there was a decline of turning time in the total time. It could be caused by the decrease of ’patience’ of searching for food nearby which could be a result of decreasing internal energy substance such as sugar or negative phototaxis of *Drosophila* larvae Kane et al. (2013). Hence, we assume that at least one of the KCs represents the information, which is modelled by Equation 3.5.

3.3.2 Neural circuit model and larval body model

The model consists of KCs, a DAN, an MBON, a CPG and a larva body model (Fig 3.5). KCs are implemented with Izhikevich neuron model (Izhikevich, 2007), and the MBON is implemented with a Leaky integrate-and-fire (LIF) neuron model. Other neurons are firing-rate models. The downstream neural circuit of the MBON is a CPG model by Wystrach et al. (2016) which control. The larval body is modelled as two rigid sticks connected by a joint and actuated by muscles.

As analysed above, the KCs are fed with time-correlated random noise, which mimics sensitive sensors in low SNR environment and other complex internal signals. One of the KCs represents the decaying internal sugar level for modelling the behaviour of decreasing turning intentions during observations. MBONs integrates the KC outputs, hence when MBONs start to fire is uncertain, and the synaptic strength between KCs and the MBON can affect the possibility of firing. Hence the time when a turn starts is uncertain, and the possibility of turning is correlated with the synaptic strength. For simplification of our model, only the necessary MBON for turning is modelled. Each fire of the MBON causes the release of neurotransmitter which sends a turning signal to its downstream neural circuit.

The downstream neural circuit is a central pattern generator (CPG) by Wystrach et al. (2016). The MBON outputs modulate the CPG by feed neurotransmitter signal to the input of the CPG and controlling the amplifying of the output of CPG proportional to the amount of neurotransmitter. The CPG is modelled according to *Drosophila* larva taxis behaviours, controls turning during locomotion. It has internal variables with periodic dynamics, and its output control *Drosophila* larva's "head" oscillate continuously. The dynamics could be disturbed by inputs and results in asymmetric oscillation. Hence a spike from the MBON can cause different turning amplitude depending on the internal state of the CPG when the signal arrives. A noise signal is also modelled for representing other inputs of the CPG, which causes a more significant variation of the turning amplitude.

There are two outputs of the CPG which controls the contraction of muscles turning to different directions. The muscles and body were modelled as a second-order dynamic system with springs and dampers. The muscles drive the relative turning between the two segments of the body model. The body model has mass and inertia, and each segment has the same fixed length. The head half turns around the tail half when the muscles apply force around the joint connecting them. The direction of the head leads the direction of crawling which is the direction of the joint's motion. The tail half gradually follows the direction of the head during crawling. The relative turning between "head" and "tail" is modified from (Fung, 2013). The dynamics of the body model also contributes to the variation of turning amplitude.

The KC neurons are based on Izhikevich neuron model with parameters from work by Wessnitzer et al. (2012). Izhikevich neuron model has a good balance between biological plausible and computational expense for large-scale simulation. It has 2 continuous dynamics and one discrete dynamics:

$$\mathrm{C_K}\dot{v}_K = \mathrm{k}\left(v_K - \mathrm{v_{rK}}\right)\left(v - \mathrm{v_{tK}}\right) \tag{3.1}$$

$$\dot{u} = \mathrm{a}\left\{\mathrm{b}\left(v_K - \mathrm{v_{rK}}\right) - u\right\} \tag{3.2}$$

$$\text{if } v_K \geq v_{\mathrm{peak_K}}, \text{then } v_K \leftarrow c, u \leftarrow u + d \tag{3.3}$$

where v_K is the membrane potential, u the recovery current, the membrane capacitance $\mathrm{C_K} = 4$, the rheobase $\mathrm{k} = 0.015$, the resting membrane potential $\mathrm{v_{rK}} = -85$, the instantaneous threshold potential $\mathrm{v_{tK}} = -25$, the recovery time constant $\mathrm{a} = 0.01$, the input resistance $\mathrm{b} = -0.3$, the voltage reset value $\mathrm{c} = -65$, the total amount of outward minus inward currents activated during a spike $\mathrm{d} = 8$. Time unit is millisecond.

There are 2 KCs in the model representing the two major types of inputs: noise and interest of turning.

The noise is time correlated continuous random numbers to mimic the time correlated neuron activities:

$$\rho_i = \alpha X_\alpha - \beta X_\beta(\rho_{i-1} - \rho_c) \tag{3.4}$$

where $X \sim U([-1,1])$, the uniform distribution from -1 to 1, α the slope coefficient with value 3, β the pull back coefficient with value 0.001, ρ_c the centre of the random value ρ_i the ith random number.

The decreasing of interest in turning is tuned to fit the experimental result:

$$s(t) = 34 + 20/(1 + \exp(3(\frac{t}{T} - 0.1))) \tag{3.5}$$

where t is the time of the calculation and T is the total time of simulation.

When a KC fires, the corresponding neurotransmitter is released. The amount of neurotransmitter decays after being released :

$$\dot{n_t} = -\mathrm{k_{n_{t_d}}} n \tag{3.6}$$

$$\text{if neuron fires, } n_t \leftarrow n_t + \mathrm{n_{t_r}} \tag{3.7}$$

where n is the amount of neurotransmitter released in a synapses between a KC and a MBON, the amount of neurotransmitter released in during a spike $\mathrm{n_{t_r}} = 0.4$, the neurotransmitter decay speed factor $\mathrm{k_{n_{t_d}}} = 0.02$.

The MBON is modelled based on LIF neuron. The input current of the neuron is:

$$I_{MI} = \sum \mathrm{k_{\rho I}} W_i (n_t)_i \tag{3.8}$$

where I_{MI} is input current of the MBON, W_i the synaptic strength of ith h input synapse of MBON, $\mathrm{k_\rho I}$ a factor of input conductive, n_t the amount of neurotransmitter in the ith synapse. $\mathrm{k_\rho I} = 0.22$.

The leak current of the MBON is:

$$I_{ML} = \rho_\mathrm{L} (\mathrm{v_{rM}} - v) \tag{3.9}$$

where I_{ML} is leak current of the MBON, ρ_{L} leak conductive, n_t the amount of neurotransmitter in the ith synapse. $\rho_{\mathrm{L}} = 0.002$.

The dynamics of the MBON is:

$$\mathrm{C_M}\dot{v}_M = I_{MI} + I_{ML} \tag{3.10}$$

$$\text{If } v_M \geq \mathrm{v_{tM}},\, v_M \leftarrow \mathrm{v_{rM}} \tag{3.11}$$

where the membrane capacitance $\mathrm{C_M} = 4.5$, the threshold potential $\mathrm{v_{tM}} = 30$, the resting membrane potential $\mathrm{v_{rK}} = -60$.

When the MBON fires, neurotransmitter is release and affects the downstream circuits. The dynamics of the amount of the neurotransmitter is same as Equation 3.6 and 3.7, whereas $\mathrm{n_{t_r}} = 1$, $\mathrm{k_{n_{t_d}}} = 0.0003$.

For the detail of the CPG model, please see work by Wystrach et al. (2016). The noise signal inputs to the CPG is the same as the noise signal for the KC. The overall input of CPG is:

$$A = 18 + \rho - 15 n_{MBON} \tag{3.12}$$

where ρ is the noise signal and n_{MBON} the amount of neurotransmitter released by the MBON.

There are two outputs of CPG controlling the force of two muscles, respectively, which drive the turning motions of the maggot body model. The outputs are modulated by the MBON output signals:

$$E_L = 0.22\, E_{LO}\, n_{MBON} + 0.02 \tag{3.13}$$

$$E_R = 0.22\, E_{RO}\, n_{MBON} + 0.02 \tag{3.14}$$

where E_{LO} and E_{RO} are original CPG outputs on two sides, respectively; n_{MBON} the amount of neurotransmitter released by the MBON; E_L and E_R are muscle control signals after modulation.

The body model for the turning angle:

$$\ddot{\theta}_r = -2\xi\dot{\theta}_r - \mathrm{k_B} \tan\theta_r + (E_L - E_R) \tag{3.15}$$

where θ_r is the relative angle of head from its straight direction and with anticlockwise the positive direction, ξ the damping ratio, $\mathrm{k_B}$ the stiffness coefficient, E_L and E_R the forces to left and to right,respectively. $\xi = 3$, $\mathrm{k_B} = 10$.

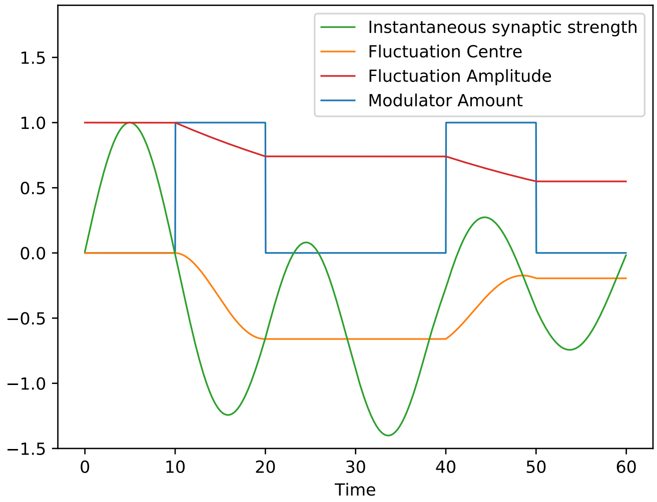


Figure 3.6: Schematic diagram of the dynamics of the synaptic plasticity model. With the presence of the modulator, the synaptic strength fluctuation centre bias to the side of instantaneous synaptic strength. The learning is bidirectional depending on the relative position of the instantaneous synaptic strength and the synaptic strength fluctuation centre. The amplitude of the fluctuation decreases with the presentation of the modulator.

3.3.3 Synaptic plasticity model

Synaptic plasticity models can be divided into two classes: phenomenological models, which are very simplified to capture the effect on neural circuits, and biophysical models, which are more detailed including internal biophysics. The former is usually based on the theoretical analysis, and the latter is usually based on controlled synaptic plasticity experiment. The synapse model we used is the former inspired by the latter. It is inspired by the dynamics of internal biophysics, especially neurotransmitter trafficking which causes spontaneous variation of synaptic strength. The variation of the synaptic strengths is a continuous exploration of the parameter space of a neural network. When the output of the neural network causes reward, the neural modulator is released. The exploration range approaches the transient synaptic strength when a synapse gets the modulator, hence gradually to a more optimised state. As shown in Fig 3.6, synaptic strength fluctuates around its fluctuation centre with a specific amplitude. With the presence of the modulator, the fluctuation centre is biased to the instantaneous synaptic strength, and the fluctuation amplitude decreases. The learning is bidirectional.

Different from most phenomenological synaptic plasticity models which have to be specified for either firing rate based or spike time-based neurons, our model can be applied to both. It is a post-synaptic learning rule controls the sensitivity of the post-synaptic region to neurotransmitters. The biophysical basis of the model is the variation of the number of neurotransmitter receptors in the post-synaptic region (Cingolani and Goda, 2008), which is a key factor in synaptic strength (Sheng and Hoogenraad, 2007). The average amount of the receptor in the region depends on the capacity of the region to contain them. However, because of dynamical receptors trafficking, the instantaneous amount fluctuates. With modulator, synapse with receptor more than its average capacity increases the capacity, synapses with receptor less than its average capacity decrease the capacity. At the same time, the resistance of receptor trafficking increase. Hence the synapse strength is gradually optimised. For more detail, please see section 2.

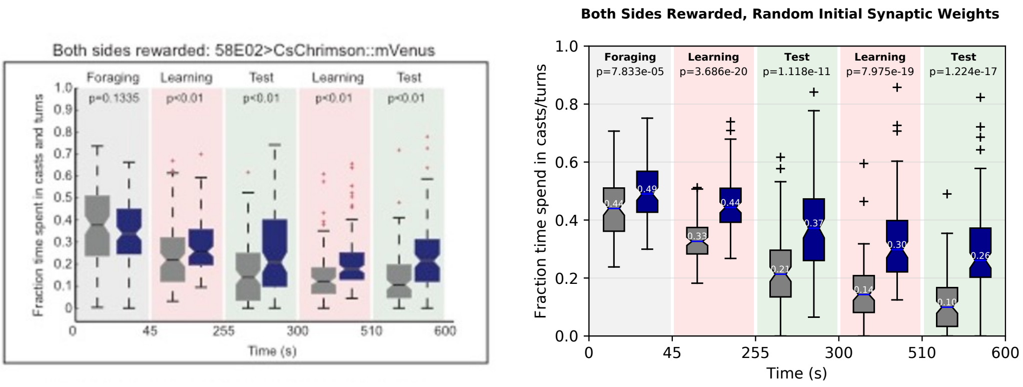
3.4 Results

The neural circuit model and body model are tested with the training protocols used for the real larvae. Overall, the simulation results agree with the experiments. For comprehensive comparisons, the results are analysed in different aspects. We analysed, firstly, the proportion of time spent in turns for each phase. The expected phenomenon is that the experimental group will have a higher proportion of time than the observation group as the experimental group learns to turn. Then we analysed the accumulative distribution of run duration. The expected phenomenon is that the runs would be broken into shorter runs because larvae learn to turn more frequently. In the following, results are organised according to the experimental protocols.

3.4.1 All turns rewarded

In this group of experiments, a larva in the experiment group got reward whenever its body was bending angle reaching the outside of ±30 °.

Figure 3.7 shows the proportion of time spend in turns for each phase time. Figure 3.7a shows the result of biological experiment and Figure 3.7b shows the result of simulation. The biological results show overall decreases of both experimental group (blue) and the yoked control group(grey). It could be a result of lost patience due to the decreasing internal energy substance or negative phototaxis. There is an increasing



(a) Biological experiment

(b) Model simulation

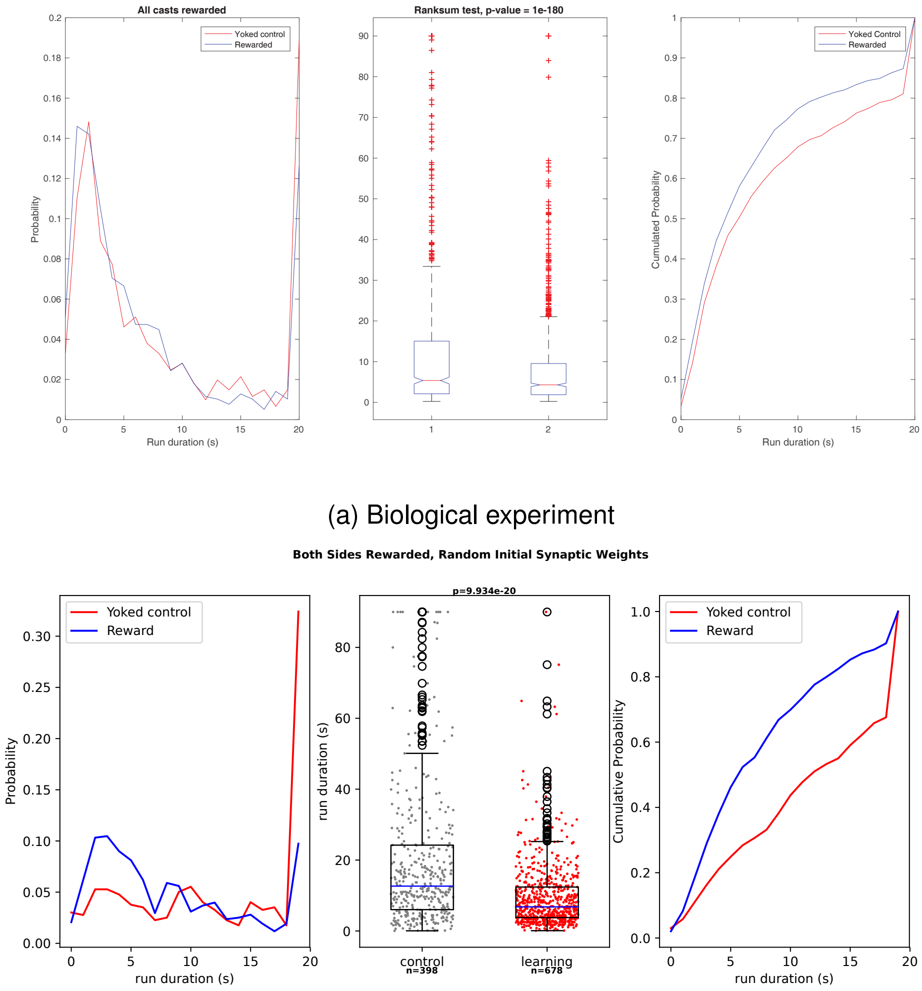
Figure 3.7: Fraction time of turns in all-cast-rewarded condition. The experiment group is in blue, and the yoked control group is in grey.

difference between the experimental group and the yoked control group showing the experimental group learns to spend more time in turns. The simulation reproduces both of the phenomena with a quantitative agreement.

Figure 3.8 shows the distribution of time spent in each runs in the final phase. In the biological experimental results, the experimental group has more runs distributed in a shorter time. The accumulated distribution plot more clearly shows the results. The model simulation qualitatively reproduced the results (please note the scales are different between Figure 3.8a left and Figure 3.8b left).

Figure 3.9 shows the box plot of average duration of turns per larva. Although distributions in simulation results are more concentrated than the biological experimental results, the middle values and distributions reproduced by simulation are close to the biological observation, such as the overall decreasing of the turn duration and the experiment group has higher middle value than the yoked control group (please note the difference in the scale of the figures).

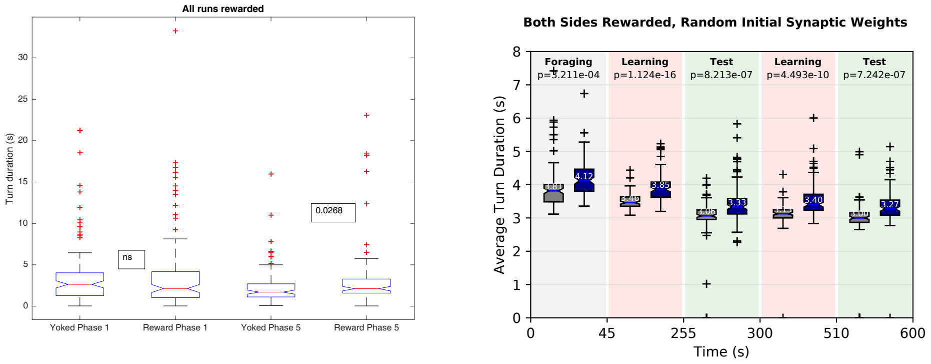
Figure 3.10 shows the box plot of average duration of runs per larva. There is an overall increase in the average run duration, which could be a result of lost ”patience” or negative phototaxis. In the first phase, the experiment group and the control group are similar. After learning, the distribution and middle values of the control group are significantly wider and higher than those of the experiment group. It may be caused by that the yoked control group receives light rewards uncorrelated to their motions so that they could get more random rewards for runs. The simulation reproduced the result (please note the difference in the scale of the figures).



(a) Biological experiment

(b) Model simulation

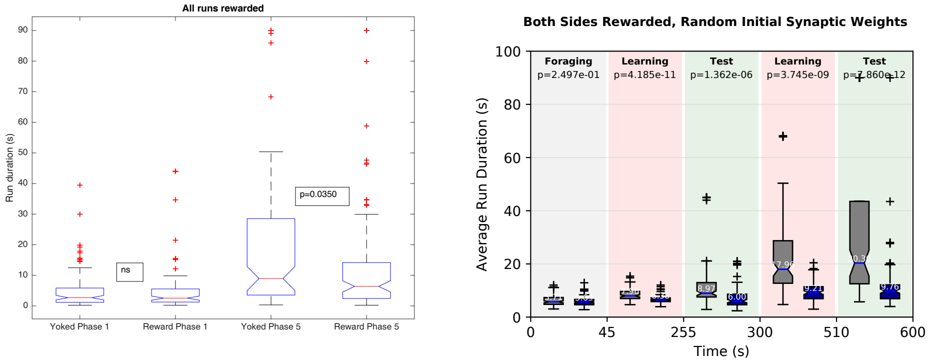
Figure 3.8: The distribution of runs in the final phase after training by rewarding turnings in both sides. Blue is the experiment group, and grey is the yoked control group.



(a) Biological experiment.

(b) Model simulation.

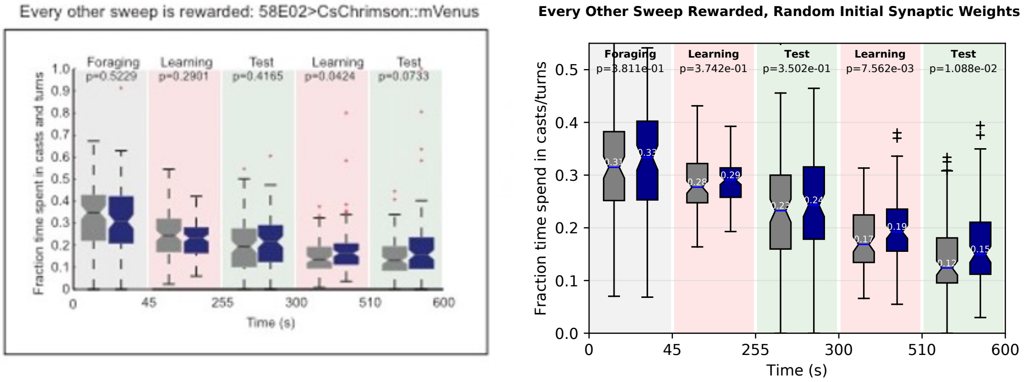
Figure 3.9: The average duration of turns before and after training by rewarding turnings in both sides. (a) Biological experiment. Only phase 1 and phase 5 are shown. (b) Model simulation. All phases are shown. Blue is the experiment group, and grey is the yoked control group.



(a) Biological experiment.

(b) Model simulation.

Figure 3.10: The distribution of runs before and after training by rewarding turnings in both side. (a) Biological experiment. Only phase 1 and phase 5 are shown. (b) Model simulation. All phases are shown. Blue is the experiment group, and grey is the yoked control group.



(a) Biological experiment

(b) Model simulation

Figure 3.11: Fraction time of turns of the larvae after training by rewarding every other turn. The experiment group is in blue, and the yoked control group is in grey. Please notice the difference in the scales.

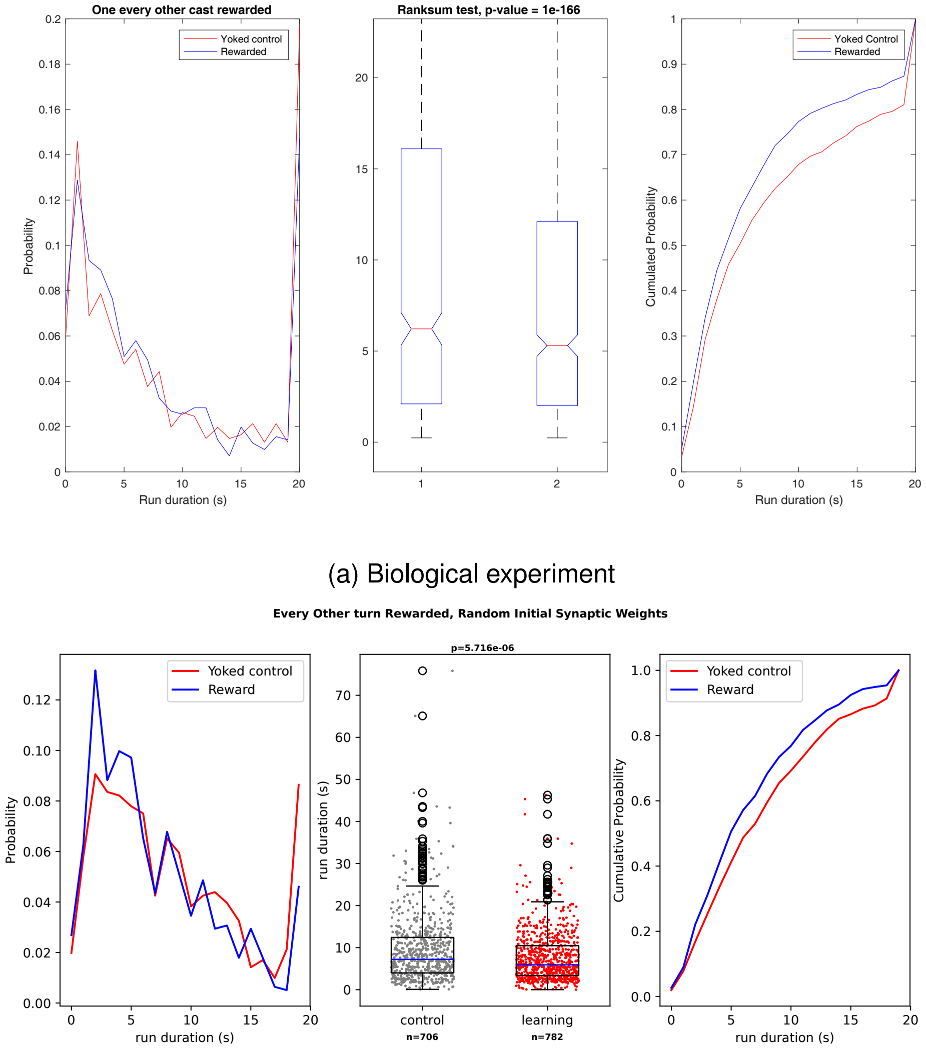
3.4.2 Every second turn rewarded

In this group of experiments, every other turn is rewarded. Because light rewards in this group are less than those in the all-turns-rewarded group, the effects of rewards are expected to be weaker in this group. As shown in Figure 3.11a, the proportion of time spend in turns decrease less then that in Figure 3.7a. Besides, the differences between experiment group and yoked control group is smaller than those in Section 3.4.1, such as the distance of middle values in phase 5 between experimental and yoked control group. The model simulation reproduced these differences, as shown in Figure 3.11b.

Figure 3.12 shows the distribution of time spent in each runs in the final phase. As excerpted, the experimental group have shorter run durations than the control group, and the model simulation reproduced the results.

3.4.3 One side rewarded

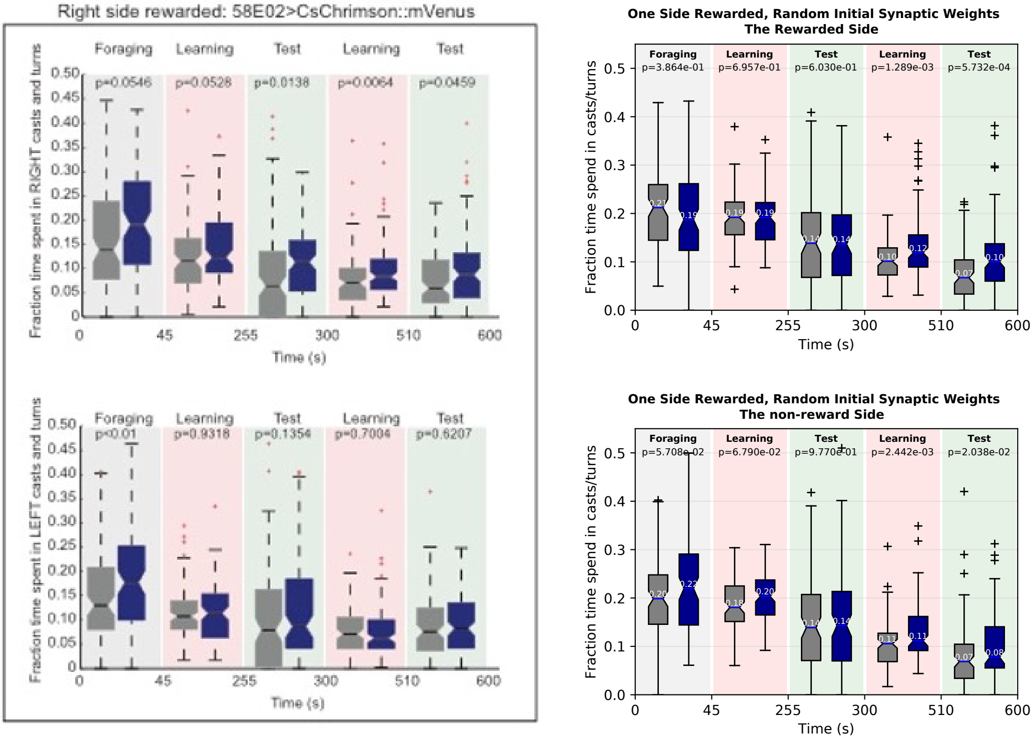
In this group of experiments, the larvae only got rewards when they turn to a specific side (relative to their body axis), to test if larvae can learn to turn to one side preferentially. Two features are observed in this group. The first is that, as shown in Figure 3.13, the biological results show a small difference between the reward side (the right side of a larva) and non-reward side (the left side of a larva). That is, the differences between the proportional time of the experiment group and yolked control group are higher on the reward side than those on the non-reward side. The difference is a (weak) evidence for side specific learning. The second feature is that, compared with



(a) Biological experiment

(b) Model simulation

Figure 3.12: The distribution of runs in the final phase after training by rewarding turns in both side. Blue is the experiment group, and grey is the yoked control group.



(a) Biological experiment

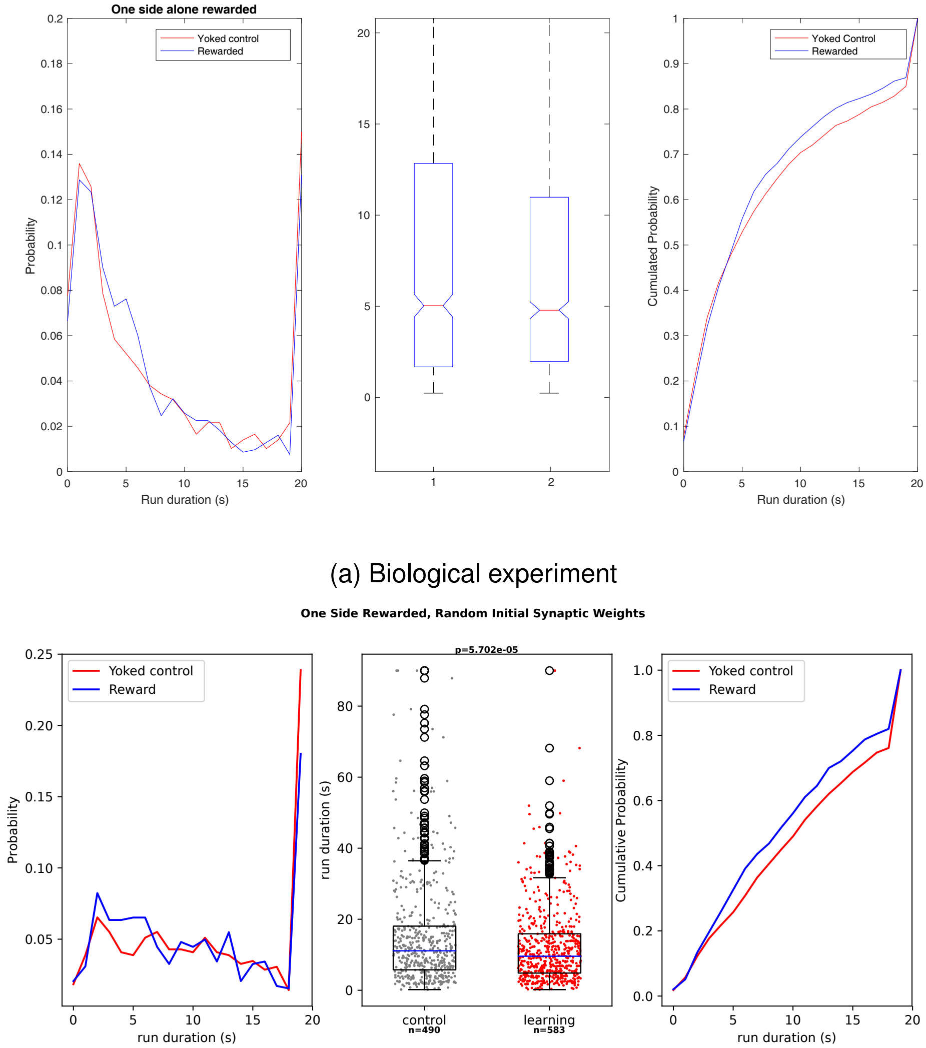
(b) Model simulation

Figure 3.13: Fraction time of turns in one-side-rewarded condition. The experiment group is in blue, and the yoked control group is in grey. Figure (a) shows the results of the biological experiment and the figure (b) are the results of the model simulation. The top plots of Figure (a) and (b) are the results of the reward side, and the bottom plots are the results of the non-reward side.

the both-side-rewarded group, the differences between experiment and yolked control in this group are small, which can be explained by that the overall rewards are less in this group.

The model simulation reproduces the second feature but not the first feature. It is because the model does not include the mechanisms, such as sensory of turning angle feeding to KC, for the learning of differences between two sides. To capture the first feature, a model should include more factors that modulate the learning in *Drosophila* larvae.

Figure 3.14 shows the distribution of run duration in the final phase. Similar to the group that every other turn is rewarded, the difference between the run duration of the experimental group and the control group is smaller than that of the all-turn-rewarded group. The model simulation also reproduced the features.



(a) Biological experiment

(b) Model simulation

Figure 3.14: The distribution of runs in the final phase after training by rewarding turnings in one side. Blue is the experiment group, and grey is the yoked control group.

3.5 Discussion

In this chapter, an operant learning model based on the mushroom body (MB) and the dynamic synapse is proposed and tested. The previous mushroom body models are for associative learning of sensory inputs, such as olfactory information and vision information, with the presence of unconditional stimulations, but not for operant learning. The mushroom body model presented in this chapter is the first mushroom body model for operant learning, based on the results of biological experiments with optogenetic *Drosophila* larvae, whose dopaminergic neurons (DANs) in MB are controlled with light. In the simulation, the DANs in the MB model are controlled by the light sequences identical with the biological experiments. The model simulation results agree with the biological experimental results, which supports the hypothesis that the MB is capable of operant learning and the dynamic synapses model presented in chapter 2 is plausible for realistic operant learning behaviours.

The MB model is simplified in comparison to the real MB in insects. The parts of MB that are not essential to reproduce the learning behaviours are ignored. For a real MB in an L1 *Drosophila* larva, there are more than 200 KCs, 23 MBONs and 7 DANs Eichler et al. (2017). In the model, only two KCs, one MBON and one DAN are modelled. Only 2 KCs are modelled for two different types of inputs to MB, the inputs to decrease the interests of turning behaviours, which are observed in both the experimental group and control group, and other signals, which are noisy as the larvae were in an environment lacking sensory stimuli with patterns. One MBON is modelled because an MBON is capable of high-level motion control, which is adequate to control the only observed learning behaviour, changing of turning. One DAN is modelled because there is usually one pair of MBON and DAN in a compartment of MB and that DANs of larvae in the experiments got the same stimulation carrying the same information. It is a level of simplification such that the key structure of the MB is kept, whereas any further simplification cannot keep the structure.

The “other signals” which is fed to one of the KC, provides noise that affects when the MBON spikes. Each spike indicts an intention to turn in the high level, then the dynamic of CPG is disturbed by the spike, resulting in a higher possibility for turning. Hence, the “other signals” results in a random interval between turns. Without the “other signals”, the model turns when the total dynamic synaptic strength is higher than a specific level. There is another noise signal that feeds to CPG, which simulates unknown inputs to the CPG. This noise affects the size of turning. Without the noise, the model turns with very similar sizes.

The MB model can be measured with seven dimensions proposed by Webb (2001) for differing simulation models:

1. Relevance: The MB model is closely relevant to the biology, as the tests use the same protocol with biological experiment, and the generated hypotheses, such as additional rewards during running can attenuate synaptic strength between the KCs and the MBON, are applicable to biology.

2. Level: The MB model is a model on the level of synapse and neurons.

3. Generality: As insects have MB, so the model, including the MB, CPG and the body, can be generalised to all insects with the larval stage.

4. Abstraction: The KCs and MBON in the MB model are modelled with spiking neuron models, hence the model has spikes that indicate when to turn like the real MB. The CPG is a firing-rate model as it is adequate to reproduce the dynamic of CPG. The larval body model has two segments as the turning behaviour can be represented with the segments.

5. Structure accuracy: The model has three levels, the behaviour level MB model, the action level CPG model and the agent level body model. The levels are similar to real larvae, who have MB lobe nerves to ventral nerve cord, where CPG is located (Berni, 2015; Clark et al., 2018), then the motor neurons in the VNC control the muscle contraction. The structure of the MB model also keeps the fundamental structure of MB.

6. Performance match: The model captured the decrease of the turning ratio and the increase of differences between experimental groups and yoked control groups which are observed in the *Drosophila* larvae turning experiment. The simulation result qualitatively reproduced the motions and behaviours, supporting the hypothesis that the mushroom body with dynamic synapses can play a key roles in the operant learning of turning behaviour.

7. Medium: The model is implemented as a numerical simulation of differential equations, based on the published neuron, CPG and larval body models.

The model is open to adding further neurons for more functions. In this version of the model, we only considered spontaneous turning behaviours and modulation for them. In real larval, however, behaviours and motions also depend on the inputs of their proprioceptors (Hughes and Thomas, 2007), which sense the states of their body and provide feedback for motion neurons. It could explain the difference between the biological experimental results and model simulation results with the protocol that only those turns in one side are rewarded, during which proprioceptors provide the information with the directions the larvae turn. Integrating feedback neurons into the model, such as the connections from proprioceptors to the KCs, can be the next step.

Chapter 4

Reinforcement learning for Bipedal Robot locomotion

4.1 Background

In chapter 2, a synaptic plasticity model based on neurotransmitter receptor trafficking was proposed. Using the learning rule based on the model, which is called dynamic synapse learning rule in this chapter, neural circuits can explore synaptic strength and be optimised with reinforcement signals. In this chapter, the synaptic plasticity model is abstracted and simplified for engineering application while the principle of learning rule is kept. The simplification is for reducing computation amount by avoiding numerical integration and for a wider range of exploration including negative numbers. The simplified model is applied to a dynamic neural network to control a planar bipedal robot, and then the result is compared with other reinforcement learning algorithms. The learning rule outperforms previous reinforcement learning rules in the task.

4.1.1 Robot learning

Controlling a robot to conduct a task with classical approaches requires a human programmer or controller who understands the robot system, the environment that the robot works in, and detailed procedures of the task. However, in some cases, it is not practical for a human to know the details. For example, (a) the movements of soft robots exists across the body which introduces infinite degree of freedoms(DoFs) hence they are hard to bemodelled accurately (Rus and Tolley, 2015); (b) The information of some environments cannot be fully acquired before the robot enter the situation, and the communication delay can impact real-time remote control for human-in-loop decision making, thus robot learning by itself became a valuable option; (c) service robots may be expected to manipulate various non-standard objects, hence it is not practical to program all possibilities (Kemp et al., 2007).

Robot learning is an interdisciplinary research field aiming to endow robots with learning capabilities (Sigaud and Peters, 2012). With robot learning, robots could be programmed to conduct tasks without specific knowledge for them. It is closely related to reinforcement learning (Kober et al., 2015; Wiering and Van Otterlo, 2012), learning from demonstration (Argall et al., 2009; Zhu and Hu, 2018; Hussein et al., 2017), and cognitive developmental robotics (Asada et al., 2009; Lungarella et al., 2003), and so on. The approach presented in this chapter belongs to the category of robot reinforcement learning, especially those using neural networks. Related works are reviewed in the next section. Other types of robot learning applied to bipedal robot control are also reviewed in the following section.

4.1.2 Reinforcement learning with neural networks

4.1.2.1 Reinforcement learning

Reinforcement learning is based on the psychological concept of Instrumental Learning or Operant Learning (Saksida et al., 1997), which is learning of actions or behaviours according to their consequences (Skinner, 1938; Hull, 1943; Staddon and Cerutti, 2003).

In computer sciences, reinforcement learning is defined using the frameworks of Markov decision processes(Sutton and Barto, 1998), which are discrete stochastic processes (Puterman, 2014). With this definition, for a reinforcement learning task, there is an agent in an environment executing actions and receiving information of states of the world and values of actions. At each time step, an agent chooses and excuse an action from a set of finite actions according to a policy; as a result, the states of the environment change, and a value is calculated according to the new states and the goal of the task; based on the value, the policy is updated. The policy is probabilities to choose actions according to the states.

Existing reinforcement learning models based on discrete stochastic process has achieved excellent performance in simple and discrete tasks, such as a robot in a grid world. However, it is not suitable for robot control tasks, which are dynamic, continuous, sophisticated and with a set of infinite actions. Given dimension and discrete state and action spaces, the learning challenge is polynomial scaling in the size of the spaces (Kearns and Singh, 2002), while with continuous state and action spaces, the learning challenge scales exponentially in the dimensions of the spaces, which is also known as the curse of dimensionality (Bellman, 1957).

There are existing efforts to control continuous systems, including robots, with reinforcement learning. Significant progress has been made by combining reinforcement learning with neural networks, which have excellent performance in fitting continuous functions. The approach is also known as deep reinforcement learning.

4.1.2.2 Deep reinforcement learning

Deep reinforcement learning is an interdisciplinary research field combining deep learning and reinforcement learning. Deep learning(Lecun et al., 2015), which rises in recent years, has powerful function approximation abilities, proving new tools for reinforcement learning(Arulkumaran et al., 2017).

Deep learning is a sub-field of Artificial Neural Networks (ANNs). ANNs branched from computational neuroscience in the middle of the last century toward to machine learning and statistics. They are performance-oriented and free from following the biological neural circuits. Neural networks with many layers were believed hard to be trained with error back-propagation. In this century, because the development of computational ability and some efforts in training tricks, training neural networks with multiple layers become possible (Schmidhuber, 2015) and people start to use 'Deep neural networks' as the name of the type of neural networks.

Deep Q-learning (DQN) is the first successful combination of deep learning and reinforcement learning(Mnih et al., 2013). DQN uses a feedforward neural network to replace the look-up table for Q value, which indicates the anticipation of values in the future given current states and action. The non-linear function approximation ability of FNN provide convergence guarantees. It also uses experiments reply, which smooths the training distribution by randomly samples previous transitions. The approach solved seven Atari 2600 games with the same architecture, among them six outperforms previous approaches, and three of then surpasses human experts. The algorithm is improved by introducing target Q as a stabilising method, and achieve human experts level performance in 39 Atari games (Schmidhuber, 2015). Although DQN has outstanding performance in playing video games, it requires a discrete and low-dimensional action space (Lillicrap et al., 2015), which is not suitable for robot control in a continuous space.

Deep deterministic policy gradient (DDPG) is an actor-critic algorithm (Silver et al., 2014) that uses an actor network to generate actions instead of searching the action with the highest Q value. Because action searching in continuous space, which has infinite possible actions, is expensive in time, DDPG is more efficient in these tasks than DQN. DDPG has been applied to continuous control tasks such as a cart-pole swing-up task and bipedal locomotion tasks (Lillicrap et al., 2015).

Asynchronous Advantage Actor-Critic (A3C) (Mnih et al., 2016) is an actor-critic algorithm different from DDPG in that A3C does not use experience replay but asynchronously execute multiple agents in parallel. The multiple independent agents decrease the possibility to converge in local minima. A2C is the version that the data of the agents collected and processed together (Schulman et al., 2017).

There is also a type of DRL use trust region, in which the step size of optimisation is limited. Trust region helps to avoid oversized update of weight and increase the stability during learning (Arulkumaran et al., 2017). The step size of optimisation can be measured by Kullback-Leibler divergence (KLD) (Kullback and Leibler, 1951), which also known as the relative entropy measuring the difference of one probability distribution from a reference probability distribution. This type of algorithms include Trust region policy optimisation (TRPO)(Schulman et al., 2015) and proximal policy optimisation (PPO)(Schulman et al., 2017; Heess et al., 2017). PPO can generate complex parkour-like motions in rich environment (Heess et al., 2017).

Deep reinforcement learning is the type of RL approach in computer sciences closet to computational neuroscience. Algorithms of the approach have able to solve some continuous control tasks. However, there are still limitations.

The action exploration is based on action space noise from random number generators, which is not time correlated and different from continuous physical movement. Although some effort for generating temporally correlated exploration has been made, such as the Ornstein-Uhlenbeck process (Lillicrap et al., 2015) and a method to generate autocorrelated noise by (Wawrzyński, 2015), the generated signals are still not as smooth as the typical motions of robots or animals. Robot reinforcement learning based on jittering actions is not efficient.

The actions exploration is based on action space noise, with which the generated actions during exploration are not correlated to sensory inputs. Recent research by Plappert et al. (2018) shows that introduced parameter space noise generate actions more effectively and achieve higher scores than those with action space noise. With the parameter space noise, action exploration is based on the sensory inputs by changing the functions of response to the sensory input, so the state-action pairs are highly correlated. The generated actions are also based on the information of neural networks, so the state-action pairs are easy to learn by the same neural network. Parameter space noise is thus worth to be applied to new algorithms for robot reinforcement learning.

Learning from action explorations needs extra works for solving the 'credit assignment' problem. Training a neural network with gradient descent (GD) and error backpropagation (BP) needs a large amount of data while finding a suitable action by action exploration needs luck. The two factors together result in long time exploration for collecting enough suitable state-actions pairs to train a neural network. Although parameter space noise is introduced Plappert et al. (2018) for action exploration, it does not directly contribute to learning. Existing deep learning algorithms still need the calculation to speculate the updating of parameter space according to the actions space. If the parameter noise can be utilised to indicate the direction of updating parameter space directly, learning could be more efficient.

Deep reinforcement learning algorithms based on GD and BP cannot efficiently utilise the neural networks with internal dynamics, such as Central Pattern Generator (CPG) or biological plausible neural models. These are some other efforts to apply CPG with neural networks in bipedal robot control. For example, Mori et al. (2004) developed a neural network called CPG-actor-critic model for a planar Biped Robot. However, the approach takes CPGs out from the networks and treat them as part of the environment and does not directly address the problem of training neural networks with internal dynamics. If a new learning rule can enable reinforcement learning to utilise neural networks with internal dynamics, computational models in existing robot control approaches and computational neural science can be available for robot reinforcement learning.

By abstracting and simplify the learning rule presented in chapter 2, the learning rule presented in this chapter aims to generate smooth actions that correlated to sensory inputs by directly exploring the parameter space which indicates the direction of updating during learning and avoids the credit assignment problems. The learning rule also aims to have a similar advantage of trust region by constraining the size of parameter exploration.

The implementation of the work reviewed above is substantially based on feedforward neural networks (FNNs) instead of recurrent neural networks (RNNs). FNNs are more commonly used than RNNs because FNNs are more robust during optimisation with backpropagation (BP) and gradient descent (GD) than RNNs, which suffers from error vanishing or exploration while the error is recurrently passed throughout the network during optimisation. Long short term memory (LSTM), in which a memory cell has an input gate, a forget gate and an output gate to control the recurrent flow and the memorised values in the cell, is a special case of RNN that largely avoids the problem noted above (Hochreiter and Schmidhuber, 1997). LSTM has been applied to reinforcement learning, for instances Mnih et al. (2016) implemented A3C with LSTM for Atari games, Heess et al. (2017) used distributed proximal policy optimization (DPPO) with LSTM for a random-target reaching task with a simple 2 DoF robotic arm, Song et al. (2017) applied LSTM in the bipedal walker task same to the one in section, Peng et al. (2018b) applied Hindsight Experience Replay (HER) (Andrychowicz et al., 2017) and Recurrent Deterministic Policy Gradient (RDPG) (Heess et al., 2015) with LSTM to the fetch arm task.

DRL with RNN can achieve better performance than DRL with FNN in tasks for which the states cannot be fully observed every time, such as exploration in a maze. In the real world, full observation of every state is not practical, in which case the Markov property no longer keeps, thus with only the observed states at one step are not sufficient for prediction of latter states. The information from earlier observations can be kept in RNN but not in FNN. Hence, RNN has more potential in real-world tasks.

Although LSTM facilitates the training of RNN, LSTM is only a special case. LSTM does not solve the problem of how to train RNNs generally, such as RNNs with simpler cells or biochemically plausible neurons. Hence, existing robot control models, such as CPG, and biologically plausible neural networks, which reference the successful neural circuits in the nature, need further approaches for training.

4.1.2.3 Other Robot Learning methods used in bipedal robot control

Imitation learning, or learning from demonstration, is relatively well developed and straightforward alternative to direct programming. It has a demonstrator providing examples for robots to learning. To teach a robot, the demonstrator can show his motion to motion observation system (Hwang et al., 2016), operate a robot by remote control (Teleoperated Demonstration) (Kukliński et al., 2014), or interact with the robot in the same environment(such as Kinesthetic Demonstration (Li and Fritz, 2015)). As this type of learning needs labelled training data and learning to fit them, it can be seen as a subset of supervised learning (Argall et al., 2009). The problems or imitation learning can be broken down into who, what, when and how to imitate (Billard et al., 2008). Two types of approaches that are popular in solving how to imitate are Neural Networks and Statistical Learning (such as (Calandra et al., 2014)). Nakanishi et al. (2004a) presented the idea of using the rhythmic movement primitives based on phase oscillators as a CPG to learn biped locomotion from demonstrations. Duan et al. (2017) proposed a framework based on neural networks for one-shot imitation learning in manipulation tasks. Calandra et al. (2014) used Rhythmic Motor Primitives (RMPs) to generate trajectories for control of an under-actuated three link bipedal robot and used Bayesian optimisation to solve trajectory imitation and optimise trajectory.

The combination of reinforcement learning and imitation learning, sometimes termed as adversarial imitation learning or apprenticeship learning (Abbeel and Ng, 2004; Kober et al., 2015), emphasises the need for learning both from a teacher and by practice. It can provide prior knowledge to achieve appropriate behaviour in fewer trials than pure reinforcement learning (Kober et al., 2015), which is important for robot learning as a robot wears after a long time of operation. It also helps in producing more natural motions in reinforcement learning and using demonstration data more efficiently in imitation learning. For example the neural network GAIL(Generative adversarial imitation learning) (Ho and Ermon, 2016), Hwang et al. (2016) reproduce human-like motion using partially observed state features. Schaal et al. (2005) trained a planar bipedal robot to walk using a framework based on Dynamic Movement Primitives, by which the robot initialised trajectory pattern by imitation learning of human motion data and optimised the locomotion by reinforcement learning. Peng et al. (2018a) proposed a deep neural network named DeepMimic which utilising motion capture data as part of target function in reinforcement learning and demonstrated it with simulations of several models including a humanoid model and an Atlas robot model.

4.2 Optimising dynamical neural networks for robot control

This section details the simplified dynamic synapse learning rule, the neural network with intrinsic dynamics and the experiments with the bipedal locomotion task.

The task involves applying reinforcement learning in a planar bipedal robot that has locomotion on terrain with slight slopes. The robot has four degrees of freedom with torque control. Its sensors are for the joint angles, the joint angular velocity, the ground contact, the angle and the angular velocity of the body to the world, as well as the horizontal and vertical velocity of the body.

The neural network is dynamic because it has two sets of recurrent connections and two central pattern generators. It is hard to be trained with the error back propagation and gradient descent algorithm. The dynamic synapse learning rule proposed in chapter 2 is simplified and applied to the training of the neural network. In the experiment, the neural network successfully learned to control the bipedal robot for locomotion. It is the first known algorithm that achieves the learning target using reinforcement learning.

For more details of the neural network, the experiments and results, please see the following paper, *A Bio-inspired Reinforcement Learning Rule to Optimise Dynamical Neural Networks for Robot Control*(Wei and Webb, 2018b), which was published at the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2018). It is about hypothesis 4, result 4,and highlight 4 mentioned in Chapter 1. The co-author, Barbara Webb, advised on the work and the writing of the paper.

A Bio-inspired Reinforcement Learning Rule to Optimise Dynamical Neural Networks for Robot Control*

Tianqi Wei[1] and Barbara Webb[2]

***Abstract*— Most approaches for optimisation of neural networks are based on variants of back-propagation. This requires the network to be time invariant and differentiable; neural networks with dynamics are thus generally outside the scope of these methods. Biological neural circuits are highly dynamic yet clearly able to support learning. We propose a reinforcement learning approach inspired by the mechanisms and dynamics of biological synapses. The network weights undergo spontaneous fluctuations, and a reward signal modulates the centre and amplitude of fluctuations to converge to a desired network behaviour. We test the new learning rule on a 2D bipedal walking simulation, using a control system that combines a recurrent neural network, a bio-inspired central pattern generator layer and proportional-integral control, and demonstrate the first successful solution to this benchmark task.**

I. INTRODUCTION

Neural networks have been applied to robot control long before the development of deep learning, e.g., the learning of inverse kinematics [1] [2], and bio-inspired central pattern generators (CPGs) [3]. Recent developments demonstrate many application in robotics [4] such as supervised learning for robot vision and reinforcement learning in robot motion control [5][6]. Due to these successes, and the ability of deep learning to fit "arbitrary" functions, we might expect the approach should also be applicable for motor learning. However, the neural control of motor systems is essentially dynamic, which can pose problems. For example, the learning rules we have for deep learning are yet not suitable for learning of neural networks with CPGs, which are undifferentiable. Yet CPGs are found widely in animals [7] and have been productively adopted for robot motion control. Although some reinforcement learning models include both CPGs and neural networks [8], the CPG is placed between the neural network and the physical model of the robot and is treated as part of the physical model, so that the neural network is differentiable.

Obviously, animals are able to learn with undifferentiable neural circuits. If such learning rules could be introduced to robot learning, there will be more options for the reinforcement learning of robot motion control. In particular, a bridge can be built between learning methods and previous research into neural network dynamics, such as CPGs, for robot control. The learning rule proposed in this paper is aimed at learning of continuous motion in an single robot using neural network architectures that are beyond the scope of backpropagation methods.

In most neural network models, both the activation functions (mimicking the processing in the soma of a neuron) and weights (mimicking input synapses) are static. Although some of the parameters are updated during training, they are time invariant during calculations. However, biological neural networks are always dynamic [9], even without learning. Neural spiking codes information in a complex hybrid of discrete and analog electrical dynamics. When the information passes between neurons through synapses, neurotransmitter is released from the pre-synaptic regions and received by receptors in the post-synaptic regions. Both regions have inherent dynamics that alter the effective synaptic strengths [10].

The model presented here is inspired by the dynamics in post-synaptic regions, which includes the motion of neurotransmitter receptors. A receptor can be transported between different regions by thermodynamics or active transportation by the neuron [11]. As receptors contribute differently to synaptic strengths (or weights) according to their locations, the transportation causes spontaneous fluctuations of the weights [12]. Because the dynamics of transportation includes random and chaotic processes, the weights are not accurately predictable and not phase-locked. Hence, weights in a real neural circuit always explore adjacent values [13]. Here we propose that if the exploration is controllable, the neural network weights could be optimised. Specifically, exploration should become wider with punishment but narrower with reward; and the centre of the exploration should move towards values that coincide with rewards. Fig 1 shows the concept with an example.

In the following sections we first introduce the proposed learning mechanism, and then describe its application to a complex, hybrid neural circuit to control a bipedal walking simulation.

II. MODELS AND METHODS

A. Learning rule

The neural network learning rule consists of 2 parts: a mechanism to produce spontaneous synapse dynamics that result in exploration of the weight space; and an updating rule to control these dynamics according to the rewards obtained. In other work [14] we have considered a biologically plausible model of the synaptic receptor transport mechanisms

*This work was supported by FP7 FET-Open project 'Minimal'.

[1] Tianqi Wei is a PhD candidate in the Insect Robotics Group, Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh, Edinburgh, United Kingdom and the Stokes Research Group, the Institute for Integrated Micro and Nano Systems, School of Engineering, University of Edinburgh, Edinburgh, United Kingdom Tianqi-Wei@outlook.com

[2] Barbara Webb is Professor of Biorobotics and Director of the Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh, Edinburgh, United Kingdom B.Webb@ed.ac.uk

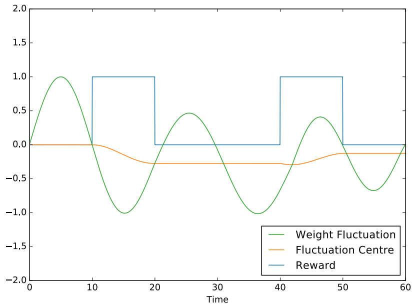


Fig. 1. Principle of the learning rule. A synaptic weight (green) fluctuates around a centre (red). When reward is received (blue) the centre is shifted in the direction of the ongoing fluctuation. The fluctuation amplitude decreases with positive reward for convergence of learning.

that could produce suitable dynamics, but here we present a simple version that is sufficient to provide the following properties:

- Spontaneous fluctuation should be free from the direct effect of information the synapse conveys, so that the input information does not limit the exploration in the synapse;
- Phases of the fluctuations in different synapses should not be locked, to avoid periodic exploration and instead help to sample the weight space densely;
- The fluctuation should be locally symmetric to be resistant to random biases, so that only rewards that correlate to the weight explorations contribute to the learning; and
- The periods of the fluctuations should be much longer than the periods of the learning objectives, such that when the new weights cause an effect and the reward arrives later, the weights should still be near the region that produced a reward.

1) Spontaneous Dynamics: Weight fluctuation is based on a sinusoid function with variable periods. For a single synapse:

$$W(t) = A\sin(2\pi\frac{t-\sum_0^i T_i}{T_i}) + C \quad \text{if}\, T_i < t < T_{i+1} \tag{1}$$

where t is time, A the amplitude of fluctuation, T_i the ith period, and C is the centre of fluctuation.

When a fluctuation crosses its centre from a specific direction, a new period is calculated by a Gaussian process:

$$T_i \sim \mathcal{N}(\mu, \sigma^2) \tag{2}$$

where μ is the centre of the distribution periods and the σ^2 the variance.

The fluctuations of weights in different synapses are independent. Hence, the weight space can be well explored. Fig 2 and Fig 3 show the exploration driven by the fluctuation in 2D and 3D weight spaces respectively.

2) Control of the dynamics: In this approach, learning consists in controlling the spontaneous dynamics according to the rewards generated. Given the use of a sinusoid function as the basis of the dynamics, we want to modulate two variables as a function of the reward (which could be positive or negative): the centre of fluctuation C which changes the average weight of the synapse; and the fluctuation amplitude A which balances exploration and convergence.

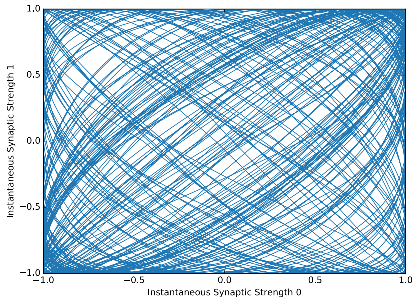


Fig. 2. Exploration Trajectory of 2 synapses

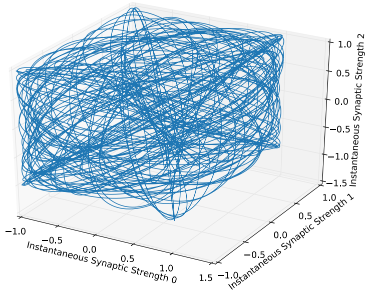


Fig. 3. Exploration Trajectory of 3 synapses

The fluctuation centre C is updated according to the reward and the current value of the fluctuating weight:

$$\dot{C} = \alpha(W(t) - C)R(t) \tag{3}$$

where α is learning rate, (here $1.2 \times 10^{-5}\mathrm{ms}^{-1}$), R the reward. When the reward is positive, the fluctuation centre shifts in the same direction as the on-going fluctuation. If the reward is negative, the fluctuation centre shifts in the opposite direction.

The fluctuation amplitude A, which is positive, is updated according to the reward only:

$$\dot{A} = -\beta R(t) \tag{4}$$

where β is the convergence rate (here $1 \times 10^{-9}\mathrm{ms}^{-1}$). When the reward is positive, the fluctuation converges; when the reward is negative, the fluctuation expands to explore a wider space.

B. Experiment

1) Simulation Environment: The experiment is based on the continuous robot motion control tasks BipedalWalker-v2 and BipedalWalkerHardcore-v2 available within the reinforcement learning environment OpenAI Gym [15]. The first task is a side-scrolling video game style environment with a

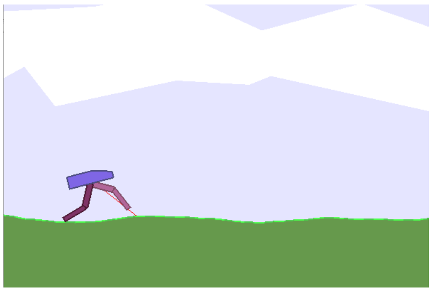


Fig. 4. BipedalWalker-v2 screenshot

2D bipedal robot moving on terrain with small slopes (Fig 4), the second task is the same type of robot but on terrain with stairs, boxes and trenches. OpenAI Gym provides the control API to the robot, which is torque that is applied to 4 joints of the robot (the values feed to the API are called Actions). The API also provides step by step states of the robot and a 10-point Lidar input, and reward values according to the robot's motions. In the experiments, we adopted the original reward provided by the API, which favours the robot's moving forward, keeping its head straight and decreasing the motors' torque. The reward is -100 when the robot falls down. The criterion of solving the task is getting average rewards of 300 over 100 consecutive trials.

For the details of the robot and tasks, please see `https://github.com/openai/gym/wiki/BipedalWalker-v2`, `https://gym.openai.com/envs/BipedalWalker-v2/` and `https://gym.openai.com/envs/BipedalWalkerHardcore-v2/`.

2) Control system: The architecture of the control system consists of a recurrent neural network (RNN) with multiple layers including FitzHugh-Nagumo Oscillators (Fig 5). There are two types of connections between layers as shown in Fig 5: buses of one-to-one connections with fixed weights; or full all-to-all connectivity where the learning rule is applied. Each neuron has one weighted sum input and one output, but different activation functions are used in different layers. Exceptionally, a Fitzhugh-Nagumo Oscillator has two states and we use both of them as outputs.

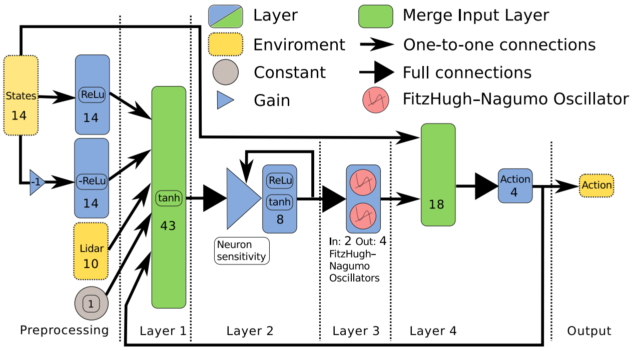


Fig. 5. The architecture of the control network (see text)

There are 5 major parts in the architecture: preprocessing and Layers 1 to 4.

In the preprocessing, each of the robot's 14 states provides two inputs: $[value, 0]$ for positive values and $[0, -value]$ for negative values. These are combined with ten Lidar values, a constant and a feedback signal of the 4 output actions. They are individually normalized according to their possible range and fed into Layer 1. Layer 1 thus has 43 neurons and uses Hyperbolic Tangent as the activation function.

The outputs of Layer 1 are fed into Layer 2 with full connectivity. The connections have uniformly distributed $U(-0.1, 0.1)$ random initial weights and are changed by the learning rule. Layer 2 has 8 neurons and serially uses Hyperbolic Tangent and Rectified Linear Unit as the activation function. To let the neurons in this layer remain sensitive to input changes, it has a slow adaptive gain to control the neuron sensitivity:

$$\dot{g_j} = (0.3 - |o_j|)\gamma \tag{5}$$

$$p_j(t) = g_j \sum o_i(t) W_{ij} \tag{6}$$

where i is the index of neurons in the Layer 1, j the index of neurons in the Layer 2, g the neuron sensitivity, γ the update rate, which is $1 \times 10^{-6}\text{ms}^{-1}$, p the input to Layer 2, o the output of Layer 1, W_{ij} the weights from neurons i in Layer 1 to neuron j in Layer 2 (please note we use W instead of w for weights to avoid confusing with the state w in FitzHugh-Nagumo oscillators).

The outputs of Layer 2 are fed into Layer 3 with full connectivity. The connections have uniformly distributed $U(-0.1, 0.1)$ random initial weights and are changed by the learning rule. The 2 neurons in this Layer are FitzHugh-Nagumo oscillators [16], a simplified model of spiking neuron dynamics, widely applied in CPGs for robot control [3]. However, unlike a typical CPG, the 2 neurons are not coupled but get input from Layer 2 individually, hence their phases are adjusted individually by the upstream layers. In our model, the dynamics are scaled to an appropriate period by τ to fit the period of the walk.

$$\tau\dot{v} = v - v^3 - w + I \tag{7}$$

$$\tau\dot{w} = a(bv - cw) \tag{8}$$

where τ is the time constant for scaling, which is 0.02; w and v are 2 states; I is the input; a, b, and c are constants, among them $a = 0.08$, $b = 0.2$ and $c = 0.8$. v and w are used as outputs to the next layer. Fig 6 shows the dynamics of the oscillator with increasing input from -2 to 2, and Fig 7 shows the phase portrait of the oscillator. With low or high input, the oscillator is stable; with input from about -0.6 to 1.1, the oscillator is unstable.

The outputs of Layer 3 are combined with the 14 states of the robot and fed into Layer 4 with full connectivity and application of the learning rule. Layer 4 has 4 neurons without activation functions (i.e. output is just weighted sum of the input). Their outputs are used as torque of the 4 joints of the robot respectively. The initial weights for the

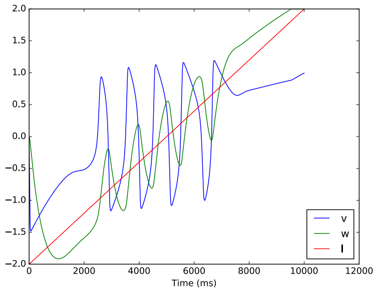


Fig. 6. A FitzHugh-Nagumo oscillator with increasing input

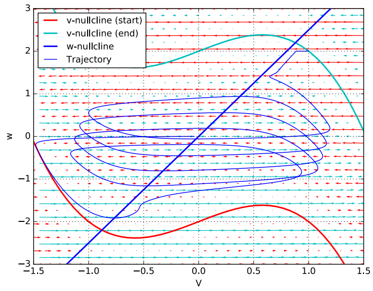


Fig. 7. Phase Portait of the FitzHugh-Nagumo oscillator

layer use the state inputs v and $-w$ to control knee and hip respectively, and take the joint speed and joint angle as the feedback for PI control. The weight values do not need to be carefully selected but depend on the learning rule for tuning. The 4 outputs of Layer 4 also feedback as inputs to Layer 1.

III. Results

The control system is trained with the learning rule on a computer with Intel®Core$^{\text{TM}}$i5-5200U CPU. It took about 11 hours wall clock time, corresponding to 47956 Episodes to solve the BipedalWalker-v2 (see the associated video `https://youtu.be/B7mLVY1NKgI`). We note that according to the official Leaderboard of the task (`https://github.com/openai/gym/wiki/Leaderboard`) on Feb. 27, 2018, no previous solution has been obtained. The source code of the model and experiment are available online: `https://github.com/InsectRobotics/DynamicSynapseSimplifiedPublic.git`.

As shown in Fig 8, the average episode reward had a quick increase from -100 to 200 during episodes 2000 to 14000, then gradually reached up to more than 300 after Episode 47956. As getting average rewards of 300 over 100 consecutive trials is the threshold of solving the task, our approach solved the task. Fig 9 shows a Poincare Map of the exploration, which gradually converges to smaller space and became denser. With negative reward more than positive reward, the fluctuation amplitude increases for the first 30 hours of simulated time, which leads to broader exploration

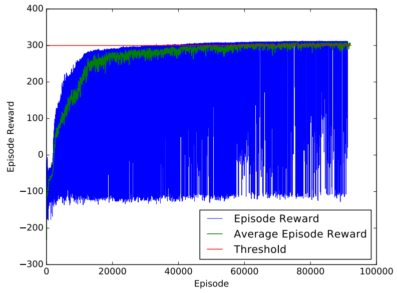


Fig. 8. Episode reward and its average in every 100 episodes. Success on the task is defined as an average above 300.

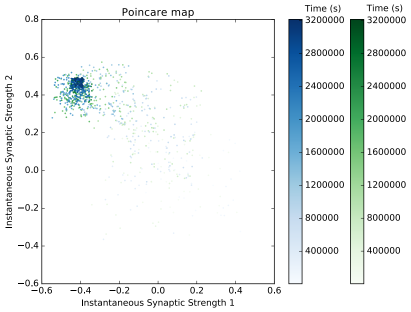


Fig. 9. A Poincare Map of the exploration. The points are intersections of the exploration trajectory and a section of the weight space. Two colours mark the side the trajectory came from, and the gradients of the colours indicate the time of intersection.

of weights. Then it gradually decreases approaching zero, which leads to convergence of the learning. As shown in Fig 10, the weight fluctuations generally explore wider around their centres at the beginning than later. The fluctuation centres moved to optimized values after the training.

The control system and learning rule were also tested on the BipedalWalkerHardcore-v2. The robot learnt to walk up and down stairs, stride over small boxes but was not successful in striding over large boxes and trenches. Striding over large boxes requires the robot to jump, so the control system might not able to support the two distinguished gaits. Striding over trenches requires the robot to control the feet to

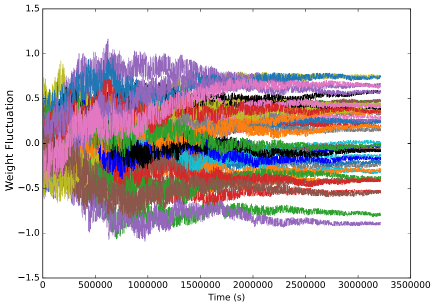


Fig. 10. Fluctuation of the weights: the amplitude and centre of fluctuation is altered by reward and converges on a solution.

correct footholds accurately, so the control system is either unable to accurately control the feet to a specific point or unable to find the correct footholds. Overcoming these limitations may require a change to the architecture of the control system.

We additionally tested whether successful learning can occur if the learning rule is only ‘attractive’, that is, the fluctuation centre is shifted towards the current value of the fluctuation when positive reward is received, but there is no ‘repulsion’, i.e., shifting away from the current value with negative reward. We also examined the contribution of the CPG input weight adaptation, using either no adaptation, linear adaption or non-linear adaption. The combined results of these two manipulations are summarised in (Figure 11).

Note that (as for other reinforcement learning algorithms), random seeds of the simulator can affect the process of learning (Figure 11 (A) without control of random seeds; (B) and (C) with control of random seeds, which show the difference between different configurations more clearly). We also notice random seeds also affect various learning configurations differently (see figure 11 (B) and (C)). Hence, we did multiple groups of experiments and show some typical results here.

Among all the experiments, those with both repulsive learning and nonlinear CPG adaptation have the highest possibility to solve the tasks, providing the only solution in Figure 11 (B, green line), and the first solution in Figure 11 (C, green line). The experiments lacking either repulsive learning or adaptation have the smallest possibility to solve the tasks. In Figure 11 (B), the configuration lacking both never reached 200 (red line). In Figure 11 (C), the configuration without repulsive learning and with linear adaptation (purple line) failed to solve the task.

Normally, with same neural network configuration and random seeds, the learning rule without repulsive learning works less well than the learning rule with it (see Figure 11 (B) and (C), respectively). A possible reason is that repulsive learning provides disturbances to push the exploration centre away from the region of weight space that leads to negative rewards. When the weight fluctuation reaches the region of weight space that leads to positive rewards, the attractive learning provides an attractor to higher reward region. Without repulsive learning, if the system under training initialised in a punishment region and cannot reach the reward region with fluctuation, the system is not able to learn.

IV. Discussion

Inspired by the dynamics of biological synapses, a new learning rule is developed. With this learning rule, we trained a hybrid control system for a bipedal 2D walker traversing terrain with small-slopes, stairs and small boxes. The architecture of the control system is different from typical reinforcement learning using neural networks that it includes not only feed-forward connections, but also feedback connections in different levels, a CPG, and a layer that work as a complex PI control. In summary, it is a mix of neural networks and classical robot control that cannot be optimised with backpropagation. Our learning mechanism instead uses spontaneous weight fluctuations which are modulated by reward to converge to the region of state space that maximises reward. We show that this can produce successful robot control.

TABLE I
Highest average scores reported

Algorithm	Highest average score	Window size	Source
NEAT	54	50	[17]
NES	-23	50	[17]
CMA	-75	50	[17]
P3O	161	50	[17]
CA3C	129	50	[17]
D3PG	90	50	[17]
PPO	251	100	[18]
PPO(8 actors)	221	100	[18]
PPO-ER	270	100	[18]
PPO-ER(8 actors)	285	100	[18]
TRPO	238	100	[18]

Our approach exceeds the performance of previous attempts at solving the bipedalWalker-v2 (see Table I). Methods that have been explored include deep reinforcement learning methods such as Continuous Asynchronous Advantage Actor-Critic, Parallelized Proximal Policy Optimization, and Distributed Deep Deterministic Policy Gradient; and evolutionary methods such as Covariance Matrix Adaptation Evolution Strategy, NeuroEvolution of Augmenting Topologies, and Natural Evolution Strategy[17]. Our approach also exceeds the performance of the proximal policy optimization algorithm with multi-batch experience replay scheme [18]. We believe there are three main reasons for the improvement in performance:

- Typical neural networks and existing robot control approaches cannot be optimised using the same rule continuously, but this is possible with our learning rule, thus our architecture can take advantage of existing robot control approaches (such as CPG and PI control) in a hybrid network.
- The sampling happens in the parameter space instead of the action space, which decreases the complexity of the optimized generated random numbers, which become a fixed-length vectors instead of unfixed-length sequences.
- For the same reason as above, the output actions are continuous and smooth during exploration, while previous approaches typically use random number generators to explore the action space, which introduces noise and impacts the quality of actions.

The main existing alternative approach to this kind of problem is based on genetic algorithms. However, they require transient changes of variables, which is less suitable than our learning rule when applied to cases in which a robot is hard to reset, such as an actual physical robot.

The gait produced by our learning method for the Bipedal-Walker is successful for locomotion, but does not resemble a typical bipedal walking gait, i.e., the legs do not swing past

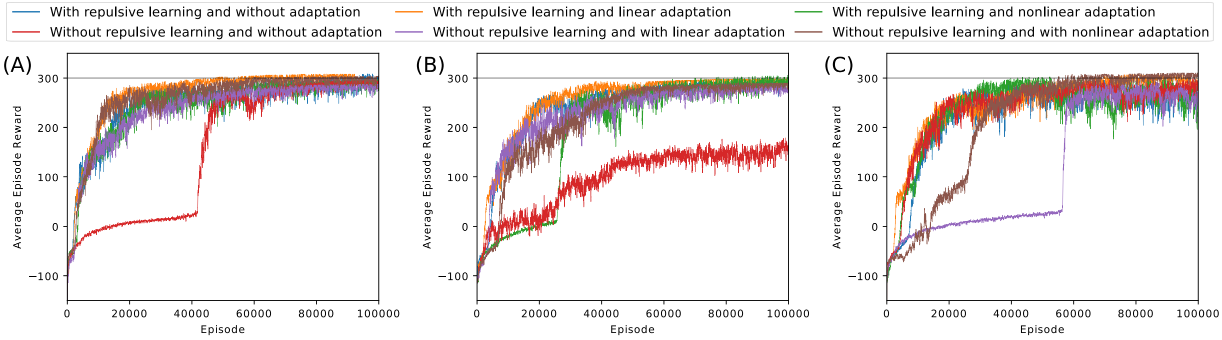


Fig. 11. Comparison between trainings with/without repulsive learning or CPG input weight adaptation. (A) Without control of random seeds. (B) and (C) With control of random seeds, each group of experiments have the same random seed.

each other in alternation. We believe this is due to the fact that the robot is planar, and as such does not need to maintain lateral balance. Keeping one leg always in front and the other at the rear is the most stable configuration for forward-backward balance, particularly in early stages, and the robot then optimises its gait for this configuration. The same gait is often found as a solution in this task. Alternatively, the observed movement can be compared to the bounding gait naturally exhibited by some quadrapeds, where the two front and two hind legs are moving in unison.

Our learning rule can be applied to more complex and dynamical neural networks than backpropagation, because it can be applied to a wider range of architectures and dynamics. This could be particularly valuable in scaling to more complex tasks, as the curse of dimensionality might be addressed by introducing richer internal structures, e.g., internal rewards that structure learning towards distant final rewards. Other applications system identification, dynamic model based robot control and so on. Moreover, it can be used in hybrid systems that combine classical robot control approaches and neural networks, thus facilitating the introduction of neural networks into robot control tasks.

References

[1] Guo and Cherkassky, “A solution to the inverse kinematic problem in robotics using neural network processing,” in *International Joint Conference on Neural Networks*. IEEE, 1989, pp. 299–304 vol.2. [Online]. Available: http://ieeexplore.ieee.org/document/118714/

[2] S. Tejomurtula and S. Kak, “Inverse kinematics in robotics using neural networks,” *Information Sciences*, vol. 116, no. 2-4, pp. 147–164, jan 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025598100981

[3] A. J. Ijspeert, “Central pattern generators for locomotion control in animals and robots: A review,” *Neural Networks*, vol. 21, no. 4, pp. 642–653, 2008.

[4] H. A. Pierson and M. S. Gashler, “Deep learning in robotics: a review of recent research,” *Advanced Robotics*, vol. 31, no. 16, pp. 821–835, 2017. [Online]. Available: http://doi.org/10.1080/01691864.2017.1365009

[5] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep Reinforcement Learning for Robotic Manipulation with Asynchronous Off-Policy Updates,” oct 2016. [Online]. Available: http://arxiv.org/abs/1610.00633

[6] D. Ribeiro, A. Mateus, P. Miraldo, and J. C. Nascimento, “A real-time Deep Learning pedestrian detector for robot navigation,” in *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. IEEE, apr 2017, pp. 165–171. [Online]. Available: http://ieeexplore.ieee.org/document/7964070/

[7] I. A. Rybak, K. J. Dougherty, and N. A. Shevtsova, “Organization of the Mammalian Locomotor CPG: Review of Computational Model and Circuit Architectures Based on Genetically Identified Spinal Interneurons,” *eNeuro*, vol. 2, no. 5, 2015. [Online]. Available: http://eneuro.sfn.org/cgi/doi/10.1523/ENEURO.0069-15.2015

[8] T. Mori, Y. Nakamura, M.-a. Sato, and S. Ishii, “Reinforcement Learning for a CPG-driven Biped Robot,” *Aaai 2004*, pp. 623–630, 2004.

[9] E. M. Izhikevich, *Dynamical Systems in Neuroscience*. The MIT Press, 2007, vol. 25, no. 1. [Online]. Available: http://www.izhikevich.org/publications/dsn.pdf

[10] D. Choquet and A. Triller, “The dynamic synapse,” *Neuron*, vol. 80, no. 3, pp. 691–703, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.neuron.2013.10.013

[11] K. Czondor, M. Mondin, M. Garcia, M. Heine, R. Frischknecht, D. Choquet, J.-B. Sibarita, and O. R. Thoumine, “Unified quantitative model of AMPA receptor trafficking at synapses,” *Proceedings of the National Academy of Sciences*, vol. 109, no. 9, pp. 3522–3527, 2012. [Online]. Available: http://www.pnas.org/cgi/doi/10.1073/pnas.1109818109

[12] L. a. Cingolani and Y. Goda, “Actin in action: the interplay between the actin cytoskeleton and synaptic efficacy,” *Nature Reviews Neuroscience*, vol. 9, no. 5, pp. 344–356, 2008. [Online]. Available: http://www.nature.com/doifinder/10.1038/nrn2373

[13] A. Minerbi, R. Kahana, L. Goldfeld, M. Kaufman, S. Marom, and E. Noam, “Long-Term Relationships between Synaptic Tenacity , Synaptic Remodeling , and Network Activity,” *PLOS Biology*, vol. 7, no. 6, 2009.

[14] T. Wei and B. Webb, “A model of operant learning based on chaotically varying synaptic strength,” in preparation.

[15] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “Openai gym,” 2016.

[16] R. FitzHugh, “Impulses and Physiological States in Theoretical Models of Nerve Membrane,” *Biophysical Journal*, vol. 1, no. 6, pp. 445–466, 1961. [Online]. Available: http://dx.doi.org/10.1016/S0006-3495(61)86902-6

[17] S. Zhang and O. R. Zaiane, “Comparing Deep Reinforcement Learning and Evolutionary Methods in Continuous Control,” no. Williams 1992, 2017. [Online]. Available: http://arxiv.org/abs/1712.00006

[18] S. Han and Y. Sung, “Multi-Batch Experience Replay for Fast Convergence of Continuous Action Control,” vol. 34141, pp. 1–10, 2017. [Online]. Available: http://arxiv.org/abs/1710.04423

4.3 Further discussion

The above paper presents a learning rule abstracted from the synaptic plasticity model proposed in chapter 2. The learning rule trained a neural network with internal dynamics and recurrent connections, which is hard to be trained with backpropagation because of vanishing / exploding gradients during error back-propagation through time (Sato, 1990).

The result supports that the learning rule can be applied to reinforcement learning (RL) with neural networks for robot control tasks. It is the first reinforcement learning model that solved the benchmark test, which is hard since the robot should perform energy efficiently without any faulty action, such as falling down, in 100 successive tests. The perfect performance in the task requires fine-tuning of the neural network by RL to minimise the possibility of entering faulty actions. The learning rule perform better than other algorithms at the time of writing this thesis). Although the resulting gait is different from human, it can be interpreted as the bounding gait, as described in the discussion of the inserted paper.

4.3.1 Parameter Space Exploration and Action Space Exploration

Parameter space exploration (PSE), a key characteristic of the learning rule, contributes to the outstanding performance by improving the exploration efficiency. PSE, as discussed in the above paper, changes the mapping from the sensory inputs to the actions, generating actions correlated to the sensory input. It is different from action space exploration (ASE), by which new actions are generated without sensory information.

As the right actions always correlate to the sensory inputs, PSE can have a higher possibility than ASE to find the correct action if the neural network is adequate for the task. ASE, however, can generate random actions that the neural network is never able to output, which are unnecessary explorations. In other words, for continuous tasks, parameter space can be smaller and less complicated than the action space, thus parameter space exploration can be more effective than action space exploration. Considering a neural network with N parameters, which controls a robot with D degrees of freedoms, for action with time length T and resolution R per unit time, the complexity of parameter space is N, and the complexity of the action space is D^{TR}. Hence, the complexity of action space is sensitive to the time length and the resolution. Although, in general, $N >> D$, with sufficiently large T and R, the action space can be more complicated than the parameter space. For the learning tasks targeting at fine action tuning, such as the benchmark task, which needs high resolution, parameter exploration is more feasible than action space exploration.

This learning rule has similarities with the parameter space noise for action exploration (Plappert et al., 2018), which is mentioned in section 4.1.2.2, in that they are explorations in parameter space. However, they are different in that the former explores in the region neighbouring the point the neural network is at, while the latter does not constrain the region of exploration. The exploration in the neighbour region has similar advantages with the trust region in parameter updating (Schulman et al., 2017; Heess et al., 2017), which improves the learning stability by limiting the step size optimisation. This learning rule is also different to the parameter space noise for action exploration, in that the latter needs an extra step to map the action back to the parameters by backpropagation, while the former directly updates the centres of parameters according to the instantaneous explorations of parameters. As the backpropagation can lead to parameters different from the parameters that generate the action, this learning rule is more robust in learning.

PSE generates actions with the coupled dynamics between the neural network robot, so the generated actions are correlated to the dynamics. For continuous systems with non-linear dynamics, chaos can exist. With chaos, the distribution of actions from the dynamics is fractal. PSE generates actions correlated to the dynamics, thus the actions can be fractal, too, and feasible to be learned by the neural networks. As a comparison, ASE is based on a stochastic process, which does not have the corresponding fractal structure which increases the difficulty in fitting the mapping with the neural network.

4.3.2 Comparison of this work with some bipedal robot control models

Although the work presented in this chapter proves that the proposed learning rule can be applied to robot control with dynamic neural networks, it is nevertheless to compare the neural network with other bipedal robot control models, especially those for learning to control robots who have similar configurations.

The Runbot Geng et al. (2006a,b) is a physical bipedal robot that has a similar configuration with the bipedal walker of the benchmark task. The robot is controlled by a biologically inspired sensor- and motor neuron models, which form a neural network that drives the motors with reflex. It is different from the dynamic neural network proposed in this chapter in that it does not have any CPG and recurrent connections. There is neither an algorithm for generating gait trajectory, nor position or speed control for tracking a trajectory. Hence, the controller only relies on reflex to control the robot, by which the motor output, action, sensory input, and the neural network forming a control loop. There is no internal feedback to pass downstream information to upstream neurons, thus any failure in the only loop can cause a fault. The number of neurons in the Runbot is 18 (Geng et al., 2006b) or 28 (Geng et al., 2006a), less than the neurons in the dynamic neural network. The connections between neurons in the Runbot is pre-designed and there is no full connection between layers, hence the learning complexity is reduced compared to the dynamic neural network. The learning rule of the Runbot is Isotropic Sequence Order Learning (ISO) (Porr and Wörgötter, 2003; Porr et al., 2003), which is based on a differential Hebb rule, learning to use the predictor sensory input to minimise the disturbance of the reflex sensory input to the outputs. This learning rule is different from the dynamic synapse learning rule in that the former does not need reward whilst the latter does, and that the former calculates the correlation between the inputs and the derivative of the output while the latter calculates the correlation between synaptic strength and rewards.

Nakanishi et al. (2004b) proposed a framework for learning biped locomotion with a central pattern generator, whose dynamic is decorated by locally weighted learning (Schaal and Atkeson, 1998) to be used as Rhythmic dynamical movement primitives. This framework is similar to the dynamic neural network in that they use CPGs. However, the “CPG” the former uses are very artificial, in that the learning of the CPG is done by adding numbers of local models to the CPG. As a comparison, a biologically plausible CPG does not change its output by adding local models, but by changing its parameters or inputs. The dynamic synapse learning rule can enable the learning of CPGs in a similar way, as shown in chapter 2 and this chapter.

Chapter 5

Soft Maggot Robot

5.1 Background

5.1.1 Motor system and motion of larvae

5.1.1.1 Motion of larvae

Motions of *Drosophila* larvae include forward, backwards, sweep, turn, and roll. The motion that has been the focus of most research is forward motion. Heckscher et al. (2012) studied characters of larvae crawling forward and backwards and tried to link muscle contraction patterns to the motions. Motion of crawling forward can be divided into 2 phases: (1) visceral piston phase, when tail, head, and viscera pushed forward by gut suspension muscles, but body walls of middle body segments keep the position; and (2) wave phase, tip of head hook substrate, body wall peristalsis forward. Figure 5.1 shows the motions of the body wall and visceral during crawling forward.

Figure 5.2 shows similar graphs for backward motion. It is based on a video by Heckscher in 2013 (https://www.youtube.com/watch?v=S6TOJJeOtoY). The original pictures are captured from the video. Muscle contractions, body wall without relative displacement with the substrate, and position of the gut are signed.

5.1.1.2 Muscle of larvae

Muscles of *Drosophila* larvae are in three kinds of orientations: dorso-ventral, anterior-posterior and oblique. Antero-posterior muscles are located more inner than Dorso-ventral muscles. During locomotion, fluids in a larval body facilitate deformation of the body. Lahiri et al. (2011) studied muscle contraction during larval locomotion.

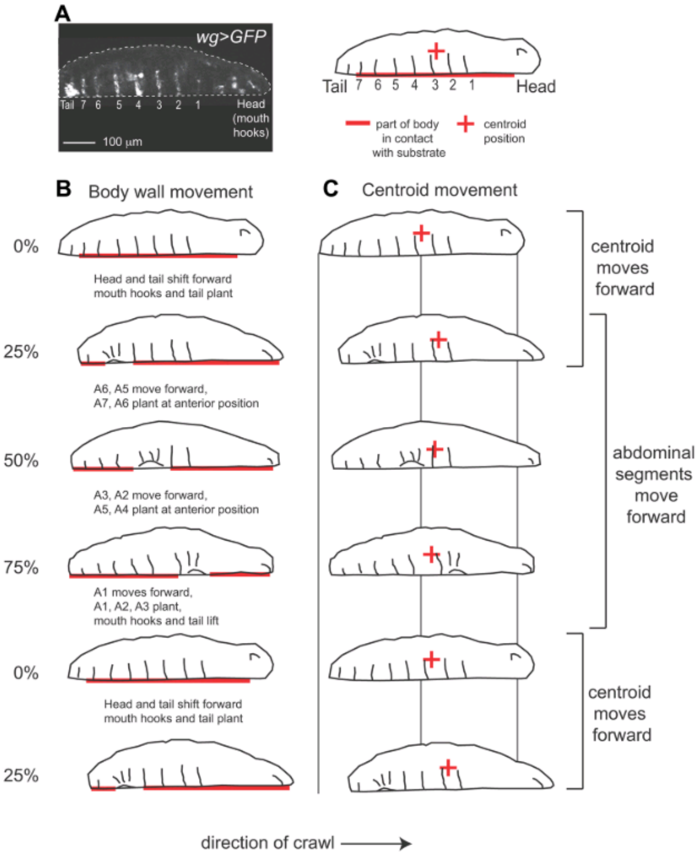

A
wg>GFP
Tail 7 6 5 4 3 2 1
Head (mouth hooks)
100 µm
Tail 7 6 5 4 3 2 1 Head
part of body in contact with substrate
centroid position
B Body wall movement
C Centroid movement
0%
Head and tail shift forward mouth hooks and tail plant
25%
A6, A5 move forward, A7, A6 plant at anterior position
50%
A3, A2 move forward, A5, A4 plant at anterior position
75%
A1 moves forward, A1, A2, A3 plant, mouth hooks and tail lift
0%
Head and tail shift forward mouth hooks and tail plant
25%
centroid moves forward
abdominal segments move forward
centroid moves forward
direction of crawl

Figure 5.1: Motions of body wall and visceral during crawling forward (schematic diagrams). Adopt from Heckscher et al. (2012), with permission.

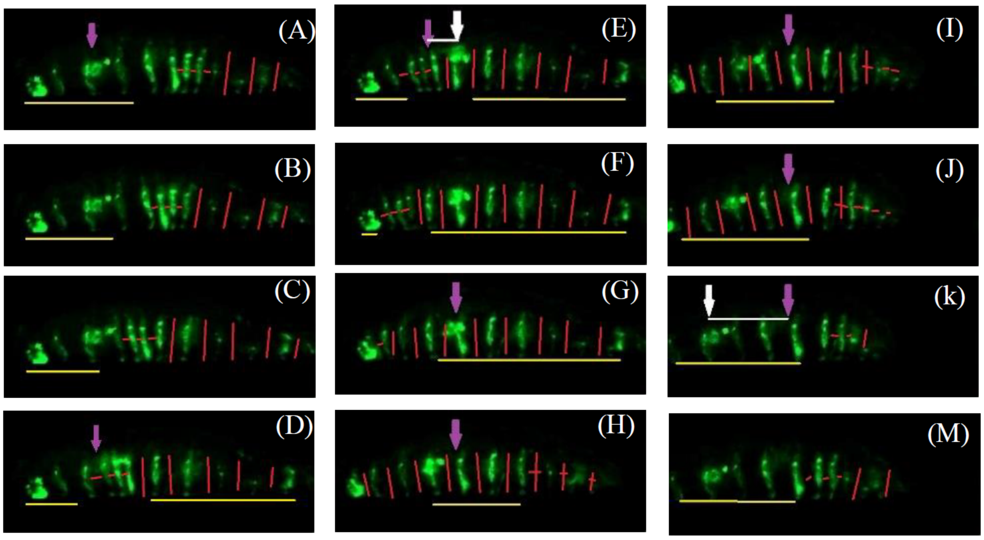


Figure 5.2: Motions of the body wall and visceral during crawling backwards (GFP photos). Amended from Heckscher (2013). Short red horizontal lines mark contracting body segment, long red vertical lines mark relaxed body segment, yellow horizontal lines mark the body walls without relative displacement with the substrate, violet arrows are the original position of the gut, and white arrows a new position of the gut. (M) and (A) to (F) are wave phase; and (G) to (K) are visceral piston phase

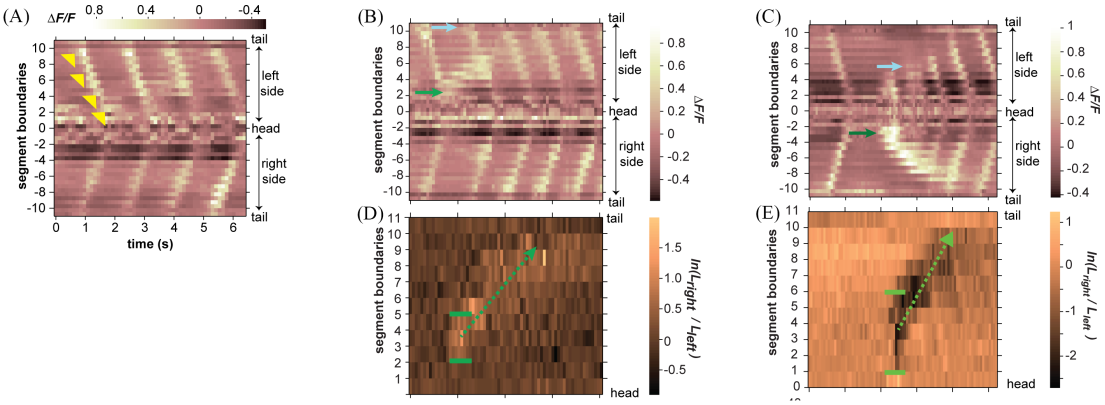


Figure 5.3: Muscle contraction during larval locomotion. In (A), (B) and (C), the middle is head, above the middle are fluorescence on the left side from head to tail, and below the middle are fluorescence on the right side from head to tail. (D) and (E) shows fluorescence on left side subtract fluorescence on the right side. (A) shows fluorescence in crawling forward, (B) and (D) show fluorescence with small head sweeps, (C) and (E) shows fluorescence with large crawling forward. Adopt from Lahiri et al. (2011), with permission.

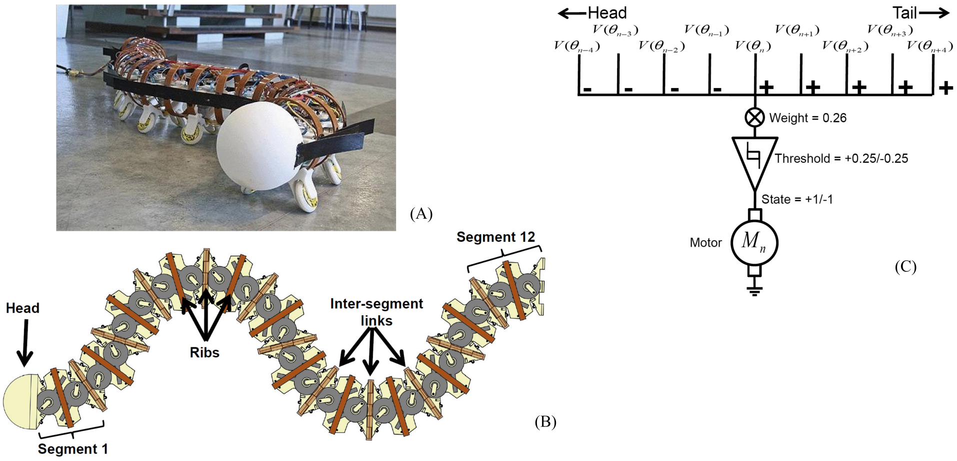


Figure 5.4: A C. elegans inspired robot. Adopt from Boyle et al. (2013), with permission.

They observed transgenic larvae with fluorescent muscle fibres. The fluorescence intensity is relevant to the intensity of contraction. Figure 5.3 shows results of observation. In Figure 5.3 (A), (B) and (C), the middle is head, above middle are fluorescence on the left side from head to tail, and below the middle are fluorescence on the right side from head to tail. Figure 5.3 (D) and (E) shows fluorescence on left side subtract fluorescence on right side. Figure 5.3 (A) shows fluorescence in crawling forward, Figure 5.3 (B) and (D) show fluorescence with small head sweeps, Figure 5.3 (C) and (E) shows fluorescence with large crawling forward. During crawling forward, peristalsis waves that travel from tail to head on two sides symmetrically. During a small head sweep, a peristalsis wave travels from head to tail superpose on crawling forward peristalsis waves. The peristalsis wave travels from head to tail is asymmetric, as Figure 5.3 (D) shows. During a large head sweep, continuous peristalsis waves that travel from tail to head are interrupted. New waves start from the body segments that backward wave have travelled.

5.1.2 Existing worm-like robots

Worm-like Robots, which have serial multi-body segments without no leg in general, can move in the ways similar to *Drosophila* larvae. For example, C. Elegans inspired Robot (Figure 5.4 (A) and (B)) made by Boyle et al. (2013) can locomote and avoid obstacles by bending the body. The control system (Figure 5.4 (C)) of the robot is inspired by the C. Elegans nervous system.

Conradt and Varshavskaya (2003) designed a worm-like robot (Figure 5.5) which is controlled based on Central Pattern Generator. The robot can behave planar horizon-

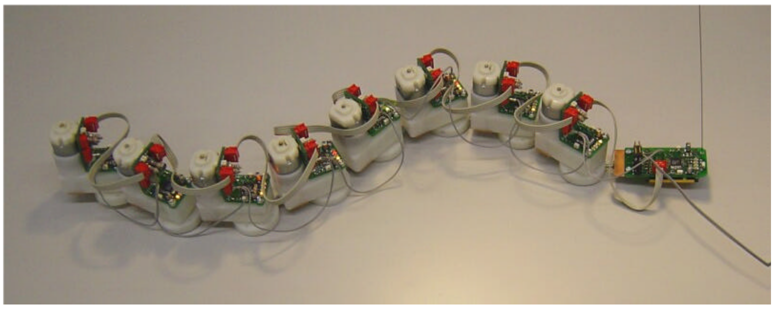


Figure 5.5: The WormBot. Adopt from Conradt and Varshavskaya (2003).

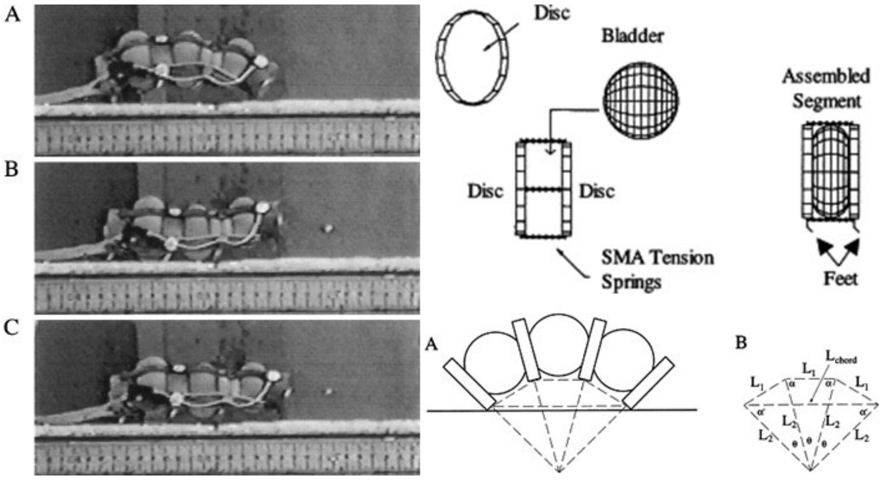


Figure 5.6: The hydrostatic robot. Adopt from Vaidyanathan et al. (2000), with permission.

tal locomotion contacting ground by venter or planar vertical locomotion contacting ground by lateral. The robot is actuated by DC motors. The control method of the robot is a distributed central pattern generator control. Every body segment is provided with a microcontroller Atmel Mega 8 which runs a local CPG and actuate the corresponding motor using PWM. The CPG is biased by current position and torque. Sensor readings are shared by all controllers.

Vaidyanathan et al. (2000) designed a hydrostatic robot with a hydrostatic skeleton that locomotes underwater (Figure 5.6). The robot consisted of three-segment. Fluid-filled bladder and wooden discs are alternately arranged. Four Shape-memory alloy (SMA) tension springs attached to two adjacent wood discs. The robot can crawl at a speed of 0.6 cm/s and capable of locomotion in straight or curved paths. The robot is controlled by four binary controllers, which switched manually.

A similar robot (Figure 5.7) was developed by Menciassi et al. (2006). Its locomotion is inspired by the peristaltic motion of Annelids, and earthworm. Its body consists

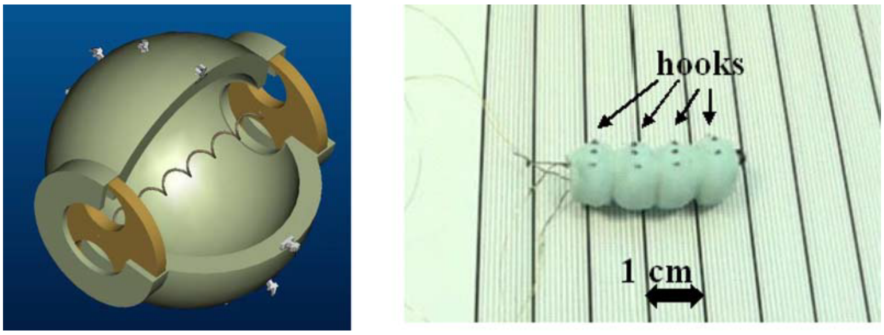


Figure 5.7: The Biomimetic Miniature Robotic Crawler. Adopt from Menciassi et al. (2006), with permission.

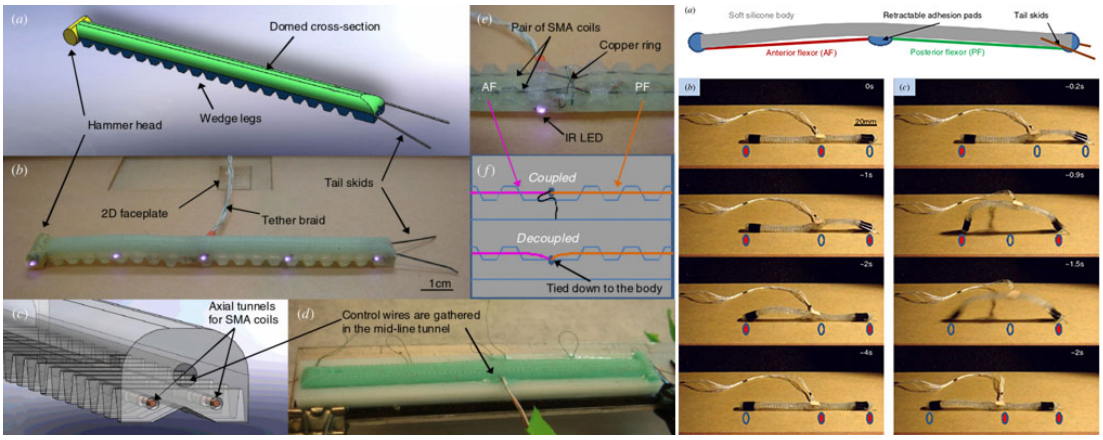


Figure 5.8: The GoQBot. Adopt from Lin et al. (2011), with permission.

of 4 modules, five brass disks and a silicone shell. SMA wire is inside the body. The robot locomotes on the ground, with miniature hooks that offer anisotropic friction coefficients.

Lin et al. (2011) studied an escape repertoire of some caterpillars that curl their body into a wheel and roll away, then designed a robot named GoQBot (Figure 5.8) that able to mimic the unique way of locomotion. The body of GoQBot consists of several kinds of silicone rubbers, in which are two axial tunnels for SMA coils and one tunnel for wires. The robot can speed up to more than 0.5m/s within the first 200ms and reaches 1 G of acceleration within 50ms in rolling movements. The robot is controlled by frequency-modulated stimuli of fixed voltage. For rolling movements, the actuators are controlled by sustained DC pulses for maximum power.

A lot of similar robots have been built, such as works by Akbarzadeh and Kalani (2012); Ávila et al. (2006); Arena et al. (2006); Wang et al. (2008). Therefore, worm-like robots are well developed and can provide a variety of platforms to test motor control and decision making of neuron circuits.

Worm-like or snake-like robots can be divided into two groups by stiffness of material: hard robots and soft robots. They have distinct characteristics in shape, gait, actuator and control system. Soft robots mimic real larvae more accurately than hard robots in the mechanical property of tissues and deformation during motion. As a control system is based on its control object, a more realistic body offers a better platform for a bionic control system.

At the present stage, for most worm-like or snake-like robots, their control strategies are relatively simple, especially for the soft robots. They usually only able to move with pre-preprogrammed motions, and thus lack the ability to learn the environment or adapt to its own body. A more realistic control system inspired by the nervous system may improve the adaptability.

5.1.3 Drives and actuators of worm-like robot

There are a variety of actuators that apply to worm-like robots. If they are classified according to the type of motions, they are mainly of 2 types: linear motion and rotation. If they are classified according to the type of driving methods, they are mainly of 5 types: electromagnetic drive, shape memory drive, chemical drive, hydraulic drive, and pneumatic drive.

5.1.3.1 Electromagnetic Drive

The electromagnetic drive is a relatively mature method when it mainly refers to electric motors. A series of types and control circuits, accurate mathematic models and control algorithm are available.

Servo motors are usually divided into two groups: direct current (DC) motors and alternating current (AC) motors. The most significant difference of structures between them is that DC motors have electric brushes but AC motors not. By closed-loop control with feedback sensors such as photoelectric encoder and Hall sensor, servo motors are capable of outputting accurate velocity and angle.

C. Elegans inspired robot made by Boyle et al. (2013) and the worm-like robot designed by Conradt and Varshavskaya (2003) adopts DC motors as actuators, to drive angle between two segments.

Steering Servo is a type of specific servo motors that integrates angular transducer and a closed-loop feedback circuit which controls its output to be a corresponding

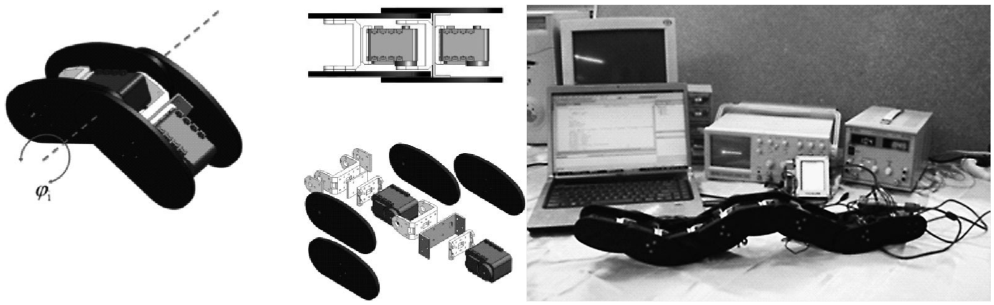


Figure 5.9: FUM Snake-3. Adopt from Akbarzadeh and Kalani (2012), with permission.

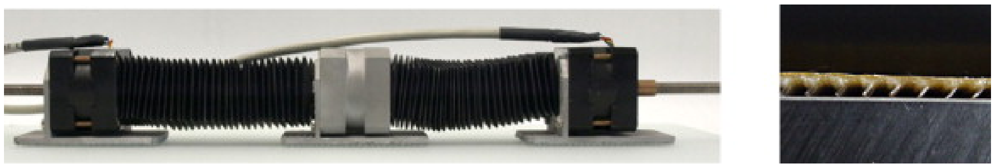


Figure 5.10: An artificial segmented worm. Adopt from Steigenberger and Behn (2011), with permission.

angle for a specific duty ratio input. As its ease of use and integration, steering servos are used in a lot of miniature robots.

The snake-like robot FUM Snake-3 (Figure 5.9) developed by Akbarzadeh and Kalani (2012) uses steering servos as actuators to control angles between every two adjacent segments. The motors could provide a maximum torque of 1.5 Nm and a maximum speed of 360 °/s.

A step motor keeps an angle with constant input and turns when the input currents of different phases switch. It is designed to achieve the functions of servo motors using a low-cost way. As stable positions of a rotor are discrete, step motors are capable of rotating an accurate distance in the absence of the feedback loop. Some step motors work as linear motors of which spindles replace by screw rods.

Steigenberger and Behn (2011) designed an artificial segmented worm, which mainly composes of linear stepper motors (Figure 5.10).

There are also some special approaches to actuate worm-like robot by electromagnetic drives, such as the inchworm mobile robot (Figure 5.11) using electromagnetic linear actuator Lu et al. (2009), and a Bio-Inspired robot named SEMOR (Figure 5.12) using a voice-coil as an actuator Cotroneo et al. (2008).

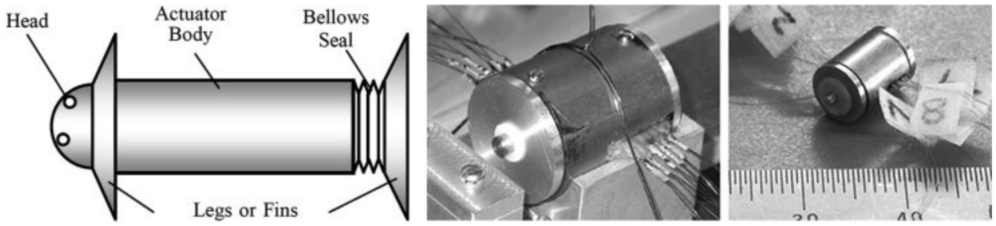


Figure 5.11: The Inchworm mobile robot using electromagnetic linear actuator. Adopt from Lu et al. (2009), with permission.

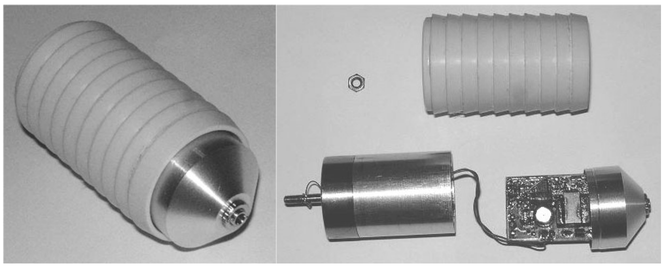


Figure 5.12: The SEMOR. Adopt from Cotroneo et al. (2008), with permission.

5.1.4 Shape memory material drive

Shape Memory drive adopts shape memory materials as actuators. This kind of materials can return from deformed shapes to their original shapes by stimulations, such as temperature (Mohd Jani et al., 2014). Shape memory materials include Shape-memory alloy (SMA) and shape-memory polymer.

Utilise the characteristic, shape memory materials are adopted as a kind of actuators in robots. Robots developed by Lin et al. (2011); Menciassi et al. (2006); Vaidyanathan et al. (2000) use SMA as actuators. Advantages of shape memory material actuators include small size and flexible arrangement, as they usually are made into wires. However, it has shortages include poor frequency response compared with other drive approaches and low energy efficiency, as most or energy is consumed in creating conditions for transition, such as heat up to the transition temperature, then dissipate heat for reverse motion.

5.1.5 Hydraulic drive

The hydraulic mechanism utilises liquid as a medium of the power supply. Hydraulic actuators usually have a high power-to-weight ratio, as hydraulic mediums can transmit high power generated from the prime motor.

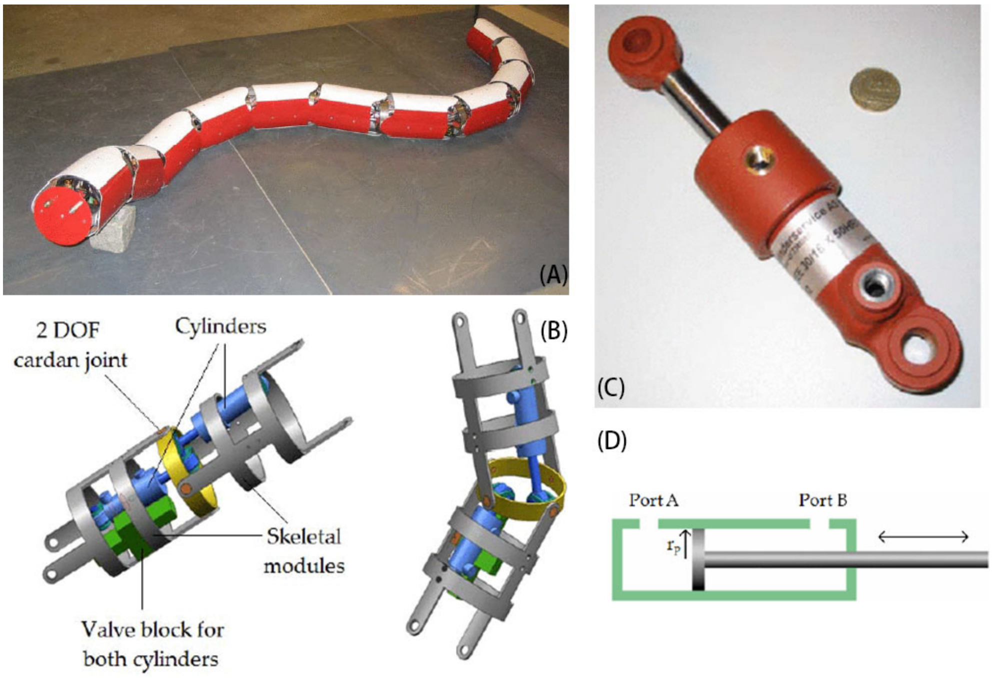


Figure 5.13: The SnakeFighte. Adopt from Liljebäck et al. (2006), with permission.

Liljebäck et al. (2006) design a snake-like robot named SnakeFighter 5.13. The robot is actuated by hydraulic cylinders. Two cylinders actuate adjacent two segments and offer 2 degrees of freedom, yaw and pitch direction, respectively.

5.1.6 Pneumatic drive

Similar to the hydraulic mechanism, the pneumatic mechanism utilises fluid as a medium of the power supply except the fluid is air. Compared with the hydraulic mechanism, a pneumatic mechanism usually has a higher frequency response and lower price, but more difficulties in control, such as nonlinear and delay of air compression.

Pneumatic artificial muscle (PAM) is an essential part of a pneumatic actuator applying to robotics. A variety of pneumatic artificial muscles have been developed during the past decades.

Braided Muscles by Henri (1953) (Figure 5.14) are the most frequently used artificial pneumatic muscles. Braided sleeving covers on an elastic tube or blade. Fibres of Sleeving are specially woven that the angles between the fibres and longitudinal axis are coincident. When the tube is inflated, the diameter of tube increase, the angle of fibres change, and the muscle becomes shorter.

Daerden (1999) developed pleated PAM (Figure 5.14 (B)). Villegas et al. (2012) improve it. The maximum contraction of this muscle was experimentally found to be

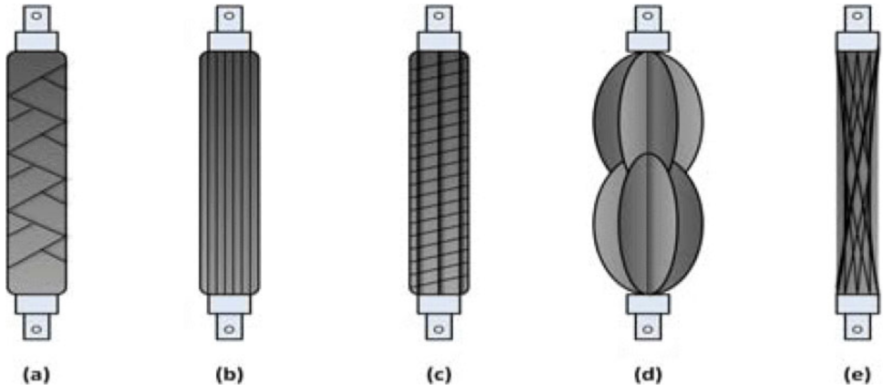


Figure 5.14: Pneumatic muscles. (a) McKibben Muscle/Braided Muscle, (b) Pleated Muscle, (c) Yarlott Netted Muscle, (d) ROMAC Muscle and (e) Paynter Hyperboloid Muscle. Adopt from Kelasidi et al. (2011), with permission.

Figure 5.15: Festo fluidic muscle. From brochure of fluidic muscle DMSP/MAS by Festo. Adopt from `www.festo.com`.

41.5%.

Yarlott Muscle(Figure 5.14 (C)) is described in US patent No. 3645173 (Yariott (1972)). It is composed of a flexible thin-wall shell with strands on latitude and longitude. An advantage of this type of muscle is energy is less consumed in deformation of the chamber.

An Axially Contractable Actuator, which is usually referred to as ROMAC (Figure 5.14 (C)), is composed of membrane covers on a frame built by jointed non-stretchable flexible sticks.

Paynter Hyperboloid Muscle (Paynter, 1988), as shown in Figure 5.14(D), is also composed of braid and membrane chamber, but braid is outside the chamber and arranged on a hyperboloid.

The pneumatic ’Fluidic muscle’ development by Festo (Figure 5.15) is similar to braided muscles. Its contraction is only up to 25% of the nominal length.

There are also several kinds of pneumatic actuator applied in worm-like robots. An inchworm-like microrobot for pipe inspection (Figure 5.16) designed by Lim et al. (2008) utilises resistance of air when it flows through a small hole to actuate the robot

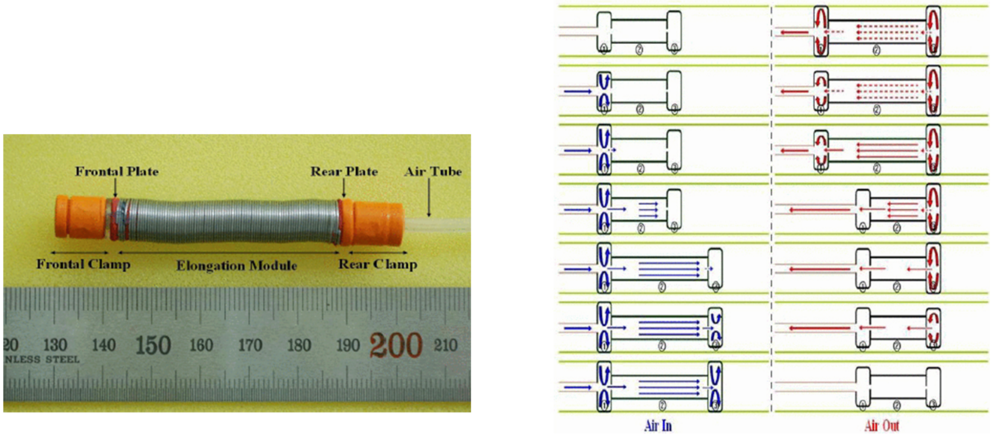


Figure 5.16: An inchworm-like micro robot for pipe inspection. Adopt from Lim et al. (2008), with permission.

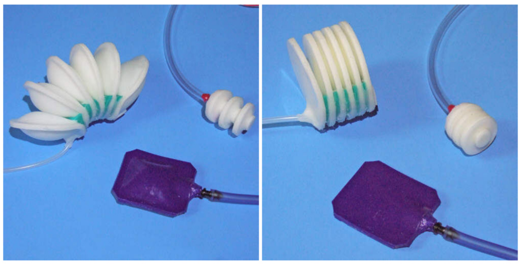


Figure 5.17: Robot Air Muscles made from Oogoo. Adopt from `www.inklesspress.com/robots.htm`

with only one airway tube. The robot is divided into three chambers: head and tail are for contact and fixation, middle for elongation. When pressed air is injected from the tail, the chamber is inflated firstly, then the air inflow to the middle chamber and the body of robot extend, and at the last, air inflow into the head chamber and block in the pipe. When pressure released from the tail, the chamber deflate first, then the body becomes shorter, at the last the head released. By alternately inflated and deflated, the robot moves forward.

The Robot Air Muscles show in Figure 5.17 is made by mikey77 and published on the website `www.inklesspress.com/robots.htm`. The muscle is made from Oogoo, a material made from silicone caulk and gorilla instant glue.

Different drivers or actuators are suitable for different robot bodies. Electromagnetic drives, such as electromotors, have mature bottom layer control method and high accuracy, a hydraulic cylinder can generate huge pressure, and an air cylinder has fast response, but they are only suitable for hard robots. For soft robots, soft hydraulic or pneumatic drives are more appropriate, since they are possible to fit into a soft body with miniature size. More importantly, as this kind of drivers can be designed as artificial muscles which are similar to animals muscles, a robot with them can reproduce realistic motions of animals.

5.2 The soft maggot robot

In this subsection, the design of a soft maggot robot is proposed and detailed. The robot uses integrated pneumatic muscles made from Ecoflex, which is a type of silicone. The pattern of the muscles mimics the muscle pattern of *Drosophila* larvae. An embedded control system is designed for the robot. Experiments show the motions of the robot.

The following paper presents the biological background, design and tests of the robot. The paper is titled "A Soft Pneumatic Maggot Robot" (Wei et al., 2016), published in *The 5th International Conference on Biomimetic and Biohybrid Systems* (Wei et al., 2016). It is about hypothesis 5, result 5, and highlight 5 in Chapter 1. Adam Stokes and Barbara Webb are the co-authors of the paper, who advised on the work and the writing of the paper.

A Soft Pneumatic Maggot Robot

Tianqi Wei[1(✉)], Adam Stokes[2], and Barbara Webb[1]

[1] School of Informatics, Institute of Perception, Action and Behaviour,
University of Edinburgh, Edinburgh EH8 9AB, UK
chitianqilin@163.com, B.Webb@ed.ac.uk

[2] School of Engineering, Scottish Microelectronics Centre, University of Edinburgh,
Edinburgh EH9 3FF, UK
A.a.stokes@ed.ac.uk

Abstract. *Drosophila melanogaster* has been studied to gain insight into relationships between neural circuits and learning behaviour. To test models of their neural circuits, a robot that mimics *D. melanogaster* larvae has been designed. The robot is made from silicone by casting in 3D printed moulds with a pattern simplified from the larval muscle system. The pattern forms air chambers that function as pneumatic muscles to actuate the robot. A pneumatic control system has been designed to enable control of the multiple degrees of freedom. With the flexible body and multiple degrees of freedom, the robot has the potential to resemble motions of *D. melanogaster* larvae, although it remains difficult to obtain accurate control of deformation.

1 Introduction

We have designed a robot to mimic *Drosophila melanogaster* larvae (maggots), as a platform to test and verify their learning and chemotaxis models. *Drosophila* as a model system has a useful balance between relatively small number of neurons yet interestingly complex behaviours [10]. Many genetic techniques, such as GAL4/UAS systems developed by Brand and Perrimon [2], facilitate research on the connectivity and dynamics of the circuits. As a result, a number of necessary components of neural circuits for sensorimotor control and learning are being found and modelled. Currently, the models are tested by comparing between wildtype and genetic mutation lines, or using simulations of neural circuits and comparing output with biological experimental recordings. To test models in a wider environment, more similar to a larva, a physical agent that copies properties of the larval body is important.

Larvae have high degrees of freedom (DOFs) and flexible bodies. As a result, they are able to do delicate and spatially continuous motion. Simplified in mechanics, a larval body consists of body wall attached to the muscles and body fluids inside the body wall. The 2 parts works together as a hydrostatic skeleton [5]. The skin has regular repeating symmetrical folds, which are essential for its deformation and friction, forming its segments. The muscles of Drosophila larvae are in 3 orientations: dorso-ventral, anterioro-posterior and

N.F. Lepora et al. (Eds.): Living Machines 2016, LNAI 9793, pp. 375–386, 2016.
DOI: 10.1007/978-3-319-42417-0_34

oblique. Antero-posterior muscles are located nearer the interior than dorsoventral muscles. The body wall muscles are segmentally repeated, and in each abdominal half segment there are approximately 30 of them ([1]) (Fig. 1).

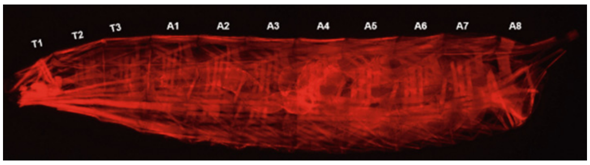


Fig. 1. A Drosophila larva expressing mCherry (a type of photoactivatable fluorescent proteins [14]) in its muscles. From Balapagos (2012).

Based on the property of their bodies, Drosophila larvae are able to do several motions, such as peristaltic crawling, body bending and rolling. Forward peristaltic motion is best described. In the centre of the body, viscera suspended in hemolymph is essential for limiting body wall deformation and produces piston motion. During the 'piston phase' of peristalsis, muscles on the tail contract and push the viscera forward. The second 'wave phase' involves a wave of muscle contraction travelling through the bodywall segments from tail to head [4]. To mimic various and motions of a Drosophila larval, it is important to utilize this anatomical structure and avoid oversimplifying the high DOFs.

Some soft robots have been developed as bionic robots. The main materials are silicone, rubber, or other flexible and stretchable materials. They are usually actuated by Shape-Memory Alloy (SMA) or pneumatically, such as Biomimetic Miniature Robotic Crawler [7], GoQBot [6], Multigait soft robot [13], and a fluidic soft robot [11]. These robots only have several degrees-of-freedom (DOFs) and usually only have one type of motion, which is not sufficient to mimic larval motion. Although SMA is widely applied on soft robots, it has a significant shortcoming. As SMAs deform according to temperature, their response is limited by control of temperature. Because soft robots are usually not sufficient in heat dissipation, heat accumulates inside the robots, and response times of SMAs get too long so that continuous actuation is infeasible. The shortcoming does not exist on pneumatic actuation. Hence, pneumatic actuators are a feasible option as they have a faster response and longer effective working time. However, the main action most of soft pneumatic actuators is off-axis bending, and the axial elongation and contraction are only side effects. For examples: Micro Pneumatic Curling Actuator- Nematode Actuator [9], Pneu-net [13]), and Robot Air Muscles made from Oogoo [8]. As axial contraction is necessary for some motion (such as peristalsis), we designed a new type of pneumatic actuators.

2 Methods

The robot is made from soft silicone rubber, instead of rigid material, because: (1) motions of Drosophila larva are based on continous body deformation; (2)

soft materials have more similar properies to biological tissue than rigid material, such as nolinear elasticity and hysteresis, which are suitable to simulate dynamic characteristics of the muscle; and (3) defomation of Drosophila larval body wall is one method to control friction between body and contacted surface.

Figure 2 shows a sketch of a possible structure of a maggot robot. The robot has repeating modular body wall segments, with a water bag or air bag inside. Here we described the construction and control of 4 body segments. At present, the control system and pneumatic system are placed off board because of limited space and load.

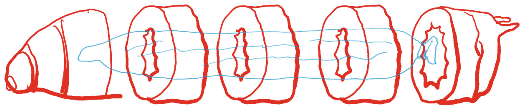


Fig. 2. Sketch of the soft maggot robot. A central bag of fluid is surrounded by muscle segments.

2.1 Design of the Actuator and Body Wall of the Soft Robot

Pneu-nets (Fig. 3(a) and (b)) are usually made from 2 different soft materials: (1) flexible and stretchable material, such as Ecoflex, to form chambers to inflate and expand; (2) flexible but less or not stretchable material, such as Polydimethylsiloxane (PDMS). Thus, when pneu-nets are inflated, the actuator bends to the side made from less stretchable material. Pneu-nets are not suitable for tubular body wall because the stretchable layer limits axial bending, hence we have modifeid the design to produce a new actuator type, which we called Extensible Pneu-nets.

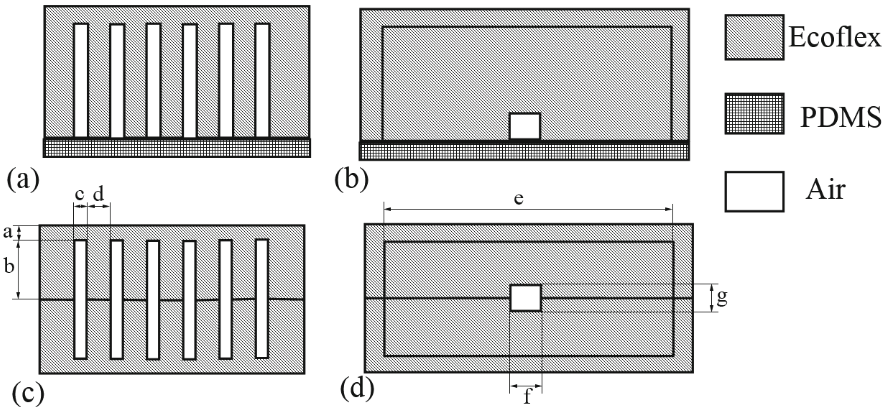


Fig. 3. Structure of Pneu-nets and Extensible Pneu-nets (a) and (b) are longitudinal and transverse sections of Pneu-nets, respectively; (c) and (d) are longitudinal and transverse sections of Extensible Pneu-nets, respectively.

Extensible Pneu-nets (Fig. 3(c) and (d)) use only 1 stretchable material. Small air chambers are connected by air tunnels to form a muscle. Different muscles are isolated. When an air chamber is inflated, it expands in all directions, and the direction with maximum expansion is the direction with the maximum cross sectional area. To limit deformation in the unwanted direction, thickness of the inner walls between chambers and thickness of the outer walls are carefully selected and tested. As stretchable material allows not only bending but also expansion along the surface, tubular body wall based on Extensible Pneu-nets are possible to axially bend.

To make the air chambers and tunnels inside, the actuator is divided into 2 layers which are cast separately. The moulds can be manufactured in conventional machining process or by 3D printing. Then the 2 layers are glued together with the same material. Finally, tubes for injecting pressed compressed air are inserted and glued. By including more air chambers and tunnels on a model, a body wall with multiple pneumatic actuators can be cast.

The first attempt at a muscle pattern was designed according to real muscle pattern on dissected and flattened body wall of Drosophila larva (Fig. 4). Dorsal oblique (DO) muscles, lateral transverse (LT) muscles, oblique lateral (LO) muscles, ventral longitudinal (VL) muscles and ventral acute (VA) muscles are simplified and mapped on the muscle pattern of the body wall. However, the adjacent muscles limited each others motions, especially when they have different orientations. The cause of limitation is that inner walls between air chambers limit transverse deformation, which is the direction that the adjacent muscles are designed to deform. Thus adjacent muscles should either be parallel, or should not be contiguous.

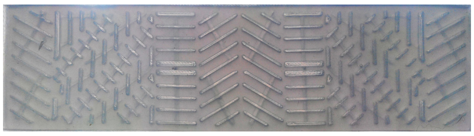


Fig. 4. Body wall of a body segment with Extensible Pneu-nets designed according to real muscle pattern on dissected and flattened body wall of Drosophila larva.

The design of the prototype evaluated in this paper is a body wall with 4 body segments (Fig. 5, left). Each body segment has 3 transverse muscles and 3 longitudinal muscles. These 2 types of muscles are connected perpendicularly and only connected on corners, leaving gaps between them to avoid limitation of deformation between each other (Fig. 5, right). Body segments are connected in series by longitudinal muscles. Figure 6 shows the mould for the body wall. After a flat body wall was made, it was folded end to end and clamped by 2 specially cut boards. Through the window of the board, the end was carefully aligned and glued together. By this process, the flat body wall is formed into a hollow cylinder shape (Fig. 7).

In this 4 body segment version, because the limited resolution of the 3D printer we use (Wanhao Duplicator with 0.4 mm nozzle) and resistance of air flow in tube, the dimensions of air chamber, as shown in Fig. 3(c) and (d), are: a = 1.2 mm, b = 3.0 mm, c = 0.8 mm, d = 1.2 mm, e = 18 mm (longitudinal muscles) or 28 mm (transverse muscles), f = 2.0 mm, g = 3.0 mm. In a curved single body segment, the longitudinal length is 40 mm, the diameter is of the robot is about 50 mm. The total length of the 4 body segment body wall is about 175 mm.

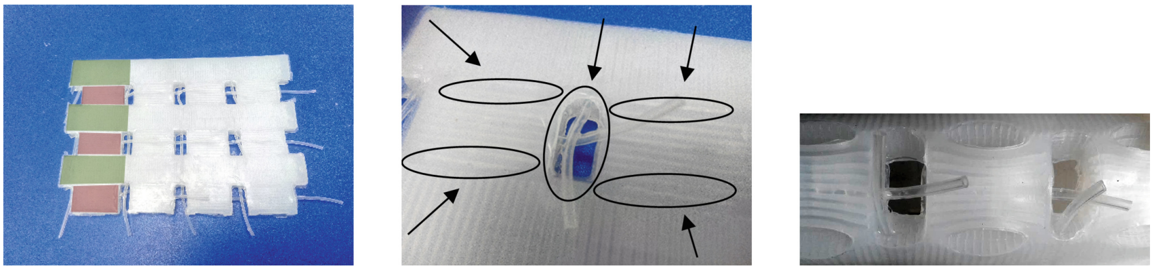


Fig. 5. (left) Prototype design of a body wall with perpendicular arrangement of muscles. Transverse muscles and longitudinal muscles of the first body segment are highlighted in red and green, respectively. (centre) A closer view of the flatten body wall shows gaps and spaces between the muscles to allow expansion. (right) The gaps and spaces when the body wall curved. (Color figure online)

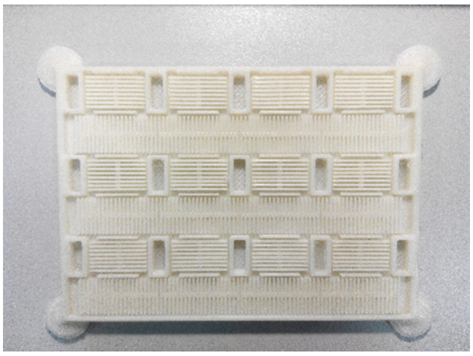


Fig. 6. The 3D printed mould for body wall casting.

2.2 Pneumatic Actuation and Control System

The pneumatic actuation and control system controls the robot by controlling air pressures of air chambers. Air pressure sensors measures pressure in every muscle, pumps and valves control the air flow.

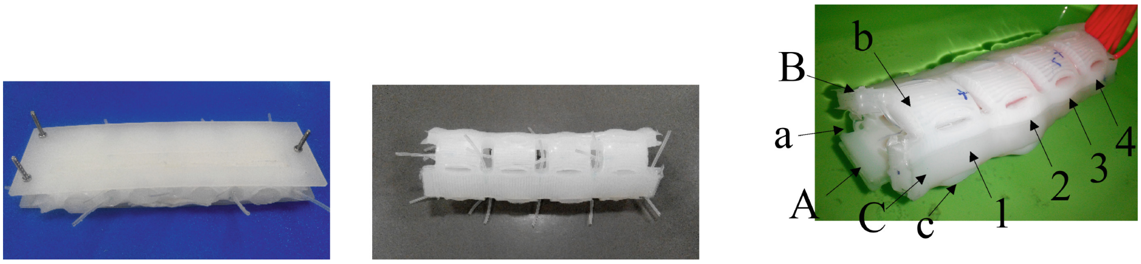


Fig. 7. (left) The body wall is clamped and glued. (centre) Formed into a hollow cylinder. (right) Names of muscles: body segments are numbered, longitudinal muscles named in capital letters, transverse muscles named in lower case letters.

Pneumatic Control System. A pneumatic control system has been designed for the robot. The system is located off board and connects to the robot with rubber tubes. As the robot has more DOFs than previous pneumatic soft robots mentioned above, the size of the pneumatic control system is designed to be compact.

The main component of the system is a valve island with 24 pairs of miniature 2 way solenoid valves (Fig. 8). The size of solenoid valves is 10 mm × 11 mm × 23 mm. Overall, the size of the valve island is 120 mm × 91 mm × 60 mm. The valves are installed the 3D main structure by interference fit. The main structure of the valve island consists of layers of 3D printed parts. The upper layer made form Acrylonitrile Butadiene Styrene (ABS), which offers Mechanical strength to fix valves, and lower layer made from Thermoplastic Elastomer (TPE), which has build in air channels with air-tightness. Every channel connects 4 ways, which are 2 valves, a pressure sensor, and an air chamber on the robot. The other 2 ways of each pair of valves are connected to compressed air and open to air, respectively. As the solenoid valves speed up to 100 Hz, the air flow can be finely controlled.

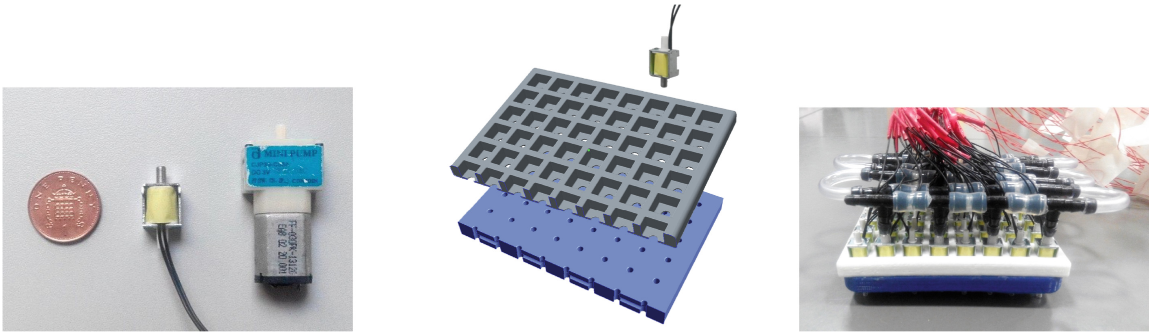


Fig. 8. (left) A valve and pump in the system. (centre) Structure of the 3D printed valve island. (right) Pneumatic valve island with 24 pairs of valves

Embedded Control System. An Embedded Control system has been designed for control and actuation. The control system is a hierarchical control system consisting of 1 main controller and 3 slave controllers. Their micro controllers are

STM32F411RE by STMicroelectronics. They are based on Cortex-M4 by ARM with digital signal processor (DSP) and floating-point unit (FPU). The main controller receives commands from a computer, and distributes them among the slave controllers by Universal Synchronous/Asynchronous Receiver/Transmitters (USART). On each of the slave controllers, 16 hardware Pulse-width modulation (PWM) channels and 8 Analog-to-digital converters (ADC) are configured to control 8 muscles. The PWMs control Darlington transistor arrays (ULx2003 by Texas Instruments). On each slave control board, 3 of them are adopted to drive valves. MPS20N0040D-D, which is an air pressure sensor to measure pressure in air chambers, is adopted to measure the pressures.

Algorithm. At present stage, the robot is controlled by feedforward preprogrammed motion. According to a approximate linearization between deformation and pressure at the initial state of equilibrium, the pressure is utilized as feedback of motion of muscle.

3 Experiments

The robot was tested for individual control of every muscle and coordination between them. Three motions are programmed and tested on the robot (A video of the experiments: https://youtu.be/aFE9dANHowk). The muscles are named based on their location. As show in (Fig. 7), body segments are numbered, longitudinal muscles named in capital letters, transverse muscles named in lower case letters.

Turn. In this motion, muscle a and B on every body segment was actuated at the same time, then pressure released. To minimize friction and show relevance between pressure and deformation, the robot is tested while floating on water. Figure 9 shows the motion of the robot. The pressure of the muscles is shown in Fig. 10.

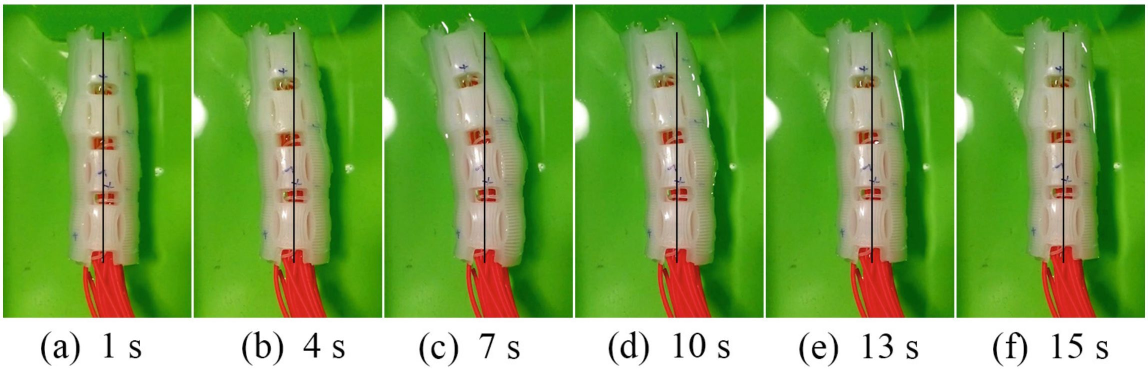


Fig. 9. Turning left. The black lines show the initial central axis.

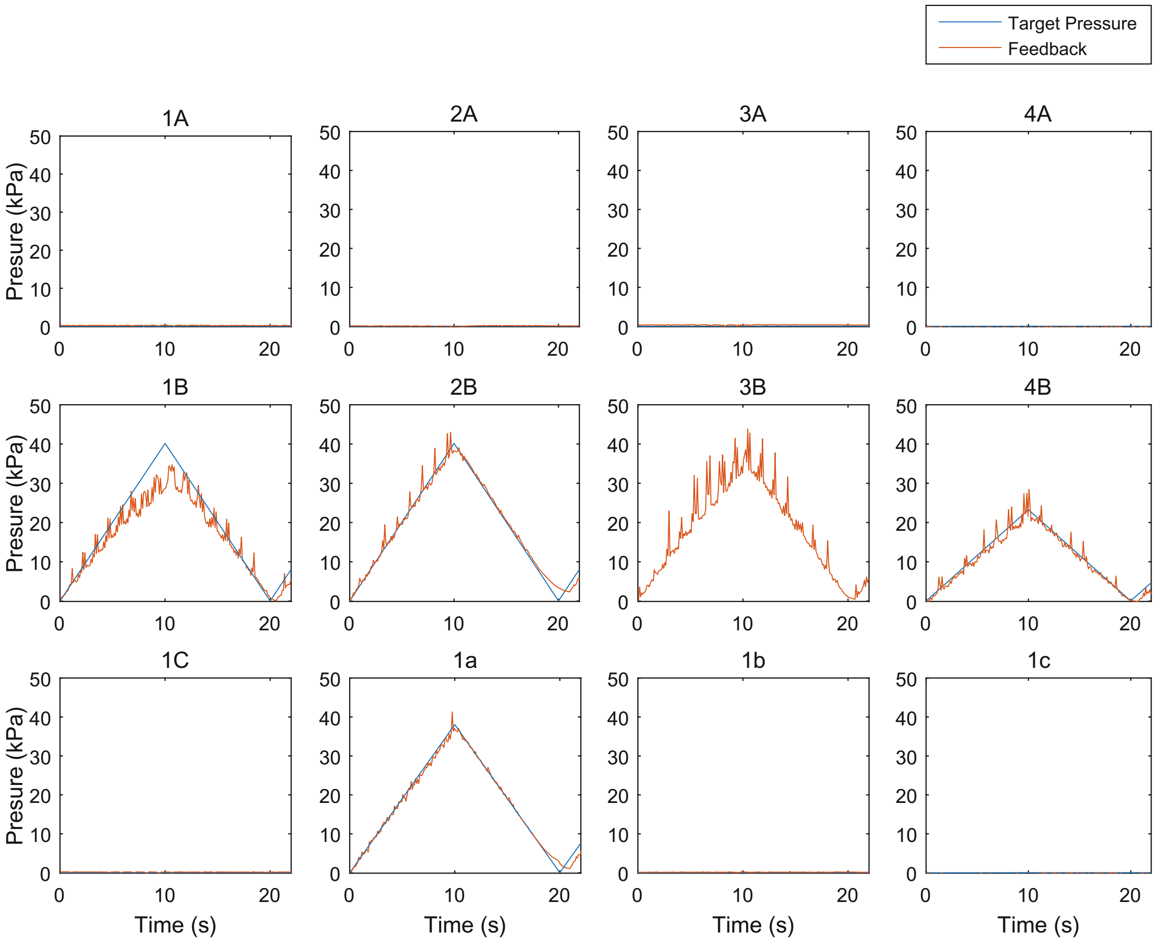


Fig. 10. Pressure of muscles in the first body segment, muscles A and muscles B during turning. (Color figure online)

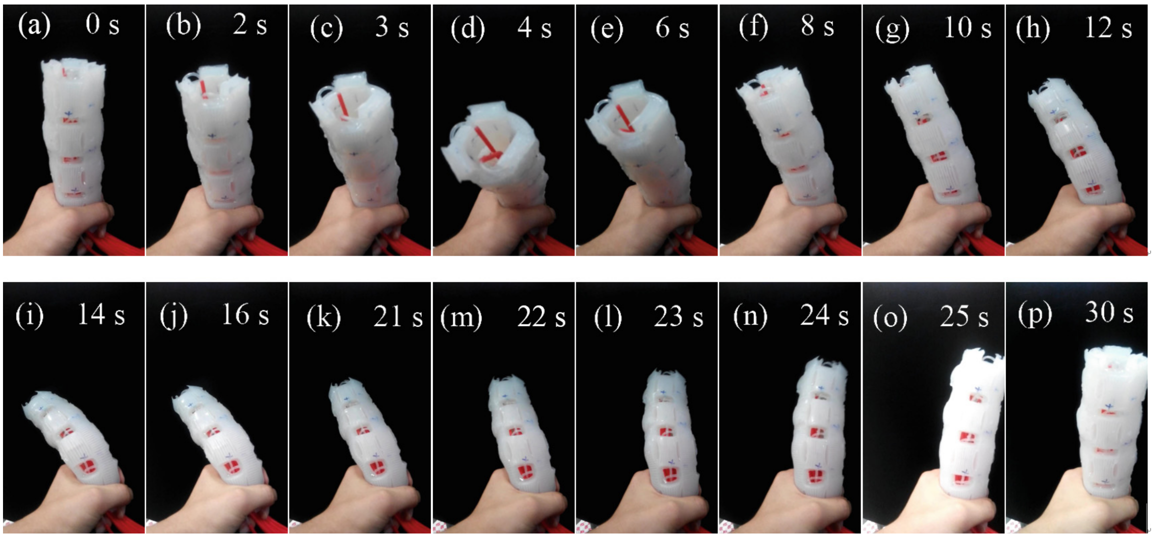


Fig. 11. Roll.

Roll. In this motion, muscle a and B, b and C, c and A, are inflated alternately. As the bundle of the tubes flowing the robot impact the rolling on a surface in water, the robot is hold on its tail vertically during test. Figure 11 shows the motion of the robot. The pressure of muscles show in Fig. 12.

Peristalsis. In this simplified peristalsis, all the muscles on a body segment inflate at same time and muscles of different segments inflate alternately.

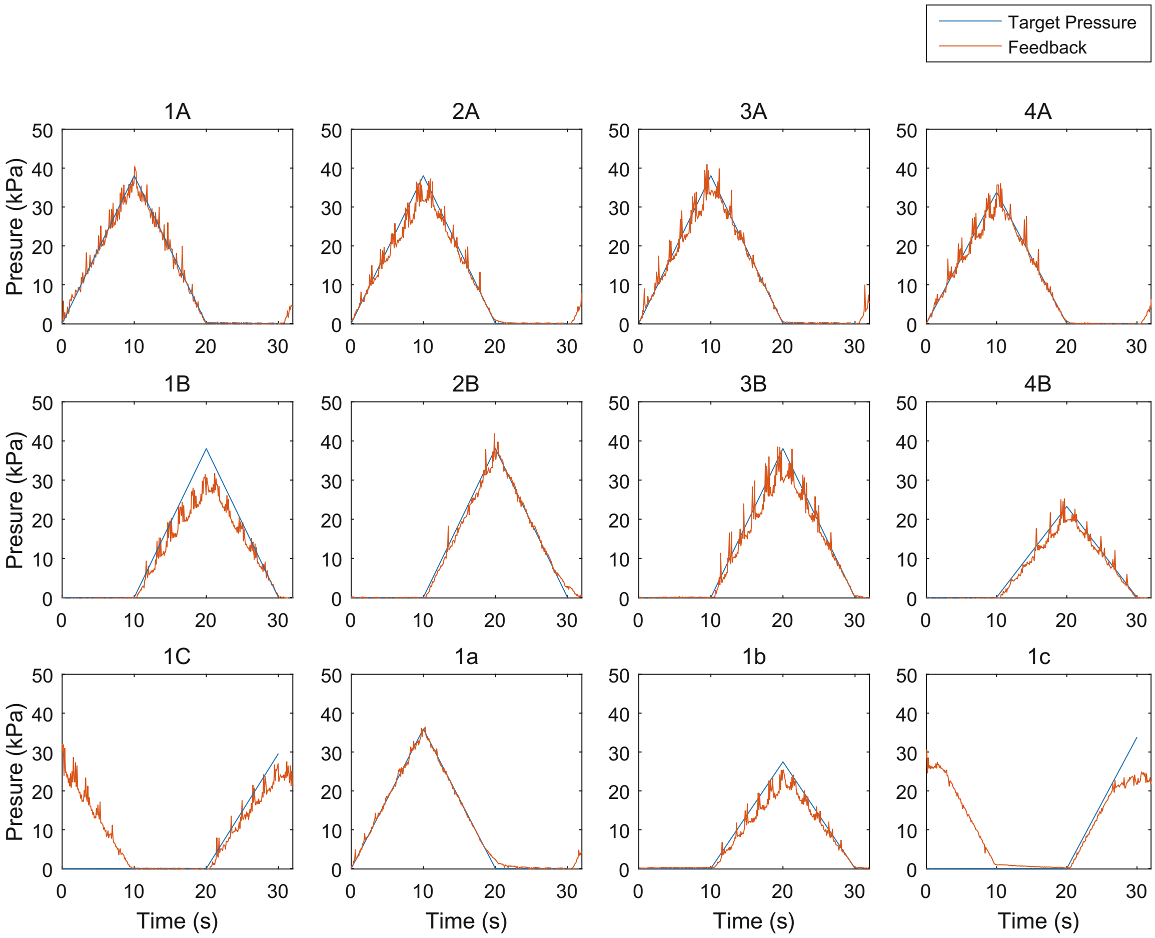


Fig. 12. Pressure of muscles in the first body segment, muscles A and muscles B during rolling. (Color figure online)

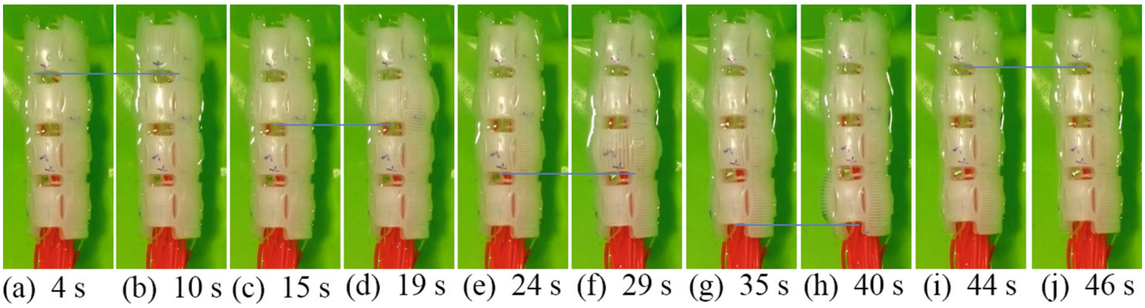


Fig. 13. Peristalsis. The parts on the blue lines were expanding. (Color figure online)

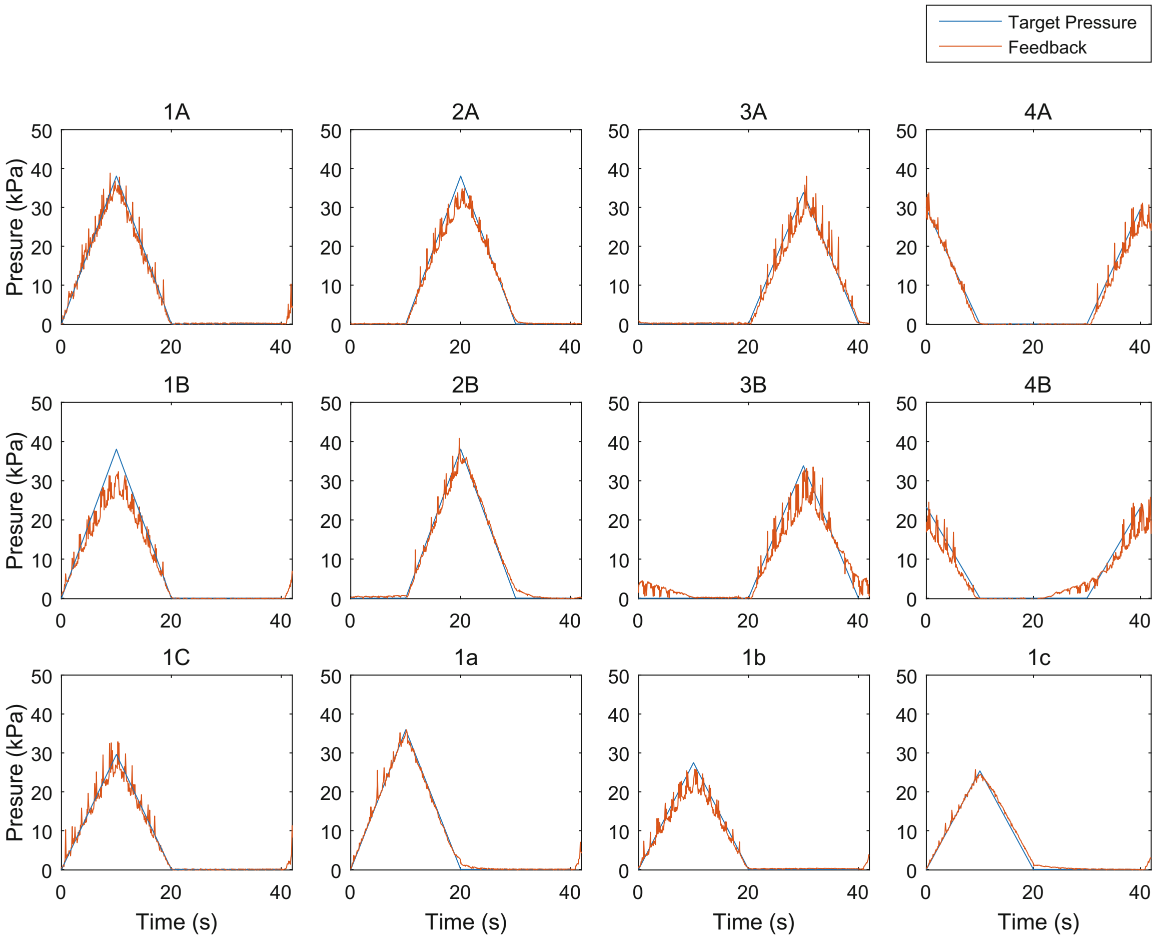


Fig. 14. Pressure of muscles in the first body segment, muscles A and muscles B during peristalsis. (Color figure online)

The motion is tested on water. Figure 13 shows the motion of the robot. The pressure of muscles show in Fig. 14.

4 Discussion

In the tests above, the robot produced three different motions from muscles actuated individually in different orders. The system was able to control the pressures according to the control signal, although the pressures have some noise. However, the three motions are not accurate. Deformation for the same pressure is different between the muscles. That is because muscles are slightly different and the relationship between deformation and pressure is not ideally linear. When an air chamber is inflated to a given range, the pressure does not change much even with obvious deformation. Hence, applying the same pressure to different muscles can result in different deformations. Thus, deformation sensors will be important to precise control of the robot.

Hence our immediate aim for future work is to develop and install sensors on the robot for deformation feedback. As the sampling density of the deformation sensors is limited, different deformations may map to the same output, hence a model or method to learn the relationship between sensor outputs and posture is necessary. We should then be able to explore more thoroughly the movement capabilities of the current design. Some additional redesign of the pneumatic muscle and body wall may be necessary, for example, surface processes to mimic denticles on Drosophila larval skin which generate asymmetric friction so that peristalsis produces forward locomotion [12].

5 Conclusion

Our longer term aim for this robot is to use it as a platform to test neural circuit models of *Drosophila* larvae. Initially this could focus on the motor circuits that generate and control peristalsis and bending. In particular these circuits could form the basis of an adaptative method for learning the control signals needed to adjust to the irregularities and non-linearities in the actuators and their interactions, in the same way that maggots are able to adapt to rapid change and growth in their body while maintaining efficient locomotion. Ultimately we would like to add sensors for environmental signals and investigate the sensorimotor control and associative learning involved in, e.g., odour search [3].

References

1. Bate, M.: The mesoderm and its derivatives. The Development of Drosophila Melanogaster, pp. 1013–1090. Cold Spring Harbor Laboratory Press, New York (1993)
2. Brand, A.H., Perrimon, N.: Targeted gene expression as a means of altering cell fates and generating dominant phenotypes. Development (Cambridge, England) **118**(2), 401–415 (1993)
3. Gomez-Marin, A., Louis, M.: Multilevel control of run orientation in Drosophila larval chemotaxis. Front. Behav. Neurosci. **8**, 38 (2014)
4. Heckscher, E.S., Lockery, S.R., Doe, C.Q.: Characterization of Drosophila larval crawling at the level of organism, segment, and somatic body wall musculature. J. Neurosci. **32**(36), 12460–12471 (2012)
5. Kohsaka, H., Okusawa, S., Itakura, Y., Fushiki, A., Nose, A.: Development of larval motor circuits in Drosophila. Dev. Growth Differ. **54**(3), 408–419 (2012)
6. Lin, H.T., Leisk, G.G., Trimmer, B.: GoQBot: a caterpillar-inspired soft-bodied rolling robot. Bioinspir. Biomim. **6**(2), 026007 (2011)
7. Menciassi, A., Accoto, D., Gorini, S., Dario, P.: Development of a biomimetic miniature robotic crawler. Auton. Robots **21**(2), 155–163 (2006)
8. mikey77: Soft robots: Making robot air muscles. Webpage (2012). http://www.instructables.com/id/Soft-Robots-Making-Robot-Air-Muscles/
9. Ogura, K., Wakimoto, S., Suzumori, K., Nishioka, Y.: Micro pneumatic curling actuator - Nematode actuator. In: 2008 IEEE International Conference on Robotics and Biomimetics, pp. 462–467 (2009)
10. Olsen, S.R., Wilson, R.I.: Cracking neural circuits in a tiny brain: new approaches for understanding the neural circuitry of Drosophila. Trends Neurosci. **31**(10), 512–520 (2008)

11. Onal, C.D., Rus, D.: Autonomous undulatory serpentine locomotion utilizing body dynamics of a fluidic soft robot. Bioinspir. Biomim. **8**(2), 026003 (2013). http://www.ncbi.nlm.nih.gov/pubmed/23524383
12. Ross, D., Lagogiannis, K., Webb, B.: A model of larval biomechanics reveals exploitable passive properties for efficient locomotion. In: Wilson, S.P., Verschure, P.F.M.J., Mura, A., Prescott, T.J. (eds.) Living Machines 2015. LNCS, vol. 9222, pp. 1–12. Springer, Heidelberg (2015)
13. Shepherd, R.F., Ilievski, F., Choi, W., Morin, S.A., Stokes, A.A., Mazzeo, A.D., Chen, X., Wang, M., Whitesides, G.M.: From the cover: multigait soft robot. Proc. Nat. Acad. Sci. **108**(51), 20400–20403 (2011)
14. Subach, F.V., Patterson, G.H., Manley, S., Gillette, J.M., Lippincott-Schwartz, J., Verkhusha, V.V.: Photoactivatable mCherry for high-resolution two-color fluorescence microscopy. Nat. Methods **6**(2), 153–159 (2009)

5.3 Further discussion

The soft maggot robot has more details of the *Drosophila* larval motor system than other robots. It has advantages in replicate larva motor control, especially subtle motions which need coordinate control of multiple muscles. The more simplified simulated animal body or robots, such as rigid body robots, miss the details of the larval motor system. Although at the behaviour level, such as chemotaxis, subtle motions are not necessary, so as the detailed motor system, for low-level actions which need fine motion control, such as efficient peristalsis gait, missing the details could result in qualitative differences. Hence, the soft maggot robot can be applied for behavioural level research as well as action level research.

5.3.1 The balance between fidelity and simplification

Although the soft maggot robot has more details than other robots, the design of the robot is still a trade-off of *Drosophila* larva motor system and existing manufacturing technology for soft robots, which seems unavoidable according to the attempts at the designs.

In our first attempt at designing of the body wall, as shown in Figure 5.18 (A), the muscle pattern of *Drosophila* larvae is copied with the maximum accuracy of the fabrication process. Five groups of muscles are captured in the silicone body wall, as marked in colour shown in Figure 5.18 (A) and (B). However, the body wall does not work as expected. As mentioned in the above paper, the adjacent muscles in different directions limit each other's deformations because they are contiguous. In a real larva, as shown in Figure 5.18 (B) and (C), the muscles are not contiguous and can slide relative to each other. In the later design the body wall takes this important feature and leaves some gaps between muscles that are in different directions. Although, the number and pattern of muscles are simplified due to the limitation of fabrication, the robot wall functions better.

In these attempts of the design, the manufacturing technology limits the fidelity of the biorobot body, which causes a range of problems of the biorobot approach. E.g. that trying to be like the animal in some factors ends up less like the animal in other factors, such as put multiple muscles at limited location constrain locomotion capability. With the limitation of existing fabrication technology, copying the animal's structure without appreciating the properties of the technology are not practical. trade off has to be made. The ultimate solutions to these problems may rely on the development of

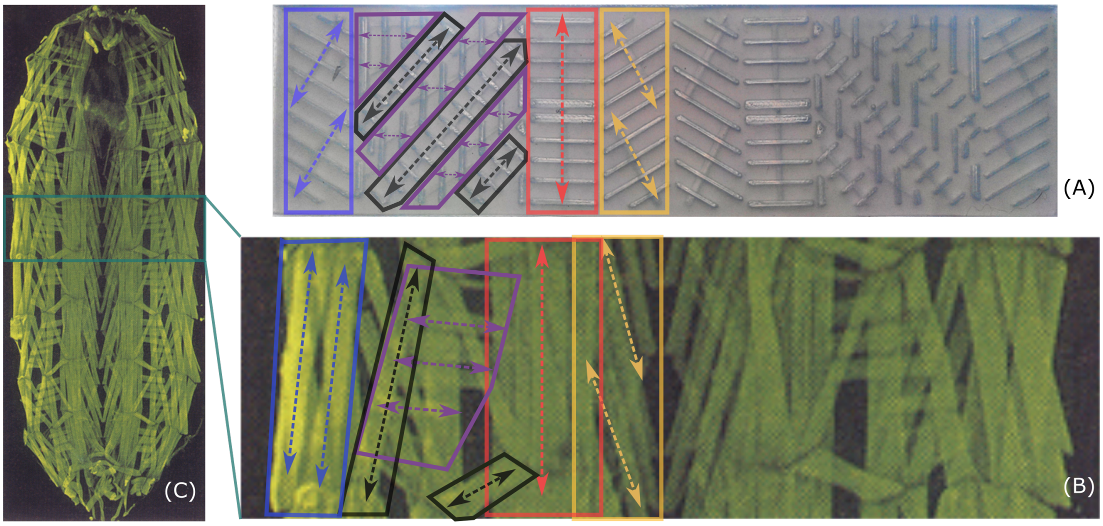


Figure 5.18: Compare between the first design of the body wall and larval body wall. (A) is the same in Figure 4 in the above paper, in addition to the left half of the muscles are marked. The muscles with same extension directions are marked with the same colours. The dotted arrows show the extension directions. (B) and (C) are photos of a *Drosophila* larva body wall, and (C) is part of (B). The left half of the muscles on (B) are marked in the same way with (A). Some of the muscles in (B) are not captured in (A) due to the limitation of fabrication and design. **Blue**: the dorsal acute (DA) muscles. **Black**: the DA muscles or the segment border muscles (SBM). **Purple**: the lateral transversal (LT) muscles. **Red**: the ventral longitudinal (VL) muscles. **Yellow**: the ventral oblique (VO). (B) and (C) are modified from the work by Itoh et al. (2016).

the fundamental technology, such as manufacturing technology of soft materials and development of new materials.

5.3.2 More about future works

In the inserted paper, some future work is mentioned. Due to the limited time of developing the robot, the sensors for body wall deformation are not yet designed and applied. Stretchable sensors, as reviewed by Liu et al. (2018), are options for measuring the deformation. The sensor made with Ecoflex, the type of silicone used in the robot, and EGaIn (eutectic gallium-indium), a type of liquid alloy when it is home temperature, can be the best option for the robot. The same type of silicone has the same deformation property and thus less possibility of limiting the motion of the robot. With liquid metal, the sensor can have a wide functional range. With the sensor as feedback, the dynamic synapse learning rule proposed in chapter 2 and neural networks with the similar architecture of the neural network proposed in chapters 3 and 4 can be applied to the maggot robot for reinforcement learning of behaviours or actions.

Chapter 6

Summary and Discussion

Robot control or decision making for a robot is usually very dynamic due to the properties of robot bodies and realistic environments. However, most existing reinforcement learning models are based on statistics of static data, thus are not efficient in robot reinforcement learning for these tasks. As animals can learn dynamic tasks in dynamic environments efficiently, *Drosophila* larva is studied in this work as a model for robot reinforcement learning. A soft robot inspired by *Drosophila* larva is also designed for the advantage that soft robots are safer than traditional rigid-body robots in motion explorations.

6.1 Key contributions

A key question in this work is how a learning rule can be compatible with neural circuit architectures which are more like biological systems and suitable for robot learning of dynamic tasks. These neural circuits usually have internal dynamics, such as dynamical neuron models and recurrent connections, and could have complex network topological structures, which cannot be optimised using conventional approaches for optimising neural networks.

Chapter 2 demonstrates that the micro-level properties of neural systems can support an alternative learning rule. Based on the study of the dynamics of synaptic neurotransmitter receptors, a synaptic plasticity model for operant learning is proposed and discussed. With the synapse model, a variety of neural circuit models, including spiking neural networks, firing-rate neural networks, feedforward neural networks and recurrent neural networks, can be optimised in operant learning tasks. The relationship between the model and previous synaptic plasticity models, such as LTP-STP models, are discussed.

Chapter 3 shows that the proposed synaptic plasticity model can reproduce lifelike learning. The model was applied to a mushroom body model to explain the operant learning of *Drosophila* larval turning with light-stimulated Dopamine neuron as rewards. The model reproduced the characteristics of the biological experiments, shows that the synaptic plasticity model is compatible with biologically plausible neural circuits and can provide learning ability in realistic tasks.

Chapter 4 shows that the proposed synaptic plasticity model can be applied to a robot reinforcement learning task. The synaptic plasticity is abstracted and simplified for the convenience of engineering application, such as for less computational resource occupation and straightforward dynamics control. The simplified model was applied to a dynamical neural network with internal dynamics and recurrent connections, which controlled a planar bipedal robot learning to walk in a standard reinforcement learning benchmark. The is no previously published model that can solve the task.

Another key question of the study is how to replicate the characteristics of the *Drosophila* larva motor system with a soft robot. There have been soft robots designed mimicking the shapes and gaits of worms, but not yet their motor system, as reviewed in Chapter 5. My study paid more attention to the mechanics of *Drosophila* larval motor systems for motions, and design a soft pneumatic maggot robot closely based on characteristics of the motor system. The pneumatic muscle configuration is simplified from *Drosophila* larval muscle patterns. With a novel pneumatic muscle integration process, the robot body contains a high density of pneumatic muscles. With the high DOFs provided by the biometric muscle system, the robot can reproduce lifelike motions such as peristalsis, turning and rolling.

6.2 Future work

6.2.1 Dynamic synapse

There is a detailed discussion of the synaptic plasticity model in chapter 2. Here are three key directions for the next step of the model.

The extension of the dynamic synapse model: The dynamic synapse model provides a framework with the ability of reinforcement learning, based on a learning rule that the centre of oscillation is biased to the instantaneous state when the state causes reward. The framework keeps the flexibility to be extended for more functions, such as 3-factor learning, as discussed in chapter 2. There are more potential extensions worth to be modelled and simulated, such as LTD/LTP, and compared with biological observation results. More biological experiments and observation of synaptic plasticity can be helpful for this comparison, especially those experiments that can densely track and record multiple types of dynamics, such as receptor trafficking, PSD, and size of synapses, in the same synapses and at the same time.

Chaotic oscillation model that has non-drift first-order integral: In an oscillation period of the dynamic synapse model, the integrated values of the differences between instantaneous synaptic strength and the centre of oscillation on each side are not equal. This causes a drift of the first-order integral and reduces the resistance of the learning rule to modulators that are not correlated with rewards. A more idealised oscillation should have non-drift first-order integral. Hence the noise in modulator or reward will have a smaller possibility to impact the centre of oscillation. As the model is simplified and abstracted from real synapses which have more detailed mechanics, it could be the simplification that causes the unequal oscillation. A model that has non-drift first-order integral could be developed by including more mechanics in synapses or using a different approach of simplification.

Modelling of chaotic explorations that integrates dynamic synapse and other chaotic processes: Chaotic spontaneous behaviours have been observed in animals, such as *Drosophila* (Maye et al., 2007). Because there are a variety of dynamics in animals can be sources of chaos, such as neuron membrane potentials (Olsen and Degn, 1985) and animal body dynamics (Loveless et al., 2018), the behaviour or action exploration can be from them as well besides synaptic chaos. Exploration of a learning system with both dynamic synapse and chaos from other sources would be an exciting direction, for the interaction between these chaotic systems and their impact on the learning process.

6.2.2 Neural circuit/network models

In Chapter 3, a mushroom body (MB) model is built based on previous mathematical models and the dynamic synapse models. The model reproduced the biological experiments of *Drosophila* larva operant learning of turning. As the complexity of the operant learning task only needs a small section of MB, the model in Chapter 3 is simplified from the intact MB. More sophisticated learning tasks, such as locomotion, need MB models with a larger scale due to the higher DOFs of actions and non-linear relations between actions and rewards.

An attractive opportunity provided by the dynamic synapse model is to explore more neural circuit architectures that have not been explored because of the restrictions of previous synapse models, such as architectures of other brain regions and other hypothetical artificial neural network architectures. An interesting target is to find the architectures that can solve or weaken the Curse of Dimensionality (Kober et al., 2014; Arulkumaran et al., 2017). The curse includes:

1. Parameter space would be too large to explore in a short time, given a limited response time of system and reward.

2. Assuming the size of regions in parameter spaces that can lead to rewards are constant, higher dimensions of parameter space will results in sparser reward during exploration.

3. The exploration in more dimensions at the same time would make the time taken for identifying relations between dimensions and results increasingly longer.

New neural circuit/network architectures with curiosity, attention and internal rewards can help solve the problems.

For complex tasks, taking locomotion as an example, the external reward is usually only provided according to whether the agent is moving in a specific direction, which is the criterion for successful movements. However, the simple reward signal is not adequate to guide the details of learning for more reusable knowledge in multiple levels. Successful locomotion also requires lower level knowledge including how to keep balance, the relations between joint angle and location of limbs, the boundary between safe and dangerous states, and actions for recovering from abnormal states. However, the external reward would not provide a direct indication of this knowledge. If a neural architecture could have 'curiosity', such as a willing for learning new arbitrary knowledge it finds, it can learn some basic knowledge without external reward. For example, when a robot notice an abnormal motion of a limb then reward the memory the association among the motion, effort of muscles and sensory inputs, it can learn a lower level control of it limb. As the lower level knowledge can be elements for building higher-level knowledge, learning with curiosity can utilise the exploration that does not result in rewards and save time spent in exploration.

The lower-level knowledge learnt with curiosity can be elements for building an internal representation of states. With the representation, links between states and actions can be built within a neural circuit/network, and then the links can be a map for calculating internal rewards by calculating the distances between the current states and rewarding states. The internal reward can be used to improve the performance when the external rewards are sparse. The map is similar to model-based learning in that they all builds internal models of the world, but it starts with model-free learning that prior knowledge of the world is not necessary to initiate the learning.

Because dynamic synapse does not require global reward and information of global synapse strengths (which is required by error back-propagation), neural circuit/network architectures with dynamic synapse can update locally. It is similar to attention in that only a small set of knowledge is updated. Moreover, the internal reward can be calculated not only based on the global state by also local states, so it can provide different rewarding criteria for different circuits to learning different knowledge, which increases the learning efficiency. It is similar to attention in that learning is based on a subset of information.

The MB could be an inspiration for architectures with these abilities. The Kenyon Cells (KCs) in MB receive an apparently random collection of sensory inputs Caron et al. (2013), mapping lower dimension sensory inputs to higher dimension spaces representation, which is a type of sparse coding. Hence, each KC can represent an initially random but specific subset of states, which provides a base for learning at an early stage. For example, an artificial architecture can facilitate the learning with curiosity by remembering the result of explored actions as transfers of the states. The transfer of the states could be coded by KCs to KCs connections, as recently discovered that KCs get extensive synaptic input from KCs(Takemura et al., 2017). The research also finds there are direct synaptic connections from KC to Dopaminergic neurons (DANs) (Takemura et al., 2017), which suggests that KCs can influence the activity of DANs. If a KC that have not ever fired actives and causes a DAN to release modulator, which enable learning but weaken synapse from KCs to DANs, the circuit can provide an internal reward for curiosity. MB consists of multiple compartments that have similar architecture (Cohn et al., 2015). Each of the compartments usually has one DAN and one Mushroom body output neuron (MBON) and is believed to be a semi-autonomous information processing units (Aso et al., 2014a). Existing MB models only capture one of the compartments, such as Wessnitzer et al. (2012),Ardin et al. (2016) and the model proposed in chapter 3. If a model can include multiple compartments, each of MBONs corresponds to a type of behaviour, and each of DANs modulates the learning related the behaviour, the model could learn with attention to different behaviours and have better exploration efficiency.

6.2.3 Robot reinforcement learning

Regarding applications of the works, there are three directions for improvements: higher processing ability of sensory input, richer behaviours and safer exploration for risky actions. To elaborate on it, the bipedal robot in Chapter 4 is taken as an example here.

In Chapter 4, the learning rule based on dynamic synapse is applied to a planar bipedal robot. Although it is just a preliminary attempt of application of the learning rule to robot reinforcement learning, the performance exceeds performances of other algorithms people have applied to the task. There is another similar task with the same robot, but the terrain is more complicated with ladders, stumps, pitfalls. To pass through the task, the neural network controlling the robot needs to rely more on the Lidar sensor data for the perception of the obstacles and adjustment of the gaits. Hence, the neural network should include more architectures for processing Lidar data. Different obstacles might not be overcome with the same gait. Hence, the neural network also should be able to explore, learn, and retrieve a variety of gaits for different obstacles. The strategy presuming stability for higher reward in early stage implied in the neural network proposed in chapter 4 could be not practicable in this case. Hence, the learning process should not only presume higher external reward but allowed learning of explored action and results without high external reward. It requires novel neural network architectures, which is as described in the above section.

The learning rule can also be applied to many other robot tasks, such as robot arm control and motion planning, manipulation, soft robot control and control of whole body motions. Among them, soft robot control is a key section because soft robots are hard to control with conventional robot control approaches but are safer than rigid body robots during action exploration. Hence, combining the learning rule with soft robot control is potentially a productive research direction.

6.2.4 Soft robot

In chapter 5, the design and evaluation of a soft pneumatic robot are presented. The robot has bio-inspired muscle patterns and can mimic some of the *Drosophila* larval motions. The muscles are driven by compressed air, and the pressure in the muscles are carefully controlled. However, because of the nonlinear elasticity of silicone, control of muscles is only reliable with small deformation. For large deformation, direct sensory feedback of deformation is necessary for control accuracy and avoiding exploration of the muscles. However, there is no existing sensor for the measurement. A strain gauge can measure the deformation of surfaces but only suitable for materials with high stiffness and small deformation (Window, 1992). There are some recently developed strain sensors using Polydimethylsiloxane (PDMS) as a substrate, such as strain sensors utilise strain-induced resistance of aligned single-walled carbon nanotube (SWCNT) thin film (Yamada et al., 2011), and strain sensors utilise deformation of Eutectic Gallium-Indium (EGaIn, a type of liquid alloy) (Kramer et al., 2011; Majidi et al., 2011; Gozen et al., 2014). Some of these sensors can deform up to 300%. However, their stiffness is too high for the maggot robot, which is made from Ecoflex, impacting the abilities of the robot in deformation and motion. A possible solution is to design a new type of soft deformation sensor with EGaIn and Ecoflex. It needs a new procedure of manufacture, as the sensor should be very thing avoiding impact the deformation, and new technologies of sealing, as the stiffness of the substrate and electrode are so different that easily causes adhesive failure with large deformation. One of my recent work with my colleagues has improved the reliability of sealing between PDMS and conductive thread, which could be applied to Ecoflex and need more tests. A procedure of manufacture the sensor has been explored and a prototype of the maggot robot with the sensor is made, which also needs more tests.

With sensors that can measure the deformation of the robot body, the robot can be a platform to test robot reinforcement learning. Because the robot is made from soft materials, the robot is safe to the environment and itself during learning. Mushroom body model with dynamic synapse can be applied to the robot for reinforcement learning of motions. The neural network proposed in Chapter 4 can be applied to the robot by a little modification such as include more CPGs. The robot can also be a physical agent to test the operant learning model proposed in Chapter 3.

6.3 Closing remark

The study is motivated by the observation that (a) existing robot reinforcement learning approaches base on idealise simplification and need a large amount of computation for learning of simple tasks, (b) while insects can have complex learning behaviours and capacities with their small brain and body. Deep learning, the rising approach in robot learning, always emphasises the significance of larger and deeper networks, while ignoring the aspects, such as neurodynamics, that could enable small neural networks with richer abilities. By modelling the dynamics inside synapses and a dendrite, I found that the dynamics can be potentially chaotic. Based on the model and a simple learning rule, the dynamics in synapses enables reinforcement learning at the neuron level. It is the core of the work presented in this thesis. With the model, three different types of neural networks were trained with reinforcement learning. The types include the feedforward neural network, the recurrent neural network and the spiking neural network. The model also explains operant learning of *Drosophila* larvae turning behaviour with optogenetic control of rewards. I also proposed a dynamic neural network with recurrent connections and CPGs. With the simplified dynamic synapse model, the neural network becomes the first algorithm that solved the bipedal worker task of the OpenAI Gym, a standard benchmark for reinforcement learning. It provides a tool of reinforcement learning with parameter space exploration that are compatible with feedforward neural networks, biologically plausible neural networks, and recurrent neural networks. It can be applied to neural networks with more complex architectures, and with potential to facilitate reinforcement learning problems with complex dynamics by introducing biological neural circuits to robot reinforcement learning. A soft maggot robot is proposed based on the observation of *Drosophila* larva body wall and locomotion. The body wall of the robot is designed with multiple versions for finding the balance between the fidelity of muscle patterns and feasibility with existing fabrication technologies. The resulting robot can execute realistic motions like a larva with feasible design.

Bibliography

Abbeel, P. and Ng, A. Y. (2004). Apprenticeship learning via inverse reinforcement learning. In *Twenty-first international conference on Machine learning - ICML '04*.

Akbarzadeh, A. and Kalani, H. (2012). Design and Modeling of a Snake Robot Based on Worm-Like Locomotion. *Advanced Robotics*, 26(5-6):537–560.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, O. P., and Zaremba, W. (2017). Hindsight experience replay. In *Advances in Neural Information Processing Systems*, pages 5048–5058.

Ardin, P., Peng, F., Mangan, M., Lagogiannis, K., and Webb, B. (2016). Using an Insect Mushroom Body Circuit to Encode Route Memory in Complex Natural Environments. *PLoS Computational Biology*, 12(2):1–22.

Arena, P., Bonomo, C., Fortuna, L., Frasca, M., and Graziani, S. (2006). Design and control of an IPMC wormlike robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(5):1044–1052.

Argall, B. D., Chernova, S., Veloso, M., and Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38.

Asada, M., Hosoda, K., Kuniyoshi, Y., Ishiguro, H., Inui, T., Yoshikawa, Y., Ogino, M., and Yoshida, C. (2009). Cognitive developmental robotics: A survey. *IEEE transactions on autonomous mental development*, 1(1):12–34.

Aso, Y., Grübel, K., Busch, S., Friedrich, A. B., Siwanowicz, I., and Tanimoto, H. (2009). The mushroom body of adult drosophila characterized by gal4 drivers. *Journal of neurogenetics*, 23(1-2):156–172.

Aso, Y., Hattori, D., Yu, Y., Johnston, R. M., Iyer, N. A., Ngo, T.-T. B., Dionne, H., Abbott, L. F., Axel, R., Tanimoto, H., and Rubin, G. M. (2014a). The neuronal architecture of the mushroom body provides a logic for associative learning. *eLife*, 3:e04577.

Aso, Y., Sitaraman, D., Ichinose, T., Kaun, K. R., Vogt, K., Belliart-Guérin, G., Plaçais, P.-Y., Robie, A. a., Yamagata, N., Schnaitmann, C., Rowell, W. J., Johnston, R. M., Ngo, T.-T. B., Chen, N., Korff, W., Nitabach, M. N., Heberlein, U., Preat, T., Branson, K. M., Tanimoto, H., and Rubin, G. M. (2014b). Mushroom body output neurons encode valence and guide memory-based action selection in Drosophila. *eLife*, 3(3):1–42.

Ávila, E. a., Meléndez, a. M., and Falfán, M. R. (2006). An inchworm-like robot prototype for robust exploration. *Proceedings - Electronics, Robotics and Automotive Mechanics Conference, CERMA 2006*, 1:91–96.

Bellman, R. (1957). *Dynamic programming*. Courier Corporation.

Berni, J. (2015). Genetic dissection of a regionally differentiated network for exploratory behavior in drosophila larvae. *Current Biology*, 25(10):1319–1326.

Billard, A., Calinon, S., Dillmann, R., and Schaal, S. (2008). Robot Programming by Demonstration. In Siciliano, B. and Khatib, O., editors, *Springer Handbook of Robotics*, pages 1371–1394. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bongard, J. C. (2013). Evolutionary robotics. *Communications of the ACM*, 56(8):74–83.

Boyle, J. H., Johnson, S., and Dehghani-Sanij, A. a. (2013). Adaptive undulatory locomotion of a C. elegans inspired robot. *IEEE/ASME Transactions on Mechatronics*, 18(2):439–448.

Brand, A. H. and Perrimon, N. (1993). Targeted gene expression as a means of altering cell fates and generating dominant phenotypes. *Development*, 118(2):401–415.

Brembs, B. (2003). Operant conditioning in invertebrates. *Current Opinion in Neurobiology*, 13(6):710–717.

Brembs, B. (2009). Mushroom Bodies Regulate Habit Formation in Drosophila. *Current Biology*, 19(16):1351–1355.

Brembs, B. and Heisenberg, M. (2000). The Operant and the Classical in Conditioned Orientation of Drosophila melanogaster at the Flight Simulator. *Learning & Memory*, 7(2):104–115.

Calandra, R., Gopalan, N., Seyfarth, A., Peters, J., and Deisenroth, M. P. (2014). Bayesian gait optimization for bipedal locomotion. In *International Conference on Learning and Intelligent Optimization*, pages 274–290. Springer, Cham.

Caron, S. J. C., Ruta, V., Abbott, L. F., and Axel, R. (2013). Random convergence of olfactory inputs in the Drosophila mushroom body. *Nature*, 497(7447):113–7.

Chiang, A.-S., Lin, C.-Y., Chuang, C.-C., Chang, H.-M., Hsieh, C.-H., Yeh, C.-W., Shih, C.-T., Wu, J.-J., Wang, G.-T., Chen, Y.-C., Wu, C.-C., Chen, G.-Y., Ching, Y.-T., Lee, P.-C., Lin, C.-Y., Lin, H.-H., Wu, C.-C., Hsu, H.-W., Huang, Y.-A., Chen, J.-Y., Chiang, H.-J., Lu, C.-F., Ni, R.-F., Yeh, C.-Y., and Hwang, J.-K. (2011). Three-Dimensional Reconstruction of Brain-wide Wiring Networks in Drosophila at Single-Cell Resolution. *Current Biology*, 21(1):1–11.

Chiel, H. J. and Beer, R. D. (1997). The brain has a body: Adaptive behavior emerges from interactions of nervous system, body and environment. *Trends in Neurosciences*, 20(12):553–557.

Cingolani, L. a. and Goda, Y. (2008). Actin in action: the interplay between the actin cytoskeleton and synaptic efficacy. *Nature Reviews Neuroscience*, 9(5):344–356.

Clark, M. Q., Zarin, A. A., Carreira-Rosario, A., and Doe, C. Q. (2018). Neural circuits driving larval locomotion in drosophila. *Neural Development*, 13(1):6.

Cohn, R., Morantte, I., and Ruta, V. (2015). Coordinated and Compartmentalized Neuromodulation Shapes Sensory Processing in Drosophila. *Cell*, 163(7):1742–1755.

Conradt, J. and Varshavskaya, P. (2003). Distributed central pattern generator control for a serpentine robot. In *Proceedings of the International Conference on Artificial Neural Networks (ICANN)*, pages 338–341.

Cotroneo, A., Vozzi, G., Gerovasi, L., and De Rossi, D. (2008). A new bio-inspired robot based on senseless motion: Theoretical study and preliminary technological results. *Multidiscipline Modeling in Materials and Structures*, 4(1):47–58.

Daerden, F. (1999). *Conception and realization of pleated pneumatic artificial muscles and their use as compliant actuation elements.* PhD thesis, Vrije Universiteit Brussel.

Deisenroth, M. P., Neumann, G., and Peters, J. (2011). A Survey on Policy Search for Robotics. *Foundations and Trends R in Robotics*, 2(1-2).

Duan, Y., Andrychowicz, M., Stadie, B. C., Ho, J., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. (2017). One-Shot Imitation Learning. In *Advances in neural information processing systems*, pages 1087–1098.

Duffy, J. B. (2002). Gal4 system in drosophila: a fly geneticist's swiss army knife. *genesis*, 34(1-2):1–15.

Eichler, K., Li, F., Litwin-Kumar, A., Park, Y., Andrade, I., Schneider-Mizell, C. M., Saumweber, T., Huser, A., Eschbach, C., Gerber, B., Fetter, R. D., Truman, J. W., Priebe, C. E., Abbott, L. F., Thum, A. S., Zlatic, M., and Cardona, A. (2017). The complete connectome of a learning and memory centre in an insect brain. *Nature*, 548(7666):175–182.

Eschbach, C. (2011). *Classical and operant learning in the larvae of Drosophila melanogaster.* PhD thesis, Graduate School of Life Sciences, Julius-Maximilians-Universität Würzburg.

Farris, S. M. (2011). Are mushroom bodies cerebellum-like structures? *Arthropod Structure and Development*, 40(4):368–379.

Fremaux, N., Sprekeler, H., and Gerstner, W. (2010). Functional Requirements for Reward-Modulated Spike-Timing-Dependent Plasticity. *Journal of Neuroscience*, 30(40):13326–13337.

Fung, Y.-c. (2013). *Biomechanics: mechanical properties of living tissues.* Springer Science & Business Media.

Galloway, K. C., Becker, K. P., Phillips, B., Kirby, J., Licht, S., Tchernov, D., Wood, R. J., and Gruber, D. F. (2016). Soft Robotic Grippers for Biological Sampling on Deep Reefs. *Soft Robotics*, 3(1):23–33.

Geng, T., Porr, B., and Wörgötter, F. (2006a). Fast biped walking with a sensor-driven neuronal controller and real-time online learning. *The International Journal of Robotics Research*, 25(3):243–259.

Geng, T., Porr, B., and Wörgötter, F. (2006b). A reflexive neural network for dynamic biped walking control. *Neural computation*, 18(5):1156–1196.

Gerber, B. and Stocker, R. F. (2007). The drosophila larva as a model for studying chemosensation and chemosensory learning: A review. *Chemical Senses*, 32(1):65–89.

Gozen, B. A., Tabatabai, A., Ozdoganlar, O. B., and Majidi, C. (2014). High-density soft-matter electronics with micron-scale line width. *Advanced Materials*, 26(30):5211–5216.

Heckscher, E. (2013). Drosophila Reverse Crawling Segment and Gut Movements.

Heckscher, E. S., Lockery, S. R., and Doe, C. Q. (2012). Characterization of Drosophila Larval Crawling at the Level of Organism, Segment, and Somatic Body Wall Musculature. *Journal of Neuroscience*, 32(36):12460–12471.

Heess, N., Hunt, J. J., Lillicrap, T. P., and Silver, D. (2015). Memory-based control with recurrent neural networks. *arXiv preprint arXiv:1512.04455*.

Heess, N., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, A., and Riedmiller, M. (2017). Emergence of Locomotion Behaviours in Rich Environments. *arXiv preprint arXiv:1707.02286*.

Henri, M. A. (1953). Elastic diaphragm. US Patent 2,642,091.

Herculano-Houzel, S. (2009). The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in Human Neuroscience*, 3:31.

Hige, T., Aso, Y., Modi, M. N., Rubin, G. M., and Turner, G. C. (2015). Heterosynaptic Plasticity Underlies Aversive Olfactory Learning in Drosophila. *Neuron*, 88(5):985–998.

Ho, J. and Ermon, S. (2016). Generative Adversarial Imitation Learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hughes, C. L. and Thomas, J. B. (2007). A sensory feedback circuit coordinates muscle activity in Drosophila. *Molecular and Cellular Neuroscience*, 35(2):383–396.

Hull, C. L. (1943). *Principles of behavior: an introduction to behavior theory*. Appleton-Century.

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. (2017). Imitation Learning: A survey of Learning Methods. *ACM Computing Surveys*, 50(2):1–35.

Hwang, C. L., Chen, B. L., Syu, H. T., Wang, C. K., and Karkoub, M. (2016). Humanoid robot's visual imitation of 3-D motion of a human subject using neural-network-based inverse kinematics. *IEEE Systems Journal*, 10(2):685–696.

Itoh, K., Komatsu, A., and Nishihara, S. (2016). Electrophysiological recording in the Drosophila larval muscle.

Izhikevich, E. M. (2007). *Dynamical Systems in Neuroscience:The Geometry of Excitability and Bursting*. MIT press.

Kane, E. A., Gershow, M., Afonso, B., Larderet, I., Klein, M., Carter, A. R., de Bivort, B. L., Sprecher, S. G., and Samuel, A. D. T. (2013). Sensorimotor structure of Drosophila larva phototaxis. *Proceedings of the National Academy of Sciences*, 110(40):E3868–E3877.

Kearns, M. and Singh, S. (2002). Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232.

Kelasidi, E., Andrikopoulos, G., Nikolakopoulos, G., and Manesis, S. (2011). A survey on pneumatic muscle actuators modeling. In *2011 IEEE International Symposium on Industrial Electronics*, pages 1263–1269. IEEE.

Kemp, C. C., Edsinger, A., and Torres-Jara, E. (2007). Challenges for robot manipulation in human environments [Grand challenges of robotics]. *IEEE Robotics and Automation Magazine*, 14(1):20–29.

Kober, J., Bagnell, J. A., and Peters, J. (2014). Reinforcement Learning in Robotics: A Survey. *Springer Tracts in Advanced Robotics*, 97:9–67.

Kober, J., Bagnell, J. A., and Peters, J. (2015). Reinforcement learning in robotics. *The International Journal of Robotics Research*, 32(11):1238–1274.

Kramer, R. K., Majidi, C., Sahai, R., and Wood, R. J. (2011). Soft curvature sensors for joint angle proprioception. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1919–1926. IEEE.

Kukliński, K., Fischer, K., Marhenke, I., Kirstein, F., Maria, V., Sølvason, D., Krüger, N., and Savarimuthu, T. R. (2014). Teleoperation for learning by demonstration: Data glove versus object manipulation for intuitive robot control. In *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2014 6th International Congress on*, pages 346–351. IEEE.

Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Lahiri, S., Shen, K., Klein, M., Tang, A., Kane, E., Gershow, M., Garrity, P., and Samuel, A. D. T. (2011). Two alternating motor programs drive navigation in Drosophila larva. *PLoS ONE*, 6(8).

Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Lee, C., Kim, M., Kim, Y. J., Hong, N., Ryu, S., Kim, H. J., and Kim, S. (2017). Soft robot review. *International Journal of Control, Automation and Systems*, 15(1):3–15.

Li, W. and Fritz, M. (2015). Teaching robots the use of human tools from demonstration with non-dexterous end-effectors. *IEEE-RAS International Conference on Humanoid Robots*, 2015-Decem:547–553.

Liljebäck, P., Stavdahl, Ø., and Beitnes, A. (2006). SnakeFighter - Development of a water hydraulic fire fighting snake robot. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lim, J., Park, H., Moon, S., and Kim, B. (2008). Pneumatic robot based on inchworm motion for small diameter pipe inspection. *2007 IEEE International Conference on Robotics and Biomimetics, ROBIO*, pages 330–335.

Lin, A. C., Bygrave, A. M., de Calignon, A., Lee, T., and Miesenböck, G. (2014). Sparse, decorrelated odor coding in the mushroom body enhances learned odor discrimination. *Nature neuroscience*, 17(4):559–68.

Lin, H.-T., Leisk, G. G., and Trimmer, B. (2011). GoQBot: a caterpillar-inspired soft-bodied rolling robot. *Bioinspiration & biomimetics*, 6(2):026007.

Liu, Y., Wang, H., Zhao, W., Zhang, M., Qin, H., and Xie, Y. (2018). Flexible, stretchable sensors for wearable health monitoring: sensing mechanisms, materials, fabrication strategies and features. *Sensors*, 18(2):645.

Loveless, J., Lagogiannis, K., and Webb, B. (2018). Mechanics of exploration in drosophila melanogaster. *bioRxiv*.

Lu, H., Zhu, J., Lin, Z., and Guo, Y. (2009). An inchworm mobile robot using electromagnetic linear actuator. *Mechatronics*, 19(7):1116–1125.

Lungarella, M., Metta, G., Pfeifer, R., and Sandini, G. (2003). Developmental robotics: a survey. *Connection Science*, 15(4):151–190.

Majidi, C., Kramer, R., and Wood, R. J. (2011). A non-differential elastomer curvature sensor for softer-than-skin electronics. *Smart Materials and Structures*, 20(10):105017.

Marchese, A. D., Katzschmann, R. K., and Rus, D. (2015). A Recipe for Soft Fluidic Elastomer Robots. *Soft Robotics*, 2(1):7–25.

Martinez, R. V., Glavan, A. C., Keplinger, C., Oyetibo, A. I., and Whitesides, G. M. (2014). Soft actuators and robots that are resistant to mechanical damage. *Advanced Functional Materials*, 24(20):3003–3010.

Maye, A., Hsieh, C. H., Sugihara, G., and Brembs, B. (2007). Order in spontaneous behavior. *PLoS ONE*, 2(5):443.

Menciassi, A., Accoto, D., Gorini, S., and Dario, P. (2006). Development of a biomimetic miniature robotic crawler. *Autonomous Robots*, 21(2):155–163.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. (2013). Playing Atari with Deep Reinforcement Learning. *arXiv preprint arXiv:1312.5602*, pages 1–9.

Mohd Jani, J., Leary, M., Subic, A., and Gibson, M. a. (2014). A review of shape memory alloy research, applications and opportunities. *Materials and Design*, 56:1078–1113.

Mori, T., Nakamura, Y., Sato, M.-a., and Ishii, S. (2004). Reinforcement Learning for a CPG-driven Biped Robot. In *Aaai 2004*, pages 623–630.

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. (2004a). Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2-3):79–91.

Nakanishi, J., Morimoto, J., Endo, G., Cheng, G., Schaal, S., and Kawato, M. (2004b). Learning from demonstration and adaptation of biped locomotion. *Robotics and autonomous systems*, 47(2-3):79–91.

Olsen, L. F. and Degn, H. (1985). Chaos in biological systems. *Quarterly Reviews of Biophysics*, 18(02):165.

Olsen, S. R. and Wilson, R. I. (2008). Cracking neural circuits in a tiny brain: new approaches for understanding the neural circuitry of Drosophila. *Trends in Neurosciences*, 31(10):512–520.

Paynter, H. M. (1988). Hyperboloid of revolution fluid-driven tension actuators and method of making. US Patent 4,721,030.

Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M. (2018a). Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):143.

Peng, X. B., Andrychowicz, M., Zaremba, W., and Abbeel, P. (2018b). Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE.

Pierson, H. A. and Gashler, M. S. (2017). Deep learning in robotics: a review of recent research. *Advanced Robotics*, 31(16):821–835.

Plappert, M., Houthooft, R., Dhariwal, P., Sidor, S., Chen, R. Y., Chen, X., Asfour, T., Abbeel, P., and Andrychowicz, M. (2018). Parameter Space Noise for Exploration. In *International Conference on Learning Representations*, pages 1–18.

Porr, B., Ferber, C. v., and Wörgötter, F. (2003). Iso learning approximates a solution to the inverse-controller problem in an unsupervised behavioral paradigm. *Neural Computation*, 15(4):865–884.

Porr, B. and Wörgötter, F. (2003). Isotropic sequence order learning. *Neural Computation*, 15(4):831–864.

Puterman, M. L. (2014). *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Putz, G. and Heisenberg, M. (2002). Memories in Drosophila heat-box learning. *Learning & memory*, 9(5):349–359.

Rescorla, R. A., Wagner, A. R., et al. (1972). A theory of pavlovian conditioning: Variations in the effectiveness of reinforcement and nonreinforcement. *Classical conditioning II: Current research and theory*, 2:64–99.

Rus, D. and Tolley, M. T. (2015). Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475.

Saksida, L. M., Raymond, S. M., and Touretzky, D. S. (1997). Shaping robot behavior using principles from instrumental conditioning. *Robotics and Autonomous Systems*, 22(3-4):231–249.

Sato, M. (1990). A real time learning algorithm for recurrent analog neural networks. *Biological Cybernetics*, 62(3):237–241.

Schaal, S. and Atkeson, C. G. (1998). Constructive incremental learning from only local information. *Neural computation*, 10(8):2047–2084.

Schaal, S., Peters, J., Nakanishi, J., and Ijspeert, A. (2005). Learning movement primitives. *Robotics Research*, 15(2005):1–10.

Schmidhuber, J. (2015). Deep Learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, pages 1–12.

Seung, S. (2003). Learning in Spiking Neural Networks by Reinforcement of Stochastics Transmission. *Neuron*, 40:1063–1073.

Sheng, M. and Hoogenraad, C. C. (2007). The postsynaptic architecture of excitatory synapses: a more quantitative view. *Annual review of biochemistry*, 76:823–847.

Shixin Mao, Erbao Dong, Shiwu Zhang, Min Xu, and Jie Yang (2013). A new soft bionic starfish robot with multi-gaits. In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 1312–1317. IEEE.

Sigaud, O. and Peters, J. (2012). *Encyclopedia of the Sciences of Learning*, chapter Robot Learning, pages 2869–2871. Springer US, Boston, MA.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). Deterministic Policy Gradient Algorithms. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 387–395.

Skinner, B. F. (1938). *The behavior of organisms: An experimental analysis.* BF Skinner Foundation.

Song, D. R., Yang, C., McGreavy, C., and Li, Z. (2017). Recurrent network-based deterministic policy gradient for solving bipedal walking challenge on rugged terrains. *arXiv preprint arXiv:1710.02896*.

Staddon, J. E. R. and Cerutti, D. T. (2003). Operant Conditioning. *Annual Review of Psychology*, 54(1):115–144.

Steigenberger, J. and Behn, C. (2011). Gait generation considering dynamics for artificial segmented worms. *Robotics and Autonomous Systems*, 59(7-8):555–562.

Steingrube, S., Timme, M., Woergoetter, F., and Manoonpong, P. (2011). Self-organized adaptation of a simple neural circuit enables complex robot behaviour. *Nature Physics*, 6(3):16.

Stokes, A. A., Shepherd, R. F., Morin, S. A., Ilievski, F., and Whitesides, G. M. (2014). A Hybrid Combining Hard and Soft Robots. *Soft Robotics*, 1(1):70–74.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. The MIT Press.

Takemura, S. y., Aso, Y., Hige, T., Wong, A., Lu, Z., Xu, C. S., Rivlin, P. K., Hess, H., Zhao, T., Parag, T., Berg, S., Huang, G., Katz, W., Olbris, D. J., Plaza, S., Umayam, L., Aniceto, R., Chang, L. A., Lauchie, S., Ogundeyi, O., Ordish, C., Shinomiya, A., Sigmund, C., Takemura, S., Tran, J., Turner, G. C., Rubin, G. M., and Scheffer, L. K. (2017). A connectome of a learning and memory center in the adult Drosophila brain. *eLife*, 6:1–43.

Trivedi, D., Rahn, C. D., Kier, W. M., and Walker, I. D. (2008). Soft robotics: Biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics*, 5(3):99–117.

Vaidyanathan, R., Chiel, H. J., and Quinn, R. D. (2000). Hydrostatic robot for marine applications. *Robotics and Autonomous Systems*, 30(1):103–113.

Villegas, D., Van Damme, M., Vanderborght, B., Beyl, P., and Lefeber, D. (2012). Third generation Pleated Pneumatic Artificial Muscles for Robotic Applications: Development and Comparison with McKibben Muscle. *Advanced Robotics*, 26(11-12):1205–1227.

Wang, W., Wang, Y., Qi, J., Zhang, H., and Zhang, J. (2008). The CPG control algorithm for a climbing worm robot. *2008 3rd IEEE Conference on Industrial Electronics and Applications, ICIEA 2008*, pages 675–679.

Wawrzyński, P. (2015). Control Policy with Autocorrelated Noise in Reinforcement Learning for Robotics. *International Journal of Machine Learning and Computing*, 5(2):91–95.

Webb, B. (2001). Can robots make good models of biological behaviour? *Behavioral and brain sciences*, 24(6):1033–1050.

Wei, T., Stokes, A., and Webb, B. (2016). A Soft Pneumatic Maggot Robot. In Lepora, N. F., Mura, A., Mangan, M., Verschure, P. F. M. J., Desmulliez, M., and Prescott, T. J., editors, *Biomimetic and Biohybrid Systems: 5th International Conference, Living Machines 2016, Edinburgh, UK, July 19-22, 2016. Proceedings*, pages 375–386. Springer International Publishing, Cham.

Wei, T. and Webb, B. (2018a). A model of operant learning based on chaotically varying synaptic. *Neural Networks*, 108:114–127.

Wei, T. and Webb, B. (2018b). A bio-inspired reinforcement learning rule to optimise dynamical neural networks for robot control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 556–561. IEEE.

Weng, J. (2004). Developmental Robotics: Theory and Experiments. *International Journal of Humanoid Robotics*, 01(02):199–236.

Wessnitzer, J., Young, J. M., Armstrong, J. D., and Webb, B. (2012). A model of non-elemental olfactory learning in Drosophila. *Journal of Computational Neuroscience*, 32(2):197–212.

Wiering, M. and Van Otterlo, M. (2012). Reinforcement learning. *Adaptation, learning, and optimization*, 12:3.

Window, A. L., editor (1992). *Strain Gauge Technology (2nd ed.)*. Butterworth-Heinemann.

Wystrach, A., Lagogiannis, K., and Webb, B. (2016). Continuous lateral oscillations as a core mechanism for taxis in drosophila larvae. *eLife*, 5:e15504.

Yagi, R., Mabuchi, Y., Mizunami, M., and Tanaka, N. K. (2016). Convergence of multimodal sensory pathways to the mushroom body calyx in drosophila melanogaster. *Scientific reports*, 6:29481.

Yamada, T., Hayamizu, Y., Yamamoto, Y., Yomogida, Y., Izadi-Najafabadi, A., Futaba, D. N., and Hata, K. (2011). A stretchable carbon nanotube strain sensor for human-motion detection. *Nature Nanotechnology*, 6(5):296–301.

Yariott, J. M. (1972). Fluid actuator. US Patent 3,645,173.

Zhu, Z. and Hu, H. (2018). Robot Learning from Demonstration in Robotic Assembly: A Survey. *Robotics*, 7(2):17.