Self-Supervised Learning of the Biologically-Inspired Obstacle Avoidance of Hexapod Walking Robot

Petr Čížek, Jan Faigl

Faculty of Electrical Engineering, Czech Technical University in Prague,
Technicka 2, 166 27, Prague, Czech Republic;

E-mail: {petr.cizek|faiglj}@fel.cvut.cz

Abstract. In this paper, we propose an integrated biologically inspired visual collision avoidance approach that is deployed on a real hexapod walking robot. The proposed approach is based on the Lobula Giant Movement Detector (LGMD) that is a neural network for looming stimuli detection that can be found in visual pathways of insects, such as locusts. Although a superior performance of the LGMD in the detection of intercepting objects has been shown in many collision avoiding scenarios, its direct integration with motion control is an unexplored topic. In our work, we propose to utilize the LGMD neural network for visual interception detection with a Central Pattern Generator (CPG) for locomotion control of a hexapod walking robot that are combined in the controller based on the Long Short-Term Memory (LSTM) recurrent neural network. Moreover, we propose self-supervised learning of the integrated controller to autonomously find a suitable setting of the system using a realistic robotic simulator. Thus, individual neural networks are trained in a simulation to enhance the performance of the controller that is then experimentally verified with a real hexapod walking robot in both collision and interception avoidance scenario and navigation in a cluttered environment.

Keywords: Lobula Giant Movement Detector, Central Pattern Generator, Collision Avoidance, Hexapod Robot, Self-Supervised Learning

Submitted to: *Bioinspir. Biomim.*

1. Introduction

Timely detection of looming stimuli and subsequent collision avoidance are vital survival abilities for many animals. Similarly, for a mobile robot moving from one place to another, the contact with a fixed or moving object may have fatal consequences. An abundant source of inspiration to develop a system for avoiding such situations can be taken from biological systems that nurtured by millions of years of evolution. In particular, we are focused on modeling artificial visual systems for collision avoidance as such dedicated neural structures to detect looming stimuli have been found in visual pathways of insects and vertebrates. The most studied examples of the looming-sensitive neurons that selectively reacts to the divergence of image edges are the Lobula Giant Movement Detector (LGMD) described by Wilson (1961), Directional Selective Neurons (DSN) (Judge & Rind 1997) of the locusts, and Mauthner cell (Preuss, Osei-Bonsu, Weiss, Wang & Faber 2006) of goldfish. Besides, correlational motion detectors of fruit flies have been recently discovered by Zabala, Polidoro, Robie, Branson, Perona & Dickinson (2012) whose work has been adopted by Chalupka, Dickinson & Perona (2016) to provide a geometrical analysis of the observed behavior.

Further studies reveal similar looming detectors in the neural structure of other animals which approve that the dedicated neural pathways for the detection of looming stimuli are common in biological systems. However, in the biological systems, the exact mapping between the looming stimuli perception and muscle action is most elusive. Works by Fotowat, Fayyazuddin, Bellen & Gabbiani (2009) and De Vries & Clandinin (2012) suggest the looming stimuli mediate the escape behavior in Drosophila; however, the escape behavior is distinct to collision avoidance (Chalupka et al. 2016) where the visual stimuli are used directly in the sensory-motor connection. The direct sensory motor connection to achieve the collision avoidance behavior has been suggested by Blanchard, Rind & Verschure (2000) and Preuss et al. (2006); however, the underlying biological mechanisms are yet to be discovered.

In mobile robotics, the problem of collision avoidance has been studied since the mobile robots appeared. Hence, there are many different approaches using a variety of sensors and processing techniques (Hoy, Matveev & Savkin 2015). The importance of low-complexity collision avoidance grows as fields such as drone flight control are rapidly developing while the traditional techniques often require significant computational power. Due to its simplicity and superior performance, the LGMD has been modeled and promoted (Blanchard et al. 2000, Yue & Rind 2013, Salt, Howard, Indiveri & Sandamirskaya 2017), and deployed on various robotic platforms including wheeled (Yue & Rind 2006, Fu, Yue & Hu 2016), legged (Čížek, Milička & Faigl 2017, Čížek, Faigl & Bayer 2017) and flying (Salt et al. 2017, Badia, Pyk & Verschure 2005, Zhao, Hu, Zhang, Wang & Yue 2018) robots, but also considered for bio-inspired vehicle collision detection system (Hartbauer 2017). However, the existing approaches are focused on the ability of the system to detect looming stimuli and only trigger the escape behavior, omitting the underlying locomotion control which drives the robot. In our approach, we follow the hypothesis of the direct sensory motor connection (Blanchard et al. 2000, Preuss et al. 2006) and combine the LGMD with a bio-inspired locomotion control in a fully mobile robot navigation stack relying on visual stimuli only. The proposed approach

Figure 1. Utilized hexapod walking robot with mounted wide-angle lens stereo camera.

is supposed to drive a real hexapod walking robot shown in Figure 1 through cluttered environments while avoiding static and dynamic obstacles based on visual stimuli only.

The proposed approach forms an end-to-end neural network that maps the visual perception represented by the stereo camera image on the control command given as the robot joint angles. For that, it utilizes a pair of LGMD neural networks for visual interception detection (Blanchard et al. 2000) with a Central Pattern Generator (CPG) (Szadkowski, Čížek & Faigl 2018) for locomotion control of a hexapod walking robot. The LGMD networks and CPG locomotion control are combined in the controller (Čížek, Faigl & Bayer 2017) based on the Long Short-Term Memory (LSTM) recurrent neural network. The ability to reliably avoid obstacles in the environment is enabled by the herein proposed learning approach that uses self-supervised learning based on the evaluation of the Braitenberg vehicle (Braitenberg 1986) behavior which constitutes the basic cognitive AI sensory-motor model. The controller is learned in a realistic simulation of the utilized hexapod walking robot (Nguyenová, Čížek & Faigl 2019) to imitate the behavior of a type 3c Braitenberg vehicle, and its performance is verified in real-world experiments.

In comparison to our previous work (Čížek, Milička & Faigl 2017, Čížek, Faigl & Bayer 2017) we have reconsidered the previously employed locomotion control that relies on inverse kinematics to provide the desired joint angles and developed a new CPG-based locomotion control that provides the desired joint angles directly. Besides, the herein proposed self-supervised learning improves the real world behavior of the robot. In particular, the contributions of the presented work regarding the previous work and the related approaches are considered as follows.

- Novel complete biologically-inspired visual collision avoidance system.
- Self-supervised learning of the developed biologically-inspired visual collision avoidance system using a cognitive model of the sensory-motor connections.
- Experimental evaluation of the proposed system in real-world experiments,
- particularly, with a deployment of the LGMD collision avoidance in cluttered environments.

The remainder of the paper is organized as follows. An overview of the most related neural-based collision avoidance approaches is summarized in Section 2. Section 3 details the individual building blocks of the proposed control architecture. A detailed description of the proposed self-supervised learning approach together with the description of the real and simulated hexapod robot is presented in Section 4. Section 5 reports on the performed simulated and real-world verification experiments. Concluding remarks and suggestions for future work are dedicated to Section 6.

2. Related Work

In the presented work, we propose a biologically-inspired control system that drives the hexapod walking robot through an environment while avoiding obstacles using only the visual perception. The proposed architecture consists of the looming stimuli detector and the locomotion controller connected in a learnable navigation system. Therefore, we report on the most related work on the looming stimuli detectors and also multi-legged locomotion control.

Reported biological findings support that there are neural pathways explicitly dedicated to the detection of the looming stimuli. Such neural networks that have also been promoted in robotics are the LGMD (Wilson 1961) and DSN (Judge & Rind 1997). Although the DSN better estimates the direction of the interception, the LGMD has shown to provide superior ability in overall looming stimuli detection (Yue & Rind 2013). The problem of direction estimation from the LGMD has been solved by splitting the visual perception between the left and right visual pathways with separate LGMDs (Fu et al. 2016). Similarly, Chalupka et al. (2016) have shown the importance of the division of the perception between the left and right hemifield in the Generalized Regressive Motion (GRM) approach. Hence, in our work, we utilize a stereo camera with a wide field of view, with two LGMDs to provide the robot with direction sensitivity to collision detection.

Multi-legged locomotion control is another field that builds on the biological findings to a great extent. In particular, approaches based on the CPGs (Ijspeert 2008) and locomotion templates (Miller & Clark 2015) are getting increasing attention in robotics, due to their robustness and unnecessity to construct the kinematic model of the robot. On the other hand, the major drawback of the most bio-inspired approaches is difficult parametrization which we have solved recently by introducing the backpropagation into the learning of the network of Matsuoka CPG oscillators (Szadkowski et al. 2018) that we use to drive our hexapod walking robot. It is a major improvement in comparison to our previous work (Čížek, Faigl & Bayer 2017) that utilizes single CPG and inverse kinematics to transform the CPG output to joint commands.

Moreover, the crucial part of the proposed approach is the learnable navigation controller that provides coupling between the visual perception and locomotion of the robot. In the previous approaches utilizing LGMD with different robotic platforms such as (Yue & Rind 2006, Fu et al. 2016, Salt et al. 2017, Badia et al. 2005, Zhao et al. 2018), the LGMD has been used only to trigger the collision evasion maneuver, while in the herein proposed approach, we are following the continuous mapping approach suggested by Blanchard et al. (2000) and Preuss et al. (2006).

The end-to-end mapping of the visual perception on the robot control command using neural networks has been utilized in multiple methods, mainly with imitation learning where neural networks learn from human operated trials (Muller, Ben, Cosatto, Flepp & Cun 2006, Loquercio, Maqueda, del Blanco & Scaramuzza 2018, Kelchtermans & Tuytelaars 2017). Different architectures including a six-layer convolutional neural network (Muller et al. 2006), deep convolutional neural network (Loquercio et al. 2018) or recurrent neural network (Kelchtermans & Tuytelaars 2017) have been presented, which are not strictly bio-inspired. However, all of these methods require a large amount of learning data that are usually provided by human-guided examples.

In (Gandhi, Pinto & Gupta 2017), the authors tackle the particular problem of the importance of the correct selection of the learning examples as the positive human guided trajectories are the bottleneck in the machine learning of steering policies. It is because there is not usually enough training data for high-capacity learning of the convolutional neural networks and the provided trajectories mostly contain only positive examples. Besides, the human operator exploits a high-level semantic knowledge of the environment, which is usually unknown to the robot.

In contrast to the listed approaches such as (Muller et al. 2006, Loquercio et al. 2018, Kelchtermans & Tuytelaars 2017, Gandhi et al. 2017), we are exploiting biologically-inspired detection of the looming stimuli by utilizing the LGMD neural network that simplifies the architecture of the proposed controller and which only needs a fraction of training examples to learn a proper control policy. In our previous work (Čížek, Faigl & Bayer 2017), imitation learning is also used to learn the collision avoiding behavior; however, imitation learning has shown to be ineffective for the herein presented full navigation task. Therefore, we investigated learning in the realistic robotic simulator and propose a self-supervised learning approach based on the cognitive model of the Braitenberg vehicle (Braitenberg 1986), which has also been used by Chalupka et al. (2016) for collision avoidance in simulated GRM experiments.

3. Neural-based Control Architecture

The proposed interception detection and collision avoidance system is considered with a hexapod walking robot, and it comprises of three building blocks: locomotion controller, detection of intercepting objects, and neural-based navigation controller. The locomotion control is based on a network of interconnected chaotic oscillators (Matsuoka 1987) that produce a rhythmic patterned output that resembles the joint angles values to control the locomotion of the hexapod robot. The CPG locomotion controller is parametrized by differential steering commands v^{left} and v^{right} that represent the velocities of the left and right legs, respectively. The direction of the interception is determined from the visual stimuli using the LGMD neural network (Blanchard et al. 2000) that responds selectively to objects approaching the robot on a collision course. Two LGMD networks are utilized for separate processing of the signals to detect interception in the left and right hemifield, respectively, and thus determine the direction of the interception selectively responding to looming stimuli by the excitation of the u^{left} and u^{right} outputs. Finally, the LGMD outputs u^{left} and u^{right} are fused in a neural-based navigational controller which is implemented using the Recurrent Neural Network (RNN). The controller affects the behavior of the CPG locomotion controller and drives the robot through the environment by setting appropriate values of the steering commands v^{left} and v^{right}. The RNN-based approach is utilized as it follows on the idea of the direct sensory-motor map-

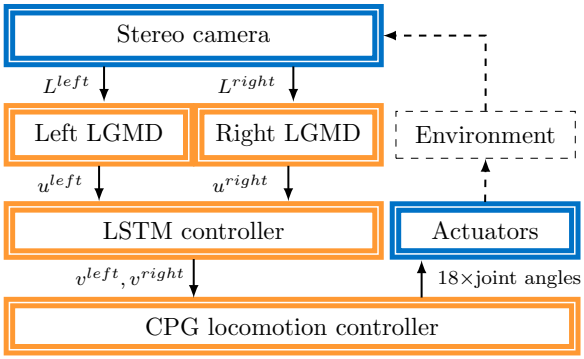
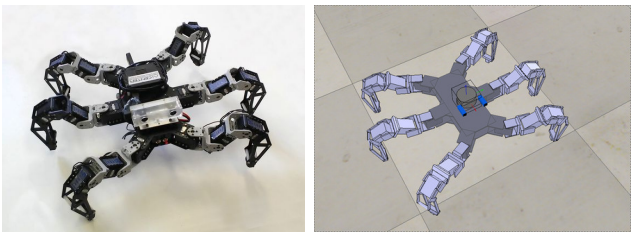


Figure 2. Overview of the proposed control architecture.



(a) (b)

Figure 3. (a) Real hexapod robot in the default configuration with the mounted wide-angle lens stereo camera and (b) model of the hexapod robot in the realistic robotic simulator V-REP. The model of the robot is equipped with the simulated stereo camera and simulated laser-scanner utilized in the proposed self-supervised learning.

ping (Blanchard et al. 2000) and it has already shown superior performance in comparison to the simple feed-forward solution (Čížek, Faigl & Bayer 2017). The controller is learned using a self-supervised approach presented in Section 4. The overall schema of the proposed system is depicted in Figure 2. Following sections describe the utilized robotic platform together with the simulation model, which is essential in the proposed self-supervised learning, and individual building blocks of the proposed neural architecture.

3.1. Robotic Platform and Simulator

In the proposed self-supervised learning of the collision avoidance behavior, we consider a small and affordable electrically actuated hexapod walking robot built of off-the-shelf components. The robot is approx. 45×40 cm large when standing in the default configuration, see Figure 3a. The robot has six legs, each with three actuators, attached to the trunk that hosts the controller and sensors. The robot is equipped with a wide-angle lens (170°) stereo camera with 6.5 cm baseline, 320×240 resolution, and 30 fps that provides the visual input (L^{left}, L^{right}) to the left and right LGMD networks.

A high-fidelity model of the hexapod robot (see Figure 3b) has been developed (Nguyenová et al. 2019) in the realistic simulator V-REP (Rohmer, Singh & Freese 2013). The simulated robot is equipped with a wide-angle lens stereo camera with the same parameters as the real camera and a 2D laser scanner sensor with the scanning angle 170° and scanning density 2 points per 1° that is placed 3 cm above the geometric center of the simulated robot. The simulated laser scanner plays a crucial role in the proposed self-supervised learning approach, detailed in Section 4 because it provides the learning feedback as a distance of the robot to nearby obstacles.

3.2. CPG-based Locomotion Control

The hexapod walking robot employs a CPG-based controller for the locomotion control. The CPGs are biologically-inspired locomotion controllers (Ijspeert 2008) that generate synchronized rhythmic signals to control locomotion of multi-legged robots. In our work, we used the CPG based on (Szadkowski et al. 2018) with backpropagation learning of the CPG network parametrization.

The controller consists of a set of $N = 6$ symmetrically interconnected CPGs, where each CPG (per each of the six robot legs) is a pair of Matsuoka's (Matsuoka 1987) adaptive neurons: extensor neuron and flexor neuron. Each CPG is connected through a simple output shaping network to shape the CPG oscillations directly to the desired joint angle values as every leg consists of three actuators. Hence, the CPG network is designed to maintain stable oscillations and inter-leg synchronization (Frigon & Rossignol 2006) whereas the output shaping network is

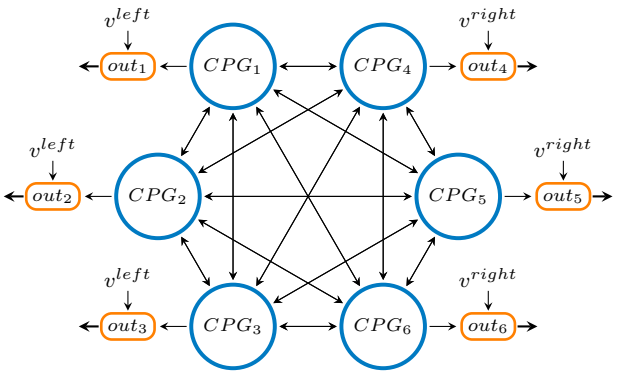


Figure 4. The overall scheme of the CPG-based locomotion controller. Individual CPGs are connected with symmetric inhibitive connections. The output of each CPG is shaped using a simple neural network to produce joint angle values directly used to control the robot actuators. The shaping networks are parametrized by the steering command.

responsible for the control of the leg joints and allows a simple differential steering by the requested v^{left} and v^{right} velocities for left and right half of the body, respectively. In comparison to the approaches based on the inverse kinematics for the output shaping of the CPG network outputs (Yu, Gao, Ding, Li, Deng & Liu 2016, Zhong, Chen, Jiao, Li & Deng 2018), the utilized solution supports greater motion capabilities (Xiong, Wörgötter & Manoonpong 2016) and does not require identification of the robot kinematic model. The proposed differential steering is mainly utilized to verify the feasibility of the presented self-supervised learning of the locomotion controller. The scheme of the locomotion controller is depicted in Figure 4.

The dynamics of the CPG network containing N units can be described by the following equations.

$$T_r \dot{u}_i^e = -u_i^e - w_{fe} g(r_i^f) - \beta v_i^e - \sum_{j=1}^{N} w_{ij} g(u_j^e) + c_i^e, \quad (1)$$

$$T_a \dot{v}_i^e = g(u_i^e) - v_i^e, \quad (2)$$

$$T_r \dot{u}_i^f = -u_i^f - w_{fe} g(r_i^f) - \beta v_i^f - \sum_{j=1}^{N} w_{ij} g(u_j^f) + c_i^f, \quad (3)$$

$$T_a \dot{v}_i^f = g(u_i^f) - v_i^f, \quad (4)$$

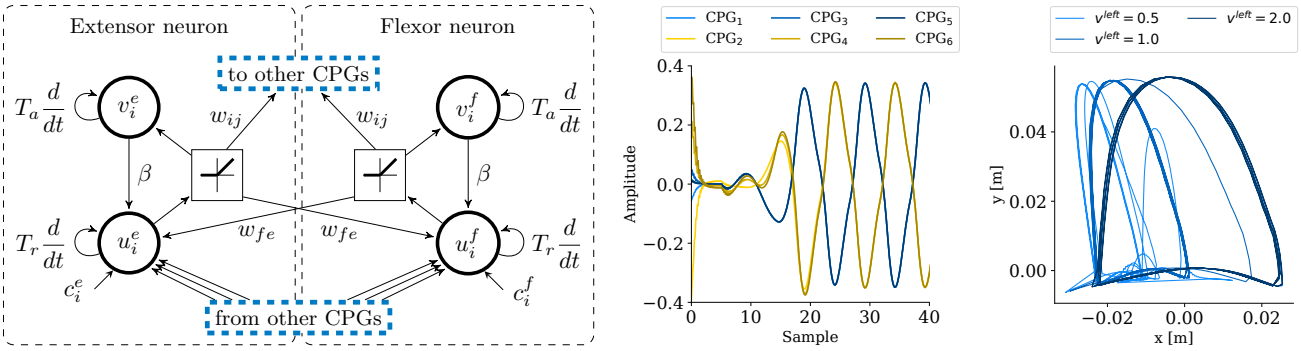
where $i \in N$ is an index of the particular CPG. The variables u_i^e and v_i^e represent the activation and self-inhibition of the extensor neuron, respectively. Besides, u_i^f and v_i^f describe the dynamics of the flexor neuron. The function g is the rectifier activation function

$$g(x) = \max(0, x). \quad (5)$$

Each neuron inhibits itself through the variable v_i scaled by the parameter $\beta > 0$. The extensor-flexor pair (i.e., the CPG unit) mutually inhibits itself through the symmetric connection with the weight $w_{fe} > 0$. Besides, each two CPG oscillators i and j in the CPG network are interconnected with symmetric inhibitive connection of the weight $w_{ij} \in W$ ($w_{ij} \geq 0$, $w_{ii} = 0$), where W is a symmetric matrix. Hence, the matrix W is responsible for synchronization of the individual CPGs to produce the desired gait pattern. The only source of excitation for the CPG network is the tonic input c_i^e, c_i^f (≥ 0) which is given externally and in our work, it is set to $c_i^e = c_i^f = 1$. $T_r > 0$ and $T_a > 0$ are the reaction times for their respective variables. The structure of a single CPG unit is visualized in Figure 5a.

The output of the i-th oscillator is given as the difference between the extensor and flexor neuron activation

$$CPG_i = u_i^e - u_i^f. \quad (6)$$



(a) (b) (c)

Figure 5. (a) The scheme of a single CPG unit. (b) CPG oscillations for a tripod gait when three legs swing at a time. Note, due to a random state initialization, there is a transition effect before the CPG network enters stable oscillations. (c) Leg foot-tip trajectory of the first leg for velocities $v^{left} \in \{0.5, 1.0, 2.0\}$. Notice the CPG transition effect on the leg foot-tip trajectory.

For the CPG network weight matrix

$$W = \begin{bmatrix} 0 & 0.04 & 0.40 & 0.04 & 0.40 & 0.04 \\ 0.04 & 0 & 0.04 & 0.40 & 0.04 & 0.40 \\ 0.40 & 0.04 & 0 & 0.04 & 0.40 & 0.04 \\ 0.04 & 0.40 & 0.04 & 0 & 0.04 & 0.40 \\ 0.40 & 0.04 & 0.40 & 0.04 & 0 & 0.04 \\ 0.04 & 0.40 & 0.04 & 0.40 & 0.04 & 0 \end{bmatrix},$$

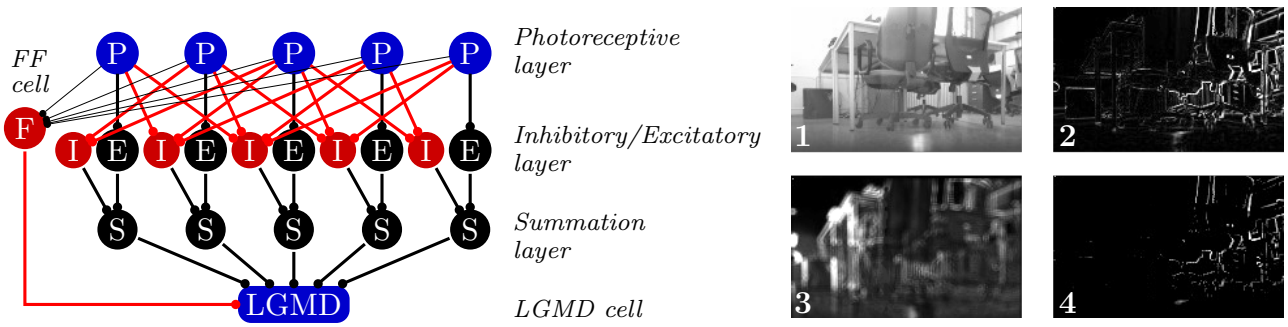
$T_r = 1.7$, $T_a = 1.5$, $\beta = 7$ and $w_{fe} = -4$; the CPG network produces stable oscillations that resemble a tripod locomotion gait with two opposite waves produced per three CPGs as it is visualized in Figure 5b.

The CPG output signal has to be further shaped to be directly used as the desired values of the joint angles to control the robot actuators. Besides, in the presented work, we are simplifying the learning of the proposed end-to-end locomotion controller by allowing simple differential steering given by the velocities of the left and right legs, respectively. Hence, there is a signal-shaping neural network that takes the output (6) of a single CPG unit and the respective v^{left} or v^{right} velocity on the input, and produces the triplet of joint angle values to be directly fed to the actuators on its output. The signal-shaping network has a single fully connected layer with six neurons, and it uses a rectifier activation function.

The output-shaping network is learned by the back-propagation algorithm to generate tripod gait motion pattern (Szadkowski et al. 2018) based on the regular locomotion gait (Mrva & Faigl 2015). The resulting trajectory of the leg foot-tip is visualized in Figure 5c for a single leg and velocities $v^{left} \in \{0.5, 1.0, 2.0\}$. It can be observed that with a larger velocity, the stride length extends which results in differential steering of the robot.

3.3. LGMD Interception Detection

The LGMD is a neural network found in the visual pathways of insects, such as locusts (Wilson 1961), which response selectively to objects approaching the animal on a collision course. The computational model of the LGMD as presented by Blanchard et al. (2000) is composed of four groups of cells: *Photoreceptive*, *Excitatory*, *Inhibitory*, and *Summation* arranged in three layers; and two individual cells: *Feed-forward inhibitory* and *Lobula Giant Movement Detector*, as it is visualized in Figure 6a.



(a) LGMD network scheme

(b) LGMD layer responses

Figure 6. (a) LGMD neural network model. (b) Visualization of responses of the LGMD layers to visual stimuli with particular responses: 1. the input image; 2. a response of the *Excitatory* layer; 3. a response of the *Inhibitory* layer; and 4. a response of the *Summation* layer. It can be observed that closer obstacles to the right of the image cause a stronger response in the *Summation* layer that further influences the *LGMD* cell output.

The *Photoreceptive layer* processes the sensory input from the camera. Its output is the difference between two successive grayscale camera frames with the size (w, h) and it is computed as

$$P_f(x, y) = L_f(x, y) - L_{f-1}(x, y), \tag{7}$$

where L_f is the current frame, L_{f-1} is the previous frame, and $x \in [1, w]$, $y \in [1, h]$ are the pixel coordinates. The *Photoreceptive layer* forms the input to the following two groups of neurons – the *Inhibition layer* and *Excitatory layer*.

The response of the *Inhibition layer* can be computed as

$$I_f(x, y) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} \left(P_{f-1}(x + i, y + j) \cdot w_I(i + n, j + n)\right), \tag{8}$$

where $n = 2$ and w_I represents the inhibition weights set with respect to the LGMD description in (Blanchard et al. 2000) as

$$w_I = \begin{bmatrix} 0.06 & 0.12 & 0.25 & 0.12 & 0.06 \\ 0.12 & 0.06 & 0.12 & 0.06 & 0.12 \\ 0.25 & 0.12 & 0 & 0.12 & 0.25 \\ 0.12 & 0.06 & 0.12 & 0.06 & 0.12 \\ 0.06 & 0.12 & 0.25 & 0.12 & 0.06 \end{bmatrix}. \tag{9}$$

The main purpose of the *Inhibition layer* is to provide lateral inhibition of the stimuli to the neighboring cells.

The *Excitatory layer* is used to delay the output of the *Photoreceptive layer* and it is calculated as

$$E_f = |P_f(x, y)|. \tag{10}$$

The response of the *Summation layer* is computed as

$$S_f(x, y) = E_f(x, y) - |I_f(x, y)| W_I, \tag{11}$$

where $W_I = 0.4$ is the global inhibition weight. The output of the *Summation layer* is then thresholded using a given threshold T_s as

$$S'_f(x, y) = \begin{cases} S_f(x, y) & \text{if } S_f(x, y) \geq T_s, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

The threshold T_s influences the sensitivity of the LGMD network that depends on the image resolution, structure of the environment (in more cluttered environment the LGMD has to be less sensitive), camera frame rate, and velocity of the robot. Therefore T_s has to be set experimentally in such a way to avoid saturation of the LGMD output.

The excitation of the *LGMD cell* can be computed as

$$U_f = \sum_{x=1}^{w} \sum_{y=1}^{h} \left| S'_f(x,y) \right|. \tag{13}$$

Finally, the *LGMD cell* output is expressed as

$$u_f = (1 + \mathrm{e}^{-U_f n_{cell}^{-1}})^{-1}, \tag{14}$$

where n_{cell} is the total number of the cells (i.e., the number of pixels). Note, the output of u_f is in the interval $u_f \in [0.5, 1]$.

The *Feed-forward cell* suppresses the output of the *LGMD cell* in a case of fast camera movements.

$$FF_{cell} = \frac{1}{n_{cell}} \sum_{x=1}^{w} \sum_{y=1}^{h} P_{f-1}(x,y). \tag{15}$$

Hence, when there is a high excitation in the *Photoreceptive layer* caused by a rapid camera movement, the LGMD output is suppressed given

$$u_f = \begin{cases} u_f & \text{if} \quad FF_{cell} > T_f \\ 0.5 & \text{otherwise} \end{cases}, \tag{16}$$

where T_f is the suppression threshold.

In our setup, two LGMD neural networks are utilized in parallel to distinguish the direction of the interception, and thus steer the robot in the opposite direction to achieve the desired obstacle avoidance behavior. A stereo camera is utilized to feed the image pairs L^{left} and L^{right} into the respective LGMD network. Each of the LGMDs provides the output denoted u^{left} and u^{right} for the left and the right LGMD, respectively, that are passed to the LSTM-based controller to produce the desired steering command.

3.4. LSTM-based Navigational Controller

The outputs from the individual LGMDs are combined in the controller based on the Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber 1997) RNN to continuously provide suitable desired values for the parametrization of the CPG network that steers the robot. The controller is necessary mainly for two reasons. First, the outputs of the LGMD are noisy and has to be filtered as a direct translation to the steering command (Čížek, Milička & Faigl 2017) has shown to be error-prone and works well only in artificially prepared experimental arenas (Blanchard et al. 2000, Zhao et al. 2018). The second reason is that the memory effect provided by the LSTM allows the robot to continue on the avoiding maneuver and successfully pass the obstacle that is not directly visible because of the avoiding maneuver.

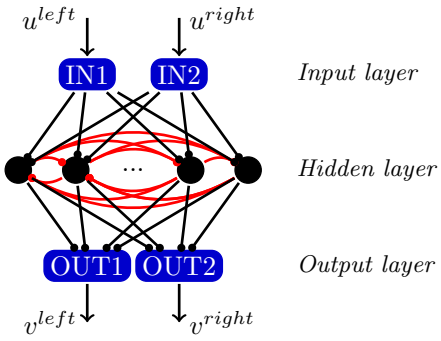


Figure 7. The scheme of the LSTM-based neural network controller.

The utilized LSTM has two inputs u^{left} and u^{right}, a single fully connected hidden layer with 32 neurons, and two outputs v^{left} and v^{right} that is visualized in Figure 7. The sigmoid function is used as the activation function for the LSTM. Finally, the LSTM is learned using the Back-propagation through time (Sutskever 2013) from the labeled data obtained using the control model based on cognitive sensory-motor model of Braitenberg vehicles (Braitenberg 1986) in the realistic robotic simulator. The description of the self-supervised learning process is presented in the following section.

4. Controller Learning

The LSTM-based navigational controller has to be learned prior deployment; hence, the mapping Φ between the looming stimuli detection (u^{left}, u^{right}) provided by the pair of the LGMD networks and the differential steering command (v^{left}, v^{right}) that parameterize the CPG locomotion controller has to be learned according to

$$\Phi : \chi(u_f^{left}, u_f^{right}, s_f^{\chi})) \rightarrow (v_f^{left}, v_f^{right}, s_{f+1}^{\chi}), \quad (17)$$

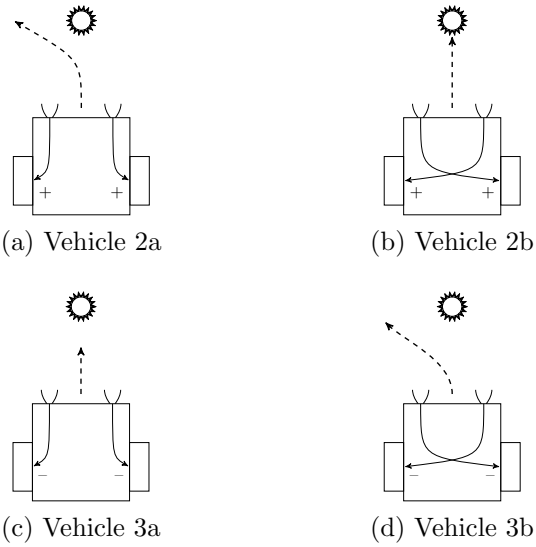
where χ is the mapping learned by the LSTM-based controller. Note, the mapping depends on the time f because, for the LSTM-based controller, the consecutive steering actions are assumed to be dependent based on the internal state s^{χ} of the controller.

The time dependence is very important as it allows filtering of the noisy LGMD outputs and avoiding obstacles that are not directly visible because of the avoiding maneuver. Besides, it has already shown a superior performance (Čížek, Faigl & Bayer 2017) in comparison to the deterministic control law (Čížek, Milička & Faigl 2017).

The mapping is learned using training data that comprise sequences $T = \{(u_i^{left}, u_i^{right}, v_i^{left}, v_i^{right}), i \in 0, \ldots, n\}$ of the length n containing the output of the LGMDs and desired steering commands. A self-supervised approach has been developed based on the robot control using Braitenberg vehicles model for collecting the training data in the realistic robotic simulator. In particular, autonomous collision-free navigation of the robot in the simulator, which measures the distance of the robot to the obstacles, has been employed to enable a deterministic control rule based on the behavior of the Braitenberg vehicles to collect the training data. In the following section, the Braitenberg vehicles model is introduced that is followed by the detailed description and discussion of the derived simulation model.

4.1. Braitenberg Vehicles Model

Braitenberg vehicles (Braitenberg 1986) are employed in the robotic simulator to provide smooth control commands and support the learning process of the proposed controller without the drawbacks of using human-guided trajectories for the learning. Braitenberg vehicles are simple autonomous agents that use basic sensory-motor connections to produce seemingly cognitive behaviors (Braitenberg 1986). The robots exhibit different behavior according to the sensory-motor connections. The connections can be based on inhibition and excitation, and four basic behaviors can be achieved, which have been evaluated in the proposed approach. The individual behaviors are visualized in Figure 8 and are as follows.



(a) Vehicle 2a (b) Vehicle 2b

(c) Vehicle 3a (d) Vehicle 3b

Figure 8. Braitenberg vehicles (Braitenberg 1986) utilized in the proposed self-supervised learning of the collision controller.

Vehicle 2a and **Vehicle 2b** represent a basic excitatory stimulation, see Figure 8a

and 8b, where both actuators speed up when exposed to the stimuli. The control rule for the Vehicle 2a is

$$\begin{aligned} v^l &= v_{base} + k_{2a}\, u^l, \\ v^r &= v_{base} + k_{2a}\, u^r, \end{aligned} \tag{18}$$

which makes the robot turn away from the stimuli, and for the Vehicle 2b

$$\begin{aligned} v^l &= v_{base} + k_{2b}\, u^r, \\ v^r &= v_{base} + k_{2b}\, u^l, \end{aligned} \tag{19}$$

makes the robot intercept the stimuli. The variable v^l and v^r are the speed of the left and right wheel, respectively, u^l and u^r are the strength of the stimuli perceived by the left and right sensor, respectively, k_{2a} and k_{2b} are the stimuli strength multipliers, and v_{base} is the base forward velocity of the robot.

Vehicle 3a and **Vehicle 3b** use the inhibitory connections when exposed to the stimuli, see Figure 8c and 8d. Therefore, Vehicle 3a orients towards the stimuli and come to rest facing it, whereas Vehicle 3b will come to rest facing away from the stimuli which resemble exploration behavior. The control rule for the Vehicle 3a is

$$\begin{aligned} v^l &= v_{base} - k_{3a}\, u^l, \\ v^r &= v_{base} - k_{3a}\, u^r, \end{aligned} \tag{20}$$

and for the Vehicle 3b

$$\begin{aligned} v^l &= v_{base} - k_{3b}\, u^r, \\ v^r &= v_{base} - k_{3b}\, u^l, \end{aligned} \tag{21}$$

where k_{3a} and k_{3b} are the stimuli strength multipliers.

4.2. Self-supervised Learning

The main purpose of the proposed self-supervised learning schema is to automate the collection of the training data necessary for learning the Φ mapping between the LGMD outputs and the differential steering command. As the objective of the robot is to avoid collisions when navigating the environment, the main idea is to turn away from the approaching obstacles. In principle, the LGMD responds to the divergence of the image edges (Chalupka et al. 2016) which depends on a combination of the camera properties, the robot velocity, and the structure of the environment. Hence, there is not a direct mapping between the stimuli strength and the distance to the obstacle. Therefore, a simulated laser scanner has been used to provide the robot with the information about the distance to the obstacles which allows using a deterministic control rule that drives the robot away of the closest obstacle, regardless the structure of the environment.

The deterministic control rule is based on the Vehicle 2a and Vehicle 3b rules that are combined in a single control rule with the parametrization that has to be tuned according to the performance of the robot. The minimal distances of the robot to obstacles in the left hemifield d_l and the right hemifield d_r are computed from the measurements provided by the laser scanner The distances are further utilized in the computation of the steering command.

$$\begin{aligned} v^l &= v_{base} + k_{2a}\frac{1}{d_l} - k_{3b}\frac{1}{d_r} \\ v^r &= v_{base} + k_{2a}\frac{1}{d_r} - k_{3b}\frac{1}{d_l} \end{aligned}, \tag{22}$$

where v^l and v^r are the differential steering velocities simulating the behavior of the Braitenberg vehicle that are fed to the CPG locomotion controller The parameters k_{2a} and k_{3b} are the collision avoidance weights for the respective vehicle type determining the speed, with which the robot turns away from the obstacles, and v_{base} is the base locomotion speed of the robot. An appropriate values of $k_{2a} = 0.8$ and $k_{3b} = 0.6$ have been determined experimentally.

The value of the base locomotion speed

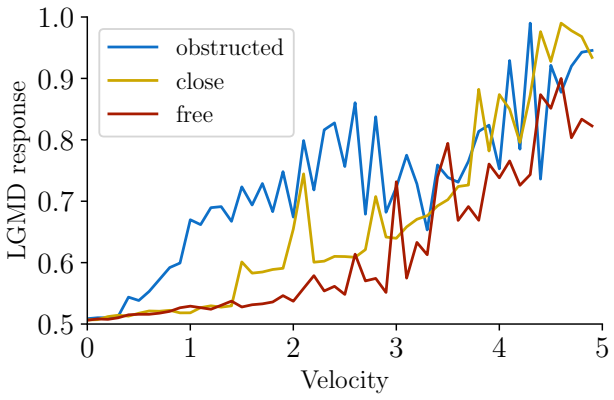


Figure 9. Mean LGMD response in the relation to the robot velocity for the scenario with the direct obstacle interception with different velocities.

v_{base} has been set concerning the mean LGMD response in the simulated environment. For a given camera configuration, we have experimentally determined that there is a given range of the robot velocities where the LGMD is selectively sensitive to the looming stimuli. The relation of the mean LGMD response and the robot velocity is depicted in Figure 9. The results have been obtained in a simulated experimental scenario where the robot approaches towards the obstacle, pass the obstacle in a close distance and then continues to the free area in cluttered parts of the environment. The forward velocity of the robot has been selected from the range 0.1 and 5.0 of the nominal robot velocity (0.05 $\mathrm{m\,s^{-1}}$) that has been identified in (Nguyenová et al. 2019). The results showed in Figure 9 indicate that the obstructed and free paths of the robot are best distinguishable for the robot velocities in the range 1.0–1.4 of the nominal velocity. Therefore, the value of the base locomotion speed has been set to $v_{base} = 1.2\mathrm{m\,s^{-1}}$.

The proposed control rule (22) has been used to autonomously navigate the robot in a cluttered simulated room inside the V-REP simulator that is visualized in Figure 10a. The robot navigates the used environment randomly. The navigation rule (22) ensures that the robot does not collide with the environment, but also provides a way how to tune the particular performance of the developed system. During the navigation, the LGMD responses (u^{left}, u^{right}) together with the steering command (v^l, v^r) have been collected and used in the learning of the LSTM-based controller using the Backpropagation through time (BPTT) algorithm (Sutskever 2013). In particular, Keras (Chollet et al. 2015) BPTT implementation in Python has been utilized with the mean squared error and RMSprop optimizer. The learning results are reported in Section 5.

Here, it is worth noting that in our experimental evaluations, we have also considered control rules based solely on the Vehicle 2a and Vehicle 3b rules that are both feasible for the collision avoidance based on the distance data. However, in comparison to the proposed combined rule, the performance of the learned controller was poor. The main insight is that for the rule based on the Vehicle 2a, the collision avoidance accelerates the respective side of the robot closer to the obstacle, which further boosts the LGMD network output on that side. Because the field of view in the left and right hemifields partially overlaps, the LGMD outputs u^{left} and u^{right} are dependent, and thus the difference between the left and right sensory inputs are not sufficient to make the robot turn away from the obstacle fast enough. On the other hand, the Vehicle 3b slows down the entire locomotion that negatively influences the selectivity of the LGMD network to looming stimuli. By the proposed combination of the both rules, we get the desired behavior in which the rule of the Vehicle 2a accelerate the respective side of the robot (the one that is closer to the obstacle) while

the speed on the opposite side is suppressed and can be even negative, which means that the robot can quickly turn on a spot.

5. Experimental Results

The feasibility of the proposed end-to-end controller and the proposed learning scheme of the novel biologically-inspired visual collision avoidance system has been verified in a set of experimental scenes. First, the system has been learned and also verified in a realistic robotic simulator, and the results are reported in the, following section. After that, a set of four real-world collision avoidance experiments has been conducted, including three experiments with the static obstacles in both indoor and outdoor environments and a single indoor experiment to examine the controller behavior with obstacles. The experimental environments are visualized in Figures 10c-e. The detailed description of the real-world experiments and the achieved results are reported in Section 5.2 and Section 5.3. The found insights and properties of the proposed solution are discussed in Section 5.4.

5.1. Collision Avoidance in Robotic Simulator

The performance of the controller has been verified in the realistic robotic simulator which has also been used for the controller learning. First, the training environment visualized in Figure 10a has been used to collect data for the learning of the navigation controller. The robot collects the LGMD responses together with the steering command given by the deterministic control rule (22) while it autonomously navigates the environment, see Section 4.2 for the detail description of the unsupervised learning. Altogether 15 minutes of continuous collision free locomotion in the simulator that equals 48 m of crawled distance has been recorded and used for the controller learning. The proposed controller with two inputs, single fully connected LSTM hidden layer with 32 neurons, and two outputs, has been learned using BPTT in Python Keras (Chollet et al. 2015) implementation with the mean squared error as the loss function and RMSprop optimizer using the collected data. The collected data has been split in half to training and testing data, and the controller has been learned for 1000 epochs. An evolution of the learning loss and validation loss is presented in Figure 11. The graph of both the training loss and testing loss indicate the ability of the proposed controller to generalize the learned behavior. Further, it suggests the LGMD output contains information about the distance to the closest obstacles, and thus its adoption for the collision avoidance in the end-to-end continuous mapping sensory-motor controller is feasible.

Figure 12 shows a subset of the learning data and the predicted steering command for a part of the learning sequence. From the top to the bottom, the figure shows the LGMD responses (u^{left}, u^{right}), the distances to the closest obstacle (d^l, d^r) measured using the simulated laser scanner, the corresponding steering command (v^l, v^r) given by the rule (22), and the output of the learned LSTM-based locomotion controller (v^{left}, v^{right}). The results indicate the controller can learn a policy that resembles the behavior of the simulated Braitenberg vehicle with the proper direction and magnitude of the steering command based on the looming stimuli only. Together with the validation loss graph in Figure 11 the results suggest the learned controller can generalize the learned behavior.

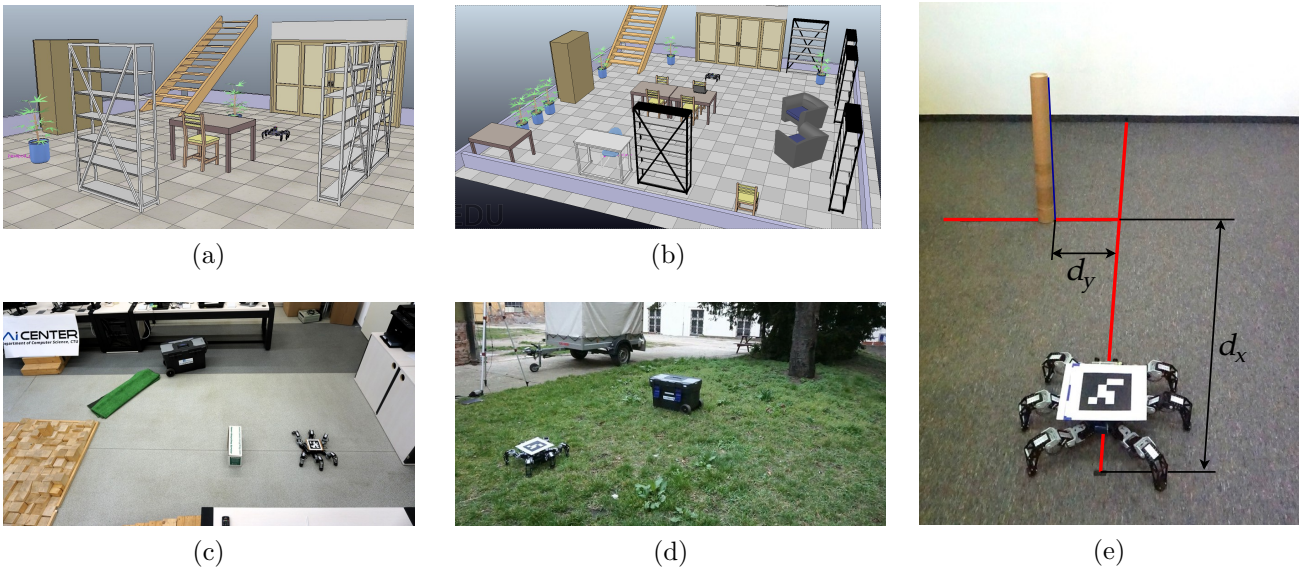


Figure 10. Environments for learning and testing of the proposed LSTM-based controller. (a) The simulated training environment. (b) The simulated testing environment. (c) The cluttered laboratory environment. (d) The outdoor environment. (e) Testing laboratory environment with single obstacle in lateral distance d_y to the heading of the robot.

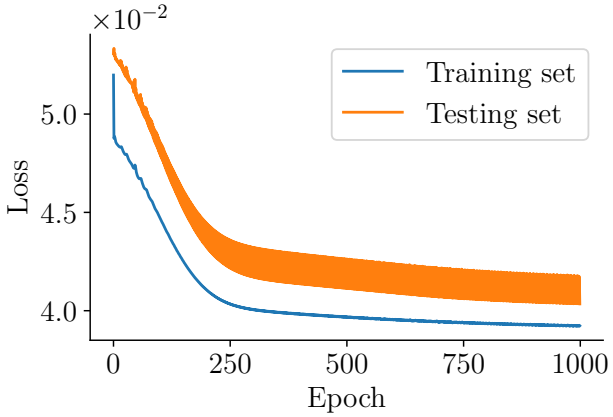


Figure 11. Controller learning and validation loss.

Next, the robot has been deployed in a more cluttered environment (see Figure 10b) to perform validation experiments of the collision avoidance behavior. Five trials have been performed in which the total traveled distance by the robot has been measured to 235.3 m. In these trials, the robot has started at the identical location $(x = 0.0, y = 2.5)$ and randomly wanders inside the environment while avoiding collisions. The resulting trajectories are shown in Figure 13 and support the feasibility of the proposed approach and the learning scheme. Therefore, to further test the learned system, it has been deployed on a real robot.

5.2. Collision Avoidance in Uncluttered Environment

We deploy the proposed system in real-world environments to make a step forward the main motivation of the presented system that is to explore the full integration of a neural and motor systems in a physical robot to construct an end-to-end neural network control system. The proposed controller has been learned in the simulated environment, and our goal is to verify that the system is ready for real-world deployed without further adjustments, modifications, or relearning of the controller using real-world data. Therefore we have set up an initial experiment, where the robot

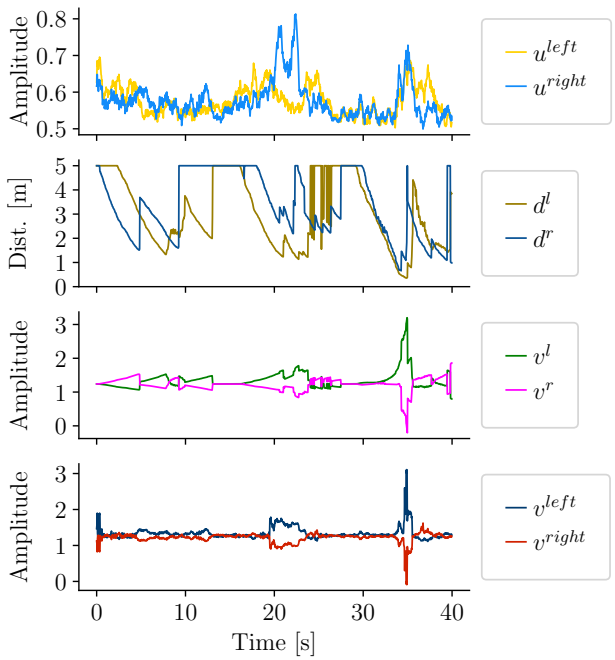


Figure 12. Training data. From top to bottom the LGMD responses (u^{left}, u^{right}), the distances to the closest obstacle (d^l, d^r) measured by the simulated laser scanner with the clipping range of 5 m, the corresponding steering command (v^l, v^r) given by the rule (22), and the output of the learned LSTM-based locomotion controller (v^{left}, v^{right}).

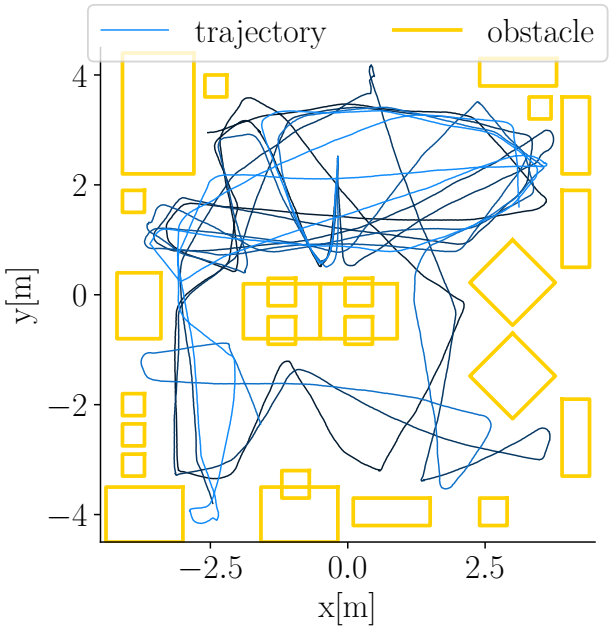
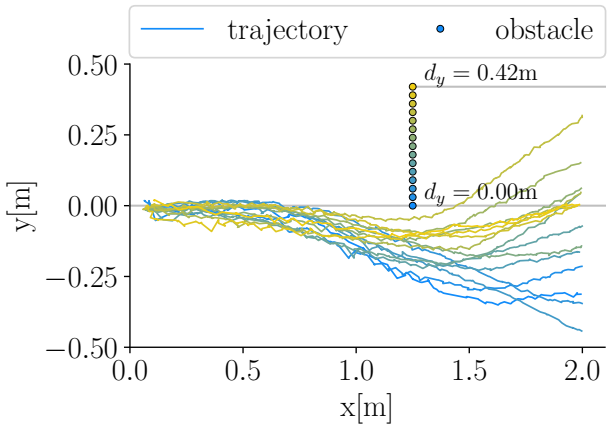


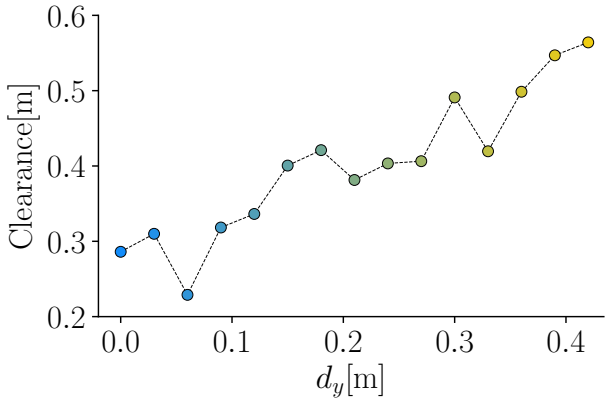
Figure 13. Robot trajectories in collision avoidance. Five testing trials in the simulated scenario.

is supposed to avoid a single obstacle in an uncluttered environment to verify the learned behavior and benchmark the performance of the controller before testing it in cluttered environments where the overall appearance of the scene can influence the robots' behavior.

In particular, we prepared a test trail with the robot placed three meters away from a plain white wall with a poster tube in the $d_x = 1.25\,\text{m}, d_y \in \{0, 0.03, 0.06, \ldots, 0.42\}\,\text{m}$ distance away from the robot, which is visualized in Figure 10e. During the evaluation, the robot has been placed at the same position with the same heading perpendicular to the white wall at the beginning of the experiment. Afterward, the locomotion controller has been turned on to steer the robot motion based on the visual input from the stereo camera. The robot movement in the environment has been tracked by a visual localization system based on the AprilTag fiducial marker (Olson 2011) attached to the robot. Individual test trails have been performed with the increased lateral distance d_y of the obstacle, and the robot position from the localization system has been recorded together with the minimum distance to the obstacle encountered during the trail run. Altogether fifteen experiments have been performed with the robot always successfully avoiding the obstacle. The results are visualized in Figure 14a and Figure 14b which show the individual trajectories and the minimum clearance between the robot and the obstacle for each trajectory respectively. The results indicate that the controller learned in the simulation can selectively react to approaching obstacles and avoid the collision, and does not need further tuning or relearning to be deployed in the real-world conditions. Therefore, further testing in cluttered environments have



(a) Robot trajectories.



(b) Trajectory clearance.

Figure 14. The results of the experimental evaluation of the collision avoidance with a single obstacle. Individual trails corresponding to the increasing lateral distance d_y of the obstacle are color coded from blue to yellow for increasing d_y.

been performed, and the achieved results are reported in the following section.

5.3. Collision Avoidance in Cluttered Environment

A set of three experiments in the indoor laboratory and outdoor environments have been performed to verify further the learned controller can be directly deployed in a real-world scenario. The evaluation scenario consists of two experiments with static obstacles and a single experiment with dynamic obstacles.

The indoor experiment with the static obstacles consists of a box placed directly in front of the robot at the distance 1.2 m, a box placed to the right of the robot, and a table to the left side of the robot, see Figure 10c. The outdoor experiment with the static obstacle has been performed at the front-yard of the university with a grass surface, multiple trees and bushes directly in the field of view of the robot together with a facade of a historic building at the background, see Figure 10d.

The experiment with dynamic obstacles has been performed in the laboratory, where the robot has crawled straight, and when it reaches a given location, an obstacle has been placed in front of it from the left side. After the robot reacts to that obstacle, a second obstacle has been placed to its field of view from the right side. The scenario is intended to examine the robot ability to respond to dynamically changing environment.

In all the experiments, the robot has been placed approximately at the same place at the beginning of each experiment. Then the locomotion controller has been turned on to steer the robot motion based on the visual input from the stereo camera. The robot position from the localization system (Olson 2011) together with the left and right LGMD responses (u^{left}, u^{right}), and the differential steering commands (v^{left}, v^{right}) have been recorded. Twelve experiments in the indoor scenario with static obstacles, ten trials in the outdoor scenario, and ten trials in the scenario with dynamic obstacles have been performed in the total. In all the cases, the robot has been able to avoid collisions with the obstacles successfully.

The robot trajectories are shown in Figure 15. Typical responses of the LGMD together with the corresponding steering commands are visualized in Figure 16a for

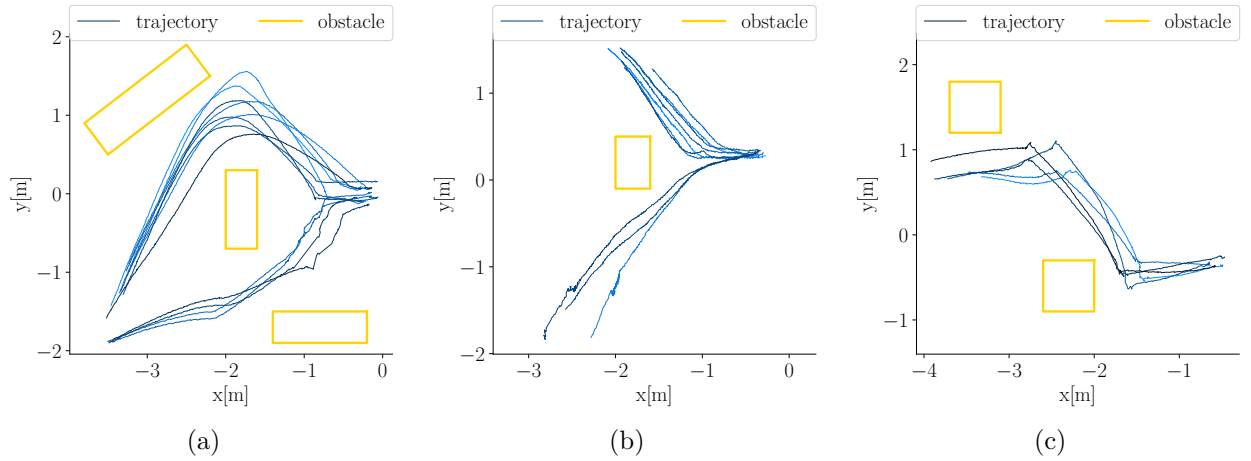


Figure 15. Robot trajectories in collision avoidance. (a) Twelve trajectories in the laboratory experiment with the static obstacles. (b) Ten trajectories in the outdoor experiment. (c) Five trajectories in the laboratory experiment with dynamic obstacles.

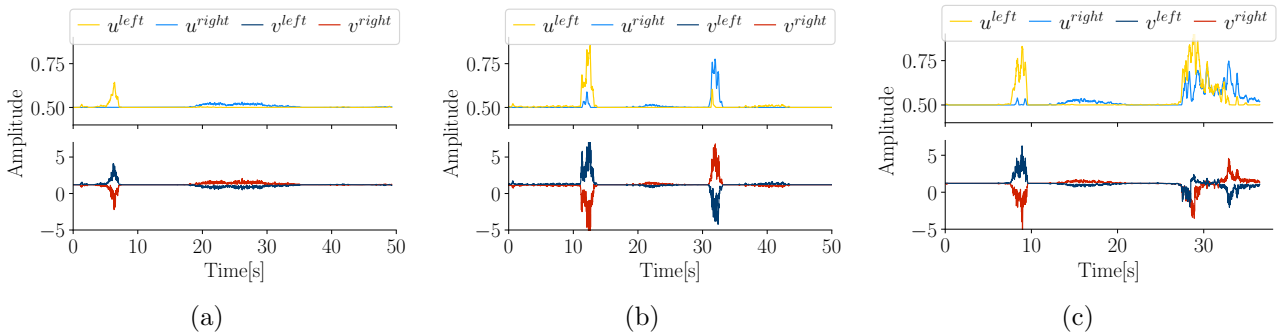


Figure 16. Example of LGMD and controller response in (a) static obstacle indoor test, (b) dynamic obstacle indoor test, and (c) dynamic obstacle indoor test with observed retreating behavior.

the static obstacles and in Figure 16b for the dynamic obstacles. It can be seen that the LGMD is less sensitive for the static case and the steering commands are less abrupt in comparison to the dynamic collision avoidance that induces a larger excitation of the LGMD given by a larger divergence of the image edges. The shape of the steering command provided by the proposed LSTM-based locomotion controller corresponds to the learned behavior given by Braitenberg vehicle steering rule (22), i.e., the base speed corresponds to $v_{base} = 1.2\mathrm{m\,s^{-1}}$ and the response to obstacles is approximately symmetric.

An interesting behavior of the robot has been observed in the experiment with the dynamic obstacles. In two out of ten trials, the robot has exhibited retreating behavior which can be noticed in the robot's steering command depicted in Figure 16c. The robot stopped when it is exposed to the second obstacle, and it did a few steps back which correspond to the situation at the time around 28 s when both the differential steering commands v^{left} and v^{right} are negative for a while. The observed behavior needs further investigation as it originates in the learned LSTM-based locomotion controller and it cannot be learned

using the rule (22) as the distance to the obstacles would have to be negative to produce negative velocities for the differential steering. Nevertheless, the herein presented results support the proposed controller can be learned to exhibit different collision avoidance and escaping strategies with a continuous mapping function between the looming stimuli detection and motion control. Besides, the simulation and real laboratory experiments support that the control scheme learned in the realistic robotic simulator using the proposed method is feasible and can be directly deployed in a real-world scenario. We leave the learning of more complex locomotion control and escaping behaviors to our future work.

5.4. Discussion of the Results

The feasibility of the proposed control system consisting of the LGMD neural network to detect the looming stimuli, CPG-based locomotion controller, and the LSTM-based navigation controller has been experimentally verified with the proposed learning approach using the realistic robotic simulator. Even though the robot successfully operated in the real experimental setup, the LGMD relies on the divergence of the scene edges; hence, a collision may occur in the case of a poor scene contrast on the objects, e.g., when walking towards a white wall. Further, the proposed system is purely reactive without path planning layer that can be utilized to guide the robot along prepared collision-free paths. Without such high-level planning, the navigation rule (22) with the v_{base} base locomotion speed inevitably leads to a situation, where the only possible forward movement is towards an obstacle. Hence, the learned controller may fail to guide the robot out of corners, where the robot gets stuck and collide. Therefore, such behavior requires further improvement of the navigation layer and its combination with path planning that is considered for future work.

6. Conclusion

In this work, we report on the developed fully autonomous agent capable of collision avoidance using an end-to-end pipeline of bio-inspired neural networks. We have shown that the proposed self-supervised training of the neural networks in the realistic robotic simulator (using the control model of simple cognitive machines) is sufficient for a reliable real-world deployment.

In our future work, we aim to exploit the enhanced traversability capabilities of the utilized multi-legged platform to enable omnidirectional movement, and thus address reliable collision-free locomotion in the dynamic environments using self-supervised learning techniques. As such, the herein proposed end-to-end pipeline of neural networks is well suited for such deployment, as the individual building blocks allow learning based on the gradient backpropagation, which significantly simplifies the learning process.

Acknowledgment

The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 15-09600Y and No. 18858S.

References

Badia, S. B., Pyk, P. & Verschure, P. F. M. J. (2005). A biologically based flight control system for a blimp-based UAV, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3053–3059.

Blanchard, M., Rind, F. & Verschure, P. F. M. J. (2000). Collision avoidance using a model of the locust LGMD neuron, *Robotics and Autonomous Systems (RAS)* **30**(1-2): 17–38.

Braitenberg, V. (1986). *Vehicles: Experiments in synthetic psychology*, MIT press.

Chalupka, K., Dickinson, M. & Perona, P. (2016). Generalized regressive motion: a visual cue to collision, *Bioinspiration & Biomimetics* **11**(4): 046008.

Chollet, F. et al. (2015). Keras, `https://github.com/fchollet/keras`. cited on 2019-01-22.

Čížek, P., Faigl, J. & Bayer, J. (2017). Enhancing neural based obstacle avoidance with cpg controlled hexapod walking robot, *Information Technologies - Aplication and Theory (ITAT), CEUR Workshop Proceedings*, Vol. 1885, pp. 65–70.

Čížek, P., Milička, P. & Faigl, J. (2017). Neural based obstacle avoidance with CPG controlled hexapod walking robot, *IEEE International Joint Conference on Neural Networks (IJCNN)*, pp. 650–656.

De Vries, S. E. & Clandinin, T. R. (2012). Loom-sensitive neurons link computation to action in the drosophila visual system, *Current Biology* **22**(5): 353–362.

Fotowat, H., Fayyazuddin, A., Bellen, H. J. & Gabbiani, F. (2009). A novel neuronal pathway for visually guided escape in drosophila melanogaster, *Journal of Neurophysiology* **102**(2): 875–885.

Frigon, A. & Rossignol, S. (2006). Experiments and models of sensorimotor interactions during locomotion, *Biological Cybernetics* **95**(6): 607–627.

Fu, Q., Yue, S. & Hu, C. (2016). Bio-inspired collision detector with enhanced selectivity for ground robotic vision system, *Proceedings of the British Machine Vision Conference*, pp. 6.1–6.13.

Gandhi, D., Pinto, L. & Gupta, A. (2017). Learning to fly by crashing, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3948–3955.

Hartbauer, M. (2017). Simplified bionic solutions: a simple bio-inspired vehicle collision detection system, *Bioinspiration & Biomimetics* **12**(2): 026007.

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

Hoy, M., Matveev, A. S. & Savkin, A. V. (2015). Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey, *Robotica* **33**(3): 463–497.

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

Judge, S. & Rind, F. (1997). The locust DCMD, a movement-detecting neurone tightly tuned to collision trajectories, *Journal of Experimental Biology* **200**(16): 2209–2216.

Kelchtermans, K. & Tuytelaars, T. (2017). How hard is it to cross the room? - training (recurrent) neural networks to steer a UAV, *CoRR* **abs/1702.07600**.

Loquercio, A., Maqueda, A. I., del Blanco, C. R. & Scaramuzza, D. (2018). Dronet: Learning to fly by driving, *IEEE Robotics and Automation Letters* **3**(2): 1088–1095.

Matsuoka, K. (1987). Mechanisms of frequency and pattern control in the neural rhythm generators, *Biological Cybernetics* **56**(5-6): 345–353.

Miller, B. D. & Clark, J. E. (2015). Towards highly-tuned mobility in multiple domains with a dynamical legged platform, *Bioinspiration & Biomimetics* **10**(4): 046001.

Mrva, J. & Faigl, J. (2015). Tactile sensing with servo drives feedback only for blind hexapod walking robot, *10th International Workshop on Robot Motion and Control (RoMoCo)*, pp. 240–245.

Muller, U., Ben, J., Cosatto, E., Flepp, B. & Cun, Y. L. (2006). Off-road obstacle avoidance through end-to-end learning, *Advances in Neural Information Processing Systems*, pp. 739–746.

Nguyenová, M. T., Čížek, P. & Faigl, J. (2019). Modeling proprioceptive sensing for locomotion control of hexapod crawling robot in robotic simulator, *Modelling and Simulation for Autonomous Systems*, pp. 215–225.

Olson, E. (2011). AprilTag: A Robust and Flexible Visual Fiducial System, *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3400–3407.

Preuss, T., Osei-Bonsu, P. E., Weiss, S. A., Wang, C. & Faber, D. S. (2006). Neural representation of object approach in a decision-making motor circuit, *Journal of Neuroscience* **26**(13): 3454–3464.

Rohmer, E., Singh, S. P. & Freese, M. (2013). V-REP: A versatile and scalable robot simulation framework, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*,

pp. 1321–1326.

Salt, L., Howard, D. J., Indiveri, G. & Sandamirskaya, Y. (2017). Differential evolution and bayesian optimisation for hyper-parameter selection in mixed-signal neuromorphic circuits applied to uav obstacle avoidance, *CoRR* **abs/1704.04853**.

Sutskever, I. (2013). *Training recurrent neural networks*, PhD thesis, University of Toronto.

Szadkowski, R. J., Čížek, P. & Faigl, J. (2018). Learning central pattern generator network with back-propagation algorithm, *Information Technologies - Aplication and Theory (ITAT), CEUR Workshop Proceedings*, Vol. 2203, pp. 116–123.

Wilson, D. M. (1961). The central nervous control of flight in a locust, *Journal of Experimental Biology* **38**(47): 471–490.

Xiong, X., Wörgötter, F. & Manoonpong, P. (2016). Adaptive and energy efficient walking in a hexapod robot under neuromechanical control and sensorimotor learning, *IEEE Transactions on Cybernetics* **46**(11): 2521–2534.

Yu, H., Gao, H., Ding, L., Li, M., Deng, Z. & Liu, G. (2016). Gait Generation With Smooth Transition Using CPG-Based Locomotion Control for Hexapod Walking Robot, *IEEE Transactions on Industrial Electronics* **63**(9): 5488–5500.

Yue, S. & Rind, F. C. (2006). Collision detection in complex dynamic scenes using an LGMD-based visual neural network with feature enhancement, *IEEE Transactions on Neural Networks* **17**(3): 705–716.

Yue, S. & Rind, F. C. (2013). Redundant neural vision systems – competing for collision recognition roles, *IEEE Transactions on Autonomous Mental Development* **5**(2): 173–186.

Zabala, F., Polidoro, P., Robie, A., Branson, K., Perona, P. & Dickinson, M. H. (2012). A simple strategy for detecting moving objects during locomotion revealed by animal-robot interactions, *Current Biology* **22**(14): 1344–1350.

Zhao, J., Hu, C., Zhang, C., Wang, Z. & Yue, S. (2018). A bio-inspired collision detector for small quadcopter, *arXiv preprint arXiv:1801.04530* .

Zhong, G., Chen, L., Jiao, Z., Li, J. & Deng, H. (2018). Locomotion control and gait planning of a novel hexapod robot using biomimetic neurons, *IEEE Transactions on Control Systems Technology* **26**(2): 624–636.