J. Hlaváčová (Ed.): ITAT 2017 Proceedings, pp. 65–70
CEUR Workshop Proceedings Vol. 1885, ISSN 1613-0073, © 2017 P. Čížek, J. Faigl, J. Bayer

Enhancing Neural Based Obstacle Avoidance with CPG Controlled Hexapod Walking Robot

Petr Čížek, Jan Faigl, and Jan Bayer

Czech Technical University in Prague, Faculty of Electrical Engineering, Technická 2, 166 27 Prague, Czech Republic
{cizekpe6|faiglj|bayerja1}@fel.cvut.cz

Abstract: Avoiding collisions with obstacles and intercepting objects based on the visual perception is a vital survival ability of any animal. In this work, we propose an extension of the biologically based collision avoidance approach to the detection of intercepting objects using the Lobula Giant Movement Detector (LGMD) connected directly to the locomotion control unit based on the Central Pattern Generator (CPG) of a hexapod walking robot. The proposed extension uses Recurrent Neural Network (RNN) to map the output of the LGMD on the input of the CPG to enhance collision avoiding behavior of the robot in cluttered environments. The presented results of the experimental verification of the proposed system with a real mobile hexapod crawling robot support the feasibility of the presented approach in collision avoidance scenarios.

1 Introduction

Avoiding collisions with obstacles and intercepting objects is a vital survival ability for any animal. For a mobile robot moving from one place to another, the contact with a fixed or moving object may have fatal consequences. Therefore, it is desirable to study the problem of collision avoidance and derive new and computationally efficient ways to trigger collision avoiding behavior.

In this work, we concern a problem of biologically inspired motion control and collision avoidance with a legged walking robot equipped with a forward looking camera only. We propose to utilize a Central Pattern Generator (CPG) approach [1] for robot locomotion control and the vision-based collision avoidance approach using the Lobula Giant Movement Detector (LGMD) [2] which are both combined in the proposed controller based on Recurrent Neural Network (RNN).

The proposed solution builds on our previous results published in [3] in which only a simple mapping function is utilized for transforming the output of the LGMD neural network directly to the locomotion control parameters of the CPG controller [1]. Such a solution works well in laboratory conditions, but, unfortunately, it is error-prone in the cluttered environment. It is mainly because of the way how the LGMD neural network processes the visual data and due to a simple mapping function. The LGMD reacts on the lateral movement of vertical edges in the image regardless their depth in the scene. In a cluttered environment, this results that the output is heavily influenced by a lot of stimuli from the distinctive edges in a far distance

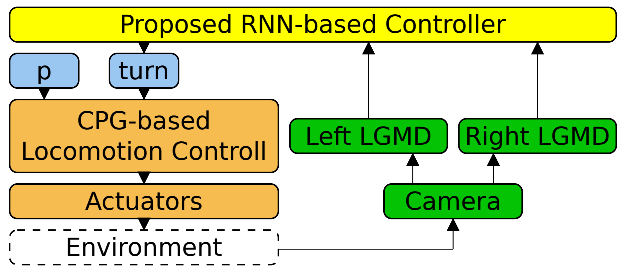


Figure 1: Overview of the proposed control system structure. Different colors discriminate the individual functional parts of the architecture.

from the robot. Moreover the mapping function translates the output of the LGMD directly to the locomotion control parameters. Hence, the reaction of the robot is based solely on the current observation of the environment which results in situations when the robot hits an obstacle from the side that has successfully avoided earlier but it is already out of the field of view. Therefore, we propose to enhance the collision avoiding behavior of the robot by incorporating a memory mechanism by means of the RNN.

The overall structure of the proposed system is depicted in Fig. 1. Regarding to the previous approaches, here, we would like to emphasize a practical verification of the proposed method on a real walking robot as the specific nature of the legged locomotion makes the problem more difficult in comparison to the wheeled [4, 5] or flying [6] robots. The main difference originates in abrupt motions of the camera induced by the locomotion of the robot which negatively influences the output of the collision avoiding visual pathway.

The reminder of the paper is organized as follows. The most related approaches on the neural-based collision avoidance using vision are summarized in Section 2. Section 3 describes the individual building blocks of the proposed control architecture. Evaluation results and their discussion are detailed in Section 4. Concluding remarks and suggestions for future work are dedicated to Section 5.

2 Related Work

The problem of collision avoidance has been studied ever since the mobile robots appeared. Hence, there is a lot of different approaches using different sensors and different processing techniques. In this work, we are focused on vision-based neural obstacle avoidance methods and the most related approaches are described in the rest of this section.

Direct mapping of the visual perception on the robot control command using a feed-forward neural network has been already utilized in several methods. The problem of road following using neural networks, which dates back to 90s, can be considered as a special case of the collision avoidance problem [7]. However, such approaches cannot be considered as biologically-based because of artificial nature of the examined roads.

In [2], the Lobula Giant Movement Detector (LGMD) neural network has been introduced in robotics to imitate the way how insects avoid collisions with an intercepting object [8]. The approach has been widely adopted for its simplicity and relatively good performance with wheeled [2, 4, 5] and flying [6] robots. However, these approaches experimentally verify the collision avoidance with a real robot either in a closed arena where it is necessary to avoid collisions with walls or in a scenario where a static robot is supposed to detect an intercepting object. Moreover, the walls of the arena or the obstacles were homogeneously distributed or coated with a high contrast artificial pattern which significantly improves the behavior of the LGMD. In our approach, we focus on the deployment of the LGMD in heavily cluttered unstructured environment, and thus evaluate the approach in more realistic scenarios.

An experimental study on the prediction of evasive steering maneuvers in urban traffic scenarios has been recently published in [9]. In this approach, the performance of the LGMD is improved by introducing so-called "danger zones" which are the image areas that will most likely indicate the incoming threat.

Another approach presented in [10] compares the performance of the LGMD and Directional Selective Neurons (DSN) in the ability to avoid collisions. Both of them are to be found in the visual pathways of insects. The reported results show that the LGMD can be trained using evolutionary techniques to outperform the DSN in the collision recognition ability.

Regarding our target scenario, the most relevant approach to the proposed solution has been presented in [11]. The authors use a biologically-inspired collision avoidance approach based on the extraction of nearness information from the image depth estimation to detect obstacles and avoid collisions. The whole system allows a simulated hexapod robot to navigate cluttered environment while actively avoiding obstacles. However, the approach uses a direct feed-forward approach for the motion control and it has not been deployed in a real-world scenario.

The herein proposed control mechanism utilizes a Recurrent Neural Network (RNN) that has been already utilized in collision avoiding scenarios using odor sensors on whiskers [12] or a set of infrared rangefinders [13]. A vision-based collision avoidance for an UAV based on the RNN has been recently presented in [14] which trains the UAV to avoid collisions during autonomous indoor flight. This work served as the inspiration for our neural-based autonomous agent.

3 Proposed Solution

Three basic functional parts can be identified within the proposed collision avoiding system. They are depicted in three different colors in Fig. 1. The first part is the locomotion control unit based on the chaotic oscillator [15] depicted in an orange color whose purpose is to control the walking pattern and to solve the kinematics. It allows to change the type of the motion gait based on the pre-set parameter p and steer the robot motion according to the input signal *turn* defining the turning radius. The second part is the visual pathway depicted in a green color which utilizes the LGMD neural network for avoiding approaching objects and triggering escape behavior. The main idea of the proposed approach is to use the LGMD outputs for setting the hexapod control parameters, in particular, the turning radius *turn* of the robot. In this work, we are proposing to use the RNN-based approach for the translation of the LGMD output to the *turn* parameter which is dedicated to the last part depicted in a yellow color. Each part is discussed in more detail in the following sections.

3.1 CPG-Based Locomotion Control

The locomotion control is based on our previous work presented in [1]. It utilizes only one chaotic CPG [15] consisting of two interconnected neurons with a control input computed solely based on the input period p. The CPG stabilizes a periodic orbit of p from the chaotic oscillation, so the output is a discrete periodic signal. The period $p \in \{4, 6, 8, 12\}$ directly determines the resulting walking pattern (motion gait): tripod, ripple, tetrapod, and wave, respectively [16].

Afterwards, the output of the chaotic oscillator is shaped and post-processed in order to obtain a signal usable for a trajectory generator and to determine the phase of individual legs, i.e., whether the leg is swinging or supporting the body. Afterwards, the output of the chaotic oscillator is thresholded and a triangle wave alternating between -1 and 1 is produced, where the upslope (swing phase) is a constant and the downslope (support phase) depends on the period p. Based on the leg coordination rules [17], individual delays are applied to the triangular wave per each leg to produce the rhythmic pattern for each leg.

The result of the post-processing module is fed into a trajectory generator, which determines the position of foot-tips according to the input signal along with the parameter *turn*, which is given by the RNN-based controller. The *turn* parameter is equal to the distance (in millimeters) from the robot center to the turning center on a line perpendicular to the heading of the robot connecting the

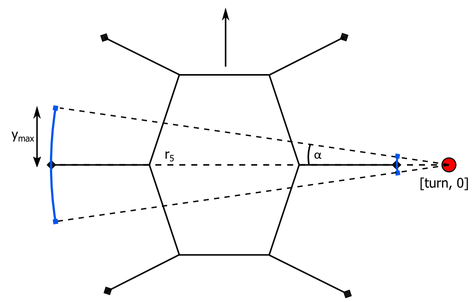


Figure 2: Trajectory generation - the turning point denoted as the small red disk is given by *turn* parameter. α is computed as the maximum angle given the turning radius and the maximum step size y_{max}.

default positions of the middle legs. Based on the *turn* parameter and the triangular wave, the trajectory generator uniquely determines the foot-tip positions of each leg on the constructed arcs which are limited by the angle α. The value of α is computed from the distance of the furthest leg from the pivotal point established by *turn* and the maximum step size y_{max}. The idea of the trajectory generator is visualized in Fig. 2. The output of the trajectory generator is transformed into the joint space using the inverse kinematics module and then performed by the robot actuators. Notice, the speed of the robot forward motion is determined by the period p, while the robot angular velocity is controlled by the *turn* parameter, which is adjusted by the RNN-based controller from the LGMD output.

3.2 LGMD Neural Network

The LGMD [2] is a neural network found in the visual pathways of insects, such as locusts [8], which responds selectively to objects approaching the animal on a collision course. It 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*. The structure of the network is visualized in Fig. 3.

The *Photoreceptive layer* processes the sensory input from the camera. Its output is the difference between two successive grayscale camera frames and it is computed as

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

where L_f is the current frame, L_{f-1} is the previous frame and (x,y) are the pixel coordinates. In principle, the *Photoreceptive layer* implements a contrast enhancement and forms the input to the following two groups of neurons – the *Inhibition layer* and *Excitatory layer*.

The response of the *Inhibition layer* is computed as

$$I_f(x,y) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} P_{f-1}(x+i, y+j) w_I(i,j), \tag{2}$$
$$(i \neq j, \text{ if } i = 0),$$

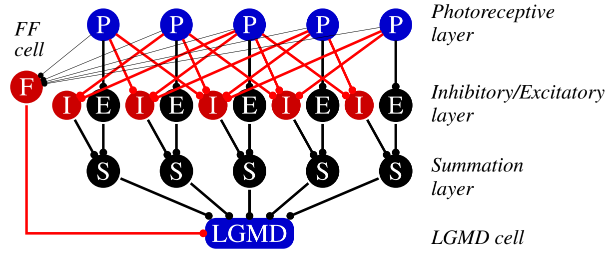


Figure 3: LGMD neural network model

where w_I are the inhibition weights set as

$$w_I = \begin{bmatrix} 0.06 & 0.12 & 0.25 & 0.12 & 0.06 \\ 0.12 & 0.06 & 0.12 & 0.06 & 0.12 \\ 0.25 & 0.12 & 0 & 0.12 & 0.25 \\ 0.12 & 0.06 & 0.12 & 0.06 & 0.12 \\ 0.06 & 0.12 & 0.25 & 0.12 & 0.06 \end{bmatrix}. \tag{3}$$

The *Inhibition layer* is essentially smoothing the *Photoreceptive layer* output values and filtering those caused by noise or camera imperfections. The inhibition weights w_I are selected experimentally with respect to the LGMD description in [2] which uses 3×3 matrix of inhibition weights, but on an image with a much lower resolution.

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

$$E = \left| P_f(x,y) \right|. \tag{4}$$

The response of the *Summation layer* is computed as

$$S_f(x,y) = E(x,y) - \left| I_f(x,y) \right| W_I, \tag{5}$$

where $W_I = 0.4$ is the global inhibition weight. Let S'_f be a matrix for which each value exceeding the threshold T_r is passed and any lower value is set to 0

$$S'_f(x,y) = \begin{cases} S_f(x,y) & \text{if } S_f(x,y) \geq T_r \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

Then, an excitation of the *LGMD cell* is computed as

$$U_f = \sum_{x=1}^{k} \sum_{y=1}^{l} \left| S'_f(x,y) \right| \tag{7}$$

and finally, the *LGMD cell* output is

$$u_f = (1 + \mathrm{e}^{-U_f n_{cell}^{-1}})^{-1}, \tag{8}$$

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

Typically, the LGMD neural network contains *Feed-forward cell* which is not utilized in the proposed scheme based on the results of the experimental evaluation. The purpose of the *Feed-forward cell* is to suppress the output of the *LGMD cell* in a case of fast camera movements.

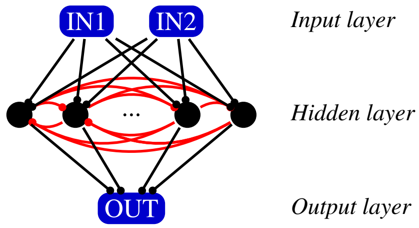


Figure 4: LSTM recurrent neural network model

However, due to the specific nature of the legged locomotion, this feature is undesirable as it makes the LGMD network less sensitive.

In our setup, two LGMD neural networks are utilized in parallel to distinguish the direction of the interception, and thus be able to steer the robot in the opposite direction to achieve the desired obstacle avoiding behavior. The input image from a single camera is split into left and right parts with the overlapping center part. Each of the LGMDs provide the output which we denote u_f^{left} and u_f^{right} for the left and the right LGMD respectively.

3.3 RNN-Based Controller

In our previous work [3], we utilized a direct mapping function between the LGMDs output tuple and the *turn* parameter of the CPG. The particular mapping function was designed as

$$\Phi(e) = \begin{cases} 100/2e & \text{for } |e| \geq 0.2 \\ 10000 \cdot \mathrm{sgn}(e) & \text{for } |e| < 0.2 \end{cases}, \qquad (9)$$

where error e is calculated as the difference of the LGMD outputs $e = u_f^{left} - u_f^{right}$.

However, the direct mapping function failed in the collision avoidance in cluttered environment. Therefore, we developed an RNN-based controller that takes the left and right LGMD outputs on its input and provides an estimate of the *turn* parameter on its output.

In the proposed controller, we utilized the Recurrent Neural Network (RNN) based on the Long Short Term Memory (LSTM) [18] with two inputs, one hidden layer, and one output that estimate the error e which is then used with the mapping function given by (9). The Backpropagation Through Time (BPTT) [19] is utilized for the RNN training, which unrolls the network over the time resulting in a feed-forward neural network. As there are only two real number inputs to the network, it is unnecessary to use sliding window approaches to the learning as it is possible to feed the data to the network in a full length. The structure of the LSTM neural network is visualized in Fig. 4.

The main idea is to connect the RNN directly to the outputs of the left and right LGMDs and let the neural network estimate the parameter e which is then translated by (9) to the *turn* parameter of the CPG-based locomotion controller.

4 Experimental Evaluation

The experimental verification of the proposed neural-based controller is focused on the ability of the hexapod walking robot to avoid collisions with the obstacles on its path. We are emphasizing the practical verification with a real walking robot to thoroughly test the proposed solution and provide insights on the achieved performance.

The experimental evaluation has been considered with the hexapod walking robot visualized in Fig. 5a. The robot has six legs attached to the trunk that hosts the sensors. In particular, the Logitech C920 camera with the field of view 78° to provide the LGMD with the visual input has been utilized. The image data fed into the LGMD neural network has been subsampled to the resolution of 176×144 pixels and divided into two parts overlapping in 10% of the image area.

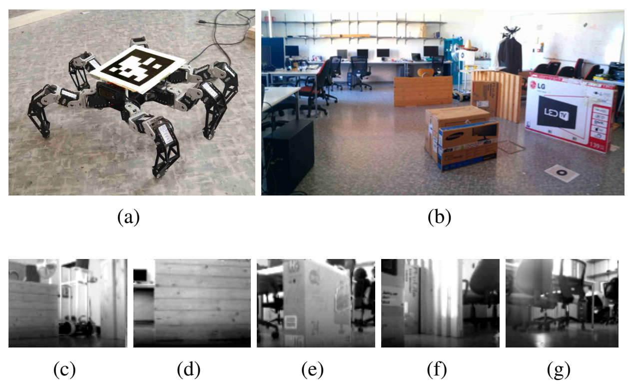


Figure 5: (a) The hexapod walking robot, (b) the laboratory test environment, and (c-g) typical images captured by the robot

The robot has operated in an arena surrounded by obstacles, which are formed by tables, chairs and boxes (see Fig. 5b). The robot movement has been tracked by a visual localization system which tracks the AprilTag [20] pattern attached to the robot, which allows to capture the real trajectory the robot was traversing. Typical images captured by the robot during traversing the arena are visualized in Fig. 5c-g. As the LGMD reacts strongly on the lateral movement of vertical edges in the image, it is much harder to avoid obstacles in the cluttered environment where the edges are distributed non-homogeneously in contrast to experiments performed in [2, 4, 6].

4.1 RNN Training Process

The LSTM neural network [18] has been trained using the BPTT technique [19]. The training process has been performed as follows. First, 10 sample trajectories have been collected by manually guiding the robot through the environment while avoiding the obstacles. The outputs of both the LGMDs have been recorded and the parameter *turn* has been adjusted manually, from which the corresponding error parameter e has been computed. The sampled trajectories contain altogether 22530 sample points. Next, the neural network has been trained with these 10 trajectories in 1000 iterations.

The herein utilized RNN has 2 inputs, 16 hidden states, and 1 output. The 16 hidden states have been selected as a compromise between the complexity of the RNN and the behavior observed during the experimental verification. As one of the problems of the former solution is the behavior of the robot when it successfully initiate the obstacle avoidance but it then hits it from the side, we selected 16 hidden states as the memory buffer to provide sufficient capacity for the robot to traverse 0.4 m given its dimensions, speed and camera frame rate.

The sigmoid function has been used as the activation function of the RNN

$$f(x) = \frac{1}{1+\mathrm{e}^{-x}}. \tag{10}$$

As the LGMD outputs are in the range $u_f \in [0.5, 1]$ and the error function $e \in [-0.5, 0.5]$, the RNN has been trained to estimate the value of $e + 0.5$ which is feasible for the sigmoid function with the range of $f(x) \in [0, 1]$.

4.2 Experimental Results

Altogether, 20 trials have been performed in the laboratory arena to verify the ability of the robot to avoid collisions. The robot has been directed to intercept different obstacles and its behavior has been observed. The algorithm failed only in 3 trials while the previous approach based on the direct control proposed in [3] is unable to operate in such a heavily cluttered environment at all. The first failed trial is specific by a direct collision with a low-textured wooden barrier (see Fig. 5d), hence the LGMDs failed to detect an approaching object. The second and third failures fall into the category of sideway interception when the robot successfully starts to avoid the obstacle but the robot hits it later from a side.

Fig. 6 shows three typical trajectories crawled by the hexapod robot in the laboratory arena. The trajectory is overlaid with the perpendicular arrows that characterize the direction and magnitude of the error e that is used for the robot steering which correspond to the direction in which the neural-based controller is sensing an obstacle. Besides, the corresponding plot of the LGMD outputs and the comparison of the control output provided by the proposed neural-based controller e_{rnn} and the direct control method e_{direct} is visualized in Fig. 7.

Further, we let the robot to continuously crawl the area and avoid obstacles. The robot has crawled the distance of approx. 140 m while colliding only 8 times.

4.3 Discussion

The presented results indicate that the proposed neural-based locomotion controller with the collision avoidance feedback provided by the LGMD neural network and the RNN-based controller is feasible. Moreover, the utilization of the RNN considerably improves the collision avoiding behavior in comparison to the direct control mechanism presented in [3]. The difference between the control principles can be best observed in Fig. 7a. It can be seen that the RNN filters oscillations in the error e which would disable the robot from avoiding the collision in a case of the direct control.

On the other hand, it is not particularly clear what is the RNN-based controller reacting to, as the dependency of the output on the distance to the closest obstacle has not been confirmed. This can be observed in Fig. 6c and the corresponding plot of the error function in Fig. 7c where the controller starts to oscillate after successfully avoiding the first obstacle. Other experimental trials have shown that these oscillations do not affect the collision avoiding behavior; however, it is unclear how and why they are produced by the neural controller.

The results indicate that the RNN calculates a weighted average of the LGMD outputs over a short period. However, further analysis of the behavior of the controller is necessary to reliably evaluate its properties.

Last but not least, the proposed controller performs only a collision avoiding behavior and does not guide the robot to any particular goal. Thus, we consider an extension of the proposed method to incorporate a higher level goal following to the architecture of the neural-based controller as a future work.

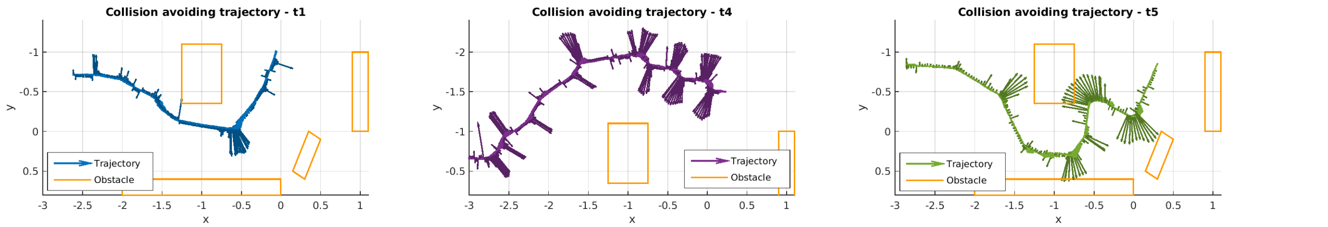
5 Conclusion

In this paper, we propose an extension of the biologically based collision avoidance approach with a Recurrent Neural Network to enhance the collision avoiding behavior of a hexapod walking robot. The proposed extension allows the robot to operate in heavily cluttered environments. The herein presented experimental results indicate feasibility of the controller which failed to avoid collision in only 3 out of 20 performed trials. The experimental results raised questions about the cause of the observed oscillations that deserve future investigation. Besides, we aim to improve the proposed biologically-based architecture to follow a specific target location, and thus developed biologically inspired autonomous navigation.

Acknowledgments – This work was supported by the Czech Science Foundation (GAČR) under research project No. 15-09600Y. The support of the Grant Agency of the CTU in Prague under grant No. SGS16/235/OHK3/3T/13 to Petr Čížek is also gratefully acknowledged.

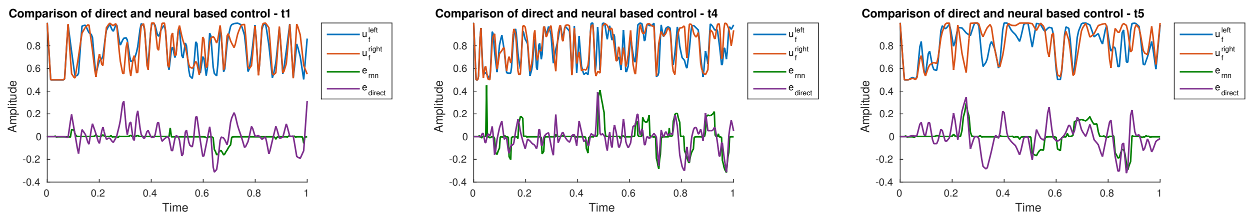
References

[1] P. Milička, P. Čížek, and J. Faigl, “On chaotic oscillator-based central pattern generator for motion control of hexapod walking robot,” in *Informačné Technológie - Aplikácie*



(a) (b) (c)

Figure 6: Collision avoiding trajectories for the experiments t_1, t_4, and t_5



(a) (b) (c)

Figure 7: Comparison of the control output provided by the proposed neural-based controller e_{rnn} and the direct method e_{direct} for the experiments t_1, t_4, and t_5

a Teória (ITAT), CEUR Workshop Proceedings, vol. 1649, 2016, pp. 131–137.

[2] M. Blanchard, F. Rind, and P. F. M. J. Verschure, "Collision avoidance using a model of the locust LGMD neuron," *Robotics and Autonomous Systems (RAS)*, vol. 30, no. 1-2, pp. 17–38, 2000.

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

[4] S. Yue and F. C. Rind, "Collision detection in complex dynamic scenes using an LGMD-based visual neural network with feature enhancement," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 705–716, 2006.

[5] Q. Fu, S. Yue, and C. Hu, "Bio-inspired collision detector with enhanced selectivity for ground robotic vision system," in *British Machine Vision Conference*, 2016.

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

[7] D. A. Pomerleau, *Neural network perception for mobile robot guidance*. Springer Science and Business Media, 2012, vol. 239.

[8] D. M. Wilson, "The central nervous control of flight in a locust," *Journal of Experimental Biology*, vol. 38, no. 47, pp. 471–490, 1961.

[9] M. Hartbauer, "Simplified bionic solutions: a simple bio-inspired vehicle collision detection system," *Bioinspiration and Biomimetics*, vol. 12, no. 2, p. 026007, 2017.

[10] S. Yue and F. C. Rind, "Redundant neural vision systems – competing for collision recognition roles," *IEEE Transactions on Autonomous Mental Development*, vol. 5, no. 2, pp. 173–186, 2013.

[11] H. G. Meyer, O. J. N. Bertrand, J. Paskarbeit, J. P. Lindemann, A. Schneider, and M. Egelhaaf, *A Bio-Inspired Model for Visual Collision Avoidance on a Hexapod Walking Robot*, 2016, pp. 167–178.

[12] J. Kodjabachian and J. A. Meyer, "Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 796–812, 1998.

[13] B.-Q. Huang, G.-Y. Cao, and M. Guo, "Reinforcement learning neural network to the problem of autonomous mobile robot obstacle avoidance," in *International Conference on Machine Learning and Cybernetics*, vol. 1, 2005, pp. 85–89.

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

[15] S. Steingrube, M. Timme, F. Wörgötter, and P. Manoonpong, "Self-organized adaptation of a simple neural circuit enables complex robot behaviour," *Nature Physics*, vol. 6, no. 3, pp. 224–230, 2010.

[16] N. Porcino, "Hexapod gait control by a neural network," in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 1990, pp. 189–194.

[17] D. M. Wilson, "Insect walking," *Annual Review of Entomology*, vol. 11, no. 1, pp. 103–122, 1966.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] I. Sutskever, "Training recurrent neural networks," Ph.D. dissertation, University of Toronto, 2013.

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