Evolving Controllers for Robots with Multimodal Locomotion

Rita Ramos, Miguel Duarte, Sancho Moura Oliveira, and
Anders Lyhne Christensen

BioMachines Lab, Instituto de Telecomunicações &
Instituto Universitário de Lisboa (ISCTE-IUL)
Lisbon, Portugal
rita_parada@iscte.pt

Abstract Animals have inspired numerous studies on robot locomotion, but the problem of how autonomous robots can learn to take advantage of multimodal locomotion remains largely unexplored. In this paper, we study how a robot with two different means of locomotion can effective learn when to use each one based only on the limited information it can obtain through its onboard sensors. We conduct a series of simulation-based experiments using a task where a wheeled robot capable of jumping has to navigate to a target destination as quickly as possible in environments containing obstacles. We apply evolutionary techniques to synthesize neural controllers for the robot, and we analyze the evolved behaviors. The results show that the robot succeeds in learning when to drive and when to jump. The results also show that, compared with unimodal locomotion, multimodal locomotion allows for simpler and higher performing behaviors to evolve.

Keywords: Evolutionary robotics, multimodal locomotion, navigation task

1 Introduction

Animals' ability to move efficiently in complex environments is crucial for key activities related to their survival, such as finding food and escaping predators. As a means to efficiently move through complex and unstructured environments, various animals exploit different modes of locomotion [11]. Birds, for example, use the aerial mode when traveling long distances, whereas the terrestrial mode is chosen for activities that require covering small distances, such as when feeding [14]. Crocodiles use terrestrial locomotion, a quadrupedal gait, when nesting and sunbathing, whilst for hunting, they rely on aquatic locomotion, primarily using undulation of the tail for propulsion [14].

Besides animals, multimodal locomotion has an important role in the field of robotics, particularly in tasks where robots may encounter distinct types of environments. Indeed, in some tasks, such as navigation, and search and rescue, it may be necessary to explore various types of terrains, which requires an adaption of movement modes, rather than just relying on one locomotion strategy [10]. Although some multimodal robots have been developed with distinct combinations of locomotion modes [1,3,20], the majority of them lacks the capacity for autonomous decision-making and are unable to decide when to use each means of locomotion.

Evolutionary Robotics (ER) is a field in which controllers for autonomous robots are synthesized by means of evolutionary computation techniques without the need for manual and detailed specification of behavior. In ER, there have been numerous studies on the evolution of controllers for robots with distinct means of locomotion, ranging from terrestrial and aerial robots, to aquatic robots [4,16,21]. Evolved controllers, however, have so far only made use of one means of locomotion. In this study, we evolve control systems for robots that have the capacity to exploit two modes of locomotion during task execution, namely driving and jumping.

For our study, we use a robot model based on the *Jumping Sumo*, a low-cost robotic platform made by Parrot. The robot has to perform a navigation task in different environments with obstacles. In order to successfully perform the task, the robot must reach a predefined destination as quickly as possible. The Jumping Sumo has a jumping mechanism that has to charge for 1 second before a jump can be executed. The need to charge prior to jumping and the fact the robot rolls stochastically after landing, make jumping slower than driving. There is thus a tradeoff, because the robot has to go around obstacles when driving, whereas the jumping locomotion, although slower, enables the robot to jump over obstacles. Taking into account the tradeoff between the two means of locomotion, we evolve control in a balanced set of environments that is fair for both locomotion modes. We compare results obtained in three distinct setups in which a robot has access to different modes of locomotion, in (i) the robot can only drive, not jump, in (ii) the robot can only jump, not drive, and in (iii) the robot is capable of both driving and jumping. We then analyze the performance and behavior of the controllers evolved in each setup. The contribution of our study is fourfold: (i) we evolve controllers that can take advantage of jumping locomotion; (ii) we demonstrate how controllers can be synthesized for multimodal locomotion, in particular, jumping and driving; (iii) we show that simpler strategies can be evolved for robots with multimodal locomotion capabilities compared with strategies evolved for robots with unimodal locomotion, and (iv) we find that the navigation strategies evolved for multimodal robots outperform strategies evolved for unimodal robots – even when only one mode of locomotion is used.

2 Related Work

In this section, we present prominent multimodal robots and discuss work related to autonomous navigation in the field of ER.

2.1 Multimodal Robots

Robots equipped with more than one means of locomotion have the potential to select which mode to use depending on the types of environment encountered, which is particularly important in tasks where terrains may not be entirely characterized prior to deployment [10]. With the growing interest in using robots for search and rescue tasks, environmental monitoring, and so forth, it is increasingly important to have robots with the capacity to exploit a variety of locomotion strategies.

Of the limited number of multimodal robots that have been developed so far, most combine aquatic and terrestrial locomotion. Examples include Aqua [8] and Salamandra Robotica I and II [3]. There is also a number of robots that combine aerial and terrestrial locomotion, such as MALV [1] and BOLT [17].

Some multimodal robots rely on the combination of jumping and wheeled locomotion, in which wheeled locomotion is the primary means of locomotion and jumping is used as a secondary means. Tsukagoshi et al. [19] developed a wheeled robotic platform with a jumping mechanism for rescue operations. The jumping mechanism uses a pneumatic cylinder and a specially designed valve that allows energy efficient and high jumps. The Jumping Sumo is another example of a wheeled robot capable of jumping that Parrot has recently developed[1], along with other multimodal robots. Other examples include the miniature Scout robot [18], a cylindrical robot with two wheels, and the mini-whegs [13].

Besides the mentioned examples, a recent survey on robotic systems equipped with multimodal locomotion can be found in [15]. Despite the interesting work done so far, most of multimodal robots are unable to autonomously decide when and how to exploit the different locomotion modes during task execution; in fact, the majority of current multimodal robots lack the capacity for autonomous decision-making altogether. In this paper, we study how to automatically synthesize controllers for a robot equipped with multimodal locomotion so that it effectively chooses which mode of locomotion during task execution based only on limited information from onboard sensors.

2.2 Evolved Navigation Behaviors

Terrestrial robots Several ER studies on wheeled robots have been carried out since the pioneering real-robot studies by Floreano and Mondada [6] and Jakobi et al. [12]. In [6], the authors evolved behavioral control that enabled a Khepera robot to locate a battery charger and periodically return to it. In [12], the authors managed to successfully evolve artificial neural network-based control for obstacle-avoidance and light-seeking tasks for a Khepera robot. Many others examples of evolved behaviors for terrestrial robots can be found in [16].

Besides wheeled robots, legged robots have also been controlled by evolved behavior. Gallagher et al. [7], for instance, carried out experiments using a

[1] Parrot MiniDrone Jumping Sumo, URL: `http://www.parrot.com/usa/products/jumping-sumo/`

neural network to control the locomotion of a real six-legged robot. Gruau and Quatramaran [9] attempted to evolve an artificial neural network with cellular encoding to control the locomotion of OCT-1, an eight-legged robot.

Aerial and aquatic robots One example of ER in aerial robots includes evolving spiking neural controllers for a flying robot which had to perform a vision-based navigation task [21]. In terms of the aquatic environment, control was recently evolved for a swarm robotics system composed of 10 surface robots, in a study that demonstrated evolved swarm control outside of controlled laboratory conditions [4].

As it is the case for all the studies discussed above, evolution of control has almost exclusively been applied to robots with one type of locomotion. In this study, we evolve control for robots capable of multimodal locomotion, in particular, jumping and driving. It should be noted that, to the best of our knowledge, no controllers have been evolved for robots capable of jumping prior to this study.

3 Robot Model and the Task

In this section, we describe the navigation task and the robot model used in our experiments. We conducted our experiments in JBotEvolver, a Java-based open-source, multirobot simulation platform and neuroevolution framework [5].

3.1 Navigation Task

In our task, the robot must navigate to a predefined target destination in an environment with different obstacle configurations. The configuration of obstacles is random, but generated according to a predefined *ratio* that determines the optimal time to complete the task by driving relative to the optimal time to complete the task by jumping. For instance, if an environment has a ratio of 2, a configuration of obstacles will be generated in such a way that the time to reach the destination will be twice as long when driving than when jumping if the respective optimal paths are followed.

We use five different types of environment with the following ratios: 1/4, 1/2, 1, 2, and 4, during the evolutionary process. In the first two types of environment, the robot can potentially reach the destination faster by driving than by jumping (ratios 1/4 and 1/2), while the opposite is true in the two final types of environment (ratios 2 and 4). In the environment with a ratio of 1, the two means of locomotion potentially allow the robot to reach the target destination equally fast. Solutions are thus evaluated in a balanced set of environments with respect to the two means of locomotion. All environments are bounded and square-shaped, with a side length of 10 m. An example of a random configuration of the five environments can be seen in Figure 1.

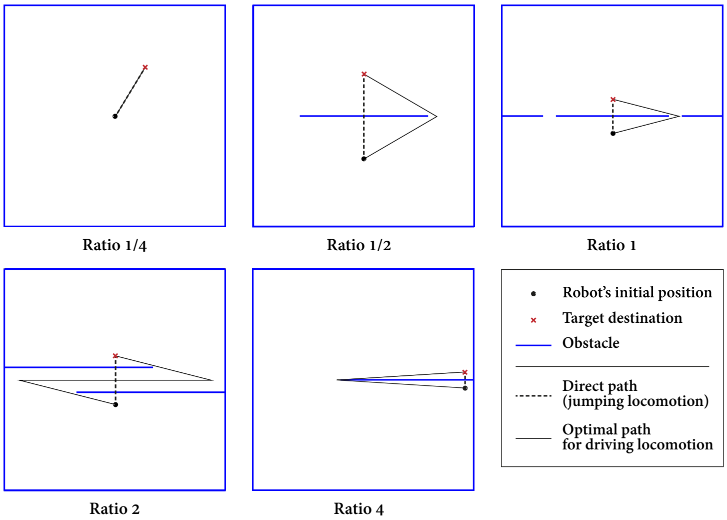

Figure 1: Examples of configurations of the five types of environment with the respective optimal paths for driving and jumping highlighted.

3.2 The Robot Model

The robot model is based on an existing physical multimodal robot, the Jumping Sumo, a differential wheeled robot capable of jumping (see Figure 2). The robot is equipped with two wheels, can move at up to 7 km/h, is able to prioritizing either height or length, and has a size of 18.5 x 15 x 11 cm.

We conducted a series of empirical tests to assess the jumping characteristics of the robot in order to model the Jumping Sumo in simulation. The jump mode with height prioritization was chosen for our experiments since it allows the robots to overcome tall obstacles. A total of 45 jumps were executed, and both the time and distance covered were recorded. In the empirical tests, we observed a jump length of 85 ± 4 cm, and a roll distance of 16 ± 7 cm after the jump. In simulation, the dynamics were modeled using two Gaussian distributions. The jumping mechanism has to charge for 1 second before a jump is executed, which was also modeled in simulation.

In our experiments, the robot was equipped with two actuators: two wheels and a jump actuator. A Gaussian noise component with a mean of 0 and a standard deviation of 5% was added to the two wheels to simulate real-world phenomena, such as imperfect motors and wheel slippage. The set of sensors includes eight destination sensors, which have a maximum range of 10 m, and eight obstacle sensors, which have a range of 4 m. The destination sensors are distributed around the chassis of the robot and the obstacle sensors are distributed on the front of the robot. All sensors have an opening angle of 60°. The robot was further equipped with a proprioceptive sensor that indicates if the robot is currently jumping or not.

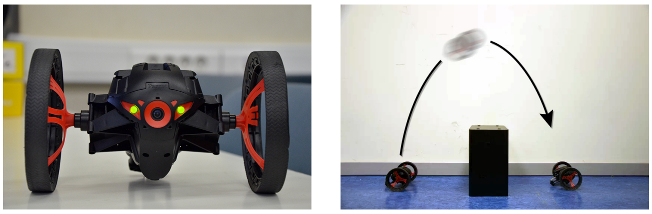

Figure 2: Left: Jumping Sumo. Right: jump trajectory.

For our experiments, we use three setups in which the robot has access to distinct locomotion capabilities: (i) Drive, where the robot is only capable of driving, (ii) Jump, where the robot can only jump and rotate on its axis, and (iii) Drive-and-Jump, where the robot is capable of multimodal locomotion, and thus can both jump and drive. In the Jump setup, the robot is considered to have reached the destination when it is within 30 cm of the target, instead of the 10 cm used for the other two setups, given that jumping is less precise than driving.

4 Control Synthesis

We evolve continuous-time recurrent neural networks [2] to control the robot. Each neural network has three layers of neurons: a reactive input layer, a fully connected hidden layer, and an output layer. The input layer is fully connected to the hidden layer, which, in turn, is fully connected to itself and to the output layer. The input layer has one neuron per input sensor and the output layer has one neuron per actuator output.

We use a simple generational evolutionary algorithm to synthesize control for the robot. Each generation is composed of 100 genomes that correspond to artificial neural networks with the topology outlined above. Genomes are evaluated over 25 samples with different initial random seeds (5 samples in each type of environment) and the average fitness is used for selection. Each sample can last up to 750 simulation time steps (19 seconds), or terminates once the robot reaches the destination. The five highest-scoring genomes are selected to become part of the next generation and to populate it. Each of the top genomes becomes the parent of 19 offspring. The genotype of an offspring is the result of applying a Gaussian noise to each gene with a probability of 10%.

The fitness function is defined as follows:

$$F(i) = R_i + P_i \tag{1}$$

where R_i is a reward component and P_i a penalty component. Value of R_i depends on whether the robot succeeded or failed to navigate to the target destination:

$$R_i = \begin{cases} 1 + (T - t)/T & \text{if robot reached the destination} \\ 1 - d/D & \text{otherwise} \end{cases} \tag{2}$$

If the robot did not reach the target destination, the fitness function has a value in $[0,1]$ depending on how close the robot got to the destination during the experiment. By means of this bootstrapping component, a faster convergence to the destination is expected. The term D represents the initial distance between the robot and the target destination, and d is the closest distance the robot came to the destination. If the robot is successful, its fitness will be in the interval $[1,2]$, depending on how long the robot took to reach the target destination. The term T is the maximum time available for the task (750 simulation steps) and t corresponds to the time needed to reach the destination.

P_i is the penalty component which is used to promote obstacle avoidance:

$$P_i = \begin{cases} nc \times -0.01 & \text{if robot reached the destination} \\ nc \times -0.001 & \text{otherwise} \end{cases} \tag{3}$$

The term nc denotes the number of collisions with obstacles. P_i also depends on whether or not the robot managed to reach the destination. A lower penalty is given when the robot was unable to reach the target destination in order to bootstrap the evolutionary process.

5 Results

A total of 30 evolutionary runs were conducted for each setup (Drive, Jump, and Drive-and-Jump), each lasting 500 generations. For the highest-scoring controller evolved in each run, we conducted a post-evaluation with 100 samples for each of the five environment types used during evolution. In this section, we present the results obtained in each setup, and we analyze the performance and behaviors of the evolved solutions.

5.1 General Performance

Figure 3(left) shows the distribution of fitness scores achieved by the highest-scoring controllers for the different experimental setups. The highest-scoring controller was found in the Drive-and-Jump setup (1.84 ± 0.06), followed by the similar performance of the Drive setup and the Jump setup (1.71 ± 0.15 and 1.71 ± 0.14, respectively). Figure 3(right) shows the distribution of the time to reach the target destination by those highest-scoring controllers. As mentioned in Section 3.1, the experiments were conducted in five types of environments that equally favor the driving locomotion and the jumping locomotion. The results show that the highest-scoring controllers evolved in the Drive setup and the Jump setup obtained a similar performance. The highest-scoring controller of the Drive setup reached the destination within a mean time of 5.69 ± 2.80 s, and the best controller of the Jump setup reached the destination with a mean time of 5.54 ± 2.96 s. The highest-scoring controller evolved in the Drive-and-Jump setup successfully reached the destination faster than controllers evolved in the other setups (3.10 ± 1.20 s). The results demonstrate that robots with multimodal locomotion capabilities can effective learn when to use each mode of locomotion based on the limited information they can obtain through their onboard sensors.

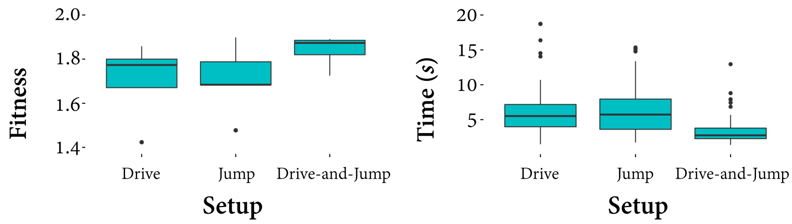

Figure 3: Left: distribution of fitness scores achieved by the highest-scoring controllers of the three setups (higher is better). Right: distribution of the time to reach the destination by the highest-scoring controllers of the three setups (lower is better).

The simplicity of evolving successful behaviors with multimodal locomotion can be seen in Figure 4(left): the evolutionary process found successful solutions around the 80th generation, after which fitness only slightly increased. The relatively low average performance displayed by the controllers evolved in the Drive setup (Figure 4(right)), can be explained by the fact that the majority of them did not succeed in reaching the destination in the final two environments (ratio 2 and 4). It is thus more challenging to evolve effective Drive behaviors than multimodal behaviors.

5.2 Behavioral Analysis

In this section, we analyze the performance and behaviors of the highest-scoring controllers for the first (ratio 1/4), middle (ratio 1) and last environment (ratio

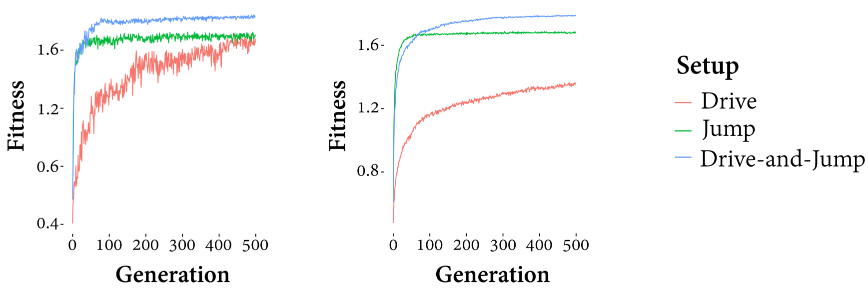

Figure 4: Left: fitness trajectories of the highest-scoring controllers in each generation. Right: average fitness trajectories of the highest-scoring controllers in each of the 30 runs.

4). Examples of evolved behaviors can be seen in Figure 5. The highest scoring controller of the Drive setup, Jump setup and Drive-and-Jump setup are hereinafter referred to as *Drive Controller*, *Jump Controller* and *Drive-and-Jump Controller*, respectively.

Ratio 1/4 The *Drive Controller* reached the destination with a mean time of 2.99 ± 1.33 s, while the *Jump Controller* achieved a mean time of 7.49 ± 1.82 s and the *Drive-and-Jump Controller* outperformed both with a mean time of 1.90 ± 0.28 s. The *Drive Controller* was not as fast as the *Drive-and-Jump Controller*, due to the fact that the controller has the general behavior of turning on the spot to find a way around potential obstacles, as can be seen in Figure 5(a), which is necessary in order to solve the navigation task in more complex environments. Whereas the *Drive-and-Jump Controller* has a more general behavior, since it is not limited to just one locomotion strategy. Successful behaviors leveraged the ability to overcome obstacles in the other environments by jumping over them, thereby using a simpler strategy in which the robots moves directly toward the destination in the environments.

Ratio 1 This environment has a ratio of 1, thus, the robot can potentially reach the destination in the same amount of time whether it drives or jumps. The *Drive Controller* reached the destination within a mean time of 4.49 ± 1.77 s, similar

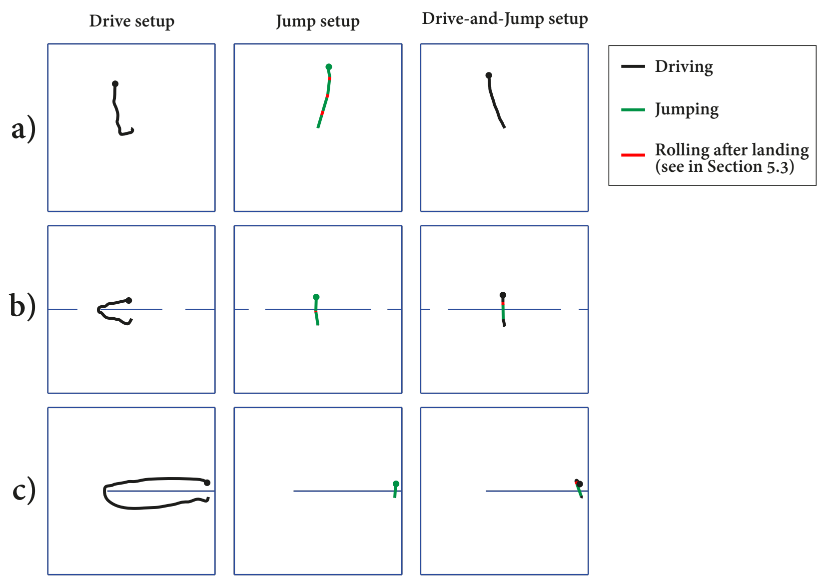

Figure 5: Example of behaviors in the environments with a) ratio 1/4, b) ratio 1, and c) ratio 4.

to the *Jump Controller* that achieved a mean time of 4.11 ± 2.11 s. As to the *Drive-and-Jump Controller*, once again outperformed the other two, achieving a mean task-completion time of 2.59 ± 0.28 s.

We observed the *Drive-and-Jump Controller* successfully combining both modes of locomotion by driving toward an obstacle, jumping over it and then driving again to the destination, therefore achieving a even better performance when compared with using just one of the locomotion strategies.

Ratio 4 The *Drive Controller* reached the destination within a mean time of 8.91 ± 3.47 s, while the *Jump Controller* achieved a mean time of 2.13 ± 0.72 s. In the Drive-and-Jump setup, the mean time to navigate to the destination was 2.83 ± 1.62 s. The reason why the *Jump Controller* had a better performance than the *Drive-and-Jump Controller* is only due to the fact that the robot is considered to reach the destination at a greater distance with the Jump setup than with the Drive-and-Jump setup, as explained in Section 3.2. The robot, therefore, needs to drive a short distance after jumping, whereas in the Jump setup, the robot reaches the destination immediately upon landing.

5.3 Generalization

In order to assess how general the evolved strategies are, we conducted an additional set of post-evaluation experiments using the highest-scoring controllers of each setup in 12 additional environments, which were not used during evolution. The distribution of ratios for the new environments were chosen to uniformly fill the gaps between the ratios of the five original environment types. Each controller was evaluated 100 times in each of the 17 environments. The distribution of how

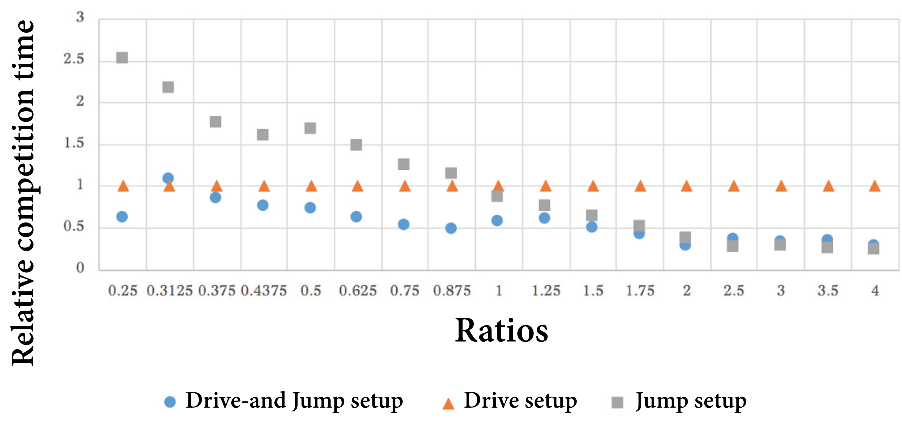

Figure 6: Distribution of task completion times of the highest-scoring controllers evolved in a range of different environments. Highest-performing controller from the Drive setup is used as baseline

long it took to complete the task in each of the 17 environments, using the *Drive Controller* as baseline, can be seen in Figure 6.

The results show that, as one might expect, the *Drive Controller* outperformed the *Jump Controller* when the ratio was less than 1. With ratios higher than 1, the *Jump Controller* achieve a higher performance than the *Drive Controller*. The *Drive-and-Jump Controller* outperformed both *Drive Controller* and *Jump Controller* in the majority of the environments.

6 Conclusions

In this study, we evolved control for robots with multimodal locomotion. To conduct our experiments, the robot had to perform a navigation task in which it had to reach a target destination as quickly as possible. The navigation task was conducted in different environments, and we compared three modes of locomotion: (i) driving locomotion, (ii) jumping locomotion, and (iii) multimodal jumping and driving locomotion.

In our simulation-based experiments, the robot equipped with multimodal locomotion, was able to adapt its locomotion strategy to a broad range of different environments. Depending on the environment, the robot navigated to the target destination either by combining jumping and driving locomotion or by exploiting the one that best suited the environment. Multimodal locomotion enabled the robot to reach the destination faster than when limited to just one locomotion strategy. The evolved behavior shows that even in environments in which only one type of locomotion is necessary, the robot can be faster by relying on a general behavior.

The concept of evolving robotic control systems that have the capacity to exploit driving and jumping locomotion opens several possibilities for others combinations of two (or more) modes of locomotion, such as jumping and flying. In our ongoing work, we are studying large-scale systems of autonomous robots with multimodal locomotion capabilities.

Acknowledgments We would like to acknowledge the financial support of ISCTE-IUL. Also, this work was partly supported by FCT – Foundation of Science and Technology under grant UID/EEA/50008/2013

References

1. Bachmann, R.J., Boria, F.J., Vaidyanathan, R., Ifju, P.G., Quinn, R.D.: A biologically inspired micro-vehicle capable of aerial and terrestrial locomotion. Mechanism and Machine Theory 44(3), 513–526 (2009)
2. Beer, R.D., Gallagher, J.C.: Evolving dynamical neural networks for adaptive behavior. Adaptive behavior 1(1), 91–122 (1992)
3. Crespi, A., Karakasiliotis, K., Guignard, A., Ijspeert, A.J.: Salamandra robotica II: an amphibious robot to study salamander-like swimming and walking gaits. IEEE Transactions on Robotics 29(2), 308–320 (2013)

4. Duarte, M., Costa, V., Gomes, J., Rodrigues, T., Silva, F., Oliveira, S.M., Christensen, A.L.: Evolution of collective behaviors for a real swarm of aquatic surface robots. PLoS ONE 11(3), e0151834 (2016)
5. Duarte, M., Silva, F., Rodrigues, T., Oliveira, S.M., Christensen, A.L.: JBotEvolver: A versatile simulation platform for evolutionary robotics. In: International Conference on the Synthesis and Simulation of Living Systems (ALIFE). pp. 210–211. MIT Press (2014)
6. Floreano, D., Mondada, F.: Evolution of homing navigation in a real mobile robot. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 26(3), 396–407 (1996)
7. Gallagher, J.C., Beer, R.D., Espenschied, K.S., Quinn, R.D.: Application of evolved locomotion controllers to a hexapod robot. Robotics and Autonomous Systems 19(1), 95–103 (1996)
8. Georgiades, C., German, A., Hogue, A., Liu, H., Prahacs, C., Ripsman, A., Sim, R., Torres, L.A., Zhang, P., Buehler, M., et al.: AQUA: an aquatic walking robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). vol. 4, pp. 3525–3531. IEEE Press (2004)
9. Gruau, F., Quatramaran, K.: Cellular encoding for interactive evolutionary robotics. In: European Conference on Artificial Life (ECAL). pp. 368–377. MIT Press (1997)
10. Ijspeert, A.J.: Biorobotics: Using robots to emulate and investigate agile locomotion. Science 346(6206), 196–203 (2014)
11. Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: a review. Neural Networks 21(4), 642–653 (2008)
12. Jakobi, N.: Minimal simulations for evolutionary robotics. Ph.D. thesis, University of Sussex (1998)
13. Lambrecht, B.G., Horchler, A.D., Quinn, R.D.: A small, insect-inspired robot that runs and jumps. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1240–1245. IEEE Press (2005)
14. Lock, R., Burgess, S., Vaidyanathan, R.: Multi-modal locomotion: from animal to application. Bioinspiration & Biomimetics 9(1), 011001 (2013)
15. Low, K., Hu, T., Mohammed, S., Tangorra, J., Kovac, M.: Perspectives on biologically inspired hybrid and multi-modal locomotion. Bioinspiration & Biomimetics 10(2), 020301 (2015)
16. Nolfi, S., Floreano, D.: Evolutionary robotics: The biology, intelligence, and technology of self-organizing machines. MIT Press (2000)
17. Peterson, K., Fearing, R.S.: Experimental dynamics of wing assisted running for a bipedal ornithopter. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5080–5086. IEEE Press (2011)
18. Stoeter, S.A., Papanikolopoulos, N.: Autonomous stair-climbing with miniature jumping robots. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35(2), 313–325 (2005)
19. Tsukagoshi, H., Sasaki, M., Kitagawa, A., Tanaka, T.: Design of a higher jumping rescue robot with the optimized pneumatic drive. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1276–1283. IEEE Press (2005)
20. Woodward, M.A., Sitti, M.: Multimo-bat: A biologically inspired integrated jumping–gliding robot. The International Journal of Robotics Research 33(12), 1511–1529 (2014)
21. Zufferey, J.C., Floreano, D., Van Leeuwen, M., Merenda, T.: Evolving vision-based flying robots. In: Biologically Motivated Computer Vision. pp. 592–600. Springer (2002)