Evolution of Controllers for Robot-Plant Bio-Hybdrids: A Simple Case Study Using a Model of Plant Growth and Motion

Mostafa Wahby[1], Mohammad Divband Soorati[1], Sebastian von Mammen[2], Heiko Hamann[1]

[1]Heinz Nixdorf Institute, Department of Computer Science, University of Paderborn, Paderborn, Germany
E-Mail: mostafa.wahby@uni-paderborn.de

[2]Organic Computing, University of Augsburg, Augsburg, Germany
E-Mail: sebastian.von.mammen@informatik.uni-augsburg.de

Introduction

In evolutionary robotics methods of evolutionary computation are applied to evolve robot controllers [1]. Evolutionary robotics is also our method of choice in the EU-funded project *flora robotica* [6, 8], which pushes research towards the evolution of a broad variety of artifacts and contraptions [4]. We are investigating how a distributed robot system and a group of biological plants can be tightly coupled to generate synergies and finally result in a bio-hybrid system. We want to create a co-dependent and self-organized system with closely linked symbiotic relationships where plants support robots, for example, by providing scaffolding and robots that direct plant growth towards desired areas. In addition to plant growth there is also plant motion which is often ignored, probably due to the fact that the motion of plants is slow compared to that of animals. It can be difficult sometimes to distinguish between a plant's motion and growth as both happen concurrently and usually in similar ways. They can differ, however, fundamentally in terms of time scales. Another important difference is the fact that motion can be reverted while growth is mostly permanent. Besides plant growth we also want to harness plant motion in our robot-plant systems.

In evolutionary robotics, the evaluation of an individual's fitness is a critical and often costly task (time, wear of robots, etc.). This is true for both options, evaluating robot controllers in simulation or via so-called embodied evolution with actual robots in hardware [17]. The potential speed-up of using simulations is diminished by the so-called reality gap problem [9]. Robot controllers evolved in simulation might perform poorly in reality due to imperfectly simulated features of reality. Still, the evaluation of a candidate controller might be performed within a matter of seconds. For example, in embodied evolution of an object avoidance controller the feedback of hitting an object can be instantly obtained from the robot's sensors and would immediately indicate a failure.

In this work, we apply artificial evolution to robot-plant bio-hybrid systems. An intuitive and simple task in this context is controlling plant growth and motion towards light (phototropism). Control here means to direct the plant tip to a certain position by switching lights on and off. It is obvious that in the case of the embodied evolution approach the feedback from the plants during an evaluation would be very slow. For example, the common bean plant (*Phaseolus vulgaris*), which is actually considered a fast grower, grows an average of 2cm to 3cm per day during an early growth stage. Hence, an evaluation period of at least three days would be required to grow the plant tip to reach a target initially 9cm away. Instead, we consider the option to priorly control plant motion. According to our experiments, bean plants bend towards a light source with a velocity of up to 4.4mm/min. This is considered relatively quick feedback and sufficient for evolving controllers in an embodied approach. Accordingly, the objective of our experiments is to maximize the plant motion within a given period of time by switching between different light settings. The resultant plant movement behavior could be useful, for instance, as a signal to draw the attention of passers-by in public spaces or to implement facades that dynamically provide visual protection. In this paper, we present our approach to create a simple growth-model of the common bean based on empirically obtained data. We then use the model as a simulation to evolve closed-loop controllers that maximize the plant's motion and grow the plant's tip to three different targets. For the future, we plan to investigate how well these evolved controllers work on real plants (reality gap), means of embodied evolution for plant motion control, and augmented user interfaces to effectively design and utilize bio-hybrid systems.

Related work and context of this research

Besides the above mentioned relation to evolutionary robotics [1] and embodied evolution of things [17, 4], our work is also related to efforts in plant modeling. In this paper we present an early preliminary plant modeling approach focused on the plant tip only. There is, however, a straightforward, simple incremental approach to extend the model by separating the plant logically into several segments. That is the idea of organizing the individual segments of a virtual plant stem in transformation hierarchies. This approach is similar to preceding works, for instance, for retracing motion dynamics of trees due to wind gusts [19]. As a result of the hierarchical organization, changes introduced to one segment of the plant affect all those segments lower in the hierarchy, or closer to the tip of the plant. In L-systems [10], which represent a wide-spread approach to model plant growth, a set of production rules is iteratively applied (in parallel) to subsume symbols of a character string which, in turn, is geometrically interpreted at each step of the simulation. The geometry of a tree and its variation have recently been inversely computed from polygon meshes of actual trees and derived abstract branching trees [19, 13]. Allowing the individual nodes in such an abstract tree to act as autonomous, reactive agents [18]. These agents, in turn, might be attracted to light sources, which provides a simple approach to model dynamic plants that react to environmental stimuli. To the knowledge of the authors, swarm grammars represent the first such implementation that also integrated the branching behavior of L-systems [15]. In fact, phototropism and reactions with the environment were among the first experiments conducted with this agent-based L-system extension. Considering the stiffness of each of the connected stem elements of the virtual plant in combination with the urge towards a light source, we can efficiently approximate the bending motion of the plant [5].

The plant growth–motion model and the evolution of controllers presented in this paper are an effort within the interdisciplinary project *flora robotica*. The project creates and investigates mixed societies of robots and natural plants and brings together scientists from plant science, architecture, zoology, robotics, and computer science. The general idea to automate gardening tasks has been addressed before [3]. Also the automation of agriculture has been studied [11]. However, our objectives go beyond the

idea of mere automation. Our motivation in *flora robotica* is to develop a bio-hybrid system that assigns equal roles to plants and robots and creates synergies between them.

A natural plant can grow structures and sense environmental features, while robots can impose artificial stimuli on the plant and add to the plant's sensing and decision-making capabilities. That way we use robots to trigger artificial growth. The idea is to leverage natural adaptive behavior in plants [7] to extend the capabilities of robots and to leverage the free programmability of robots to create artificial growth processes. Our main idea is to extend the rich variety of natural growth processes [14] with artificial growth processes. There are many potential applications but we focus on the artificial growth of potentially dynamic architectural artifacts.

The research presented in this paper is a machine learning approach to create robot controllers that influence the growth of plants. It is an initial step towards the creation of a complex bio-hybrid system, i.e. a decentralized, self-organizing multi-robot system tightly coupled with plants.

Bean growth and motion experiments

We have created a setup that allows us to conduct simple experiments with plants in order to study and explore the possibilities of controlling their growth and motion. The choice for a specific plant species was driven by the facts that (a) the speed of growth and motion is of great concern, and (b) the plant needed to grow in standard robot lab conditions and office areas. A good compromise to cope with these constraints is the common bean plant (*Phaseolus vulgaris*) in its early growth stage. In our setup, as soon as the bean plant starts sprouting, we place it inside a box of 2m height and 1,20m in width and breath, clad in black cloth from the inside to reduce light reflections and to allow for taking high contrast photos. Next, we impose a light stimulus on the plant using two light sources located 30cm above the plant, and 30cm to the left and right, respectively (see Fig. 1(a)). They are turned on alternatingly every six hours for a total period of a 72 hours.

(a) (b)

Figure 1: Simple growth and motion experiment: (a) Light from the left and right is alternatingly shone on a common bean plant. (b) Superposition of states indicating the amplitude of the plant's growth and left-right motion.

The light sources are two Adafruit NeoPixel RGB LED strips with 144 LEDs each. In our setting, each strip emits white light at full brightness which requires a current of up to 8.64A. Each LED has a power consumption of 0.24W and emits 18 lumen. A Raspberry Pi is used to operate the LED strips and a camera module[1] that takes a picture every five minutes, resulting in 864 images for each experiment, and 5184 images in total.

In Fig. 2 we show a sequence of photos from one of six experiments. The 16 photos depict growth and movement throughout an initial time period of 48 hours, with three hours between the shots[2]. After 48 hours the plant has grown to about 20cm. During the six-hour activations of each light source, the plant bends towards it while maintaining a counterclockwise turning behavior (standard climbing behavior of bean plants, not seen in the photos). In Fig. 1(b) a superposition in false color of several photos taken during the experiment is shown, clearly indicating the amplitude of the plant's left-right motion.

[1] `https://www.raspberrypi.org/products/camera-module/`
[2] Find a video at: `https://youtu.be/e-84bxhwpZo`

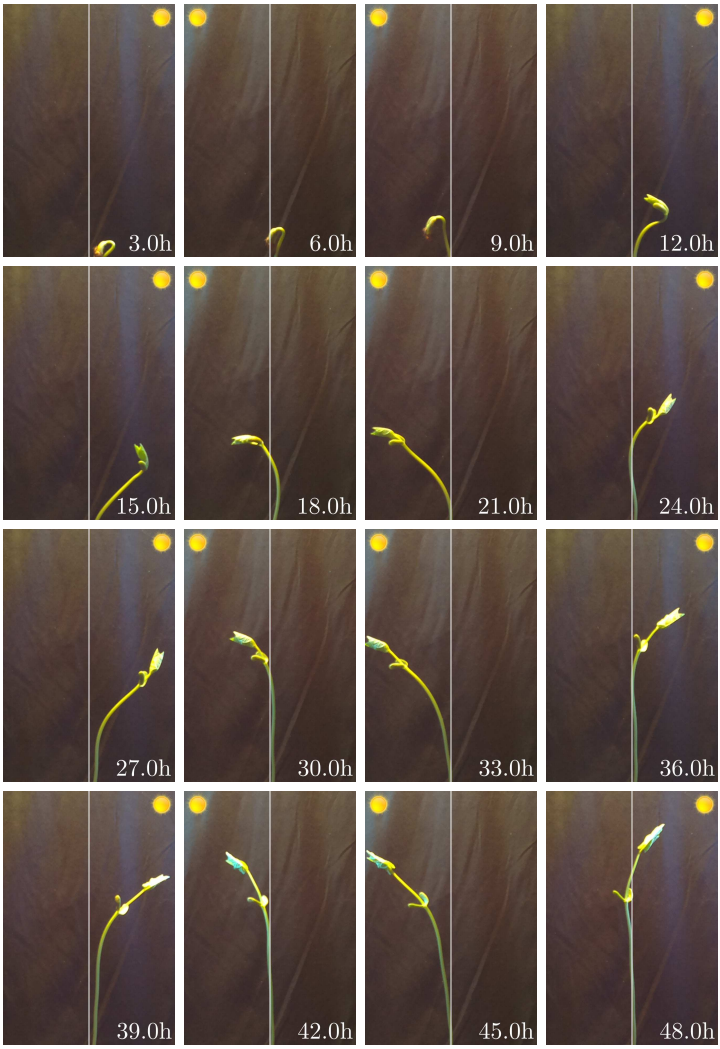


Figure 2: Photos of the bean plant at different times during the experiment showing rapid growth and motion towards light; the white line indicates the location of the roots, the sun symbolizes the activation of a light source to the left or right.

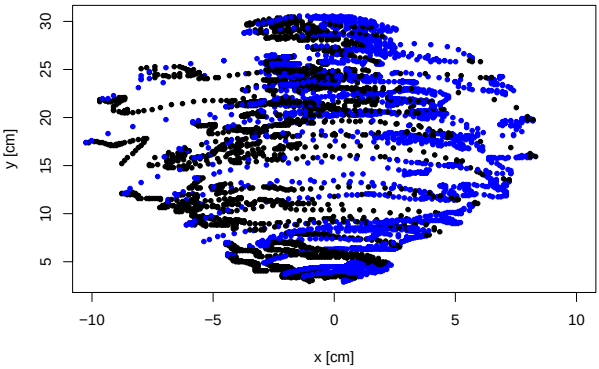
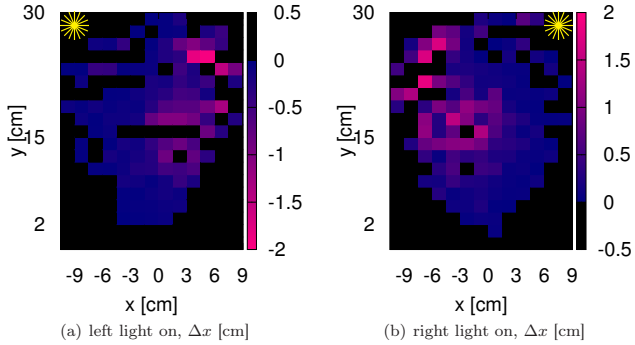


Figure 3: Plant tip positions from all our experiments. Black points denote the activation of the left, blue points the activation of the right light source.

Bean growth and motion model

We processed the obtained 2D images using the OpenCV library. Due to the high contrast between background and plant, we simply transformed the images to grayscale, applied a Gaussian filter to smoothen the images and then extracted the brightest points. The highest point of the plant (after cropping the area of actual plant growth, dismissing the pot and the light sources) is stored as its tip with the position $\mathbf{x} = (x, y)$ relative to the roots of the plant. The time series $\mathbf{x}_t$ for each experiment is a rough description of the plant's growth process and the effect of the controlled stimuli (Fig. 3).

We make use of this data to create a simple *bean growth–motion model*. We define a model that represents a plant's current tip position $\mathbf{x}_t$ and the current lighting condition L_t (boolean value indicating whether the left light is on). A current configuration of the system is then defined by $(\mathbf{x}, L)_t$. Using the collected data we derive the next tip position $\mathbf{x}_{t+1}$ for discrete time steps, each representing five minutes of real time. We assume that the two light sources are identical and also that the plant has no other bias to grow towards either of the two directions. Therefore, we mirrored the collected data to both sides (mapping $x \mapsto -x$ and keeping y identical) to



(a) left light on, Δx [cm]

(b) right light on, Δx [cm]

(c) left light on, Δy [cm]

(d) right light on, Δy [cm]

Figure 4: xy-motion of the bean plant's tip within 5min when activating the left/right light source (left-hand/right-hand column, respectively).

logically double the available data and hence to increase the precision of the model (note that the light conditions can still be distinguished: light on in the same quadrant or light on in the other quadrant).

In order to calculate the next plant tip position $\mathbf{x}_{t+1}$ for a given configuration $(\mathbf{x}, L)_t$, we define a rectangle $R = ((x - w_x, y - w_y), (x + w_x, y + w_y))$ with the plant tip position $\mathbf{x}$ at the center, width $2w_x$, and height $2w_y$ (i.e., a sliding window). Then we select all data points that are contained by rectangle R and that have the same light condition. Based on these data points we want to calculate a shift of the tip position Δx and Δy. From the selected data points and their successors in time we obtain samples of

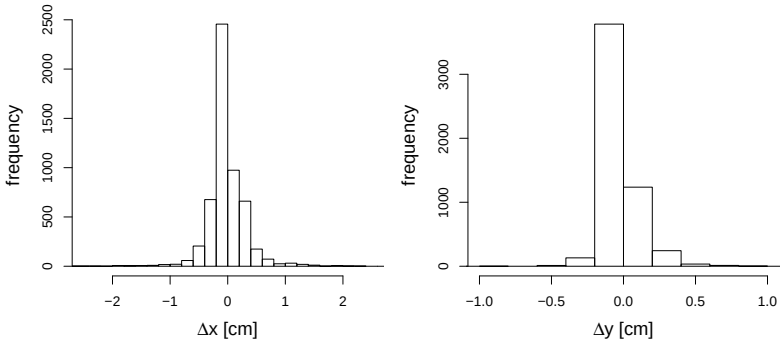


Figure 5: Histograms of plant tip position shifts acquired from all data points within 5min time period.

Δx and Δy. In Fig. 4 we show the original data of Δx and Δy within 5min time periods for all data points and both light settings before mirroring the data. The plant tip shift at the side opposite to the currently active light has the greatest absolute values. Note that, not only Δx but also Δy are subject to the plant's rotational motion which is usually centered at $x = 0$. We observe that at the bottom half ($y < 15$) the change in motion is generally smaller, probably because the plant's size limits its motion. In Fig. 5 we show the distribution of Δx and Δy for all data points. We interpret these distributions as normal distributions which can be expected for such a natural growth process.

We present and investigate three different methods to calculate the tip shift. The first is called *deterministic*, implements a deterministic model, and uses the mean values $\overline{\Delta x}$ and $\overline{\Delta y}$ of all data points inside the rectangular window. The second is called *stochastic*, implements a a stochastic model, and directly samples uniformly from the data points inside the rectangular window. The third method is called *mixed*, implements a mixture of the previous two methods, where the mean value $\overline{\Delta x}$ is used to calculate Δx, while Δy is randomly sampled from a normal distribution with a mean value of 0.04 and a standard deviation of value 0.01. The mean value of 0.04 was chosen to get overall heights of about 15cm for the chosen experiment length which corresponds to what we observed for the experiments with the natural plant.

Table 1: Used NEAT parameters.

Parameter	Value	Parameter	Value
PopulationSize	50	*CrossoverRate*	0.5
DynamicCompatibility	True	*MutateWeightsProb*	0.9
YoungAgeTreshold	15	*YoungAgeFitnessBoost*	1.0
OverallMutationRate	0.5	*WeightReplacementMax*	5.0
MinSpecies	5	*WeightMutationRate*	0.75
MaxSpecies	25	*Elitism*	0.1
SurvivalRate	0.6	*MutateAddNeuronProb*	0.04

Evolutionary approach

We use MultiNEAT [2] in combination with our simple growth–motion model to evolve closed-loop controllers in simulation. MultiNEAT is a portable software library implementing NEAT (NeuroEvolution of Augmenting Topologies) that uses the com plexification method [12] to evolve artificial neural networks (ANN). Table 1 specifies the NEAT parameters used in our experiments. These parameters are based on our experience from previous experiments where we evolved robot controllers for a parallel parking task [16]. We evolve ANN with two input neurons, a variable number of neurons in the hidden layer (determined by NEAT), one output neuron, and an unsigned step activation function.

The input of the network is the current plant tip position $\mathbf{x}_t = (x, y)$ (initially $\mathbf{x}_0 = (0, 0)$). For each time step (discrete time steps represent 5min in reality), the network's output simply determines the light condition L_t, that is, whether the left light or the right light is turned on. Next, the current configuration of the system $(\mathbf{x}, L)_t$ is passed to the simple growth–motion model in order to obtain the next tip position $\mathbf{x}_{t+1}$. This procedure is iterated for 300 time steps (representing 25 hours of real time), while evaluating the performance of the controller in respect to a certain task.

As a proof of concept, we evolve controllers for two simple tasks. (1) *maximum motion*: The task is to maximize the plant's overall motion (i.e., covered distance of the plant tip) during a time period of 25 hours, and (2) *three targets*: The plant has to approach three different desired positions

in space during the experiment. For both tasks, we conduct a set of three experiments using the above mentioned *deterministic*, *mixed* and *stochastic* method (i.e., calculation of the next plant tip position $\mathbf{x}_{t+1}$) to model the growth and motion behavior of the plant.

First, we define the fitness function F (eq. 1) to evaluate the performance of the individuals during the evolutionary processes within the maximum motion experiments. It simply accumulates the absolute value of all the plant tip position shifts Δx_t over 300 time steps:

$$F = \sum_{t=2}^{300} |\Delta x_t|, \text{ with } \Delta x_t = x_t - x_{t-1}. \tag{1}$$

For each experiment, 10 evolutionary runs of 100 generations each with a population size of 50 were performed. For the *deterministic* and *mixed* method we use only one repetition to evaluate a controller. For the *stochastic* method we do three repetitions, hence we get three fitness values F_0, F_1, F_2, and we define the controller's fitness conservatively as the minimum $F = \min\{F_0, F_1, F_2\}$. The best fitness of each run over generations for the *deterministic* method in the *maximum motion* experiment is shown in Fig. 6(a). Clearly, the task here is simple, therefore, convergence is achieved already at about generation 60. The behavior of the simulated plant tip (i.e., its trajectory) when running the best controller is shown in Fig. 7(a). The controller turns the right light on in the beginning, keeps it on to grow the plant tip to the right, then switches the left light on, and switches between the lights during the last few time steps. The fitness value of this controller is $F = 72.7$, which means the tip moved 72.7cm in total (horizontally). The best fitness for each run over generations for the *mixed* method for the *maximum motion* experiment is shown in Fig. 6(b). Similarly, the evolutionary approach converges quickly because of the task's simplicity. In Fig. 7(b) we show a trajectory of the plant tip sampled from the stochastic simulation (plant tip motion in height is stochastic, motion in width is deterministic) when running the best controller with a fitness value $F = 135.5$. The controller is turning lights on and off frequently to maximize the plant motion. The best fitness over generations for the *stochastic* method in the *maximum motion* experiment is shown in Fig. 6(c). Here, the complexity of the task is increased because both the plant tip's

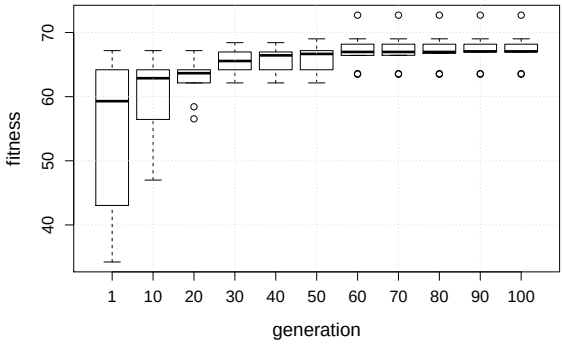
motion in height and width are stochastic and also variant because they are sampled directly now. As a result, we observe a non-monotonic increase of fitness interrupted by sudden drops. There is only a small increase in fitness within 100 generations and no saturation. We have tested the initial populations against the final populations using the Wilcoxon Rank-Sum Test to check whether the evolved controllers perform better than random. The test indicates that the controllers of generation 100 are significantly better. The simulated behavior of the tip when running the best controller of fitness value $F = 96.57$ is shown in Fig. 7(c).

Considering the objective to reach *three targets*, the fitness function is defined as

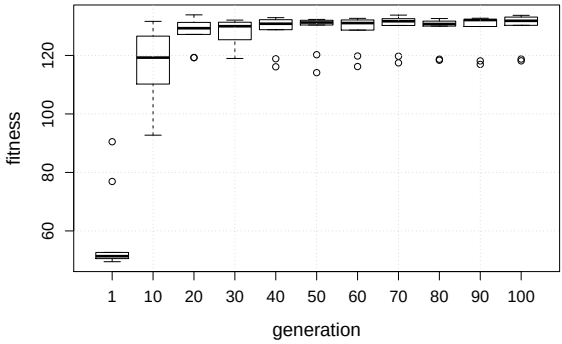
$$\begin{aligned} D(\mathbf{x}_{\text{target}}) =& |x_{\text{target}} - x_t| + |y_{\text{target}} - y_t|, \\ E =& \sum_{t=2}^{120} D(3,8) + \sum_{t=121}^{220} D(-5,11) + \sum_{t=221}^{300} D(-1,18), \\ F =& F_{\max} - E, \end{aligned} \tag{2}$$

for an assumed maximal fitness $F_{\max} = 3000$. For each target it considers a defined time interval during which it accumulates the differences between the respective target's position $\mathbf{x}_{target}$ and the current plant tip position $\mathbf{x}_t$. The first phase lasts for 120 time steps (time interval $[0, 120]$), the second for 100 time steps (time interval $[120, 220]$), and the third phase for 80 time steps (time interval $[220, 300]$).

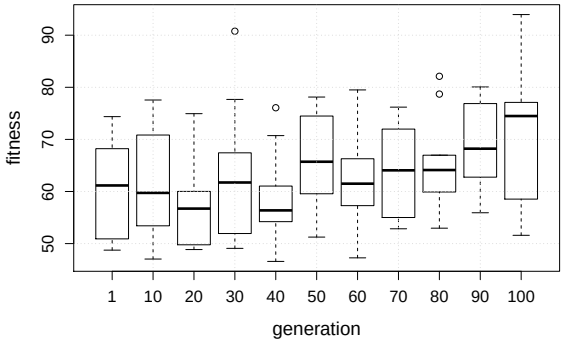
Here, for both the *deterministic* and *mixed* experiments, 10 evolutionary runs of 600 generations each with a population size of 50 were performed. Similarly, we also conducted the stochastic experiment, however, for only 150 generations (due to technical issues) but still with three repetitions and the minimum of these three values as fitness. In contrast to the maximum motion task, this task is more complex and requires longer evolutionary runs in order to evolve successful behaviors. The best fitness of each run over generations for the *deterministic* method in the *three targets* experiment is shown in Fig. 8(a). In contrast to the results from the *maximum motion* experiment, convergence is not achieved. Instead, there is a steady increase in performance through the 600 generations. This proves that the task here is more complex and indicates the presence of a room for improvement. The behavior of the simulated plant tip when



(a) deterministic

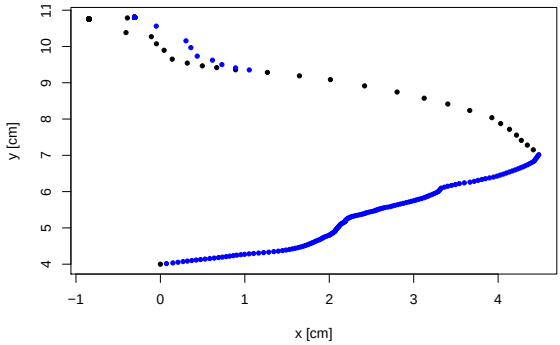


(b) mixed

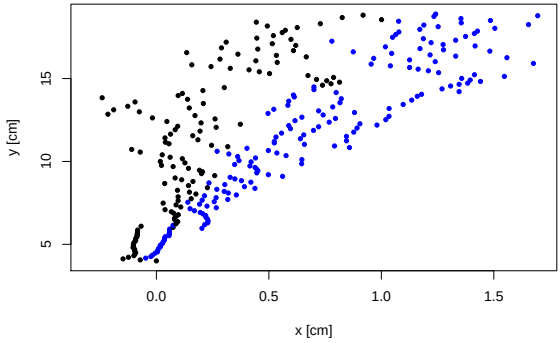


(c) stochastic

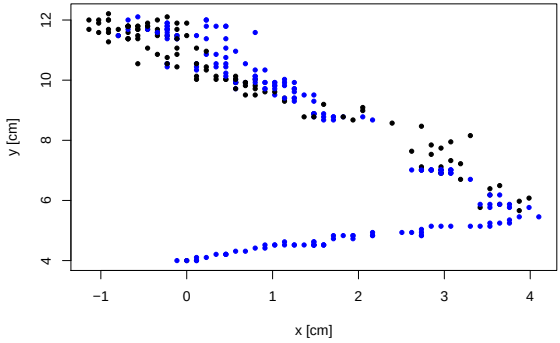
Figure 6: Maximum motion, fitness of the best controllers per generation of the 10 evolutionary runs for the maximum motion experiments.



(a) deterministic



(b) mixed



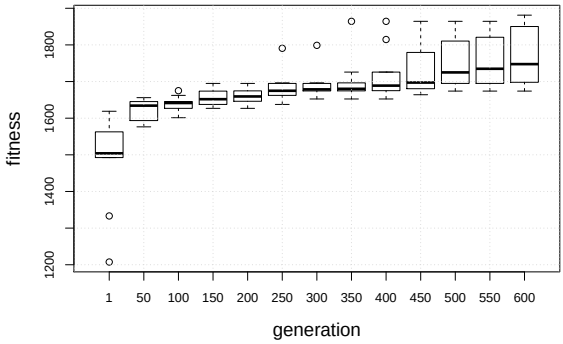
(c) stochastic

Figure 7: Maximum motion, (sampled) trajectories of the simulated plant tip (blue dot: right light is on, black dot: left light is on) for the respective best controller.

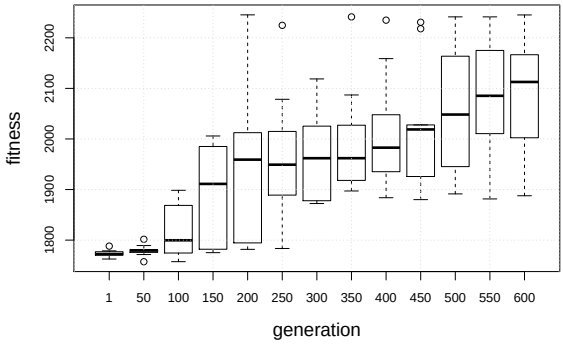
running the best controller is shown in Fig. 9(a). The controller turns the right light on in the beginning until the plant tip is close to the first target, then switches the left light on until the plant tip is close to the left target, then keeps on switching between the lights and fails to meet the third target. The fitness value of this controller is $F = 1881.2$. The best fitness of each run over generations for the *mixed* method in the *three targets* experiment is shown in Fig. 8(b). Similarly, no saturation is observed, however, the trajectory of the plant tip when running the best controller with a fitness value $F = 2255.9$ (see Fig. 9(b)) shows that the tip could successfully approach all three targets. The best fitness over generations for the *stochastic* method in the *three targets* experiment is shown in Fig. 8(c). As above, we have checked for significance between the initial and final population. According to the Wilcoxon Rank-Sum Test, the performance of controllers in the last generation are not significantly better than the initial population. However, according to our observations (see Fig. 9(c)), there is an improvement in the performance and longer evolutionary runs are planned for future work. The simulated behavior of the tip when running the best controller of fitness value $F = 1843.65$ is shown in Fig. 9(c).

Conclusion and future work

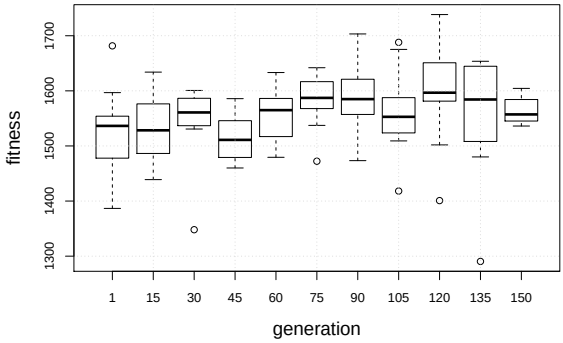
We have reported on experiments with a natural plant, the common bean (*Phaseolus vulgaris*). We have introduced a simple plant growth and motion model which focuses on the plant tip exclusively. Then we used this model to simulate the plant tip motion in several preliminary experiments. This way we evolve closed-loop controllers that maximize the plant's motion or grow the tip to three differently located targets. According to the discussion in the previous section, the results from the six experiments indicate the effectiveness of our evolutionary approach (arguably except for the experiments with the *stochastic* method). Also, in many cases the optimization process did not fully converge. Running the evolutionary process for a longer time would probably increase the performance of the evolved controllers. Therefore, we intend to investigate longer evolutionary runs in future work, probably along with a more efficient implementation. We are also motivated to investigate the impact of different parameters of



(a) deterministic

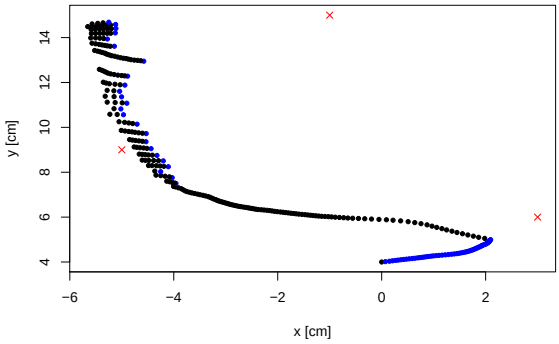


(b) mixed

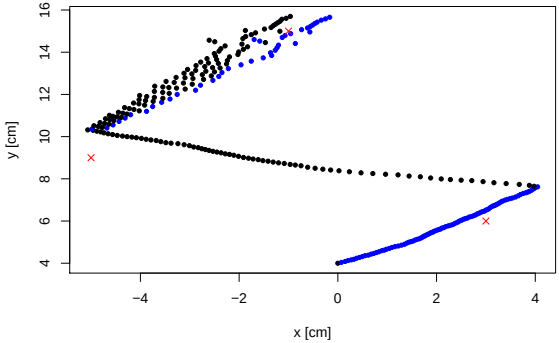


(c) stochastic

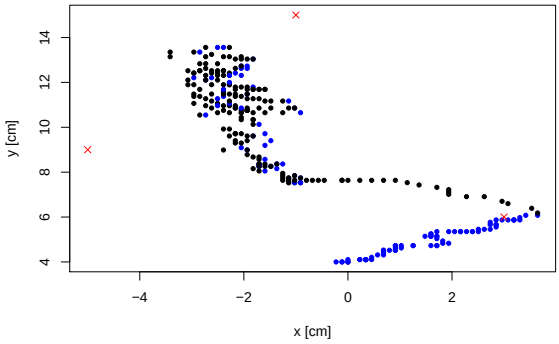
Figure 8: Three targets experiments, fitness of the best controllers per generation of the 10 evolutionary runs for the three targets experiments.



(a) deterministic



(b) mixed



(c) stochastic

Figure 9: Three targets experiments, (sampled) trajectories of the simulated plant tip (blue dot: right light is on, black dot: left light is on) for the best controllers. Red crosses indicate the positions of the three targets.

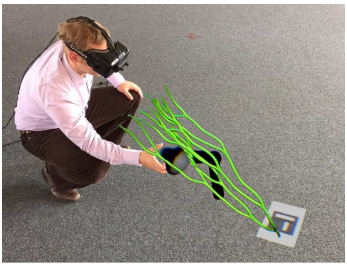


Figure 10: This photomontage illustrates the workings of our current augmented reality interface for bio-hybrid plant-robot systems.

the evolutionary algorithm (e.g., the mutation rate) in order to decrease the time required to evolve successful behaviors.

Especially the results from the mixed, *three targets* experiment are promising (see Fig. 9(c)). We plan to test these controllers, that were evolved in simulation, in the real setting and hence investigate the reality gap and the reaction of real plants. As the controllers will not perfectly transfer into reality, we expect to require additional investigations. We plan to extend the existing model of the plant tip to consider the whole plant growth and motion behavior, that is, also to model lower parts of the plant.

In addition to honing the domain model and the developmental and dynamic plant representation, we are also pushing towards the application of *flora robotica*'s results (e.g., applications in architecture). In terms of accessibility, we found it especially useful to spatially experience a bio-hybrid plant-robot system. To this end, we are developing an augmented reality interface that allows the user to project the system into arbitrary real world spaces, configure and fast forward the system's evolution. Fig. 10 shows our current prototype: (1) A stereoscopic camera (OVRVision) is mounted on a stereoscopic head-mounted display (Oculus DK2). (2) QR code markers on the ground synchronize the augmented reality projection with the lab space. (3) The user can place robotic machinery, seed plants and experience their interplay over time.

Finally, we will integrate these approaches within our project *flora robotica* to control the growth and motion of natural plants by robots and to create an adaptive bio-hybrid system.

Acknowledgment

Project '*flora robotica*' has received funding from the European Union's Horizon 2020 research and innovation program under the FET grant agreement, no. 640959.

References

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

[2] P. Chervenski and S. Ryan. MultiNEAT, project website;. `http://www.multineat.com/`.

[3] N. Correll, N. Arechiga, A. Bolger, M. Bollini, B. Charrow, A. Clayton, F. Dominguez, K. Donahue, S. Dyar, L. Johnson, et al. Indoor robot gardening: design and implementation. *Intelligent Service Robotics*, 3(4):219–232, 2010.

[4] A. E. Eiben and J. Smith. From evolutionary computation to the evolution of things. *Nature*, 521:476–482, May 2015.

[5] R. Featherstone and D. Orin. Robot dynamics: Equations and algorithms. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 1, pages 826–834. IEEE, 2000.

[6] *flora robotica*. project website, 2015. http://www.florarobotica.eu.

[7] P. C. Garzón and F. Keijzer. Plants: Adaptive behavior, root-brains, and minimal cognition. *Adaptive Behavior*, 19(3):155–171, 2011.

[8] H. Hamann, M. Wahby, T. Schmickl, P. Zahadat, D. Hofstadler, K. Stoy, S. Risi, A. Faina, F. Veenstra, S. Kernbach, I. Kuksin, O. Kernbach, P. Ayres, and P. Wojtaszek. *flora robotica* – mixed societies of symbiotic robot-plant bio-hybrids. *Proceedings of the 2015 IEEE Symposium on Artificial Life (IEEE ALIFE'15)*, 2015. (in press).

[9] S. Koos, J.-B. Mouret, and S. Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17(1):122–145, 2013.

[10] A. Lindenmayer. Developmental algorithms for multicellular organisms: A survey of L-systems. *Journal of Theoretical Biology*, 54(1):3–22, 1975.

[11] D. Slaughter, D. Giles, and D. Downey. Autonomous robotic weed control systems: A review. *Computers and electronics in agriculture*, 61(1):63–78, 2008.

[12] K. O. Stanley and R. Miikkulainen. Competitive coevolution through evolutionary complexification. *Journal of Artificial Intelligence Research*, 21(1):63–100, Jan. 2004.

[13] O. Stava, S. Pirk, J. Kratt, B. Chen, R. Měch, O. Deussen, and B. Benes. Inverse procedural modelling of trees. In *Computer Graphics Forum*, volume 33, pages 118–131. Wiley Online Library, 2014.

[14] D. W. Thompson. *On Growth and Form*. Cambridge University Press, 1917.

[15] S. Von Mammen and C. Jacob. The evolution of swarm grammars-growing trees, crafting art, and bottom-up design. *IEEE Computational Intelligence Magazine*, 4(3):10–19, 2009.

[16] M. Wahby and H. Hamann. On the tradeoff between hardware protection and optimization success: A case study in onboard evolutionary robotics for autonomous parallel parking. In *Applications of Evolutionary Computation (EvoApplications 2015)*, volume 9028 of *LNCS*, pages 759–770. Springer, 2015.

[17] R. A. Watson, S. G. Ficici, and J. B. Pollack. Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39(1):1–18, 2002.

[18] M. Woolridge. *Introduction to Multiagent Systems*. John Wiley & Sons, Inc. New York, NY, USA, 2001.

[19] A. Zamuda and J. Brest. Vectorized procedural models for animated trees reconstruction using differential evolution. *Information Sciences*, 278:1–21, 2014.