Evolution, Individual Learning, and Social Learning in a Swarm of Real Robots

Jacqueline Heinerman, Massimiliano Rango, A.E. Eiben
VU University Amsterdam, The Netherlands
Email: j.v.heinerman@vu.nl, massimiliano.rango@gmail.com, a.e.eiben@vu.nl

***Abstract*—We investigate a novel adaptive system based on evolution, individual learning, and social learning in a swarm of physical Thymio II robots. The system is based on distinguishing inheritable and learnable features in the robots and defining appropriate operators for both categories. In this study we choose to make the sensory layout of the robots inheritable, thus evolvable, and the robot controllers learnable. We run tests with a basic system that employs only evolution and individual learning and compare this with an extended system where robots can disseminate their learned controllers. Results show that social learning increases the learning speed and leads to better controllers.**

***Index Terms*—Evolutionary robotics; swarm robotics; embodied evolution; morphology evolution; individual learning; social learning**

I. Introduction

The field of Evolutionary Robotics (ER) is concerned with designing and optimizing robots by using evolutionary computing techniques [1], [2], [3], [4], [5]. In principle, evolutionary algorithms can be applied to develop the robot controller, the robot morphology, or both, but the huge majority of work in ER is concerned with evolving controllers only. The usual approach is based on simulations and follows the off-line approach. That is, the evolutionary algorithm (EA) is employed during the design stage and the evolved solution (i.e., the fittest controller found by the EA) will not change after deployment during the operational stage of the robot. This approach has two drawbacks:

- The reality gap, that is, the effect that solutions developed in simulations do not work well on the real hardware [6].
- The lack of adaptivity, that is, the fact the controller can not adjust to changing or unforeseen circumstances.

In this paper we go beyond the usual ER approach in several aspects. First, we run evolution on the real hardware, in a group of Thymio II robots [7], thus eliminating the reality gap. Second, we follow the on-line approach, which means that evolution takes place during the operational stage of the robots adapting their features on the fly. Third, we are not only evolving controllers, but morphological features as well.

The key innovation behind the system we investigate here is the 'adaptation engine' that integrates evolution, individual learning, and social learning [8], [9]. The distinction between evolution on the one hand and lifetime learning on the other hand is based on distinguishing two types of adaptable robot features: inheritable features (genome) and learnable features (memome). Inheritable features do not change during the lifetime of an individual, only from generation to generation by the *evolutionary operators* mutation and crossover. In contrast, learnable features do change during lifetime by the *learning operators*. Furthermore we distinguish two types of lifetime learning. Individual learning takes place within a single individual that changes some of its learnable features based on its own experience. Social learning requires more individuals as it amounts to changing one's learnable features based on somebody else's experience.

The main goal of this paper is to investigate an integrated three-tier adaptation engine in a swarm of physical robots. To this end we choose to make the sensory layout of the robots inheritable and the robot controllers learnable. This means that genomes encode morphological properties; robots with a different genome have a different set of active sensors. The task of lifetime learning is to obtain appropriate controllers that exploit the available sensory information and generate adequate behavior in the given environment. Here we use neural networks as controllers and an obstacle avoidance task.

Our specific research questions are related to the effects of social learning under different conditions. In a swarm where each robot has the same sensory layout, we expect that social learning increases the speed of learning and possibly results in better controllers. When the robots have different sensory layouts this is not so obvious because the learned controllers are, in principle, specific to the available sensory information. Our experiments will seek answers to the following questions:

1) What is the effect of social learning on the quality and speed of learning if the robots in the swarm have identical sensory layout (no evolution)?
2) What is the effect of social learning on the quality and speed of learning in a swarm where the sensory layout is evolving?
3) What is the effect of social learning on the evolved sensory layouts?

II. Related Work

Forced by space limitations we restrict our review of related work to studies regarding on-line evolution in swarms of physical robots. The pioneering works in this area are those of Watson *et al.* and Simoes *et al.* published in the same year. Watson *et al.* introduced "embodied evolution" in a swarm of six physical robots [10]. The robots evolved their controllers for a phototaxis task in an on-line fashion by broadcasting

values from their genome at a rate proportional to their fitness. Simoes *et al.* evolved both morphological features and the controllers in the same genome for collision-free behavior [11]. The main difference with the work presented in [10] is the guarantee that the fittest robot will survive to the next generation, implemented in a centralised way. Since then only three more papers were published showing on-line evolution in a group of physical robots. The authors of [12] want to speed up embodied evolution in a population of two robots, by adding the fitness of the controller to the broadcasted genome. They further distinguish their work by having multiple tasks: phototaxis, obstacle avoidance, and robot seeking. The essential point in [13] is that that the progress of evolution directly depends on the number of robots and the frequency of encounters. That is why they propose and island model for an obstacle avoidance task for a swarm of six robots. Both studies discussed above show an increasing fitness value for the learned task over time where the last one also shows the positive impact of communication between the islands. Last but not least, a recent paper investigates a very different evolutionary system that does not have an objective function designed by the user. The evolutionary algorithm is driven by environmental selection, robots with inferior controllers will have fewer offspring than those with a good strategy [14].

All these papers investigate a single-tier adaptive system, where evolution is the only adaptive force. Our work presented here is different because we have a three-tier adaptive system with evolution in the genome space and individual and social learning in the memome space. This is not a new concept in simulation [8], [15], [16], [17], [18] but to our best knowledge it has not been implemented in real hardware before.

III. System Description

A. Robot

The Thymio II robot includes 7 Infra-Red (IR) proximity sensors able detect to obstacles of which are 5 in the front and two in the back (values between `0` and around `4500`, where a higher value corresponds to a near obstacle). The robot can move through two differential wheels, meaning that two different speeds (range between `−500` and `500`) can be set for each wheel. For the purpose of our research, we extend the standard setup with a more powerful logic board, wireless communication, and a high capacity battery. We use a Raspberry Pi B+ (credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation) that interacts with the Thymio sensors and actuators. A WiFi dongle (Edimax 150Mbps Wireless 802.11b/g/n nano USB WiFi Adapter model EW-7811Un) attached to the Raspberry Pi ensures communication between the robots. The power is given by a Verbatim Dual USB 12000 mAh battery that allows for a total experimental time of 10 hours. The extended Thymio is shown in Figure 1.

B. Environment

The robots operate in an arena of two meters by two and a half meters with inner and outer walls that act as obstacles to avoid, cf. Figure 2. Next to this arena a WiFi router is placed, facilitating the exchange of wireless TCP messages between the robots. When a Raspberry Pi is powered on, the algorithm starts and listens for a TCP command in order to begin the experiment. This command is sent from a computer to all the robots at the same time whereafter the computer plays no active role in the experiment, it is merely collecting the data.

Figure 1: Thymio II robot, developed by The École Polytechnique Fédérale de Lausanne (EPFL) and École Cantonal d'Arts de Lausanne (ÉCAL), with Raspberry Pi B+, WiFi dongle, and external battery.

Figure 2: The environment used for the experiments.

C. Task

The task to learn is obstacle avoidance. The obstacles in the arena are represented by the inner and outer walls and the robots have to learn how avoid them and each other. The robots' performance over an evaluation period of T time steps measured by the usual formula:

$$f = \sum_{t=0}^{T} s_{trans} \times (1 - s_{rot}) \times (1 - v_{sens}),$$

where:

- s_{trans} is the translational speed (not normalized), calculated as the sum of the speeds assigned to the left and right motor;
- s_{rot} is the rotational speed, calculated as the absolute difference between the speed values assigned to the two motors and normalized between 0 and 1;
- v_{sens} is the value of the proximity sensor closest to an obstacle normalized between 0 and 1.

Using v_{sens} in f may suggest that only the active sensors in an individual's genome are included in this part. This is not the case because it can result in undesired behaviors getting high fitnesses. Suppose that all front sensors are excluded from the genome and a robot is driving with both wheels at full speed against a wall, the fitness function will result in the highest possible value. In this case s_{trans} has a value of 500, s_{rot} of 0 and v_{sens} only includes the back sensors that don't see an obstacle and therefore having a value of 0.

D. Inheritable robot features

The inheritable part of the robot makeup –represented by the genome– does not change by learning. In this study we define the genome as the active proximity sensors of a robot. An active sensor means that the robot can use the value of this proximity sensor. If a sensor is not active then its values are not available to the controller. The Thymio II robot is equipped with 7 IR proximity sensors resulting in a genome array g of length 7, where $g_i = 1$ when sensor i is active and $g_i = 0$ if it is not active ($i \in \{1, ..., 7\}$).

E. Learnable robot features

The controller of an individual is a feed forward neural network (NN) with the input nodes corresponding to the active sensors of the robot, one bias node, and two output nodes that represent the motor speed values. Each input node and the bias node is directly connected to each output node, resulting in a neural network with a maximum of 16 weights. This is implemented as an array where the weights 1 through 8 correspond to the left motor (weight 8 belongs to the bias node) and weights 9 through 16 to the right motor (weight 16 belongs to the bias node). The motor speeds are calculated every timestep the following way:

$$m_{left} = m_{max} \times \tanh \sum_{i \in \{1,...,7\} \text{ where } g_i=1, i=8} w_i \times s_i,$$

and

$$m_{right} = m_{max} \times \tanh \sum_{i \in \{1,...,7\} \text{ where } g_i=1, i=8} w_{i+8} \times s_i,$$

where:

- m_{max} is the maximum speed of the motor;
- w_j is weight j of the neural network;

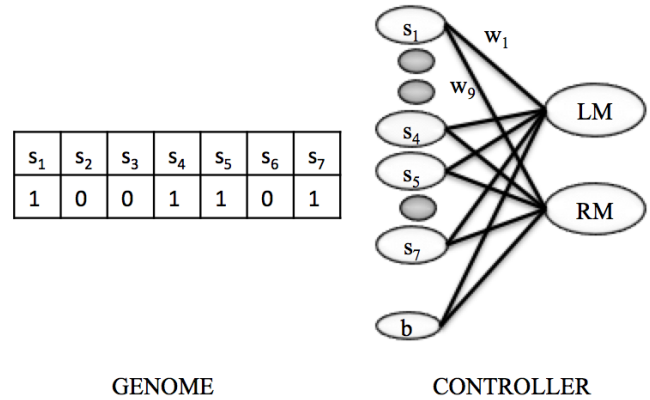

Figure 3: Relation between genomes and memomes. A genome that specifies the active sensors also defines the structure of the NN controller. The memome then consists of all weights for the given NN.

- s_i is the value of the proximity sensor i normalized between -1 and 1;
- tanh is a hyperbolic tangent activation function.

In this system we postulate that the weights of the NN controller are the learnable robot features. Hence, the memome is a vector of maximal 16 values. Note that the structure of the NN is an inheritable property as it is fully determined by the actual set of active sensors. The relation between the genome and the memome is shown in Figure 3.

F. Adaptive mechanisms

To prevent confusion in the following we need to define a specific terminology for robotic systems with evolution, learning, and social learning as studied here. We use the term *robot* to designate the physical device, in our case a Thymio II. An *individual* is a unit of selection from the evolutionary perspective. In the current system an individual is a given sensory layout of the robots we use. Individuals are represented by genomes as explained in Section III-D. Consequently, the same physical robot will be a different individual after each evolutionary step that creates a new genome for it. Given a robot and a genome the appropriate *controllers* are determined. In this paper the structure of the NN controller is specified by the genome, whereas the weights are not. These weights form the learnable features that are changed by applying the learning operators. Hence, the same robot and the same individual will have a new controller (defined by the newly learned weights) after every learning step. This is shown in Figure 4.

The general idea behind the three-fold adaptation mechanism is as follows. Each individual has a maximum lifetime. We define one year in an individual's lifetime by one controller evaluation as explained in III-C. After a controller evaluation we make a choice between performing a learning step (individual learning or social learning) or reevaluation. Reevaluation is necessary because of the noisy fitness function [19]. The

Physical	Robot_A			
Evolvable	Individual_1	Individual_2	…	Individual_N
Learnable	c_1 c_2 … c_k			

Figure 4: A physical robot is used by multiple individuals sequentially. During the lifetime of an individual a number of controllers is tested. At the end of its lifetime an individual is replaced by its offspring and the system starts to learn a new type of controller that fits the new sensory layout. We use 6 physical robots and $N = 8$ generations and $k = 100$ controller evaluations in the evolutionary experiments.

fitness value obtained by reevaluation is used to create a new fitness value for the current controller in combination with the old one with a 20-80 weight distribution (20% for the new, 80% for the old fitness value). If we choose for performing a learning step, then with a 70% probability individual learning is performed and with 30% we apply a social learning step. The age of a robot is increased by one after every new controller evaluation until the maximum lifetime is reached. Then a new genome, hence a new individual, is created by evolutionary operators.

Evolution Genomes are broadcasted after every memome evaluation together with their fitness value.[1] Individuals collect these genomes together with the corresponding fitness value in their *genome storage* with limited capacity. Only unique genomes are stored here. When the genome is already in storage, the fitness value will be replaced by the last obtained fitness value. When an individual's lifetime expires it picks a new genome through tournament selection from the genome storage. Uniform crossover and mutation are performed on the genome of the tournament winner and the current genome of the robot. When a new genome is established, the genome storage is cleaned and a new memome is created where the weights of the neural network are uniform random initialized.

Individual Learning The method for individual learning can be any algorithm that can optimize neural networks efficiently. In our system, it is a (1+1) Evolutionary Strategy based on [20]. The fitness function for this ES is the f defined in section III-C. The algorithm works on the weights of the neural network that are mutated with a Gaussian noise $N(0, \sigma)$ whose σ value is doubled when the mutated memome is not better than the current memome. Before a memome is evaluated a recovery period is introduced. During this period, the individual can move, but the fitness is not being measured, so that the individual is able to recover from a difficult starting position. When a altered memome has a higher fitness than the current, the weights of the neural network are replaced resulting in a new current memome.

Social Learning Memomes are broadcasted after every evaluation, provided that a minimum fitness threshold is exceeded. We have implemented a *no*, *medium* and a *high* value for this, in particular, 0%, 30% or 75% of the theoretical maximum of the fitness function[2]. The place where memomes from other robots are collected is called the *memome storage*. A memome is taken from the storage in a *Last In First Out (LIFO)* order and combined with the current robots memome to create an altered memome. To perform recombination, the weights of the current memome are copied into the altered memome. Thereafter the altered weights, now equal to the current weights, are overridden by those of the collected memome if these weights are applicable to the current genome (i.e. the corresponding sensory layout). After evaluation of the altered memome, it is either discarded when the fitness is lower than the current memome or promoted to the current memome.

This system is illustrated in Figure 5 that shows the genomes, the memomes, the evolutionary operators and learning operators with respect to one robot.

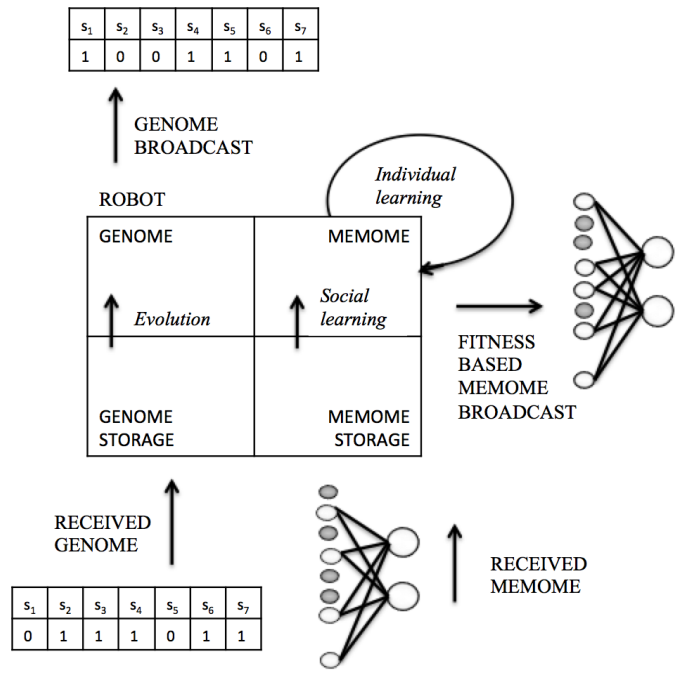

Figure 5: Overview of the three-fold adaptive engine. During the lifetime of an individual, genomes and memomes are broadcasted and stored in the corresponding storage locations. New memomes are created through social learning and individual learning. When the individual/genome reaches maximum age, a new genome is created using the current genome and a genome selected from the storage.

[1]Because of using WiFi here all robots will receive this broadcast. Using Bluetooth and a large arena recipients would be limited to those robots that are within communication range.

[2]The maximum is calculated by assuming a robot moving in a perfectly straight line with no obstacles in sight for the full evaluation period. Let us note that the practically obtainable fitness values are around 90% of this maximum.

IV. Experimental Setup

We use two different setups[3] for answering the research questions listed in the Introduction:

1) The first setup is without evolution where six robots with all sensors active undergo a lifetime learning process for a lifetime of 200 years.
2) The second setup is an evolutionary experiment of eight generations where the six individuals have a lifetime of 100 before being replaced by an offspring.

The learning only experiment takes one hour and the evolutionary experiment takes four hours. Although the external battery can ensure a longer experiment time, the robot is not suited for long experiments as the controller board gets overheated and leads to breaking robots[4].

When an experiment is completed, the robots start at the position they stopped during the previous experiment, in order to increase variability between the runs and to exclude human bias for the starting position of the robots. Human intervention has been proven necessary when (1) the robot gets stuck in the smallest part of the arena (2) the robot's wires get stuck into each other and (3) when the lego component falls off.

For each setup, we compare individual learning only and individual and social learning together. Furthermore, we vary the threshold value that regulates the quality pressure in the social learning mechanism; we experiment with three variants: no threshold, medium threshold or a high threshold. For every setup and threshold value, we do 10 repetitions with different random seeds. The list of all relevant system parameters and the values used is given below.

System parameters	
Max. evaluations	**200,800**
Maximum number of evaluations in a run.	
Max. lifetime	**100,200**
Maximum lifetime of an individual.	
Evaluation duration	**10.25**
The duration of one evaluation measured in seconds (recovery time of 1.5 sec and actual evaluation of 8.75 sec).	
Reevaluation rate	**0.2**
Chance that the current memome is reevaluated.	
Social learning rate	**0.3**
Chance that an altered memome is created by social learning.	
Evolution	
Inactive chance	**0.3**
Chance for each sensor to be inactive initialization.	
Tournament size	**2**
Size of the tournament that is held among the collected genomes.	
Mutation rate	**0.05**
Change to flip a bit in a genome.	
Genome storage size	**5**
Maximum number of uniquely collected genomes.	
Learning	
Weight range	**4**
Value of NN weights are between [-4, 4].	
Sigma initial	**1**
Initial sigma value for mutating weights.	
Sigma maximum	**4**
Maximum sigma value.	
Sigma minimum	**0.01**
Minimal sigma value.	
Memome storage size	**20**
Maximum number of memomes to store.	
Fitness	
Reevaluation weight	**0.8**
Weight of current memome fitness in reevaluation.	
Maximum fitness	**105000**
Theoretical maximum fitness value.	
Memome broadcast threshold	**0,30,75%**
Percentage of maximum fitness to exceed before sending memome.	

[3]The code for implementation is available on the web site of the first author.

[4]During the experiments several robots broke. These results will be excluded from the graphs unless mentioned otherwise.

V. Experimental Results

The experimental data for the first research question was collected under the first setup without evolution. Figure 6 shows the results. From the increase in fitness values we can conclude that the individuals are able to learn the task. Without social learning, individuals reach an average of around 65% of the maximum possible fitness value. With social learning, the reached fitness values are between 75-80%.

Figure 7 shows a re-plot of the data including 90% confidence intervals for individual learning alone and social learning with a threshold of 75%. We see non-overlapping confidence intervals, meaning that the impact of social learning is significant with the P value much less than 0.05 [21].

Similar results are obtained when comparing individual learning alone and social learning with no threshold. This is good news, indicating that the social learning method works even in cases where there is not enough information about optimal fitness values to establish a reasonable threshold.

To answer the second research question we conducted the evolutionary experiments under the second setup. Figure 8 shows the outcomes for four cases: 1) evolution and individual

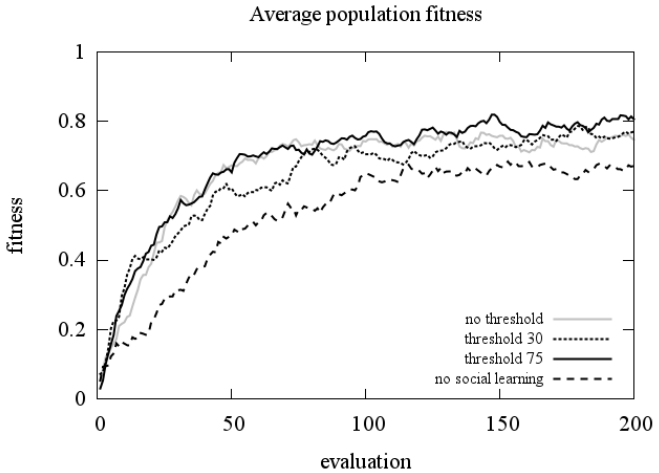

Figure 6: Lifetime learning without evolution in a group of 6 robots with identical sensory layouts. Time is measured by the number of evaluations (x-axis), fitness by the formula in Section III (y-axis) divided by the maximum fitness.

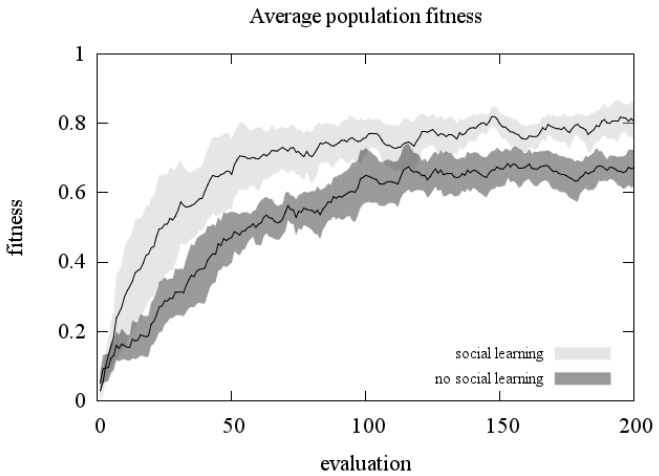

Figure 7: Lifetime learning without evolution in a group of 6 robots with identical sensory layouts. Curves show the same data as in Figure 6 for two of the four methods: individual learning alone and social learning with a threshold of 75%. The 90% confidence intervals are included over 10 repetitions using a t-distribution with 9 degrees of freedom.

learning without social learning and 2) - 4) evolution and individual learning combined with the three variants of the social learning method. These plots clearly show that social learning improves performance even if the sensory layouts, hence the number of input nodes in the NN controllers, vary over the members of the population. This result is consistent with the ones obtained in simulations using swarms of e-puck robots [9].

To get a better picture showing the impact of social learning we re-plot some of the data from Figure 8. In Figure 9 we exhibit the fitness values at the end of every generation for two algorithm variants: the one with evolution and individual learning alone and the one with evolution and individual learning with social learning using the 75% threshold. These graphs confirm the conclusions based on Figure 8.

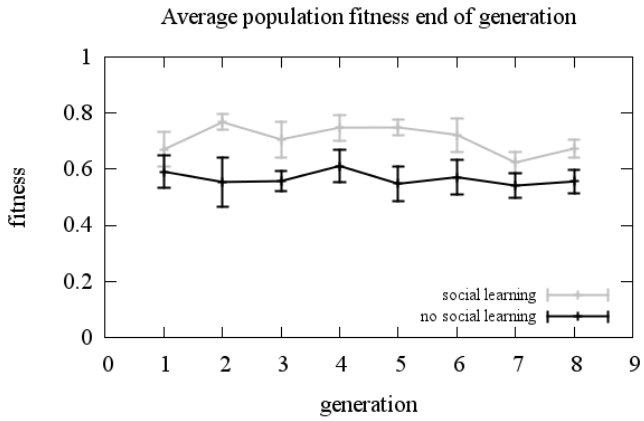

Figure 9: Average population fitness at the end of a generation. For social learning, the high threshold of 75% is used. The 90% confidence intervals are included over 10 repetitions using a t-distribution with 9 degrees of freedom

The answer to the third research question can be obtained from Figure 10. The top row shows the total number of active sensors at the end of each generation. These plots show that total number of active sensors decreases over time for all algorithm variants. This matches our intuition that expects that the "useless" sensors will be unlearned over time.

The second row in Figure 10 shows the *level of agreement* between the individuals of a given population. The level of agreement is the total number of sensors that have the same status (active or inactive) as the other individuals. For example, when the level of agreement in a certain generation is 7 this means that for all individuals the sensory layout is identical, regardless of which sensors are active or inactive.

The level of agreement can give information about the consensus regarding the best genome in the population, i.e., in the group of six Thymio robots. The faster they have a consensus, the more similar the genome and thus the neural network structure. A similar neural network structure results in more valuable information in the social learning step because less integration is necessary.

The second row of graphs in Figure 10 we can conclude[5] shows that the level of agreement is growing more rapidly with social learning than with individual learning alone. Oddly enough, social learning with the 75% threshold value breaks this pattern, and at this moment we have no explanation for this. For the other threshold values, a significant increase can

[5]Excluding data when a robot broke produces similar results. For the consistency in the confidence interval sizes we decided to include all runs.

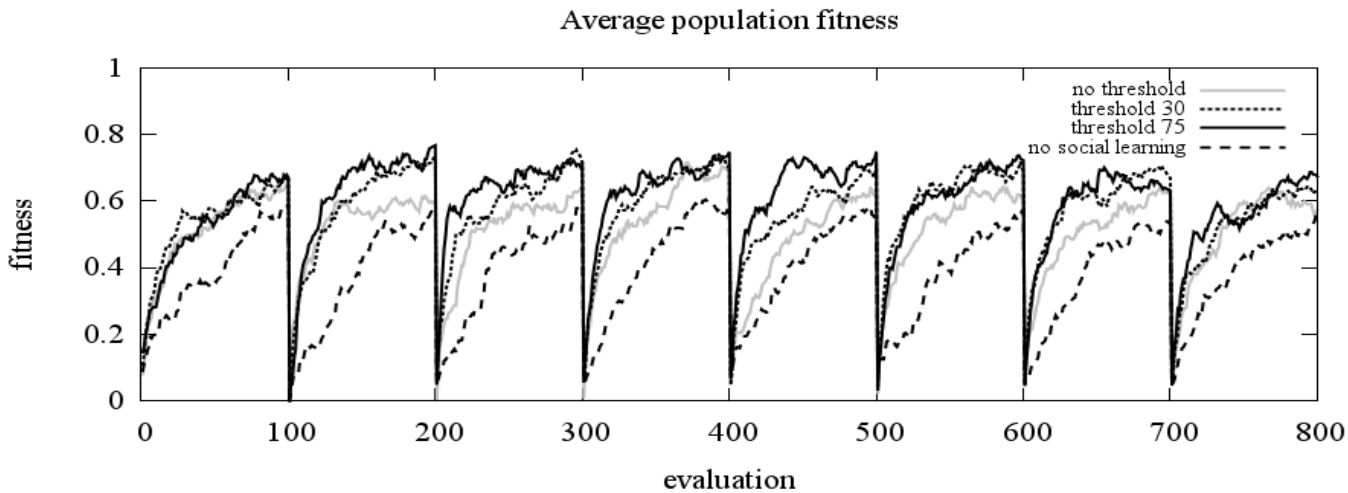

Figure 8: Evolution, individual learning, and social learning in a group of 6 robots. Time is measured by the number of evaluations (x-axis), fitness by the formula in Section III (y-axis) divided by the maximum fitness. After 100 evaluations, an evolutionary step takes place resulting in new individuals.

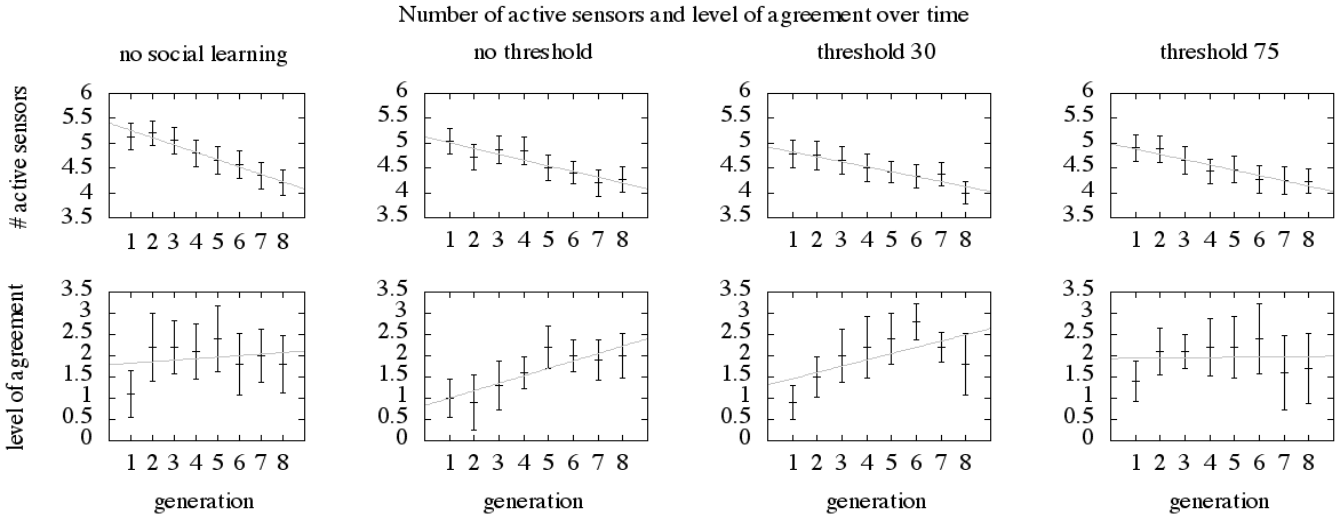

Figure 10: Top row: total number of active sensors over time including 90% confidence interval using a t-distribution with 59 degrees of freedom and a least-squares regression fit. These results are conducted over 60 observations (6 individuals per generation and 10 runs). Second row: level of agreement, number of sensors all active or all deactivate in a generation, over time. These results are conducted over 10 runs including 90% a confidence interval using a t-distribution with 9 degrees of freedom and a least-squares regression fit.

be obtained, at least for the first six generations. This means that social learning does influence evolution in the genome space. This is explainable when we think of the role of social learning. Social learning gives individuals the option to explore a different behavior space from the individual hill climber. This results in more information about the capabilities of a certain genome. Because the new genome is picked through a tournament and therefore based on these capabilities, evolution in the genome space will converge quicker to the best genomes in the population.

Convergence of the genome has the advantage of less needed integration of the memome obtained through social learning. This leads to more valuable information in the case of a high threshold value and, therefore, higher fitness values. For this reason, we hoped to see an increase in the fitness values obtained during the generations in Figure 9. We can see that this is not the case, but we can see that when the level of agreement is decreasing for the high threshold, the fitness values are decreasing as well.

This level of agreement seems to decrease because of the fast

consensus on the genome layout and the mutation probability of 5%. It would be interesting to decrease the mutation probability according to the level of agreement to see whether the level of agreement stays higher and the fitness value too.

VI. Conclusions and Further Work

In this paper, we presented and investigated a three-fold adaptive mechanism based on evolution, individual, and social learning to implement on the fly adaptation in a swarm of physical Thymio II robots. The conceptual framework is generic, separating inheritable controller features (genome) from the learnable controller features (memome) and specifying adequate evolutionary and learning operators for the adaptive processes.

A special feature in our implementation is that the genome encodes morphological properties of the robots, albeit a simple version, defining the set of active sensors. Hereby the genome (the inheritable material) partly determines the memome (the learnable material) that corresponds to the weights of the NN that controls the robot.

The experiments show a significant benefit of social learning: it makes the population learn faster and the quality of learned controllers higher. This effect is demonstrated under two different setups: for robots with identical genomes and for robots with evolving genomes. Furthermore, we have seen an indication that social learning has a guiding effect on the genome evolution by showing a significant effect in the level of agreement in the population, i.e. the number of sensors that have the same state for all individuals. For this reason, we think that social learning, combined with individual learning, results in better exploration of the memome possibilities. The memome fitness, that now better represents the memome quality, results in increasing selection pressure on the best genomes.

Due to restrictions on the total time for executing an experiment (overheating) only eight generations are investigated. Longer runs may be possible by pausing experiments regularly to cool down the robots.

There are still many interesting questions to investigate including (1) variable lifetimes, resulting in overlapping generations where the younger individuals can learn from the older ones (2) increasing the task difficulty, or having multiple tasks and (3) changing the genome/memome setup to explore the generality of the results.

VII. Acknowledgments

This work was made possible by the European Union FET Proactive Initiative: Knowing, Doing, Being: Cognition Beyond Problem Solving funding the Deferred Restructuring of Experience in Autonomous Machines (DREAM) project under grant agreement 640891. The authors would like to thank Alessandro Zonta and Evert Haasdijk for their comments.

References

[1] J. Bongard, "Evolutionary robotics," *Communications of the ACM*, vol. 56, no. 8, pp. 74–85, 2013.

[2] D. Floreano, P. Husbands, and S. Nolfi, "Evolutionary robotics," in *Springer Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer, 2008, vol. Part G.61, pp. 1423–1451.

[3] S. Nolfi and D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MIT Press, Cambridge, MA, 2000.

[4] S. Doncieux, N. Bredeche, and J.-B. Mouret, Eds., *New Horizons in Evolutionary Robotics*, ser. Studies in Computational Intelligence. Springer, 2011, vol. 341.

[5] P. Vargas, E. D. Paolo, I. Harvey, and P. Husbands, Eds., *The Horizons of Evolutionary Robotics*. MIT Press, 2014.

[6] N. Jakobi, P. Husbands, and I. Harvey, "Noise and the reality gap: The use of simulation in evolutionary robotics," in *Advances in artificial life*. Springer, 1995, pp. 704–720.

[7] F. Riedo, M. S. D. Chevalier, S. Magnenat, and F. Mondada, "Thymio II, a robot that grows wiser with children," in *Advanced Robotics and its Social Impacts (ARSO), 2013 IEEE Workshop on*. IEEE, 2013, pp. 187 – 193.

[8] E. Haasdijk, A. E. Eiben, and A. F. Winfield, *Individual, Social and Evolutionary Adaptation in Collective Systems*. Singapore: Pan Stanford, 2013, ch. 12, pp. 413–471.

[9] J. Heinerman, D. Drupsteen, and A. E. Eiben, "Three-fold adaptivity in groups of robots: The effect of social learning," in *Proceedings of the 17th annual conference on Genetic and evolutionary computation*, ser. GECCO '15. ACM, 2015 (to appear).

[10] R. A. Watson, S. G. Ficici, and J. B. Pollack, "Embodied evolution: Embodying an evolutionary algorithm in a population of robots," in *1999 Congress on Evolutionary Computation (CEC'1999)*, vol. 1. IEEE Press, Piscataway, NJ, 1999, pp. 335–342.

[11] E. Simoes and K. R. Dimond, "An evolutionary controller for autonomous multi-robot systems," in *Systems, Man, and Cybernetics, 1999. IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on*, vol. 6. IEEE, 1999, pp. 596–601.

[12] U. Nehmzow, "Physically embedded genetic algorithm learning in multi-robot scenarios: The pega algorithm," in *Proceedings of The Second International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, ser. Lund University Cognitive Studies, C. Prince, Y. Demiris, Y. Marom, H. Kozima, and C. Balkenius, Eds., no. 94. Edinburgh, UK: LUCS, August 2002.

[13] Y. Usui and T. Arita, "Situated and embodied evolution in collective evolutionary robotics," in *Proceedings of the 8th International Symposium on Artificial Life and Robotics*, 2003, pp. 212–215.

[14] N. Bredeche, J.-M. Montanier, W. Liu, and A. F. Winfield, "Environment-driven distributed evolutionary adaptation in a population of autonomous robotic agents," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 18, no. 1, pp. 101–129, 2012.

[15] A. Acerbi and D. Parisi, "Cultural transmission between and within generations," *Journal of Artificial Societies and Social Simulation*, vol. 9, no. 1, 2006.

[16] W. Tansey, E. Feasley, and R. Miikkulainen, "Accelerating evolution via egalitarian social learning," in *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, ser. GECCO'12. ACM, 2012, pp. 919–926.

[17] D. Federici, "Combining genes and memes to speed up evolution," in *2003 Congress on Evolutionary Computation (CEC'2003)*, vol. 2. Edinburgh, UK: IEEE Press, Piscataway, NJ, 2003, pp. 747–754.

[18] C. Marriott and J. Chebib, "The effect of social learning on individual learning and evolution," *arXiv preprint arXiv:1406.2720*, 2014.

[19] H. G. Beyer, "Evolutionary algorithms in noisy environments: theoretical issues and guidelines for practice," *Computer methods in applied mechanics and engineering*, vol. 186, no. 2, pp. 239–267, 2000.

[20] N. Bredeche, E. Haasdijk, and A. E. Eiben, "On-line, on-board evolution of robot controllers," in *Proceedings of the 9th international conference on Artificial Evolution*, P. Collet, N. Monmarché, P. Legrand, M. Schoenauer, and E. Lutton, Eds. Berlin: Springer, 2009, pp. 110–121.

[21] M. E. Payton, M. H. Greenstone, and N. Schenker, "Overlapping confidence intervals or standard error intervals: what do they mean in terms of statistical significance?" *Journal of Insect Science*, vol. 3, no. 1, p. 34, 2003.