Evolution of Spiking Neural Networks Robust to Noise and Damage for Control of Simple Animats

Borys Wróbel[1,2(✉)]

[1] Systems Modeling Group, IO PAN, Sopot, Poland
[2] Evolutionary Systems Group,
Uniwersytet im. Adama Mickiewicza, Poznań, Poland
wrobel@evosys.org
http://www.evosys.org

Abstract. One of the central questions of biology is how complex biological systems can continue functioning in the presence of perturbations, damage, and mutational insults. This paper investigates evolution of spiking neural networks, consisting of adaptive exponential neurons. The networks are encoded in linear genomes in a manner inspired by genetic networks. The networks control a simple animat, with two sensors and two actuators, searching for targets in a simple environment. The results show that the presence of noise on the membrane voltage during evolution allows for evolution of efficient control and robustness to perturbations to the value of the neural parameters of neurons.

Keywords: Spiking neural networks · Adaptive exponential integrate-and-fire model · Genetic algorithm · Robustness to noise · Robustness to damage

1 Introduction

One of the central mysteries of biology is the enormous robustness of complex biological systems to perturbations [7]. This robustness is paradoxical because large complexity suggests fragility. And yet biological systems are robust not only to the fluctuations of the external environment, malfunctions of internal parts, but also the steady bombardment, over generations, of genetic disturbances (mutations) resulting in slight changes in structure of these systems. For example, biological genetic networks are robust to transcriptional noise, point mutations, deletions and duplications of genes. Perhaps the most complex systems known, biological neural networks, are robust to changes at several scales—developmental variability from one generation to the next, influencing the number of cells and theirs connectivity, fluctuations over individual life in the number of cells resulting from their death of cells and formation of new ones, and at a scale smaller still—destruction and formation of synapses, changes of the neurophysiological properties of individual neurons, etc.

© Springer International Publishing AG 2016
J. Handl et al. (Eds.): PPSN XIV 2016, LNCS 9921, pp. 686–696, 2016.
DOI: 10.1007/978-3-319-45823-6_64

As the components of artificial computational systems get smaller, these systems become more difficult to build (resulting in the variability of structure) and more unreliable (with more noise and more fragility of each part). Hence the interest in building artificial systems inspired by biology, such as artificial genetic networks and artificial neural networks, which promise large computational resources, low power consumption, and robustness to silicon mismatch (for example [6]).

In this paper I investigate the interplay between the robustness to noise and to other perturbations in evolved artificial spiking neural networks. The networks are evolved to control the behavior of a simple animat, with and without the noise on membrane potential of neurons. I then analyse the robustness of these networks to changes in parameters of neurons and the functioning of the animats' actuators.

The model of evolution of networks used in this paper was built originally for artificial genetic networks. We called this approach a 'mixed paradigm' [8,9], because the encoding in the artificial genomes is inspired by the encoding of biological genetic networks, but the functioning of the networks is inspired by the networks of biological neurons in the brain. In biology, the encoding of the neural structures in the genome is much more indirect, with the number of neurons in large mammalian brains vastly larger than the number of genes. However, the computational task faced by the networks investigated here is quite simple, consisting of directional movement toward target of an animat with two sensors and two actuators, so a simple encoding is more than sufficient.

2 Model

2.1 Evolving Spiking Neural Networks

The network model used in this paper does not restrict the number of nodes or connections in the networks (more precisely, the restrictions imposed by the limited computer memory are never reached in practice; however, the task considered here does not require large networks). Each internal node is encoded in the genome as a series of *cis genetic elements* followed by a series of *trans elements* (Fig. 1). Each element in the genome has several fields (four in the version of the model used in this paper): the *type* (cis, trans, and input or output), *sign*, and two *coordinates*. Three types of connections are allowed between the nodes: input-cis—encoded by one input and one cis element—and, similarly, trans-cis, and trans-output. The signs determine if a particular connection is inhibitory (when the signs of two elements are different) or excitatory (when the signs are the same).

A connection is formed if the coordinates of two elements are such that the Euclidean distance between the corresponding points in an abstract 2-dimensional space is below a predefined threshold (5.0). The smaller the distance, the higher the weight of the connection (using the positive part of the function $\frac{10-2d}{d+1}$, where d is the distance between the elements). If more than one connection is formed between any two nodes, the weights are added, giving in the end either a positive synaptic weight (an excitatory connection) or a negative one (an inhibitory connection). For example, to calculate the weight between the second sensory neuron in the network for the animat in Figs. 1 and 3 (marked as S) to the first interneuron (which has 5 cis elements), we need to add the weights coming from the interactions between input-cis pairs $(3, 4)$, $(3, 5)$, and $(3, 8)$, the other pairs have distances higher than 5.0. This gives the weight $-0.16 - 3.07 + 2.04 = -1.19$.

Input nodes have one state variable, determined by the sensors on the animat; internal and output nodes have four state variables: membrane potential v, adaptation current w, excitatory conductance g_E, and inhibitory conductance g_I. They are governed by four differential equations, according to the adaptive exponential integrate-and-fire model of spiking neurons (AdEx, EIF) [2,5]:

$$\frac{dv}{dt} = \frac{g_L(E_L - v) + g_L\delta e^{\frac{v-V_T}{\delta}} + w + g_E(E_E - v) + g_I(E_I - v) + I_{offset}}{C} \tag{1}$$

$$\frac{dw}{dt} = \frac{a(V - E_L) - w}{\tau_w} \tag{2}$$

$$\frac{g_E}{dt} = \frac{-g_E}{\tau_E} \tag{3}$$

$$\frac{g_I}{dt} = \frac{-g_I}{\tau_I} \tag{4}$$

Euler integration was used, with 1-ms steps. The exponential term gives an upswing of the action potential (when the input current, $g_E(E_E-v)+g_I(E_I-v)+I_{offset}$, drives the membrane potential beyond V_T), which is stopped when the potential reaches 0 mV, and the downswing (in the next simulation step) results from the reset condition: v is given the value of V_r, and w is incremented by b. If the neuron has a negative (positive) connection, the inhibitory (excitatory) conductance of the postsynaptic neuron is incremented by the synaptic gain (0.003 µS) multiplied by the weight.

The values of the parameters used in this paper give tonic spiking when the input current is constant (above about 0.2204 nA): leak conductance $g_L = 0.01\,\mu\text{S}$, rest potential $E_L = -70\,\text{mV}$, slope factor $\delta = 2\,\text{mV}$, threshold potential $V_T = -50\,\text{mV}$, excitatory reversal potential $E_E = 0\,mV$, inhibitory reversal potential $E_I = -70\,mV$, offset current $I_{offset} = 0$ for internal neurons and $I_{offset} = 0.5\,\text{nA}$ for output nodes, membrane capacitance $C = 0.2\,\text{nF}$, adaptation coupling $a = 0.002\,\mu\text{S}$, adaptation time constant $\tau_w = 30\,\text{ms}$, synaptic time constants $tau_E = tau_I = 5\,ms$, reset voltage $V = -58\,\text{mV}$, and adaptation increment $b = 0\,\text{nA}$.

2.2 Animats and Their Environment

The animat has two sensors and two actuators. The state of the sensors depends on the amount of the signal received from the targets (which can be seen as,

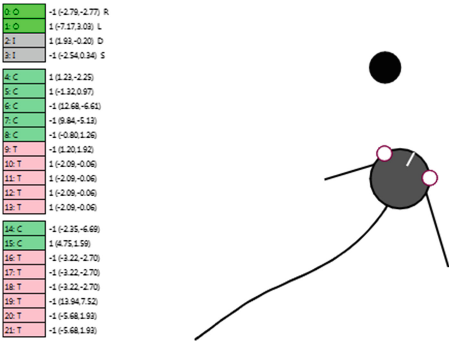

Fig. 1. The genome and the animat. The genome of animat shown in Fig. 3 is shown as an example (*left*); it consists of two sets of a series of cis (dark green) and trans (pink) elements, thus encoding two internal nodes. The four elements on top encode the outputs (green) and inputs (gray). The animat (*right*) has two sensors in front (light circles) and two actuators behind them (the direction of the thrust is shown by black straight lines); the trace of the movement for the first 500 ms is indicated by the curvy line; as the animat approaches the target on its left, the right actuator is slightly more active. (Color figure online)

for example, light intensity or scent [3]). The signal coming from a given target decreases with the Euclidean distance (d_{Euc}) from this target (as $\frac{1}{1+0.2d_{Euc}}$), and reaches maximum (1.0) at zero distance. The signal coming from all the targets is summed. When a target is reached, it disappears, and the signal field changes instantaneously. The activation of the sensor (S_L and S_R, Fig. 1) is equal to the value of signal at the sensor's location.

The sensory information is provided to the neurons that connect to the input nodes. The state of one of these nodes (S, for sum) depends on the average activation of both sensors on the animat ($\frac{2}{1+e^{-\gamma_{avg}(S_R+S_L)}} - 1$), and the state of the other (D, for difference)—on the difference in sensors' activation ($\frac{1}{1+e^{-\gamma_{dif}(S_R-S_L)}}$). In other words, the state of the node D is 0.5 when the activation of the left and right sensor on the animat is the same, and it decreases towards 0 (or increases towards 1) when the right-left difference decreases (or increases). The steepness of the sigmoid functions is set to amplify small differences or to allow for a dynamic response even when the animat is close to several targets ($\gamma_{dif} = 10$, $\gamma_{avg} = 0.5$). At each simulation step, the state of each input node is determined, rounded to the largest previous hundredth (to simulate sensors with limited precision), and this value, multiplied by the synaptic gain and the weight of the connection to a postsynaptic internal neuron to which the input node connects, is added to the excitatory (or inhibitory, if the weight is negative) conductance of this postsynaptic neuron.

The thrust forces (Fig. 1) generated by the actuators are proportional to number of spikes of the output neurons in the previous 120 ms. The directions of the forces are such that when the activations of the actuators differ, the animat turns, but even when only one actuator is active, the animat moves in a loop rather than turning on the spot. When both previously active actuators become inactive, the motion continues due to inertia until the animat is brought to a stop by drag (proportional to velocity). This drag also imposes a maximum velocity possible.

2.3 Genetic Algorithm

Each evolutionary run consisted of 250 generations of a genetic algorithm with a constant population size of 300 individuals, with binary tournament selection (draw two, keep the better one), and elitism (10 individuals). The genomes of the animats in the initial population had, apart from two input and two output elements, three series of cis and trans elements (the number of cis and trans in each series was drawn from the normal distribution with mean and standard deviation both equal to 3.0, rounded to the largest smaller integer; all numbers below 1.0 were binned to 1). Coordinates in genetic elements were determined by drawing a random direction and a random distance from $(0, 0)$ using a uniform distribution. Genetic operators were changes of coordinates (with probability 0.005 per gene; coordinate change causes the point corresponding to the element to move in the abstract 2-dimensional space by a distance drawn from a normal distribution), deletions, and duplications of individual elements. The probabilities of deletions and duplications were 0.00375 and 0.0025, respectively, creating a mutational pressure for short genomes.

The genetic algorithm aimed to minimize the average value of the fitness function over 5 random maps with 20 targets each, $f_{fit} = 1 - \frac{c_{targets}}{20})$, where $c_{targets}$ is the amount of targets reached. The animats were allowed to move for 24000 ms during evolution, but when analysing the champions after each run (and testing robustness), the fitness was re-evaluated by averaging for 1000 random maps with 20 targets and 48000 ms for each map.

Since the output nodes spike at a constant frequency without input from internal neurons (because of the positive offset current), most animats in the initial population moved, although not directionally.

3 Results and Discussion

Although the foraging task considered here is quite simple, corresponding to Braitenberg vehicle 2b [1], it is not completely trivial. This is because the number of targets gets smaller with each find, changing the activation of the sensors on the animat, and the control has to be tuned to the physics of the environment (the drag forces) and the animat (the thrust of the actuators). In fact, only about 6 % of the independent runs resulted in networks that allowed the animat to find nearly all the targets on any map (fitness function below 0.1, meaning that at most about 2 targets out of 20 were left on average). Only these champions were analysed further (9 from 159 independent runs conducted with Gaussian noise on the membrane potential with standard deviation 5 mV; 13 from 209

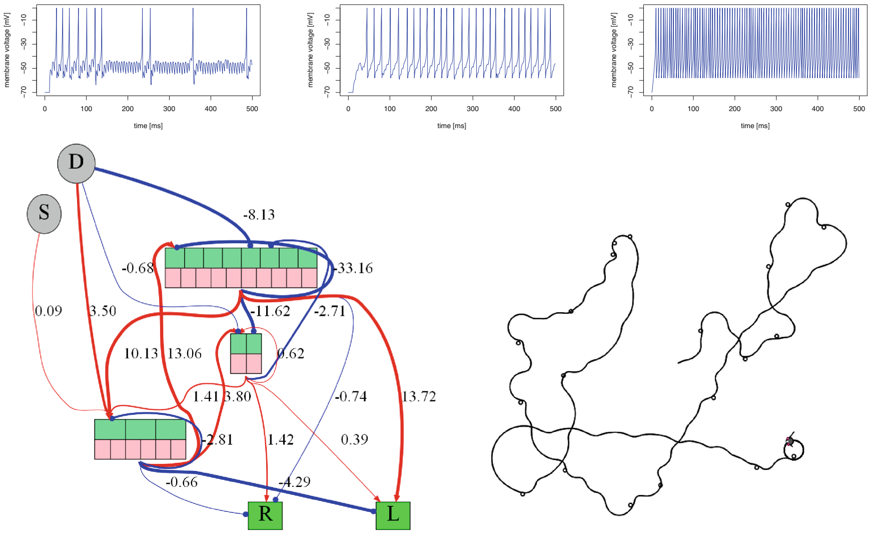

Fig. 2. The behavior of the best animat in the cohort of 13 champions evolved without noise. See text for details.

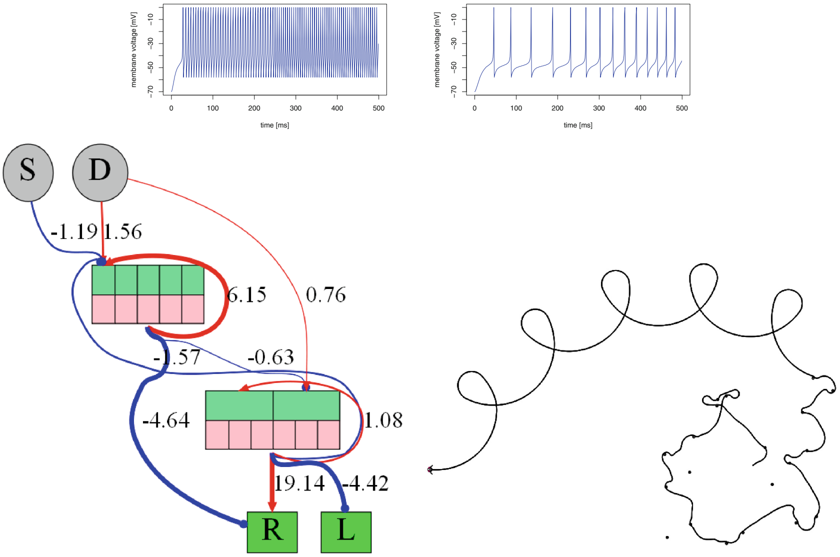

Fig. 3. The behavior of the worst animat in cohort of 13 champions evolved without noise. See text for details.

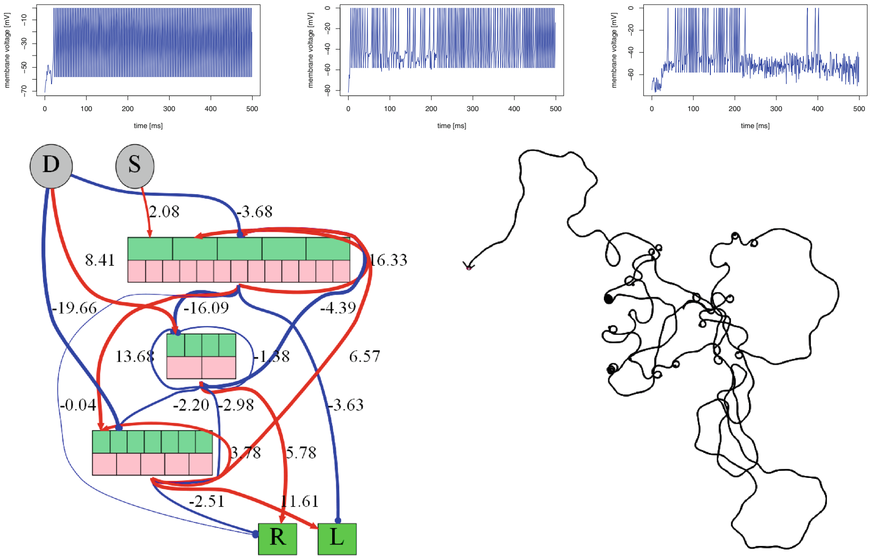

Fig. 4. The behavior of the best animat in the cohort of 9 champions evolved with noise. See text for details.

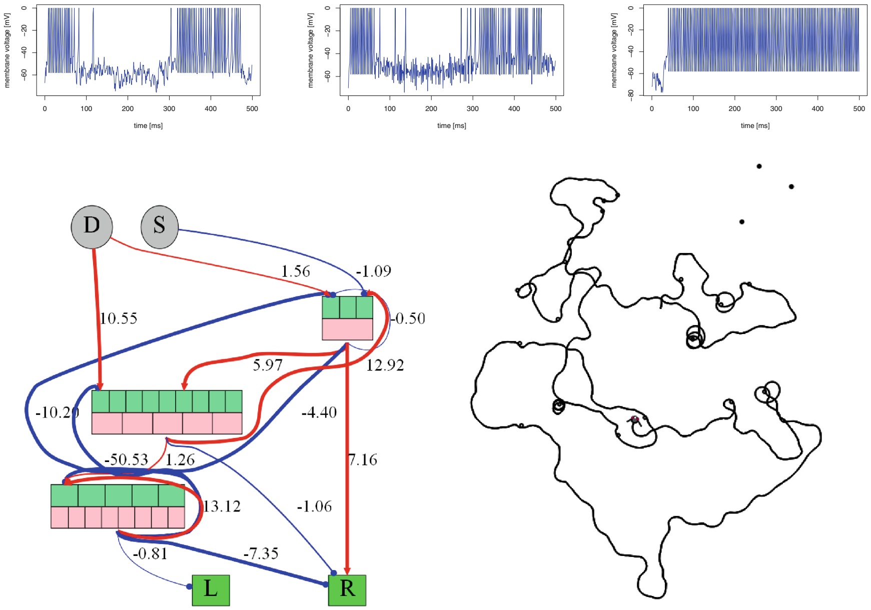

Fig. 5. The behavior of the worst animat in the cohort of 9 champions evolved with noise. See text for details.

Table 1. Comparison of the robustness of the 9 champions evolved with noise on voltage and 13 evolved without. The fitness function values (all the columns but the last 2) have been obtained for 1000 random maps and 48000 ms; the last 2 columns indicate the number of cis and the trans elements in the genomes. The values of the fitness function below 0.25 are highlighted in red, the values below 0.33 in green, and below 0.40 in grey. The last row shows the results of the comparison between 9 and 13 values in each column using the Anderson-Darling k-samples test, with values below 0.05 highlighted in red. See text for more details.

original	Min	Max	Int I	Int I	Out I	Out I	gain	gain	Vr	Vr	EL	EL	Sum	Sum
f_fit	A 15%	A 15%	-0.2	0.2	0.2	0.8	0.0025	0.0035	-68	-48	-100	-40	cis	trans
Evolved with noise on voltage (SD 5 mV)														
0.015	0.293	0.308	0.123	0.135	0.012	0.028	0.304	0.353	0.183	0.168	0.140	0.194	16	20
0.017	0.042	0.151	0.062	0.169	0.043	0.024	0.130	0.267	0.520	0.211	0.093	0.245	15	22
0.030	0.144	0.201	0.170	0.097	0.088	0.033	0.361	0.224	0.086	0.032	0.325	0.164	20	33
0.035	0.154	0.184	0.112	0.109	0.231	0.089	0.139	0.257	0.186	0.328	0.093	0.082	14	15
0.064	0.141	0.262	0.259	0.184	0.388	0.181	0.123	0.046	0.175	0.291	0.292	0.127	10	18
0.086	0.444	0.467	0.623	0.106	0.071	0.199	0.869	0.087	0.315	0.376	0.802	0.362	16	8
0.088	0.244	0.287	0.214	0.416	0.383	0.124	0.328	0.416	0.133	0.125	0.228	0.388	18	17
0.089	0.163	0.220	0.434	0.442	0.363	0.141	0.412	0.235	0.389	0.058	0.391	0.408	18	21
0.095	0.183	0.263	0.135	0.143	0.375	0.166	0.321	0.186	0.805	0.825	0.260	0.119	17	14
Evolved without noise on voltage														
0.000	0.001	0.211	0.995	0.935	0.669	0.497	0.911	0.952	0.829	0.784	0.999	0.934	13	18
0.008	0.176	0.284	0.991	0.991	0.998	0.632	0.565	0.556	0.427	0.870	0.991	0.992	14	8
0.048	0.250	0.651	0.997	0.983	0.998	0.664	0.564	0.341	0.414	0.972	0.982	0.994	8	13
0.053	0.437	0.587	0.994	0.927	0.999	0.722	0.949	0.995	0.946	0.995	0.992	0.854	14	8
0.062	0.218	0.491	0.447	0.952	0.999	0.224	0.632	0.281	0.677	0.986	0.998	0.982	10	21
0.066	0.400	0.464	0.997	0.998	0.995	0.571	0.992	0.810	0.402	0.765	0.997	0.998	15	20
0.074	0.287	0.392	0.995	0.988	0.999	0.727	0.765	0.836	0.233	0.981	0.995	0.978	15	13
0.087	0.430	0.461	0.744	0.527	0.997	0.584	0.622	0.122	0.831	0.989	0.973	0.792	16	14
0.088	0.431	0.450	0.997	0.995	0.998	0.501	0.940	0.766	0.636	0.992	0.999	0.994	7	9
0.090	0.294	0.353	0.995	0.993	0.998	0.779	0.907	0.739	0.475	0.970	1.000	0.991	11	9
0.096	0.224	0.384	0.913	0.996	0.287	0.357	0.896	0.778	0.268	0.994	0.996	0.923	18	14
0.098	0.280	0.299	0.998	0.970	0.998	0.443	0.998	0.925	0.997	0.926	0.999	0.980	11	18
0.099	0.116	0.481	0.987	0.997	0.998	0.427	0.766	0.161	0.236	0.943	0.988	0.995	7	11
AD test p-value	0.18	0.0052	5.4E-5	3.6E-5	0.00025	3.6E-5	0.0002	0.006	0.014	7.9E-5	3.8E-5	3.6E-5	0.019	0.055

independent runs without noise, Table 1; all the re-evaluations shown in the table were performed using the noise levels at which the networks were evolved).

The results clearly indicate that networks evolved with noise were much more robust to other perturbations (Table 1). Nearly all of such networks (with the exception of one) allowed to find at least two thirds of the targets when the activity of one of the actuators was increased by 15 %, and the activity of the other actuator was decreased by 15 % (Table 1, column 2 and 3). Some animats performed better when the left actuator had higher and the right lower activity; for the other animats the inverse situation resulted in better performance, so Table 1 shows the results for the better scenario in column 2 and for the worse scenario in column 3. The networks evolved without noise did not fare well in the worse scenario (see Table 1 for p-values of all the statistical tests; all tests were done using the Anderson-Darling k-sample test; I also used, when appropriate, the two-sample Kolmogorov-Smirnov test, the p-values for the two tests were very close in value).

Similarly, most of networks evolved with noise performed well when the internal neurons were given either positive or negative offset current of 0.2 nA (instead of 0; Table 1; column 4 and 5), when the offset current for the output nodes was decreased or increased by 0.3 nA (from 0.5 to 0.2 or 0.8, respectively; column 5 and 6), when synaptic gain was increased or decreased by 0.0005 µS (from 0.003 to 0.0025 or 0.0035; column 7 and 8), reset voltage decreased or increased by 10 mV (from 58 to −68 or −48; column 9 and 10), or rest potential decreased or increased by 30 mV (from 70 to −100 or −40; column 11 and 12). None of the networks evolved without noise performed well in these scenarios.

Two networks evolved with noise were not affected much even by a change from tonic spiking to tonic spiking with adaptation (by setting the adaptation increment $b = 0.1$ nA), the fitness function changed in these two cases from 0.017 to 0.119 and from 0.035 to 0.089, and two other networks could perform even with Gaussian noise with standard deviation 7 mV (the fitness function changed from 0.015 to 0.227, and from 0.030 to 0.109), while only one network without noise gave good behaviour with noise up to 2 mV (the fitness function changed from 0.074 to 0.198).

In general, the networks evolved with noise were encoded by slightly longer genomes than the networks evolved without noise (although all the networks in Table 1 had 3 internal nodes, with the exception of the one in the last row, shown in Figs. 1 and 3). This difference can be attributed mostly to a higher number of cis elements in the genomes evolved with noise (Table 1; column 13 and 14).

The detailed analysis of the topology of the networks indicates that all networks have a neuron spiking with high frequency and inhibiting one of the actuators (Figs. 2, 3, 4 and 5, the best and worst in each cohort is shown; the *top* panels in all four figures show the voltage in the first 500 ms of the three, or two, internal neurons; the order from left to right on the *top* corresponds to the order from top to bottom in the graph on the *bottom left* panel of each figure, D is for input node whose state corresponds to the difference in animat sensors, S is for the node whose state corresponds to the sum/average of the sensors, R and L indicate output nodes regulating right and left actuators, respectively, red edges correspond to excitatory, and blue to inhibitory connections; the *bottom right* shows the trajectory of the animat over 48000 ms).

For example, the network of the best animat in the cohort of 13 evolved without noise (Fig. 2) this high-frequency neuron is internal neuron 3, the one inhibiting stronger the left actuator. The network has also a self-sustaining loop between neurons 1 and 3, a weaker loop between 2 and 3, the neuron 2 activates itself very weakly, and neuron 1 inhibits itself strongly. The sustained activity of neurons 2 and 3 results in a continued circular counterclockwise movement after all the targets are reached (and the sensors are quiet); since the left actuator is inhibited (by neuron 3), the animat turns to the left.

The small network in Fig. 3 is a less efficient controller; the trajectory shows that three targets have been missed on this map (filled dark circles in the top right). With small activation of the sensors, and driven mostly by self-sustained spiking of both neurons (neuron 1 with higher frequency), the animat continues to move with the left actuator more activated than the right, and will not reach the targets if the simulation is continued.

In contrast, the animat in Fig. 4, evolved with noise, reaches all the targets, and—with no activation of the sensors—continues to move without any clear pattern, driven by noise, and sustained activity of the network (with neuron 1 at the highest frequency, much higher than the spiking frequency of the other neurons, but with neuron 3 having the most complex spiking pattern, with periods of activity separated by relative quiescence, apparent also in the first 500 ms, Fig. 4 *top*). Note that the noise prevents the animat to reach the target as precisely as is possible for the animats evolved and acting without noise (most targets are reached after an approach along a circular local trajectory).

The same is true for the worst animat in the cohort of 9 evolved with noise. This animat is relatively less efficient than the best one, but on this map it manages to collect all the missing 3 targets when the simulation is continued beyond 48000 ms shown (and used to reevaluate fitness); after all the targets are collected, the sustained activity of all neurons (with the highest frequency of neuron 3, and complex spiking pattern of neuron 2) continues.

4 Conclusion and Future Work

The main conclusion is that both an efficient control and robustness to perturbations to the value of the parameters of neurons can be evolved in the presence of noise, similarly to our previous results showing that robustness to noise in evolving genetic networks promotes robustness to damage [4]. The noise used in this paper was on one of the state variables of the spiking neural network; it will be interesting to see if other models of noise and perturbations (for example, relevant to neuromorphic hardware [6] will give similar results. Importantly, the networks in the model used here were encoded in an artificial genome in a way that does not limit in principle, and allows the evolutionary process to vary, the number of nodes and connections in the network. It remains to be seen if the evolution for more complex tasks and behaviors, possibly requiring more complex networks, is possible in this system.

Acknowledgments. This work was supported by Polish National Science Centre (project EvoSN, UMO-2013/08/M/ST6/00922). I am grateful to Volker Steuber for discussions, and to Ahmed Abdelmotaleb and Michal Joachimczak for their involvement in the development of GReaNs software platform.

References

1. Braitenberg, V.: Vehicles. Experiments in Synthetic Psychology. MIT Press, Cambridge (1986)
2. Brette, R., Gerstner, W.: Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. J. Neurophysiol. **94**(5), 3637–3642 (2005)
3. Joachimczak, M., Wróbel, B.: Evolving gene regulatory networks for real time control of foraging behaviours. In: Proceedings of the Alife XII Conference, pp. 348–358. MIT Press, Cambridge (2010)

4. Joachimczak, M., Wróbel, B.: Evolution of robustness to damage in artificial 3-dimensional development. Biosystems **109**(3), 498–505 (2012)
5. Naud, R., Marcille, N., Clopath, C., Gerstner, W.: Firing patterns in the adaptive exponential integrate-and-fire model. Biol. Cybern. **99**(4–5), 335–347 (2008)
6. Stromatias, E., Neil, D., Pfeiffer, M., Galluppi, F., Furber, S.B., Liu, S.C.: Robustness of spiking deep belief networks to noise and reduced bit precision of neuro-inspired hardware platforms. Front. Neurosci. **9** (2015). paper number 222
7. Wagner, A.: Robustness and Evolvability in Living Systems. Princeton University Press, Princeton (2013)
8. Wróbel, B., Abdelmotaleb, A., Joachimczak, M.: Evolving networks processing signals with a mixed paradigm, inspired by gene regulatory networks and spiking neurons. In: Di Caro, G.A., Theraulaz, G. (eds.) International Conference on Bio-Inspired Models of Network, Information, and Computing Systems, BIONETICS 2012. LNICST, vol. 134, pp. 135–149. Springer, Heidelberg (2014)
9. Wróbel, B., Joachimczak, M.: Using the genetic regulatory evolving artificial networks (GReaNs) platform for signal processing, animat control, and artificial multicellular development. In: Kowaliw, T., Bredeche, N., Doursat, R. (eds.) Growing Adaptive Machines. SCI, vol. 557, pp. 187–200. Springer, Heidelberg (2014)