Internal State Predictability as an Evolutionary Precursor of Self-Awareness and Agency

Jaerock Kwon
Department of Computer Science
Texas A&M University
College Station, TX 77843
Email: jrkwon@tamu.edu

Yoonsuck Choe
Department of Computer Science
Texas A&M University
College Station, TX 77843
Email: choe@tamu.edu

***Abstract*—What is the evolutionary value of self-awareness and agency in intelligent agents? One way to make this problem tractable is to think about the necessary conditions that lay the foundation for the emergence of agency, and assess their evolutionary origin. We postulate that one such requirement is the predictability of the internal state trajectory. A distinct property of one's own actions compared to someone else's is that one's own is highly predictable, and this gives the sense of "authorship". In order to investigate if internal state predictability has any evolutionary value, we evolved sensorimotor control agents driven by a recurrent neural network in a 2D pole-balancing task. The hidden layer activity of the network was viewed as the internal state of an agent, and the predictability of its trajectory was measured. We took agents exhibiting equal levels of performance during evolutionary trials, and grouped them into those with high or low internal state predictability (ISP). The high-ISP group showed better performance than the low-ISP group in novel tasks with substantially harder initial conditions. These results indicate that regularity or predictability of neural activity in internal dynamics of agents can have a positive impact on fitness, and, in turn, can help us better understand the evolutionary role of self-awareness and agency.**

I. Introduction

To build intelligent agents that can interact with their environments and also their own internal states, the agents must identify the properties of objects and also understand the properties of other animated agents [1]. One of the fundamental steps in having such abilities is to identify agents themselves from others. Therefore, finding *self* has been a grand challenge not only among cognitive scientists but also in computer scientists. Even though Feinberg and Keenan strongly suggested that the right hemisphere has a crucial role in the creation of the self [2], localizing the self does not answer many intriguing questions about the concept of the self. On the other hand, a Bayesian self-model that can distinguish *self* from *others* was proposed, and Nico, an upper-torso humanoid robot, was able to identify itself as *self* through the dynamic Bayesian model using the relationship between its motor activity and perceived motion [3]. Bongard, Zykov, and Lipson made a self-aware robot which can adapt to the environment through continuous self-modeling [4]. We believe that Autonomous Mental Development (AMD) [5] can also lead to a self-model, e.g., as in Self-organizing Autonomous Incremental Learner (SAIL) [5] and Dav [6].

Nico [3] focused more on higher level modeling of self-awareness; SAIL [5] and Dav [6] concentrated on modeling autonomous mental development; and the resilient machine [4] continuously re-modeled its physical body rather than conceptual self. As far as we know, neuronal substrate of *self* has not been discussed. It is not easy to answer the question about the self without invoking complex and controversial issues. However, an alternative way exists to address the problem: If we uncover *necessary conditions* for the emergence of self-awareness, then we might be able to make some progress. An interesting insight is that predictability in the internal state dynamics can be such a necessary condition. We postulate that the *predictability* of the neural activities in the internal dynamics may be the initial stepping stone to self-awareness. Such predictability could lead to authorship (and eventually agency and self-awareness), since a distinct property of one's own actions is that they are always perfectly predictable.

A. *Self-awareness*

Self-awareness has an important role in cognitive processes [7]. Self-aware system has an ability to distinguish itself from *others*. Being self-aware can be a good beginning to have cognitive capabilities. However why have intelligent agents such as humans evolved to have self-awareness? Is self-awareness simply an evolutionary by-product of self-representation as Menant pointed out [8]? Otherwise, if cognitive agents always have to be self-aware, there must be an associated evolutionary pressure. However, the attributes of self-awareness is still uncertain [9]. So, it is difficult to track down the roots of the emergence of self-awareness or agency. One way to circumvent the problem is to find circumstances that can serve as necessary conditions for the emergence of self-awareness, and assess their evolutionary value.

In this paper, we focus on finding these necessary conditions. One possible requirement would be the predictability of one's own internal state trajectory (another possibility is direct prediction of one's own action, as in Nolfi et al.'s work on neuroevolution combined with learning [10]). We postulate that Internal State Predictability (ISP) can have a strong impact on performance of the agents, and ISP could have lead to intelligent agents developing self-awareness.

B. Internal State

Many researchers have focused on external environments and behaviors when developing intelligent robots or agents. This was especially true when the investigations were carried out in an evolutionary context.

However, researchers started to take a serious look at the internal dynamics of an intelligent agent as well. The central nervous system models sensorimotor dynamics, and the model seems to reside in the cerebellum [11]. Exploring one's internal state can lead to a sense of self. The sense of self may be a prerequisite to building a machine with consciousness [12].

There may be a consensus that neuronal activation levels can be considered as the state of a neural system. Bakker and de Jong pointed out that the state of a neural network could be defined by the current activation levels of the hidden units [13]. Also, the system state could be viewed as consciousness, in a way [14]. There are also physiological arguments about this idea. The firing rate of each neuron in the inferior temporal visual cortex tells much about the stimuli applied to the cortex [14]. On the other hand, spiking activities from place cells in the hippocampus can be used to rebuild certain features of the spatial environment [15]. These results tell us that spiking patterns of neurons that form one's internal state might influence task performance. In sum, knowing internal state of oneself may be the first step of being conscious and internal state itself can be simply stated as spiking patterns of neurons during task performance.

The idea that self-awareness has evolutionary advantages is not new [8]. Menant hypothesized that noticing agony of conspecifics may be the first step in developing self-awareness. But as far as we understand, the precondition of identifying agony in conspecifics is self-awareness and the identification of agony is also a requirement of being self-aware. It falls into a circular argument. Namely, self-awareness is a requirement of identifying agony and also, identifying agony develops self-awareness. Moreover, Menant's argument is more like a hypothesis, without giving plausible evidence.

We present experimental results that indicate "understanding" internal states has an actual evolutionary benefit.

II. Method

We hypothesized that activation values from neurons in the hidden layer can represent the internal state of an agent. Understanding one's own internal state can be strongly linked to knowing what is going to happen in one's internal state. We quantified such an understanding as the predictability of the internal state trajectories.

In order to examine whether internal state predictability has any evolutionary value, we evolved sensorimotor control agents with recurrent neural network controllers. The neural activity in the hidden layer of the network was viewed as the internal state of an agent. A two-degree-of-freedom (2D) pole balancing task was chosen to scrutinize the internal state trajectories. The neural network controllers were trained by a neuro-evolution method. The activation values from each neuron in the hidden layer from the neural network were stored to measure the predictability of each neuron's activation trace over time. The predictability of each neuron was quantified by a supervised learning predictor which forecasted the next activation value based on the past activations. Note that any reasonable predictor can be used for this, e.g. Hidden-Markov models, and the choice is orthogonal to the main argument of this paper.

A. Two-Degree-of-Freedom Pole Balancing

Pole balancing task has been used to demonstrate complex and unstable nonlinear dynamical systems in the field of artificial neural networks for decades because it is straightforward to understand and easy to visualize. A conventional 1D pole balancing task has dealt with the following situation: A pole is hinged atop a cart that travels along a single straight line track. The pole can only move on the vertical plane along the track [16], [17]. It makes the task simple enough to be analyzed, but it is not complex enough to show interesting behavior. Here, we used 2D pole balancing where force to the cart can be applied in both the x and the y directions, so the cart moves around on a 2D plane within a boundary and a pole attached on top of the cart can fall in any direction [18], [19]. As a result, the task is more complex and difficult to master than 1D version. Figure 1 shows a 2D pole-balancing system with which we conducted our experiment. (Removing velocity information from the 1D problem could make the task more difficult and thus more interesting, but we did not pursue this direction.)

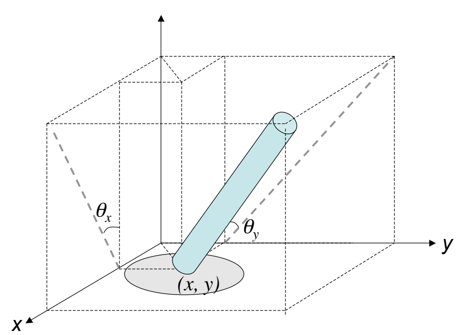


Fig. 1. Two-degree-of-freedom pole-balancing system.

The state of the cart (the gray circle on the bottom figure 1) with a pole on top is characterized by the following physical parameters: The cart position in the plane (x, y), the velocity of the cart $(\dot{x}, \dot{y})$, the angle of the pole from the vertical in the x and the y directions (θ_x, θ_y), and their angular velocities $(\dot{\theta_x}, \dot{\theta_y})$ [18]. These parameters were used as eight input values to a neural network. Fourth-order Runge-Kutta method was used to simulate the real world physics.

B. Time Series Prediction

A time series is a sequence of data from a dynamics system. The measurements of one or more variables of the system take place at a successive and regular time interval [20]. The system dynamics changes the state over time, so it can be considered as a function of the current state vector $x(t)$. A time series is a sequence of either vectors or scalars.

$$\{x(t_0), x(t_1), \cdots, x(t_i), x(t_i), x(t_i+1), \cdots\}$$

The activation level of hidden neurons in a neural network can be considered as a time series. In our case, three sets of time series exist since there are three neurons in the hidden layer of our neural network. Let us assume that we predict value x at time $t+1$ which is the very next state from present. If we can look back N time steps including the current one from time t, we can say that forecasting $x(t+1)$ means finding a function $f(\cdot)$ using a time series $\{x(t), x(t-1), x(t-2), \cdots x(t-N+1)\}$(figure 2):

$$\hat{x}(t+1) = f\left(x(t), x(t-1), x(t-2), \cdots, x(t-N+1)\right).$$

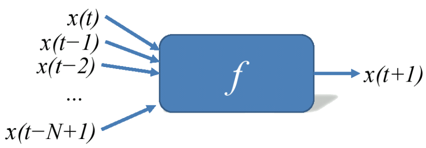


Fig. 2. Predicting future using the past.

1) Neural Network Predictors: Feed-forward neural networks have been widely used to forecast a value given a time series dataset [20]. The neural predictors use a set of N data as inputs, and a single value as an output for the target of the network. The number of input data is often called the sliding window size [20]. Figure 3 gives the basic architecture of a feed-forward neural network predictor.

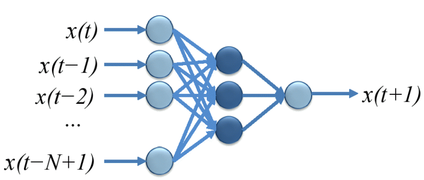


Fig. 3. A neural network predictor for a time series.

2) Adaptive Error Rates: When a neural network predictor forecasts a future state, the outcome of the predictor, which is a predicted value, should be compared with a real activation value. If a prediction error, the difference between a predicted and a real value, is greater than a certain amount (we call it minimum error threshold) then it is fair to say the prediction had failed. However, we cannot use a fixed minimum error threshold, because amplitude envelope of activation values could be different from neuron to neuron. Why does the envelope of activation matter?

As figure 4 shows, it cannot be stated that two cases have the same amount of error, although the actual amount of error

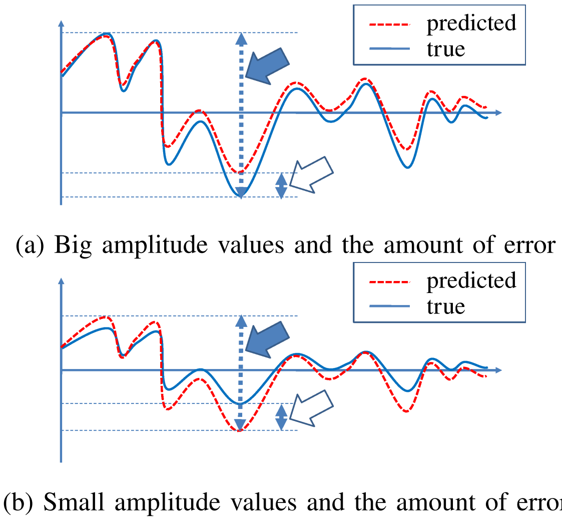


Fig. 4. An example of adaptive error rates. The big solid arrows indicate the amplitude of activation values, and the hollow big ones indicate the amount of error. When estimating the error, the activation amplitude envelope should be considered.

is almost the same in (a) and (b). The minimum error threshold value should be adapted to the variance of the time series. Namely, if the amplitude of a time series is also small, the minimum error threshold should be small, and if it is large, then the threshold should become large as well.

$$Err_{th} = |ActValue_{max} - ActValue_{min}| \times R,$$

where Err_{th} is the minimum error threshold, $ActValue$ means an activation value of a neuron in a hidden layer, and R is the adaptive rate for the minimum error threshold adjusted based on the activation amplitude as shown in figure 4.

III. Experiments and Results

We evolved agents driven by a recurrent neural network in a 2D pole balancing task, and then partitioned successful individuals into two groups: One group had high ISP, and the other had low ISP. The high-ISP group showed better performance than the low ISP group in tasks with harsher initial conditions.

A. Training the Controllers

We implemented the pole balancing agent with a recurrent neural network controller. The artificial neural networks were trained by genetic algorithms. Network connection weights of an agent were evolved to balance the pole during the training sessions.

Force between $-10N$ and $10N$ was applied at regular time intervals (10 millisecond). The pole was 0.5 meter long and was initially tilted by 0.573° (0.01 radian) on the x-z plane and the y-z plane respectively. The area where the cart moved around was $3 \times 3\ m^2$.

1) Neural Network Architecture: The configuration of a controller network was as follows: eleven input nodes (eight input values from the simulated physical environment and three context input values from the hidden layer), one hidden layer with three neurons, and two output neurons (figure 5). The eight parameters describing the current state of the cart were used as the input values, and two values from the output neurons, F_x and F_y, represented the force in the x and the y direction.

Figure 5 shows the recurrent network that we used in the experiments.

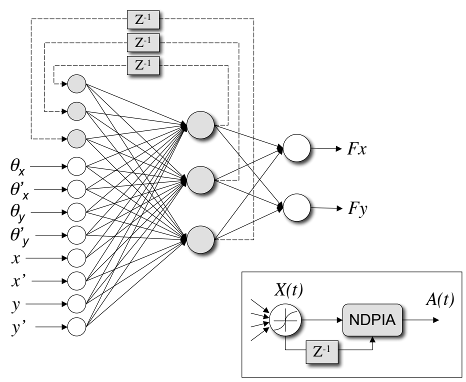


Fig. 5. Recurrent neural network controller for 2D pole-balancing. The signal flow inside each neuron is shown in the box. Z^{-1} means unit delay. [21]

2) Neuro-evolution: In training these non-linear controllers, neuroevolution methods have proved to be efficient [22], [23]. Fitness was determined by the number of time steps where a network was able to keep the pole within $\pm 15°$ from the vertical in the x and the y directions and kept the cart within the 3×3 m^2 area. The chromosome encoded the connection weights between input and hidden layer neurons, and between hidden and output neurons. Crossover occurred with probability 0.7 and the chromosome was mutated by ± 0.3 perturbation rate with probability 0.2. The force was applied in both the x and the y directions at 10 millisecond intervals. The number of networks in a population was 50 for an evolutionary epoch. If an agent balanced the pole more than 5,000 steps, we considered it as a success.

B. Training the Neural Network Predictors

Neuronal activities in the hidden layer of the recurrent neural network were viewed as the internal state of the agent. The predictability in the internal state trajectory was able to be measured using a feed forward neural network predictor.

The size of the sliding window was four. The activation values of neurons in the hidden layer formed the network input. 3,000 activation values were used as training data for each input, and a test set used the next 1,000 steps (3,001 to 4,000). Time series from 1 to 1,000 steps and from 4,001 to 5,000 steps were not used because we did not want to use the somewhat chaotic initial movements and finalized stable movements. Back-propagation algorithm was used to train the predictors (learning rate 0.2). In the test sessions, we compared the predicted value with the real activation value. We chose 10% threshold error rate to calculate the adaptive minimum error threshold when comparing the forecasted activation with the real activation value. The adaptive error rate was determined empirically, based on the performance of the time series predictor.

C. Performance Measurement in High- vs. Low-ISP groups

We evolved approximately 130 pole balancing agents. By definition, all the agents were necessarily good pole balancers during the training phase. Some of them turned out to have high ISP and others low ISP. Figure 6 shows all the agents sorted by their prediction success rates.

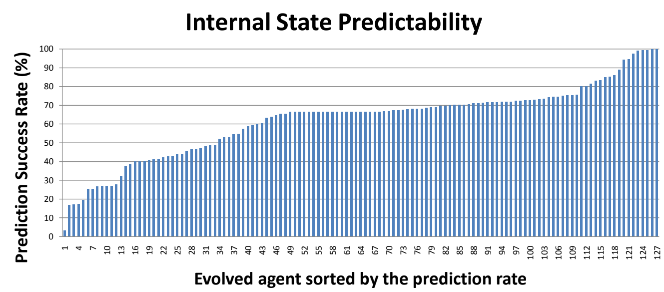


Fig. 6. All trained agents sorted by their prediction success rates. A small number of agent toward the left end show very low predictability, while those near the right end show very high predictability.

We chose pole balancers having top 10 highest ISPs, and bottom 10 lowest ISPs. High predictability means all three neurons from the hidden layer have highly predictable internal state trajectories. Most of their prediction rates in a high ISP group were over 99%, and only two pole balancers had average prediction success rates 83.30% and 88.93% ($\mu = 95.61\%$ and $\sigma = 5.55\%$). As for low ISP pole balancers, their average prediction performances from the three neurons were between 17.37% and 48.53% ($\mu = 31.74\%$ and $\sigma = 10.79\%$). Figure 7 shows the predictability in the high and the low ISP group.

The learning time of the two different groups was also investigated, but we could not find a significant difference, even though low ISPs took slightly less time than high ISP (figure 8). Note again that the performance of both groups (high and low ISP) were comparable during the evolutionary trials.

In order to further test and compare the performance between the two groups, we made the initial condition in the 2D pole balancing task harder than that during the training phase. All the neural network controllers were evolved in a condition where both projected initial angles to the x-z and the y-z plain were 0.573° (0.01 radian). In the test condition, we had those trained neural network controllers balance the pole in a more difficult initial condition where the initial projected angles were 4.011° (0.07 radian) on the $x-z$ plain, and 2.865° (0.04 radian) on the y-z plain. This strategy was used to push the controllers to the edge so that differential behavior results.

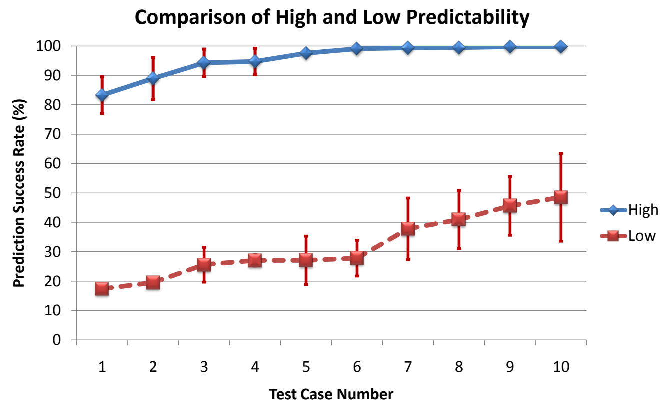


Fig. 7. A comparison of the average predictability from two groups: high ISP and low ISP. The predictive success rate of the top 10 and the bottom 10 agents from figure 6 are shown. Each data point plots the mean predictive success rate of three hidden neurons of each agent. The error bars indicate the standard deviation.

Our main results were that networks with higher ISP show better performance in harder tasks than those with lower ISP.

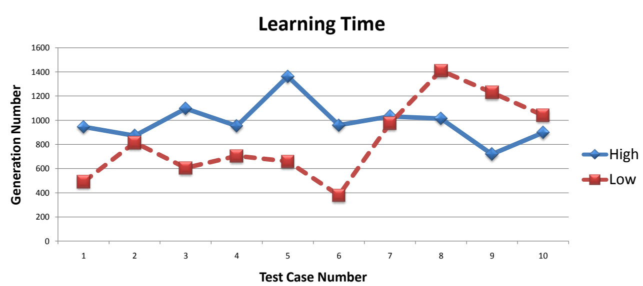


Fig. 8. Learning time in high vs. low ISP groups. Agents are ordered in the same order as in figure 7.

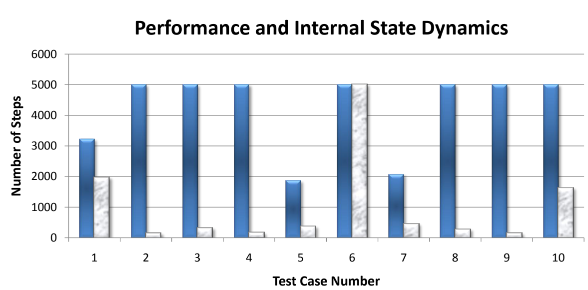


Fig. 9. Novel task performance in high vs. low ISP groups. Each test case (with low vs. high ISP results) corresponds to an experiment ran with identical initial condition, so the pairing is meaningful.

Figure 9 shows that the evolved pole balancers with higher ISP have better performance than the other group.

One might argue that this result seems straightforward, because simple internal state trajectories simply reflect behavioral properties. A trivial solution to a pole balancing problem would be to quickly make the pole stand up vertically,

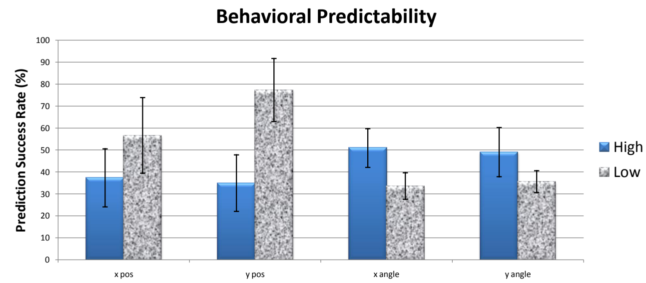


Fig. 10. Mean behavioral predictability (error bars indicate the standard deviation).

and then make minimal adjustments. But according to our experimental results (see figure 10), higher ISP does not necessarily mean that their behavioral trajectory is also simple. Figure 11 (compared to figure 13) and figure 12 (compared to figure 14) show that behavioral complexity is not necessarily directly related to the complexity of internal states. That is, even an agent having high ISP may have complex behavioral properties, and those with low ISP may have simple behavioral attributes.

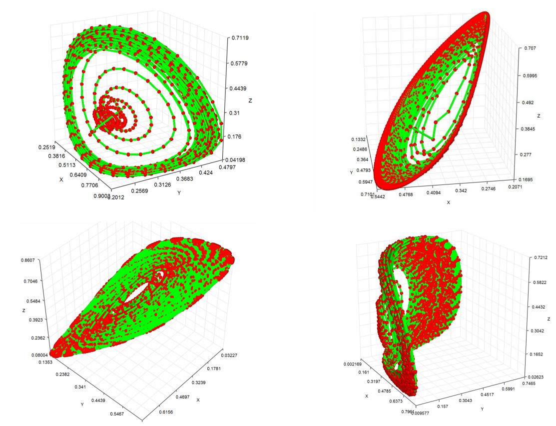


Fig. 11. Examples of internal state dynamics from the high ISP group showing smooth trajectories.

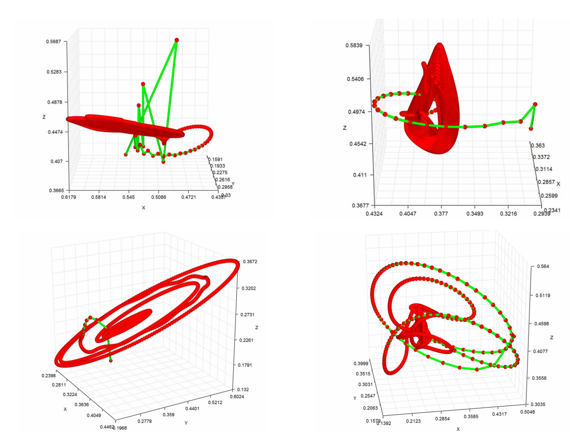
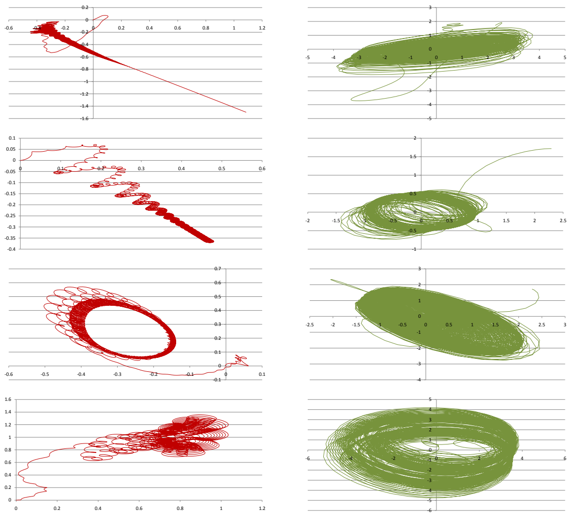
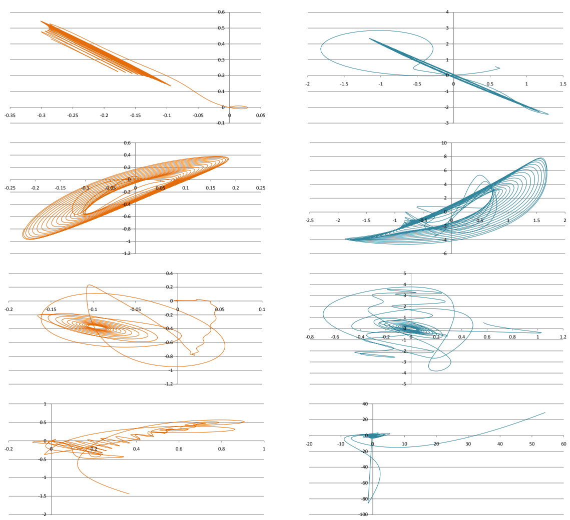


Fig. 12. Examples of internal state dynamics from the low ISP group showing abrupt, jittery trajectories.



(a) Cart x, y position (b) Pole angle

Fig. 13. Examples of behavioral trajectories of x, y positions and pole angles from the high ISP group exhibiting complex trajectories.



(a) Cart x, y position (b) Pole angle

Fig. 14. Examples of behavioral trajectories of x, y positions and pole angles from the low ISP group exhibiting complex trajectories.

IV. Conclusion

Starting with individuals evolved to give the same level of behavioral performance, we showed that those with simpler (more predictable) internal dynamics can achieve higher levels of performance in harsher environmental conditions. These results suggest that internal agent properties such as simpler internal dynamics may have a survival value. We also showed that the increased survival value is not always due to smoother behavior resulting from the simpler internal states. The implication of these findings is profound, since they show that, in changing environments, apparently circumstantial internal agent properties can affect external behavioral performance and fitness. The results also show how an initial stepping stone (or a necessary condition) in the evolutionary pathway leading to self-awareness and agency could have formed. We expect the framework we developed here to help us better address hard issues such as self-awareness and agency in an evolutionary context. Future directions include evolution of model-based prediction of both internal and external dynamics (cf. [4], [10]), and generalization of our framework to other more complex tasks.

References

[1] B. Scassellati, "Theory of mind for a humanoid robot," *Autonomous Robots*, vol. 12, pp. 13–24, 2002.

[2] T. E. Feinberg and J. P. Keenan, "Where in the brain is the self?" *Consciousness and Cognition*, vol. 14, no. 4, pp. 661–678, 2005.

[3] K. Gold and B. Scassellati, "A Bayesian robot that distinguishes "self" from "other."," in *Proceedings of the 29th Annual Meeting of the Cognitive Science Society*, Nashville, Tennessee, 2007.

[4] J. Bongard, V. Zykov, and H. Lipson, "Resilient machines through continuous self-modeling," *Science*, vol. 314, pp. 1118–1121, 2006.

[5] J. Weng, J. L. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 5504, pp. 599–600, 2001. [Online]. Available: http://www.sciencemag.org/cgi/content/full/291/5504/599

[6] J. D. Han, S. Q. Zeng, K. Y. Tham, M. Badgero, and J. Y. Weng, "Dav: A humanoid robot platform for autonomous mental development," in *Proceedings of the 2nd International Conference on Development and Learning*, Cambridge, Massachusetts, 2002, pp. 73–81.

[7] N. Block, "On a confusion about a function of consciousness," in *Behavoral and Brain Sciences*, 1995, pp. 227–247.

[8] M. C. Menant, "Proposal for an approach to artificial consciousness based on self-consciousness," November 2007. [Online]. Available: http://cogprints.org/5866/

[9] J. Taylor, "CODAM: A neural network model of consciousness," *Neural Networks*, vol. 20, no. 9, pp. 983–992, 2007.

[10] S. Nolfi, J. L. Elman, and D. Parisi, "Learning and evolution in neural networks," *Adaptive Behavior*, vol. 3, pp. 5–28, 1994.

[11] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends in cognitive science*, vol. 2, pp. 338–347, 1998.

[12] K. Kawamura, W. Dodd, P. Ratanaswasd, and R. Gutierrez, "Development of a robot with a sense of self," *Computational Intelligence in Robotics and Automation, 2005. CIRA 2005. Proceedings. 2005 IEEE International Symposium on*, pp. 211–217, 2005.

[13] B. Bakker and M. de Jong, "The epsilon state count," in *From Animals to Animats 6: Proceedings of The Sixth International Conference on Simulation of Adaptive Behavior*, 2000, pp. 51–60.

[14] E. T. Rolls, "A computational neuroscience approach to consciousness," *Neural Networks*, vol. 20, no. 9, pp. 962–982, 2007.

[15] V. Itskov and C. Curto, "From spikes to space: reconstructing features of the environment from spikes alone," *BMC Neuroscience*, vol. 8, no. Suppl 2, p. P158, 2007.

[16] C. W. Anderson, "Learning to control an inverted pendulum using neural networks," *IEEE Control Systems Magazine*, vol. 9, pp. 31–37, 1989.

[17] B. Mehta and S. Schaal, "Forward models in visuomotor control," *Journal of Neurophysiology*, vol. 88, pp. 942–953, 2002.

[18] F. Gomez and R. Miikkulainen, "2-D pole balancing with recurrent evolutionary networks," in *Proceedings of the international conference on artificial neural networks*, 1998, pp. 425–430.

[19] H. Lim and Y. Choe, "Facilitating neural dynamics for delay compensation and prediction in evolutionary neural networks," in *GECCO '06: Proceedings of the 2006 conference on Genetic and evolutionary computation*, 2006, pp. 167–174.

[20] R. J. Frank, N. Davey, and S. P. Hunt, "Time series prediction and neural networks," *J. Intell. Robotics Syst.*, vol. 31, no. 1-3, pp. 91–103, 2001.

[21] J. Kwon and Y. Choe, "Enhanced facilitatory neuronal dynamics for delay compensation," *IJCNN 2007. International Joint Conference on Neural Networks, 2007*, pp. 2040–2045, 2007.

[22] F. Gomez and R. Miikkulainen, "Active guidance for a finless rocket through neuroevolution," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2003, pp. 2084–2095.

[23] F. Gomez, "Robust non-linear control through neuroevolution," Ph.D. dissertation, Department of Computer Science, The University of Texas at Austin, Austin, TX, 2003, Technical Report AI03-303.