

Investigating a Minimal Categorical Perception Task with a Node-Based Sensorimotor Map

Felix M. G. Woolford and Matthew D. Egbert

School of Computer Science, University of Auckland, Auckland, New Zealand
fmg.woolford@gmail.com

Abstract. We present a variation on a classic evolutionary robotics experiment using an unorthodox controller model, a Node-Based Sensorimotor Map (NB-SMM) – a stateless, deterministic, continuous-time controller which directly maps an animat’s sensorimotor state to a change-in-motor-state command. Our investigation illustrates how such a simple model can be used in an evolutionary robotics context in place of a more typical neural network model, but offers its own set of practical and theoretical points of difference. Previous work has likened the role of the internal state of a recurrent neural network to the role of the brain in the brain-body-environment system. It is often assumed that in order to produce adaptive behaviour, our control systems require some internal state which modulates the dynamics of the animat’s sensorimotor loop. Our results challenge this received view by demonstrating that a stateless controller can also perform tasks that may be considered cognitive.

1 Introduction

A key benefit of the evolutionary robotics approach is the ability to strip away all but the most essential requirements of a model of cognitive behaviour. Seth’s animats, for example, illustrated a counterpoint to the assumption that an internal arbitration mechanism was required for action selection, by presenting a model with only a “set of independent sensorimotor links, and the influence of some internal state” [7]. What exactly is the role and nature of such an internal state though? How does it relate to the role of knowledge and representation in sensorimotor perception [8]? One way to clarify this is to investigate models which have no internal state whatsoever and evaluate their capabilities. To this end, we use a stateless controller model to revisit a previous investigation of a minimal categorical perception behaviour [3].

Node-Based Sensorimotor maps (NB-SMM) are a class of continuous-time controller models which operate by deterministically mapping the instantaneous sensorimotor state of an animat to a change-in-motor state output. The defining feature of an NB-SMM is that the parameters of the mapping function are determined by a limited number of *nodes* in a sensorimotor space. A benefit of the node-based approach is that the process of generating and adjusting these nodes can be altered to allow targeted investigation of particular aspects of cognitive behaviour. One approach is for nodes to be generated dynamically while the

controlled animat goes about its activity. These dynamic NB-SMMs are *stateful* – the parameters of the nodes and mapping function change in response to the model’s internal state. Alternatively, nodes may be placed through some optimisation process while the animat is offline. These static NB-SMMs are *stateless* and always give the same output for a particular sensorimotor input. Varieties of dynamic NB-SMM models have been used to explore habit-based behaviour [5, 4] and goal-oriented behaviour [10]. Our previous work with a static NB-SMM enumerated all possible configurations of 1- and 2-node NB-SMM’s and demonstrated that even those minimal systems provided a foundation of functional behaviour [9].

In this investigation, we present an example of how a simple, static NB-SMM can be used in the context of an evolutionary robotics-style experiment. We then compare the behaviour produced by the NB-SMM-controlled animats with animats controlled by stateful continuous-time recurrent neural networks (CTRNNs) evolved to perform the same behaviour.

2 Model

2.1 NB-SMM

We use the same kind of NB-SMM that we defined in [9], which in turn uses the same functions to determine a change-in-motor-state output as a related *iterant deformable sensorimotor medium model* [5, 4].

An NB-SMM generates a map in sensorimotor space, which is a construct which defines all possible sensory and motor states of an animat, with each spatial dimension representing a single motor or sensor variable of the animat. This map defines a change-in-motor-state output for every possible sensorimotor state, and thus the controller operates by continuously outputting new change-in-motor-state commands as the animat’s sensorimotor state changes. Crucially, the controller has no internal state which modulates over time the relationship between sensorimotor state and output. In other words, the only relevant property in determining the output behaviour at any moment is the immediate state of the simulated “body”. Such a system is in contrast to a *stateful* controller such as a CTRNN-based system, in which the state of internal hidden neurons typically influence the relationship between input and output.

The mapping itself is defined in terms of *nodes* which are localized in sensorimotor space. Each node has a *position* in sensorimotor space, around which its influence is strongest, and a *velocity* component which determines the direction and speed of its influence on the change-in-motor-state of the controlled animat.¹ Finally each node has a *weight* which determines its relative influence compared to other nodes. Thus each node can be expressed as a tuple $N = \langle \vec{p}, \vec{v}, w \rangle$

This particular architecture has been chosen for the sake of consistency with related work, but the motivation for its specific design principles is of limited

¹ Note that this “velocity” of the node does not refer to a rate at which the node moves through sensorimotor space – its position is fixed.

relevance here – further discussion and explanation of the following functions may be found in [5] and [9]. The key point is that the nodes are used determine the mapping via the following function:

$$\frac{d\mu}{dt} = f(\vec{r}) = \tau \frac{\sum_N \left((\omega(N_w) d(N_{\vec{p}}, \vec{r}))^2 \cdot (N_{\vec{v}} + \Gamma(N_{\vec{p}} - \vec{r}, N_{\vec{v}}))^{\vec{\mu}} \right)}{\sum_N (\omega(N_w) d(N_{\vec{p}}, \vec{r}))} \quad (1)$$

which itself is composed of the following:

$$d(\vec{x}, \vec{y}) = \frac{2}{1 + \exp(k_d \|\vec{x} - \vec{y}\|^2)} \quad (2)$$

$$\Gamma(\vec{a}, \vec{V}) = \vec{a} - (\vec{a} \cdot \frac{\vec{V}}{\|\vec{V}\|}) \frac{\vec{V}}{\|\vec{V}\|} \quad (3)$$

$$\omega(N_w) = \frac{2}{1 + \exp(-k_w N_w)} \quad (4)$$

In these functions, $N_{\vec{p}}$ is the position in SM-space for each node, $N_{\vec{v}}$ is the velocity for each node, N_w is the weight of the node, and \vec{r} is the animat’s current position in SM-space. The superscript μ in Equation 1 indicates taking only the motor component of the vector. The fixed parameters k_d and k_w respectively scale the range of influence of all nodes in SM-space and the influence of node weight. τ scales the output relative to the animat’s velocity.

2.2 Experiment setup

The NB-SMMs are evolved to guide a animat through a task involving distinguishing between two curves, one classed as “narrow” and one as “wide”. The animat must demonstrate its ability to distinguish the curves’ widths by consistently stopping at the peak of the designated target curve (i.e. always atop the narrow curve or always atop wide). The challenge to this task is that the sensor only detects the distance to the point of the curve immediately in front of it, and thus a particular sensory state is associated with multiple points in the environment, over both curves. The animat must therefore employ an exploratory strategy over time in order to distinguish between the objects, which provides an interesting challenge for a stateless controller.

The experimental setup is illustrated in Figure 1. The environment consists of a animat in a two-dimensional arena with a stimulus in the shape of two bell-shaped curves. The arena has a width and height of size 1 and periodic boundaries on the horizontal axis. The stimulus a shape such that

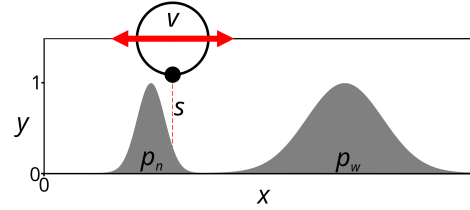


Fig. 1. Illustration of the task environment.

$$y = \max \left(\exp \left(\frac{(x - p_n)^2}{2\sigma_n^2} \right), \exp \left(\frac{(x - p_w)^2}{2\sigma_w^2} \right) \right) \quad (5)$$

where p_n and p_w are the x -positions of the centers of the narrower and wider curves and $\pm\sigma_n$ and $\pm\sigma_w$ are the maxima of the function's derivative for the narrower and wider curves. During the evolutionary process $\sigma_n = 0.03$ and $\sigma_w = 0.08$, but evolved animats are subsequently exposed to a range of widths. In each trial p_n and p_w are set randomly, with a minimum distance of 0.3 between the two to avoid significant overlap.

In each trial the animat is initially positioned with its sensor at $y = 1$ and with a random x -position. It can move along the x -axis with a velocity of v units per second such that $-0.25 \leq v \leq 0.25$. Its sensor is activated as the distance d between it and the shape at the point directly below the animat, such that its state is

$$s = 1 - d \quad (6)$$

This means that the animat's sensor state is at its maximum when it is at exactly the peak of either of the curves. The animat is controlled by an NB-SMM with a two-dimensional sensorimotor space corresponding to the single motor and single sensor. The motor state μ of the animat corresponds to its velocity but is scaled such that its value is on the interval $[0, 1]$, i.e. $\mu = 0.5$ corresponds to $v = 0$.

The parameters of the NB-SMM are defined through a genome which is optimised via a microbial genetic algorithm [6] with a population size of 100 evaluated over 220 generations with a deme size (a property specific to the microbial GA variant) of 15. The NB-SMM has 11 nodes, and the position, velocity, and weight of each node are defined in the evolutionary genome for a total of 5 genes per node. Additionally the k_d and k_w parameters are also defined in the genome. The τ parameter is fixed at $\tau = 10$. This requires a genome with $5 * 11 + 2 = 57$ genes to be evolved, where each gene is a 64-bit float from the range $[0, 1]$. The genes for k_d and k_w are scaled so that the parameter values are $2 \leq k_d \leq 20$ and $0.01 \leq k_w \leq 0.05$, and node weights are scaled so that $-300 \leq w \leq 600$. Position and velocity genes do not need scaling.

We present results for variations of the task where either the wide or narrow curve is the one that should be approached (hereafter the *approach*-curve) while the other is avoided (*avoid*-curve), and we refer to these different tasks as the wide-approach and narrow-approach variants. Each genome is tested in 108 trials lasting for 40 seconds, with the initial conditions of animat starting position and velocity selected systematically across their ranges. Trials are evaluated with a fitness function which calculates the root-mean-square error between the animat's position and the peak of the *approach* curve (either p_n or p_w), averaged over the last eight seconds of each trial, and then averaged over those 108 per-trial fitnesses. For the last 20 generations, the fitness function is adjusted such that the root-mean-square error is also multiplied by the animat's velocity. Simulations were run using Euler integration with a step size of 0.01.

2.3 CTRNN comparison

We compare the results of the NB-SMM-controlled animats with some that are controlled by minimal CTRNNs evolved to solve the same task. An explanation of CTRNNs and their use may be found in [1]. Our CTRNNs are 2-neuron networks where the first neuron receives an input and the second neuron’s state is mapped to determine the animat’s motor state. The first neuron is connected to itself, and the second is connected to itself and to the first. The input is the animat’s absolute sensor value. Note that differs from the version of the experiment presented in [3] where the CTRNN input is the time-derivative of the sensor. Ranges for biases, connection weights, and time constants are $[-32, 32]$, $[-16, 16]$, and $[0.5, 10]$ respectively. Apart from the use of CTRNNs, the experimental setup is consistent with that used with the NB-SMMs, however as the optimisation process for these experiments proved more difficult, we doubled the population and deme sizes, and the number of generations.

3 Results

3.1 NB-SMM results

An effective solution (fitness less than 0.008) was found in all 10 runs for each task variant. Wide-approach variants consistently had a superior fitness to the narrow-approach variants (0.007 versus 0.004 average fitness). All evolved NB-SMM-controlled animats display a more or less consistent behavioural strategy: When approaching either curve from one particular side (either left or right) the animat will turn back before reaching the peak, and when approaching from the other side it will pass over the peak of the *avoid* curve or come to stop at the peak of the *approach* curve. Figures 2 and 3 present visualisations of an example solution’s sensorimotor map and the phase spaces of the coupled system of the animat and its environment. Specific trajectories are highlighted on each figure for a single trial beginning from the initial conditions ($x = 0.9, \mu = 0.75$). In the sensorimotor trajectories, there are several overlapping points from which the trajectory progresses in different ways from a single state. This is possible because there are multiple states in the coupled system which produce the same sensorimotor state – when the animat is in particular positions over both the wide and narrow curve. The challenge of the task of course is that the animat must respond to these different environmental contexts appropriately, despite having a controller which reacts only to the sensorimotor state.

How do the evolved NB-SMMs solve the task? Essentially, they exploit the regularity that for any given non-zero motor state, the rate of change for the sensor state is greater as the animat passes over the narrow curve compared to the wide curve. In Figure 2 this can be seen occurring after points ψ . At each ψ we have two instances where the animat is in the same sensorimotor state, but is interacting with different curves. Furthermore the animat is in a similar position relative to each peak (i.e. to left of both in Figure 2A, and to the right of both in 2B). In other words, the divergence of the two trajectories from state

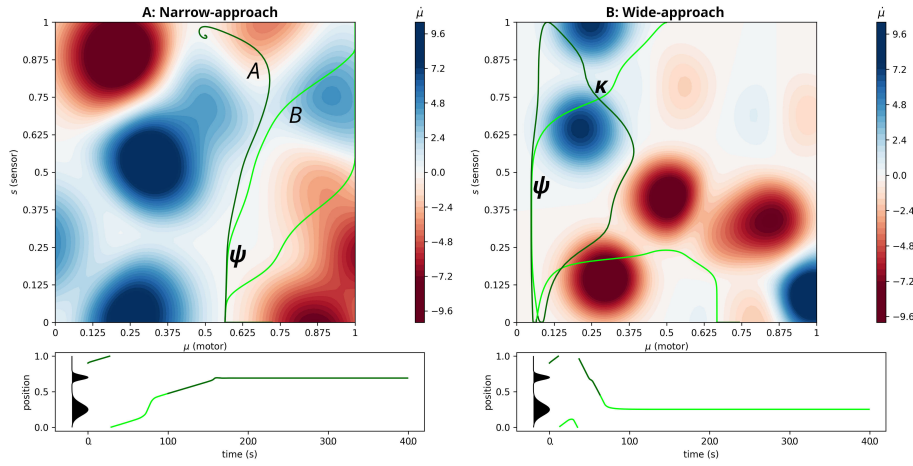


Fig. 2. Visualisations of the sensorimotor maps generated by two example NB-SMMs. For each, a sensorimotor trajectory is shown for a single trial in which the animat performs the task correctly, and a time series plot of the animats’ position during each is shown below. These same trajectories are also highlighted in Figure 3 in the coupled systems’ phase spaces. ψ , κ , A , B are discussed in the text. Note that the dark green trajectory indicates that the animat is nearer to the narrow curve than the wide.

ψ onwards is entirely a consequence of the way in which the animat interacts with different-width curves. Contrast this to point κ on Figure 2B, where the two segments of the trajectory intersect again, but the animat is on the left side of the narrow peak but on the right side of the wide peak, meaning that the difference in sensorimotor response following κ is primarily due to the contrast between moving toward a curve’s peak as opposed to moving away.

In the figures, the part of the trajectory associated with the *approach*-curve continues to approach the stable point after ψ , never intersecting with the *avoid* part again in the same way. Since the controller is stateless, it follows that the process of discriminating narrow and wide curves occurs entirely after point ψ . This is not to say that all preceding behaviour is redundant, nor the other parts of the map, which do not directly influence this aspect of the behaviour. We find that all evolved animats display a general strategy in which the animat establishes a particular sensorimotor state, (mostly) regardless of initial conditions, and from that state takes advantage of the different environmental sensory response while passing over the different curves. This can be seen in Figure 3, where trajectories from many initial conditions rapidly tend to converge.

As for the process of discriminating between the curves after ψ , how does this work within the constraints of an NB-SMM, which by definition always gives the same motor output for a particular sensorimotor state? This process is illustrated by the annotated points in Figure 2. At a given point, $\frac{d\mu}{dt}$ is consistent regardless of the environmental context, but $\frac{ds}{dt}$ varies depending on the environmental context. Therefore even as the animat makes the same motor actions, so long as

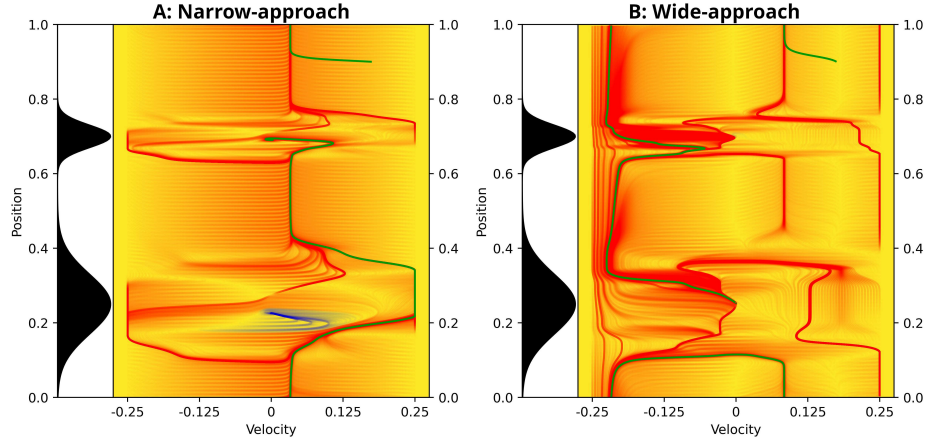


Fig. 3. Many trajectories in the phase space of the entire coupled system. Red trajectories indicate that the trajectory approaches a state which corresponds with successful task performance, while blue indicates a failure. The green trajectories match those in Figure 2.

$\frac{d\mu}{dt} \neq 0$, from a particular sensorimotor state ψ , a trajectory over a fixed time interval will arrive at different points in sensorimotor space, A and B , depending on the shape of the curve. Further, if the mapping is such that the animat always reaches state ψ , or at least approximates it, then it is guaranteed that the animat will only reach state A when it is over the narrow curve, and state B when it is over the wide curve. Ultimately, the NB-SMM’s mapping can take advantage of this by producing different motor activity for states A and B . As appropriate, one of these states can lead to the end of the *approach* part of the behaviour (i.e. come to a stop), while the other can lead to the *avoid* part of the behaviour (i.e. move away from the current curve). To take advantage of these regularities, a sensorimotor map that can solve this task must have two parts: One part of the mapping ensures that there is a region of sensorimotor space such that when the animat’s sm-state is in that region it will move to stop at the peak of the currently sensed curve; The rest of the mapping ensures that the animat’s sensorimotor state will only enter that region in the correct context for the given task. As seen in Figure 3, there is a caveat to this solution, in which the animat will fail to solve the task if it begins with initial conditions which violate the aforementioned guarantee about states A and B .

Although the general strategy is consistent, we observe a fundamental difference between the two task variants in the behaviour which occurs after ψ . In the wide-approach case, the animat immediately decelerates toward an oscillation around $v = 0$. In the narrow-approach case however, the animat first accelerates before decelerating. These differences can be seen in Figure 2 and are consistent across all evolved solutions. This highlights the peculiarities of the relationship between sensorimotor maps and the specific dynamics of each task variant. Why does this behavioural distinction develop? As we have established,

the total change in motor state will always be greater in the case of passing over the wider curve compared to passing the narrower. In the wide-approach variation, this means that an effective animat can simply decelerate from its ψ state until it approaches $v = 0$ around the same moment that it reaches the peak of the curve. The same map will cause the animat to pass over the top of the narrow curve before it has reached $v = 0$, as occurs in Figure 2B. In contrast, the narrow-approach variation uses an acceleration to distinguish the curves. When it passes over the narrow curve it accelerates slightly, but then decelerates as it passes over the top and eventually reverses before coming to a stop. When the same animat passes over the wide curve however, it accelerates more, such that it avoids the region of sensorimotor state where the mapping causes it to decelerate and double back. That the narrow-approach variation requires both an acceleration and a deceleration suggests that there is an added degree of complexity to the narrow-approach variation compared to the wide-approach.

3.2 Categorical Perception

The NB-SMMs are only exposed to a particular pair of curve widths during evolution. However the evolved animats also display an ability to respond to various pairings of widths, tested between $0.01 < \sigma_N < 0.065$ and $0.05 < \sigma_W < 0.12$. This ability produces emergent categories of “wide” and “narrow” curves defined in terms of how the animats respond to each. The boundaries of these categories are not objective, but rather they vary from animat to animat depending on the precise dynamics of each map. Figure 4 illustrate the fitness of narrow-approach and wide-approach animats across a range of width pairings. We can see that there are thresholds within which the animat’s task performance is near perfect, but beyond those thresholds there are regions where the animat’s fitness is lower

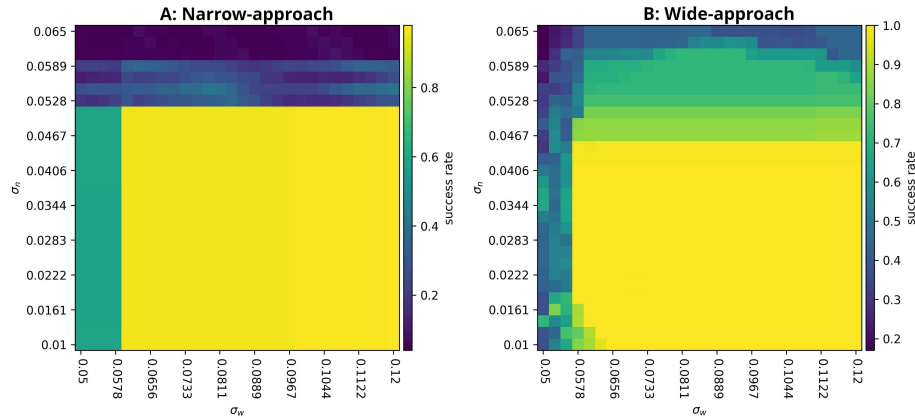


Fig. 4. Frequency with which the animat ends a trial above the approach-curve’s peak for different pairings of curve. Each pairing is systematically evaluated across 900 different initial conditions.

but not indicative of complete task-failure. There are two factors which cause the fitness of the animat to drop off past those limits. Firstly, the set of initial conditions which cause the animat to incorrectly stop atop the avoid curve becomes larger. This is because, as the widths become more similar, the deviation between trajectories after ψ become less pronounced, and this means that more initial states fall within the conditions which the NB-SMM has implicitly established as states which should only be reached when over the approach-curve. Eventually the *avoid* curve becomes so similar to the original approach curve that the animat will always stop at the avoid peak if it encounters it first. In the narrow-approach variant, this limit establishes the lower bound of “wide” curves, and similar establishes the upper bound of “narrow” in the wide-approach variant. Secondly, when the *approach* curve width breaches its own limit, the animat will no longer stop at the correct peak. However it will typically still move relatively slowly near the peak and therefore spend a larger amount of time in that area. Therefore depending on initial conditions, the animat is still likely to be near the peak of the approach curve at the end of the trial, but may also have moved away again. These two factors leads to the fuzziness of the success rates outside of the yellow regions of the plot which describe correct behaviour.

We can see that in the example narrow-approach case, the upper limit for a curve to be considered “narrow” is around $\sigma_n = 0.051$, whereas the lower bound for a “wide” curve is around $\sigma_w = 0.06$. When curves are in between those limits, the animat’s behaviour is heavily dependant on initial conditions. Similarly for the wide-approach example, the upper and lower bounds are $\sigma_n = 0.045$ and $\sigma_w = 0.056$ respectively.

3.3 CTRNN results

Figure 5 shows plots equivalent to Figure 3 for two examples of CTRNN-controlled animats, illustrating the behaviour in terms of the animat’s position and velocity. For the wide-approach variant, a two-node CTRNN which performed with comparable fitness to the NB-SMM was found consistently. The general strategy of the CTRNN-controlled animats align with that of the NB-SMM version. That is, it approaches from one side and decelerates as it passes over each curve, such that the deceleration brings it to a stop over the wide curve but not the narrow. The overall pattern of behaviour is much simpler than that of the NB-SMM-controlled animat, with the animat rapidly achieving a maximum leftward-velocity from most initial conditions.

The narrow-approach variant failed to converge on an effective solution in 10 runs. This failure is consistent with what we observed in the NB-SMM animats regarding the need for both an acceleration and deceleration in the performance of the of the narrow curve variant – resulting in a slightly more complex task – and the relative simplicity of the function approximated by the CTRNN. This result provides a contrast to that in [3], in which a time-derivative of the sensor state is used as the input to a CTRNN with the same topology, producing an effective solution with an oscillatory behaviour.

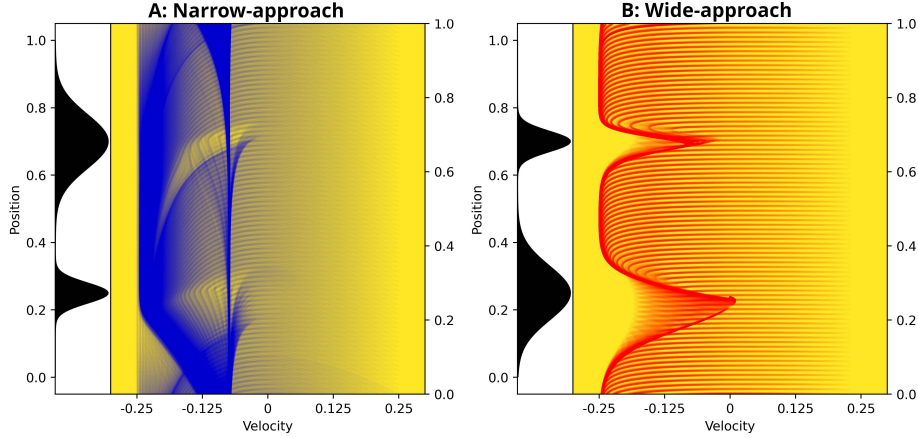


Fig. 5. For two evolved CTRNNs, trajectories of the same variables as plotted in Figure 3. Note that in these systems there is an additional variable in the internal state of the first neuron, which is not plotted here. In the narrow-approach case, the evolutionary algorithm has failed to find an effective solution.

For the wide-approach variant, Figure 6 illustrates the performance over various width pairs. Unlike the NB-SMM version, there is essentially no gap between the upper bound of the perceived “narrow” curve and lower bound of the “wide” curve.

4 Discussion

At first glance, it may seem surprising that the NB-SMM-controlled animats are capable of performing the task. On the one hand, we have a task in which the immediate sensorimotor information that is available to the animat is insufficient to distinguish between the widths of the curve. On the other, we have a reactive controller model whose output is entirely determined by that immediate sensorimotor information. The crux of the matter is that animat’s motor state develops over time in such a way that it reflects the history of the time-extended perception of the curve shape. This does not happen by accident – it requires the animat to make particular movements at particular times for the motor state to play a useful role in the task’s fulfillment. This kind of investigation, in which sensorimotor dynamics

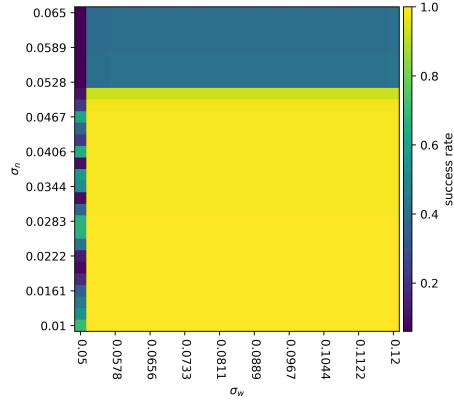


Fig. 6. Fitness results for the CTRNN wide-approach task, equivalent to Figure 4.

are simulated in isolation from neural dynamics, emphasizes the importance of embodiment in this kind of cognitive process.

Braitenberg’s *Vehicles* [2] served as an example of the way in which extremely primitive models could display behaviour with a surprising resemblance to cognitive behaviour. An NB-SMM model can be seen as an intermediate point between the Braitenberg vehicles and a stateful controller like a CTRNN: While the Braitenberg vehicle’s behaviour is a function of the sensor state (i.e. the environment); The NB-SMM-controlled animat’s behaviour is a function of the sensor and motor state (i.e. environment and body); and the CTRNN-controlled animat’s behaviour is a function of the sensor, motor, and internal state (i.e. environment, body, and brain). Exploring the behaviours that are possible with the NB-SMM, and those that are not, is a method for understanding the exact roles of the body and the brain in the context of adaptive behaviour and embodied cognition. Buhrmann, et al. analysed a CTRNN-controlled animat performing another variant of this task [3], and part of their discussion highlighted the role of the internal state of the hidden neuron which causes the system to alternate between an approach regime and an avoid regime. In this context, the controller’s internal state played a role analogous to a nervous system which modulates the sensorimotor response in accordance with the agent’s goal. By the same token, the NB-SMM model can be interpreted as simulating a system which lacks a nervous system, but which nonetheless displays the same goal-oriented property.

The difference between the behaviour of the NB-SMM and CTRNN versions, illustrated in Figures 3 and 5, demonstrate that the NB-SMM has some advantages over the more typical model, even in an evolutionary robotics context. In particular, the map’s complexity may be increased by adding nodes, without increasing the dimensionality of the entire system’s state space. Meanwhile, the CTRNN-based system with a three dimensional state space produced relatively simple behavioural patterns. The value of the NB-SMM in this case is demonstrated in the NB-SMM’s ability to produce a solution for the narrow-approach variant where the CTRNN did not. At the same time, system’s behaviour is relatively easy to visualize and interpret. In order for a CTRNN to match the complexity of the behavioural patterns generated by the NB-SMM it would need more neurons, thereby increasing the number of variables in the system and reducing its interpretability.

Finally, the difference in fitness and behavioural patterns between the two task variants is an unintuitive outcome. Each variant would seem to be simple inversions of each other, with similar performance expected. This is perhaps a misleading aspect of describing the task in functionalist terms such as identifying the curve type and moving to the top of one – it would seem to follow from this that it would be an equivalent process to identify the curve type and move to the top of the other one. However although it is attractive to interpret the behaviour of the animat as first distinguishing the curve type (i.e. moving into a particular sensorimotor state) and then responding appropriately (slowing to a stop or passing over the peak), these delineations only serve to aid the description of what is in practice a single continuous act. Over the course of this act, the

sensorimotor dynamics associated both with distinguishing the curves, and with traversing to the peak, are intertwined. The way that the two properties of the task description interact, i.e. which curve to approach, and how a successful approach is measured, appears to have made the wide-approach variant simpler than the other.

A consequence of this in terms of modelling is that seemingly trivial decisions of task specification have the potential to impact the final behaviour of the model, and this raises issues of how abstractions of sensorimotor dynamics affect our ability to extrapolate our results to natural systems. Consider for example the use of wheeled-animat styles of models as abstractions of moving organisms. It is critical to the performance of the two-curves task that the animat be altering its motor speed as it passes over each curve, so that it can use its motor state as a proxy for the time it has spent over the curve. A wheeled-robot style animat has a particular type of relationship between its motor and sensor dynamics, which would be different from that of, say, a more naturalistic legged-robot. Would such an animat be able to utilise its motor state in the same way as the one presented here?

References

1. Randall D Beer. On the Dynamics of Small Continuous-Time Recurrent Neural Networks. *Adaptive Behavior*, 3(4):469–509, March 1995.
2. Valentino Braitenberg. *Vehicles, experiments in synthetic psychology*. MIT Press, Cambridge, Mass., 1984.
3. Thomas Buhrmann, Ezequiel Alejandro Di Paolo, and Xabier Barandiaran. A Dynamical Systems Account of Sensorimotor Contingencies. *Frontiers in Psychology*, 4, May 2013.
4. Matthew Egbert and Lola Cañamero. Habit-Based Regulation of Essential Variables. In *Artificial Life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 168–175, New York, NY, United States, 2014.
5. Matthew D. Egbert and Xabier E. Barandiaran. Modeling habits as self-sustaining patterns of sensorimotor behavior. *Frontiers in Human Neuroscience*, 8(August):1–15, 2014.
6. Inman Harvey. The Microbial Genetic Algorithm. In George Kampis, István Karsai, and Eörs Szathmáry, editors, *Advances in Artificial Life. Darwin Meets von Neumann*, volume 5778, pages 126–133. Springer, Berlin, Heidelberg, 2011.
7. Anil K Seth. Evolving action selection and selective attention without action, attention, or selection. *Proceedings of the 5th Conference on the Simulation of Adaptive Behavior*, pages 1–8, 1998.
8. David Silverman. Bodily skill and internal representation in sensorimotor perception. *Phenomenology and the Cognitive Sciences*, 17(1):157–173, February 2018.
9. Felix M. G. Woolford and Matthew D Egbert. Behavioural variety of a node-based sensorimotor-to-motor map. *Adaptive Behavior*, (Special issue on Approaching Minimal Cognition), April 2019.
10. Felix M. G. Woolford and Matthew D. Egbert. Goal oriented behavior with a habit-based adaptive sensorimotor map network. *Frontiers in Neurobotics*, 16, 2022.