

Cognitive Supervisor for an Autonomous Swarm of Robots

Vladimir G. Ivancevic, Darryn J. Reid

Decision Sciences, Joint and Operations Analysis Division, Defence Science & Technology Group, Canberra, Australia

Email: Vladimir.Ivancevic@dsto.defence.gov.au, Darryn.Reid@dsto.defence.gov.au

How to cite this paper: Ivancevic, V.G. and Reid, D.J. (2017) Cognitive Supervisor for an Autonomous Swarm of Robots. *Intelligent Control and Automation*, 8, 44-65.

<https://doi.org/10.4236/ica.2017.81004>

Received: January 16, 2017

Accepted: February 21, 2017

Published: February 24, 2017

Copyright © 2017 by authors and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

As a sequel to our recent work [1], in which a control framework was developed for large-scale joint swarms of unmanned ground (UGV) and aerial (UAV) vehicles, the present paper proposes cognitive and meta-cognitive supervisor models for this kind of distributed robotic system. The cognitive supervisor model is a formalization of the recently Nobel-awarded research in brain science on mammalian and human path integration and navigation, performed by the hippocampus. This is formalized here as an adaptive Hamiltonian path integral, and efficiently simulated for implementation on robotic vehicles as a pair of coupled nonlinear Schrödinger equations. The meta-cognitive supervisor model is a modal logic of actions and plans that hinges on a weak causality relation that specifies when atoms may change their values without specifying that they must change. This relatively simple logic is decidable yet sufficiently expressive to support the level of inference needed in our application. The atoms and action primitives of the logic framework also provide a straightforward way of connecting the meta-cognitive supervisor with the cognitive supervisor, with other modules, and to the meta-cognitive supervisors of other robotic platforms in the swarm.

Keywords

Autonomous Robotic Swarm, Cognitive Supervisor, Hippocampus Path Integration and Navigation, Hamiltonian Path Integral, Modal Logic, Nonlinear Schrödinger Equation, Reasoning about Actions and Plans

1. Introduction

Recently, we have proposed in [1] a rigorous model for prediction and control of a large-scale joint swarm of *unmanned ground vehicles* (UGVs) and *unmanned aerial vehicles* (UAVs), performing an autonomous land-air operation. In that paper, we have also introduced a need for a *cognitive supervisor* for the high-

dimensional distributed multi-robotic system. Its primary task is to have a birds-eye view of the situation across the joint land-air operation, and based on the GPS locations of both the target and all included robots, to provide them with good 2D and 3D attractor fields so that they can reach the proximity of the target in the shortest possible time. The purpose of the present paper is to develop a rigorous model for this cognitive supervisor, based on recent discoveries in brain science that show us how humans navigate in 2D environments and how bats navigate in 3D environments, and to couple this with a meta-cognitive supervisor model that allows vehicles to reason about actions and construct simple plans.

The 2014 Nobel Prize in Physiology or Medicine was awarded jointly to John O’Keefe, May-Britt Moser and Edvard I. Moser “for their discoveries of cells that constitute a positioning system in the brain”, in other words, for the *hippocampus path integration and navigation system*.

Briefly, there is a part of a mammal brain, called the *hippocampal formation*, which in humans is mostly developed in taxi drivers, grows in size with their experience and can be also trained (like a muscle) using fast-action video games. The hippocampal formation provides a *cognitive map* of a familiar environment which can be used to identify one’s current location and to navigate from one place to another. This *mapping system* provides two independent strategies for locating places, one based on *environmental landmarks* and the other on a *path integration system* (see [2] [3] and the references therein), which uses information about distances traveled in particular directions. This brain navigation system exists in all mammals, while in humans it additionally provides the basis for the so-called *episodic memory* [4].¹

Two main components of the hippocampal formation (discovered by O’Keefe) are: (i) hippocampal place cells, and (ii) grid cells from the entorhinal cortex (discovered by Mosers). In particular, according to Edvard Moser, “All network models for grid cells involve continuous attractors ...”—which is similar to our attractor Hamiltonian dynamics of UGVs and UAVs, given by Equations (1)-(2) in the next section.

As inspired by this discovery in brain science, the present paper proposes a novel probabilistic spatio-temporal model for mammalian path integration and navigation, formulated as an *adaptive Hamiltonian path integral*. The model combines: (i) a cognitive map $p(t)$ performed by hippocampal place cells, (ii) an entorhinal map $q(t)$ performed by grid cells, (iii) a current of sensory (extra-hippocampal) stimuli $J(t)$, and (iv) Hebbian learning in hippocampal synaptic weights $w(t)$. This model represents an infinite-dimensional neural network, which can be simulated (using $10^6 - 10^7$ neurons) on IBM’s TrueNorth chip.

We also propose to couple this cognitive supervisor to a meta-cognitive supervisor, supporting dynamic mission planning, using on an established propositional multimodal logic framework. This approach gives robotic vehicles the

¹For technical details on the Nobel awarded work of John O’Keefe, May-Britt Moser and Edvard I. Moser, see [Appendix](#) and the references therein.

ability to construct and execute simple plans on the fly against goals, given local sensor information, state information communicated locally between vehicles, and aspects of the state of the robot itself. The coupling to the cognitive supervisor and to the outside world is through the truth value of logical atoms and using multimodal actions.

2. Affine Hamiltonian Control for a Joint (UGV + UAV) Swarm

The affine Hamiltonian control model with many degrees-of-freedom has been presented in the form of $2n$ -dimensional (2ND) Langevin-type attractor matrix equations with nearest-neighbor couplings, which represent two recurrent neural networks:

UGV-swarm:

$$\begin{aligned} \dot{q}_{jk} &= \varphi^{2D} \left(q_A^{2D} - q_{jk} - \omega_{jk} q_{jk}^2 p_{jk} \right) - \sum_{j,k} \partial_{p_{jk}} H_{jk} u_{jk} + \eta_q(t), \\ \dot{p}_{jk} &= \varphi^{2D} \left(p_A^{2D} - p_{jk} - \omega_{jk} p_{jk}^2 q_{jk} \right) + \sum_{j,k} \partial_{q_{jk}} H_{jk} u_{jk} + \eta_p(t). \end{aligned} \tag{1}$$

UAV-swarm:

$$\begin{aligned} \dot{q}_{jkl} &= \varphi^{3D} \left(q_A^{3D} - q_{jkl} - \omega_{jkl} q_{jkl}^2 p_{jkl} \right) - \sum_{j,k,l} \partial_{p_{jkl}} H_{jkl} u_{jkl} + \eta_q(t), \\ \dot{p}_{jkl} &= \varphi^{3D} \left(p_A^{3D} - p_{jkl} - \omega_{jkl} p_{jkl}^2 q_{jkl} \right) + \sum_{j,k,l} \partial_{q_{jkl}} H_{jkl} u_{jkl} + \eta_p(t). \end{aligned} \tag{2}$$

The following terms are used in Equations (1) and (2): $q_{jk} = q_{jk}(t)$ and $p_{jk} = p_{jk}(t)$ are time-evolving matrices defining coordinates and momenta of the UGV-swarm, respectively, with initial conditions: $q_{jk}(0)$ and $p_{jk}(0)$. Similarly, $q_{jkl} = q_{jkl}(t)$ and $p_{jkl} = p_{jkl}(t)$ are time-evolving tensors defining coordinates and momenta of the UAV-swarm, respectively, with initial conditions: $q_{jkl}(0)$ and $p_{jkl}(0)$. q_A^{2D} and p_A^{2D} are the $2D$ attractors for the UGV swarm, while q_A^{3D} and p_A^{3D} are the $3D$ attractors for the UGV swarm. φ^{2D} and φ^{3D} are the attractor field strengths for the UGV and UAV swarms, respectively; ω_{jk} and ω_{jkl} are corresponding adaptive weights of both swarms which can be trained by Hebbian learning, (q_{jk}, p_{jk}) and (q_{jkl}, p_{jkl}) are the initial formations of both swarms, u_{jk} and u_{jkl} are Lie-derivative controllers for both swarms, H_{jk} and H_{jkl} are affine Hamiltonians of both swarms, while $\eta_q(t)$ and $\eta_p(t)$ are zero-mean, delta-correlated, Gaussian white noises added to q and p variables in both swarms. For more technical details on affine Hamiltonian (or, similar, *port-Hamiltonian*) control of large-scale dynamical systems, see [1] and the references therein.

The purpose of the cognitive supervisor is to provide the 2D and 3D inputs to the recurrent neural nets (1)-(2), or specifically, 2D attractors (q_A^{2D}, p_A^{2D}) for the UGV-swarm and 3D attractors (q_A^{3D}, p_A^{3D}) for the UAV-swarm (see, e.g. [5]).

3. Adaptive Hamiltonian Path Integral

In his recent Nobel lecture, John O’Keefe referred to his pioneering 1976-paper [6] [7], describing the function of the *hippocampal place cells* performing Tolman’s *cognitive mapping*: “When an animal had located itself in an environment

(using environmental stimuli) the hippocampus could calculate subsequent positions in that environment on the basis of how far and in what direction the animal had moved in the interim ...” This quotation was accompanied by a *commutative diagram* depicting the *vector addition* and a suggestion that an *animal moves in a sequence of vectors*.

This extract from O’Keefe’s lecture is the motivation for the present mathematical model. Basically, any two-dimensional (2D) vector is equivalent to a complex number: $z = x + iy$, where (x, y) are Cartesian coordinates and $i = \sqrt{-1}$. The same complex number z can be also given in the polar form as: $z = re^{i\theta}$, where r is the radius vector and θ is the heading. The sequence of N vectors is the sum of complex numbers:

$$\text{particle-like animal motion} = \sum_{k=1}^N z_k = \sum_{k=1}^N x_k + iy_k = \sum_{k=1}^N r_k e^{i\theta_k}. \tag{3}$$

In this way, we can describe a particle-like animal motion in the complex plane \mathbb{C} , from some *initial point* A to the *final point* B , performed in N steps, as an integral complex number (3). This basic idea describes an animal’s motion in purely static and deterministic terms; it can be generalized into a more realistic, *probabilistic dynamics*, as follows.²

Now, instead of the complex plane \mathbb{C} , consider a particle-like animal motion in the *phase plane* $(p-q)$, where $p = p(t, x)$ represents the action of the hippocampal place cells and $q = q(t, x)$ defines the action of the entorhinal grid cells. The animal moves from some initial point A given by canonical coordinates (q_0, p_0) at initial time t_0 , to the final point B given by canonical coordinate (q_1, p_1) at final time t_1 , via *all possible paths*, each path having an equal probability (so that the sum of all path-probabilities is = 1). This most general 2D motion is properly defined by the *transition amplitude*:

$$\langle B|A \rangle \equiv \langle q_1, p_1, t_1 | q_0, p_0, t_0 \rangle$$

whose absolute square represents the *transition probability density* function:

$$PDF = |\langle B|A \rangle|^2 \equiv |\langle q_1, p_1, t_1 | q_0, p_0, t_0 \rangle|^2.$$

The transition amplitude $\langle B|A \rangle$ can be calculated via the following *Hamiltonian path integral*³

²For For simplicity reasons, we are using the same Hamiltonian symbols, q and p , for the cognitive representation of robotic coordinates and momenta, to emphasize the one-to-one correspondence between the physical robotic level and the mental supervisor level. However, while at the physical robotic level, $q = q(t)$ and $p = p(t)$ are only temporal variables, at the cognitive level, $q = q(t, x)$ and $p = p(t, x)$ represent spatiotemporal wave functions.

³The path integral (4) was formulated by R. Feynman in [10]. It has been widely appreciated that the phase-space (*i.e.*, Hamiltonian) path integral is more generally applicable, or more robust, than the original, Lagrangian version of the path integral, introduced in Feynman’s first paper [11]. For example, the original Lagrangian path integral is satisfactory for Lagrangians of the form:

$L(x) = \frac{1}{2}m\dot{x}^2 + A(x)\dot{x} - V(x)$, but it is unsuitable, e.g., for the case of a particle with the Lagrangian

(in normal units): $L(x) = -mqr\dot{1} - \dot{x}^2$. For such a system (as well as many more general expressions) the Hamiltonian path integral is more robust; e.g., the Hamiltonian path integral for the free particle: $\int \mathcal{D}[p] \mathcal{D}[q] \exp\{i \int [p\dot{q} - qrp^2 + m^2] dt\}$ is readily evaluated.

$$\langle q_1, p_1, t_1 | q_0, p_0, t_0 \rangle = \int \mathcal{D}[q] \mathcal{D}[p] \exp \left\{ i \int_{t_0}^{t_1} [p(\tau) \dot{q}(\tau) - H(p, q)] d\tau \right\}, \tag{4}$$

$$\text{with } \int \mathcal{D}[q] \mathcal{D}[p] \equiv \frac{1}{2\pi} \int \prod_{\tau} dq(\tau) dp(\tau),$$

where the integration is performed over the $p(\tau)$ and $q(\tau)$ values at every time τ , with the time-step $d\tau \equiv t_j - t_{j-1}$ and the velocity $\dot{q}(\tau) \equiv \frac{q(t_j) - q(t_{j-1})}{t_j - t_{j-1}}$.

For technical details of the derivation of the Hamiltonian path integral (4), see e.g. [8] [9] and the references therein.

Next, the *sources* from various extra-hippocampal stimuli can be incorporated into the basic transition amplitude (4) by adding some form of a bio-electric current J , as:

$$Z[J] = \int \mathcal{D}[q] \mathcal{D}[p] \exp \left\{ i \int_{t_0}^{t_1} [p(\tau) \dot{q}(\tau) - H(p, q) + J(\tau) q(\tau)] d\tau \right\}, \tag{5}$$

where $Z[J] = \langle q_1, p_1, t_1 | q_0, p_0, t_0 \rangle$ is the system's *partition function* dependent on the current J and obeying the *normalization condition*: $Z[J = 0] = 1$.

A generalization from a single particle Hamiltonian path integral (4) to the probabilistic dynamics of an N -particle system is straightforward. The phase-space functional integral that defines the transition amplitude

$\langle q_1^1 \cdots q_1^N, p_1^1 \cdots p_1^N, t_1 | q_0^1 \cdots q_0^N, p_0^1 \cdots p_0^N, t_0 \rangle$, from the initial MD point $(q_0^1 \cdots q_0^N)$ at time t_0 to the final MD point $(q_1^1 \cdots q_1^N)$ at time t_1 is given by:

$$\langle q_1^1 \cdots q_1^N, p_1^1 \cdots p_1^N, t_1 | q_0^1 \cdots q_0^N, p_0^1 \cdots p_0^N, t_0 \rangle$$

$$= \int \mathcal{D}[q] \mathcal{D}[p] \exp \left\{ i \int_{t_0}^{t_1} \left[\sum_{k=1}^N p_k(\tau) \dot{q}^k(\tau) - H(p_k, q^k) \right] d\tau \right\},$$

$$\text{with } \int \mathcal{D}[q] \mathcal{D}[p] = \frac{1}{2\pi} \int \prod_{\tau} \prod_{k=1}^N dq^k dp_k,$$

where we are allowing for the full Hamiltonian of the system $H(p_k, q^k)$ to depend upon all the N coordinates $q^k(\tau)$ and momenta $p_k(\tau)$ collectively.

Again, we can add various sources as incoming bio-currents J_k as a straightforward generalization of the single-particle partition function (5) to the system of N particles:

$$Z_N[J_k] = \int \mathcal{D}[p] \mathcal{D}[q] \exp \left\{ i \int_{t_0}^{t_1} \left[\sum_{k=1}^N p_k(\tau) \dot{q}^k(\tau) - H(p_k, q^k) + J_k(\tau) q^k(\tau) \right] d\tau \right\},$$

where $Z_N[J_k] = \langle q_1^1 \cdots q_1^N, p_1^1 \cdots p_1^N, t_1 | q_0^1 \cdots q_0^N, p_0^1 \cdots p_0^N, t_0 \rangle$ is the system's *partition function* dependent on all the incoming currents J_k and obeying the normalization condition: $Z_N[J_k = 0] = 1$, for $k = 1, \dots, N$.

Our final step is to transform the N -particle partition function $Z_N[J_k]$ into an infinite-dimensional recurrent neural network (of a generalized Hopfield type) by including the hippocampal synaptic weights $w_k(\tau)$ into it (compare with [12]), as:

$$Z_N [w_k, J_k] = \int \mathcal{D}[w] \mathcal{D}[q] \mathcal{D}[p] \exp \left\{ i \int_{t_0}^{t_1} \left[\sum_{k=1}^N p_k(\tau) \dot{q}^k(\tau) - H(p_k, q^k) + J_k(\tau) q^k(\tau) \right] d\tau \right\}, \quad (6)$$

$$\text{with } \int \mathcal{D}[w] \mathcal{D}[q] \mathcal{D}[p] = \frac{1}{2\pi} \int \prod_{\tau} \prod_{k=1}^N dw_k dq^k dp_k,$$

where the weights $w_k(\tau)$ are adapted (in a discrete time) by Hebbian-type learning:

$$w_k(\tau + 1) = w_k(\tau) + \frac{q}{\eta} [w_k^D(\tau) - w_k^A(\tau)], \quad (7)$$

where $q = q(\tau)$, $\eta = \eta(\tau)$ represent local *signal* and *noise* amplitudes, respectively, while superscripts *D* and *A* denote *desired* and *achieved* system states, respectively.

The system (6-7) defines the proposed adaptive Hamiltonian path integral model for a generic mammalian path integration and navigation. Both the sequential (Ising spin) Hopfield network [13] with its Galuber dynamics and the graded-response Hopfield network [14] with its Fokker-Planck dynamics can be considered as special cases of this general Hamiltonian neural system.

Direct computer simulations of the adaptive Hamiltonian path integral system (6-7) can be performed on the IBM TrueNorth chip (see [15] and the references therein) as a *Markov-chain Monte Carlo simulation over a grid of Hopfield nets* (which are already implemented in the TrueNorth chip, using 10^6 - 10^7 artificial neurons).

In the next section we propose a more efficient approach to simulate the path integral system (6-7).

4. A Pair of Coupled Nonlinear Schrödinger Equations

In this section, instead of the direct computer simulations on a supercomputer, we will present an indirect approach of simulating the path integral system (6-7) on an ordinary PC, represented as a pair of coupled nonlinear Schrödinger (NLS) equations. In his first paper [11], Feynman showed that his Lagrangian q -path integral was equivalent to the standard linear Schrödinger equation from quantum mechanics, given here for the case of a free particle (in natural physical units: $\hbar = 1$, $m = 1$):

$$i \partial_t \psi(t, x) = -\frac{1}{2} \partial_{xx} \psi(t, x), \quad \left(i = \sqrt{-1}; \text{ with } \partial_z \psi = \frac{\partial \psi}{\partial z} \right), \quad (8)$$

which defines the complex-valued *microscopic wave function* $\psi = \psi(t, x)$, whose absolute square $|\psi(x, t)|^2$ defines the *probability density function* (PDF).

In the last decade it was shown (see [16] and the references therein) that if the linear Schrödinger Equation (8) is put into an adaptive (iterative) feedback loop, it adds a cubic nonlinearity with a potential field $V = V(x)$ and becomes the NLS equation:

$$i \partial_t \psi = -\frac{1}{2} \partial_{xx} \psi + V |\psi|^2 \psi, \quad (9)$$

which now defines the *macroscopic wave function* $\psi = \psi(t, x)$ whose absolute square $|\psi(x, t)|^2$ still defines the PDF [17]. A variety of analytical solutions for the NLS Equation (9) have been reported in [18] [19].

Finally, to represent the Hamiltonian (q, p) -path integral, we can use the (q, p) -pair of NLS equations, as follows.

4.1. Special Case: Analytical Soliton

We start with a simple (q, p) -NLS pair representation for the path integral system (6-7), which admits the analytical closed-form solution, given by the so-called *Manakov system* (with the constant potential V):⁴

$$i\partial_t q = -\frac{1}{2}\partial_{xx}q + V(|q|^2 + |p|^2)q, \tag{10}$$

$$i\partial_t p = -\frac{1}{2}\partial_{xx}p + V(|q|^2 + |p|^2)p, \tag{11}$$

which was proven in [20], using the *Lax pair representation* [21], to be completely integrable Hamiltonian system, by the existence of infinite number of involutive integrals of motion.

The (q, p) -NLS pair (10)-(11) admits both “bright” and “dark” soliton as solutions, of which the simplest one is the so-called *Manakov bright 2-soliton* given by:

$$\begin{bmatrix} q(t, x) \\ p(t, x) \end{bmatrix} = 2b \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \operatorname{sech}[2b(x + 4at)] e^{-2i(2a^2t + ax - 2b^2t)}, \tag{12}$$

where a and b are real-valued parameters and $|c_1|^2 + |c_2|^2 = 1$.

4.2. General Case: Numerical Simulation

Now that we have introduced the simple (q, p) -NLS pair, we can define our real representation for the path integral system (6-7), as the following more general (q, p) -NLS pair:

$$i\partial_t q(t, x) = -\frac{1}{2}w_k I(t, x)\partial_{xx}q(t, x) + V(t, x)|q(t, x)|^2 p(t, x), \tag{13}$$

$$i\partial_t p(t, x) = -\frac{1}{2}w_k J(t, x)\partial_{xx}p(t, x) + U(t, x)|p(t, x)|^2 q(t, x), \tag{14}$$

including the bell-shaped (sech) spatiotemporal potentials:

$$V(t, x) = \frac{1}{2}\operatorname{sech}(a_1 t x^3), \quad U(t, x) = \frac{1}{2}\operatorname{sech}(a_2 t^3 x),$$

and the soft-step shaped (tanh) spatiotemporal input currents:

$$I(t, x) = a_3 \tanh(a_4 t^3 x), \quad J(t, x) = a_5 \tanh(a_6 t x^3),$$

together with the common initial condition:

$$\chi(x) = \frac{1}{2}\exp(ix)\operatorname{sech}\left(\frac{1}{2}x\right),$$

⁴The Manakov system has been used to describe the interaction between wave packets in dispersive conservative media, and also the interaction between orthogonally polarized components in nonlinear optical fibres (see, e.g. [22] [23] and the references therein).

the set of parameters:

$$(i = \sqrt{-1}, a_1 = 0.4, a_2 = 0.6, a_3 = 0.3, a_4 = 0.8, a_5 = 0.7, a_6 = 0.2)$$

and the set of adaptive synaptic weights w_k .

The (q, p) -NLS pair (13)-(14) has been numerically simulated in mathematica®, producing 3D plots of real and imaginary parts of the (q, p) -wave functions (see Figure 1) and density plots of attractor fields for robotic swarms (see Figure 2), using the following code:

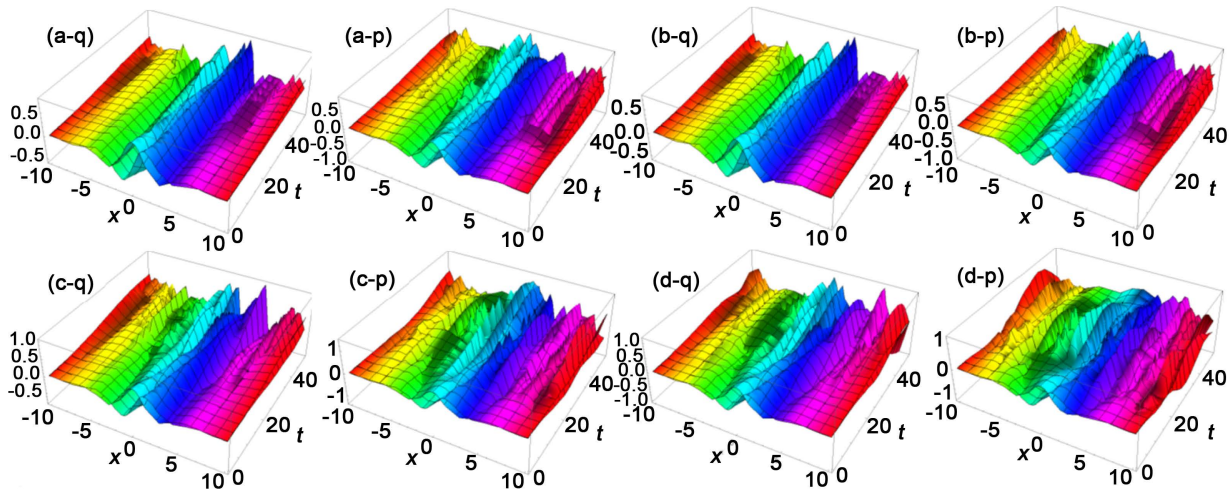


Figure 1. Simulation of the (q, p) -NLS pair (13)-(14) in mathematica®, showing the adaptation of the (q, p) -waves with the change of the synaptic weights $w_k \in [0,1]$. Each 3D plot shows the following two surfaces (real and imaginary values of the wave function): (a-q) is the q -wave plot at $w_1 = 0.1$, (a-p) is the p -wave plot at $w_1 = 0.1$; (b-q) is the q -wave plot at $w_2 = 0.3$, (b-p) is the p -wave plot at $w_2 = 0.3$; (c-q) is the q -wave plot at $w_3 = 0.5$, (c-p) is the p -wave plot at $w_3 = 0.5$; (d-q) is the q -wave plot at $w_4 = 0.7$, (d-p) is the p -wave plot at $w_4 = 0.7$.

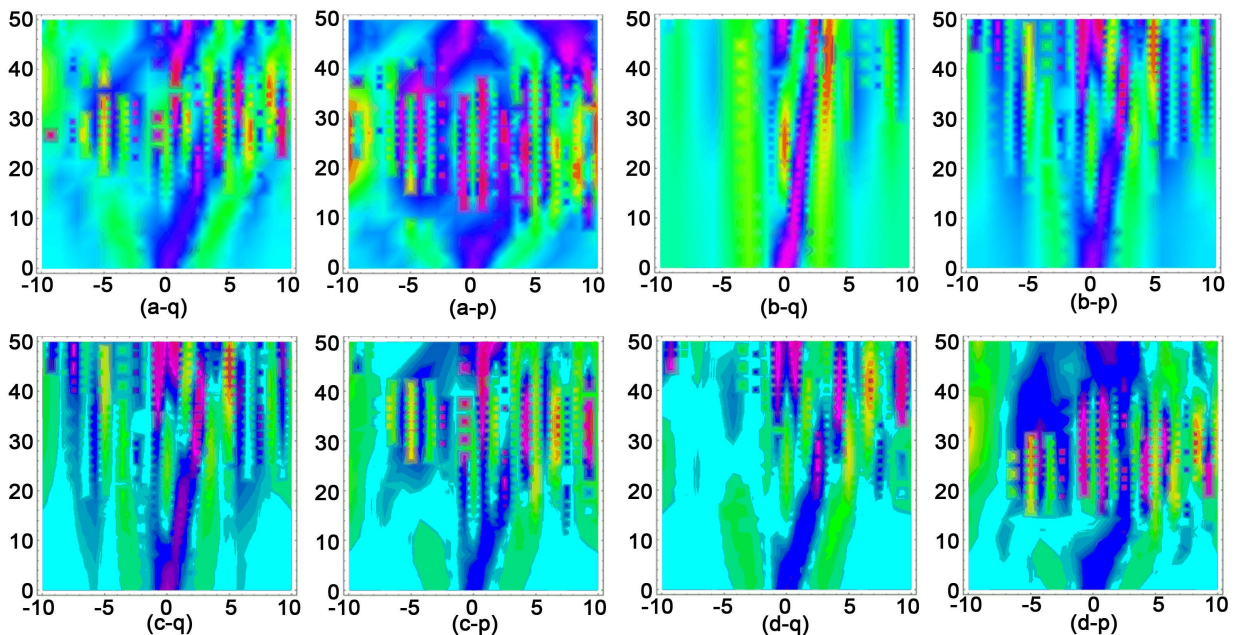


Figure 2. Density plots corresponding to 3D plots in Figure 1, representing hypothetical attractor fields for robotic swarms.

Defining potentials:

$$\begin{aligned} V[t_-, x_-] &:= \frac{1}{2} \text{Sech}[0.4tx^3]; & U[t_-, x_-] &:= \frac{1}{2} \text{Sech}[0.6t^3x]; \\ I[t_-, x_-] &:= 0.3 \text{Tanh}[0.8t^3x]; & J[t_-, x_-] &:= 0.7 \text{Tanh}[0.2tx^3]; \\ \text{IC}[x_-] &:= \frac{1}{2} \text{Exp}[ix] \text{Sech}\left[\frac{1}{2}x\right]; & \text{Tfin} &= 50; L = 10; w = 0.9; \end{aligned}$$

Defining NLS-equations:

$$\begin{aligned} \text{NLS2} = \left\{ i\partial_t q[t, x] = -\frac{w}{2} I[t, x] \partial_{x,x} q[t, x] + Vq[t, x] \text{Abs}[q[t, x]]^2 p[t, x], \right. \\ \left. i\partial_t p[t, x] = -\frac{w}{2} J[t, x] \partial_{x,x} p[t, x] + Vp[t, x] \text{Abs}[p[t, x]]^2 q[t, x] \right\}; \end{aligned}$$

Numerical solution:

$$\begin{aligned} \text{sol} = \text{NDSolve}\left[\left\{\text{NLS2}, q[0, x] = \text{IC}[x], q[t, -L] = q[t, L], \right. \right. \\ \left. \left. p[0, x] = \text{IC}[x], p[t, -L] = p[t, L]\right\}, \{q, p\}, \{t, 0, \text{Tfin}\}, \{x, -L, L\}, \right. \\ \left. \text{Method} \rightarrow \left\{\text{"MethodOfLines"}, \text{"SpatialDiscretization"} \rightarrow \right. \right. \\ \left. \left. \left\{\text{"TensorProductGrid"}, \text{"DifferenceOrder"} \rightarrow \text{"Pseudospectral"}\right\}\right\}; \end{aligned}$$

3D Plots:

$$\begin{aligned} \text{Qplot} = \text{Plot3D}\left[\left\{\text{Evaluate}\left[\text{Re}\left[q[t, x] / .\text{First}[\text{sol}]\right]\right], \text{Evaluate}\left[\text{Im}\left[q[t, x] / .\text{First}[\text{sol}]\right]\right]\right\}, \right. \\ \left. \{x, -L, L\}, \{t, 0, \text{Tfin}\}, \text{PlotRange} \rightarrow \text{All}, \text{ColorFunction} \rightarrow (\text{Hue}[\#] \&), \right. \\ \left. \text{AxesLabel} \rightarrow \{\text{"x"}, \text{"t"}, \text{"Re/Im[q]"}\}, \text{ImageSize} \rightarrow 400\right] \\ \text{Pplot} = \text{Plot3D}\left[\left\{\text{Evaluate}\left[\text{Re}\left[p[t, x] / .\text{First}[\text{sol}]\right]\right], \text{Evaluate}\left[\text{Im}\left[p[t, x] / .\text{First}[\text{sol}]\right]\right]\right\}, \right. \\ \left. \{x, -L, L\}, \{t, 0, \text{Tfin}\}, \text{PlotRange} \rightarrow \text{All}, \text{ColorFunction} \rightarrow (\text{Hue}[\#] \&), \right. \\ \left. \text{AxesLabel} \rightarrow \{\text{"x"}, \text{"t"}, \text{"Re/Im[p]"}\}, \text{ImageSize} \rightarrow 400\right] \end{aligned}$$

The bidirectional associative memory, given by the NLS-pair (13)-(14) effectively performs quantum neural computation, by giving a spatiotemporal generalization of Hopfield, Grossberg and Kosko BAM family of recurrent neural networks (see [16] and the references therein). In addition, the shock- wave and solitary-wave nature of the coupled NLS equations may describe brain-like effects: propagation, reflection and collision of shock and solitary waves (see [24]).

5. Meta-Cognitive Supervisor

The meta-cognitive supervisor model is concerned with equipping the elements of the robotic swarm with a limited capability for higher reasoning about potential consequences of its actions, using logical constraints that attempt to rule out actions leading to states that we wish to avoid. That is not to say that such states will never occur, so we also require considerable flexibility in our formulation, in the sense that it should maintain the potential to continue to operate under adverse conditions including partial system failure. Yet we require a formalism for capturing this that is as simple as possible, decidable, and computationally feasi-

ble in practice on small platforms; in this sense, we regard the Situation Calculus as too rich for our present purpose, since it is inherently first-order and consequently undecidable.

Our starting point is with the logic of actions and plans \mathcal{LAP} , which is broadly similar to the propositional dynamic logic \mathcal{PDL} [25] [26], with the difference being that the necessity operator \Box in \mathcal{LAP} is slightly less expressive (and hence less expensive) than the iteration operator $*$ in \mathcal{PDL} . This loss of expressivity does not currently appear to be a limiting factor in our application, while the compactness and strong completeness of \mathcal{LAP} -which is not shared by \mathcal{PDL} -represent a distinct advantage in the demanding highly dynamic context of our application and the limited resources of our envisaged platforms.

The logic of necessity \Box in \mathcal{LAP} is **S4** [27], and \Box has a dual possibility operator \Diamond , which is used to express goals. The logic of each modal operator $[v]$, where $v \in V$ is a countable set of verbs designating distinct actions, is the basic modal logic **K** [27]. We also include a null action $\emptyset \in V$. There is also a countable set A of atoms, which designate relevant state conditions in the environment (including within robotic system itself). The set of literals \bar{A} consists of all atoms $a \in A$ and their negations $\neg a$. We symbolize the canonical tautology as \top , which evaluates to true in all interpretations, and the canonical contradiction as \perp , which evaluates to false in all interpretations.

Formulae are then defined in the usual manner: \top and \perp are formulae, all literals a and $\neg a$ for $a \in A$ are formulae, and conjunctions $f \wedge g$, disjunctions $f \vee g$, and material implications $f \rightarrow g$ are formulae if f and g are formulae, and $[v]f$ is a formula if f is a formula and $v \in V$ is a verb. Nothing else is a formula.

For our application, we utilize a message passing approach for distributed communication, so actions for local broadcast of meta-cognitive supervisor messages are represented logically in meta-cognition as verbs and any messages received appear as atoms. Note also that this framework admits interaction with the cognitive supervisor, in both directions; specifically, some actions define initial conditions for the determination of the 2D (q_A^{2D}, p_A^{2D}) and 3D (q_A^{3D}, p_A^{3D}) attractors. Actions for attractor determination are relative to platform position rather than absolute. A formula that does not contain any of the modal operators \Box , \Diamond , or $[v]$ for $v \in V$ is described as classical.

Possibility and necessity are related as $\Diamond f \equiv \neg \Box \neg f$, and we write $\langle v \rangle f$ to mean $\neg [v] \neg f$. Goals are formulas having the form $\Diamond f$, where f is classical. For every verb $v \in V$ and formula f , we have $\Box f \rightarrow [v]f$, which means that the formula f is invariant; formulae of the form $\Box f$ constitute integrity constraints. When f in $\Box f$ is classical, the constraint is a static constraint, while $\Box f$ where f contains $[v]$ for some $v \in V$ is a dynamic constraint that describes some action law. We are especially concerned with effect constraints, which have the form $\Box(f \rightarrow [v]g)$, and describe the consequences of

performing actions. For instance, $\Box(Carrying \rightarrow [release]Delivered)$ states that if *Carrying* is true and the action *release* is executed, then *Delivered* will be made true.

Frame axioms are effect constraints of the form $\Box(f \rightarrow [v]f)$, which specifies that action $[v]$ cannot cause the condition $\neg f$. For example, $\Box(Carrying \rightarrow [travel]Carrying)$ specifies that *Carrying* will remain true after action *travel* is executed if it was true before. Formulae of the form $[v]f$ mean that after performing action v , condition f will be true, so $\Box[release]\neg Carrying$ says that *Carrying* is false after executing *release*, for instance. We can also relate alternate actions v and ψ depending on the truth or falsity of a current condition f to yield an outcome g using $((f \rightarrow [v]g) \wedge (\neg f \rightarrow [\psi]g))$, which we abbreviate as $[f?v:\psi]g$. A formula $\neg[v]\perp$ means that action v is executable, while $[v]\perp$ means that v is not executable.

The major weakness with using \mathcal{LAP} directly lies in the large number of frame axioms required; these axioms in bulk means that any action changes the truth value of relatively few formulae, which bogs down the inference procedure. Even though our current meta-cognitive definitions are small, these frame axioms would likely still be sufficiently numerous to cause problems for the limited computational resources we typically expect on the currently available small and cheap platforms our research is targeting. Consequently, our choice of logic of action and plans is a variant of \mathcal{LAP} , namely \mathcal{LAPD} [28], which employs a weak ternary causal dependence relation involving atoms, actions and conditions to overcome the frame problem. This logic also means that our meta-cognitive supervisor is able to derive indirect effects of actions, which we expect will be very important in terms of being able to reason about exposure to unacceptable failure.

The logic \mathcal{LAPD} is a variant of the earlier $\mathcal{LAP} \rightsquigarrow$ [29], which also solves the frame problem using a notion of weak causal dependence. The attraction for our problem of \mathcal{LAPD} over $\mathcal{LAP} \rightsquigarrow$ is that the former supports more compact domain descriptions without decreasing expressibility, increasing complexity or sacrificing decidability; using the latter would force us to still have to explicitly state conditional frame axioms, which have the form $\Box((f \wedge g) \rightarrow [v]g)$. In \mathcal{LAPD} . In either system we still have to state indirect dependencies.

In contrast to the ternary dependence relation provided by our choice of \mathcal{LAPD} , the dependence relation in $\mathcal{LAP} \rightsquigarrow$ is a binary relation \rightsquigarrow between actions and literals; it is actually the complementary independence relation \nrightarrow that is used here to encode frame axioms, according to $v \nrightarrow a \equiv \Box(\neg a \rightarrow [v]\neg a)$. We use the logic \mathcal{LAPD} because it includes an extra parameter in the dependence relation to capture the conditions under which actions may impact on atoms, thereby avoiding the need to state conditional frame axioms in defining the problem domain. We write the ternary contextual dependence relation as

$f|v \rightsquigarrow a$, where f is a classical formula, $v \in V$ is a verb and $a \in A$ is an atom, to mean that if f is true then action v may change the truth value of a . Note that weak contextual dependence does not mean that the action in the context causes a change in truth value of the atom, only that the change might happen. For instance, $Carrying \wedge \neg Delivered | release \rightsquigarrow Delivered$ states that executing $release$ when $Carrying$ is true and $Delivered$ is false may effect the value of $Delivered$; the conditional frame axiom

$\Box((\neg Carrying \wedge \neg Delivered) \rightarrow [release]\neg Delivered)$ is not needed in the domain description. In our case, the conditions f in $f|v \rightsquigarrow a$ are always conjunctions of literals, because disjunctions can be simply split into separate dependence statements. **Figure 3** contains a small example of a robot meta-cognition scenario in \mathcal{LAPD} .

The logic \mathcal{LAPD} is decidable, with the satisfiability problem being EXPTIME-complete, which is the same decidability and complexity as the base system \mathcal{LAP} . A tableau method for \mathcal{LAP} is simply a combination of the tableau rules for the logics **S4** and **K**, while $\mathcal{LAP} \rightsquigarrow$ and \mathcal{LAPD} require additional rules to handle their dependency relations. We use a notational variant of the definitions from [29] for tableau rules for $\mathcal{LAP} \rightsquigarrow$, with the different rules from [28] for the ternary weak dependence relation $\cdot| \cdot \rightsquigarrow$ in place of those for $\cdot \rightsquigarrow \cdot$ to describe the tableau calculus.

A labeled formula is a pair (n, f) where f is a formula and n is from a countable set of labels for possible worlds; we just use the non-negative integers \mathbb{N} . A *Skeleton* is a ternary relation $\Sigma \subseteq (V \cup \{\mathbb{N}\}) \times \mathbb{N} \times \mathbb{N}$, which represents the accessibility relations between possible worlds under actions; we write $n \Rightarrow_v m$ for $(v, n, m) \in \Sigma$. A *Tree*, which corresponds to a \mathcal{LAPD} -model, is a pair (L, Σ) consisting of a set L of labeled formulae and a skeleton Σ . A *tableau* for f is then the limit of a sequence T^n for $k = 0, 1, \dots$ of sets of trees, where $T^0 = \{(\{0, f\}, \{\})\}$ and each T^{k+1} is obtained from T^k by the application of a tableau rule (see **Figure 4**). We also use additional rules for other connectives in practice, but they amount to simple combinations of the basic rules shown here.

$$\begin{aligned}
 W = \{ & \Box \langle travel \rangle \top, \\
 & \Box (\neg Carrying \rightarrow \langle pickup \rangle \top), \\
 & \Box \langle release \rangle \top, \\
 & \Box [pickup] Carrying, \\
 & \Box [release] \neg Carrying, \\
 & \Box (Carrying \rightarrow [release] Delivered), \\
 & \neg Carrying | pickup \rightsquigarrow Carrying, \\
 & Carrying | release \rightsquigarrow Carrying, \\
 & Carrying \wedge \neg Delivered | release \rightsquigarrow Delivered \} \\
 K = \{ & \neg Carrying, \neg Delivered \}
 \end{aligned}$$

Figure 3. A simple theory in \mathcal{LAPD} . Note the absence of frame axioms, which we would have in \mathcal{LAP} , and conditional frame axioms we would still need in $\mathcal{LAP} \rightsquigarrow$. We have $\models_{\mathcal{LAPD}} (S \cup K) \rightarrow [pickup][travel][release](\neg Carrying \wedge Delivered)$.

- (\perp) : if $(n, f) \in L$ and $(n, \neg f) \in L$ then add (n, \perp) to L
 (\neg) : if $(n, \neg\neg f) \in L$ then add (n, f) to L
 (\wedge) : if $(n, f \wedge g) \in L$ then add (n, f) and (n, g) to L
 (\vee) : if $(n, \neg(f \wedge g)) \in L$ then add (n, f)
 and add $(L \cup \{(n, \neg g)\}, \Sigma)$ to T^{i+1}
 (T_{\square}) : if $(n, \square f) \in L$ then add (n, f) to L
 (A_{\square}) : if $(n, \square f) \in L$ and $n \Rightarrow_{\nu} m$ for some $\nu \in V$ or $n \Rightarrow_{\square} m$
 then add $(m, \square f)$ to L
 (K_{ν}) : if $(n, [\nu] f) \in L$ and $n \Rightarrow_{\nu} m$ then add (m, f) to L
 (\diamond) : if $(n, \neg[\nu] f) \in L$ then add $(m, \neg f)$ to L and $n \Rightarrow_{\nu} m$ to Σ ,
 where $m \in \mathbb{N}$ is new
 (\diamond) : if $(n, \neg\square f) \in L$ then add $(m, \neg f)$ to L and $n \Rightarrow_{\square} m$ to Σ ,
 where $m \in \mathbb{N}$ is new
 (RP) : if $(n, f) \in L$, where f is a literal, $n \Rightarrow_{\nu} m$, and $L \not\models (n, g)$ for all $g \mid \nu \rightsquigarrow f$
 then add (m, f) to L
 (RB) : if $(m, f) \in L$, where f is a literal, $n \Rightarrow_{\nu} m$, and $L \not\models (n, g)$ for all $g \mid \nu \rightsquigarrow f$
 then add (n, f) to L

Figure 4. Tableau rules for \mathcal{LAPD} , as a modification of the tableau rules for $\mathcal{LAP} \rightsquigarrow$, with the final two rules substituted for new rules for handling the ternary dependence relation. The notation $L \models (n, a)$ means that the literal a cannot be verified in world n from the formulae in L .

The rule (RP) states that all literals that depend on an action in a context that does not verify have to be propagated following the execution of that action. Rule (RB) is a back-propagation rule, which is required for completeness of the tableau method, and it states that literals that are true but whose truth value was not changed from the parent node in the tree must also have been true in the parent node. See [29] and [28] for further details.

Space precludes a full account of our test implementation; our system is in the pure functional language Haskell. We use a straightforward algebraic type for representing formulas, and labeled formulas are represented using a simple type synonym. The ordering of the clauses is important: formulas causing tableau branching are lower in the definition, which means that the ordering on formulas that the compiler generates by declaring the type to be an instance of “Ord” is used to prioritize reduction rule application to push branching towards the lower part of the tableau.

```

data LAPFormula = F|T|Atom String|Not LAPFormula And Formula
                |Necessary LAPFormula|Possible LAPformula
                |Cause String Formula...
                deriving (Eq, Ord)

type Formula = (Int, Formula)

```

Space precludes a complete account of our test implementation, so we offer an abbreviated description focussing on the core components. In addition to the tableau evaluation module, we also have parser modules, and a number of supporting data structures particularly for representing skeleton relations and dependency relations. The tableau is implemented in operational semantics style,

using an algebraic data type to represent a set of abstract operations on tableau branches. As a sample: “Fail” represents an error condition, ‘Result’ carries a return value, “Get” retrieves a formula for reduction, “Put” adds a labeled formula to the branch where it will be subject to further reduction, “Fresh” generates a new possible world index, “Claim” asserts accessibility between possible worlds under actions, “Close” closes the tableau branch by asserting a contradiction for a given world index, and “Split” forks the tableau into two tableau.

```
data Tableau u = Fail
    |Result u
    |Get (Formula -> Tableau u)
    |Put Formula (Tableau u)
    |Fresh (Int -> Tableau u)
    |Claim (Int, String, Int) (Tableau u)
    |Close Int (Tableau u)
    |Split (Tableau u) (Tableau u)
```

...

There are also constructors for other operations such as those for adding labeled literals to a separate list where they will not be subject to further reduction steps, for searching the skeleton relation, and for searching the dependency relations (our implementation actually supports $\mathcal{LAP} \rightsquigarrow$ as well as \mathcal{LAP}). These operations all follow the same basic pattern as those shown, so we have omitted them for brevity.

```
instance Monad Tableau where
    return = Result
    Fail      >>= f = Fail
    (Result u) >>= f = f u
    (Get g)   >>= f = Get $ \x -> g x >>= f
    (Put x t) >>= f = Put x (t >>= f)
    (Fresh g) >>= f = Fresh $ \n -> g n >>= f
    (Close n t) >>= f = Close n (t >>= f)
    (Split t t') >>= f = Split (t >>= f) (t' >>= f)
```

...

It is easy to verify that this obeys the left unit, right unit and associativity monad laws. For convenience, we use some simple wrapper functions around the constructors, rather than using them directly. For instance:

```
get :: Tableau Formula
get = Get Result
put :: Formula -> Tableau ()
put x = (Put x . Result) ()
close :: Int -> Tableau ()
close n = (Close n . Result) ()
```

The instantiation of the abstract machine uses a data type that has separate structures for holding labeled formulas that will be subject to further reduction, those that will not be further used to fire rules, the skeleton relation of accessi-

bility between possible worlds, and the ternary dependency relation. Here “BBTree” is an ordered tree type that implements the priority relation on formulas, so that non-branching formulas are preferred for reduction over those that cause branching. “Skeleton” is an indexed tree structure supporting the various kinds of searches needed on accessibility relation instances, and “Dependency” represents the ternary dependency relation and similarly allows the necessary searches on its instances.

```
data Branch = Branch {
    todo :: BBTree Formula
  , lits :: BBTree Formula
  , rho :: Skeleton
  , index :: Int
  , deps :: Dependency
} deriving (Eq, Ord)
```

The central function is the “runTab” function that defines the how the abstract operations should be applied to branch structures. The rest of the cases (not shown) follow the same basic pattern.

```
runTab :: (Ord u) => Tableau u -> Branch -> BBTree (u, Branch)
runTab Fail      _ = nil
runTab (Result y) b = singleton (y, b)
runTab (Get g)   b = case delmin (todo b) of
    Nothing -> nil
    (Just x, xs) -> run (g x) (b { todo = xs})
runTab (Put f t) b = let b' = into b f
    in runTab t b'
runTab (Fresh g) b = let n = 1 + ()index b
    b' = b {index = n}
    in runTab (g n) b'
runTab (Close n t) b = let b' = b { todo = nil, lits = single (n, F) }
    in runTab t b'
runTab (Split t t') b = (runTab t b) 'mplus' (runTab t' b)
```

...

The functions “nil” and “singleton” build trees with zero and one element, respectively. The case for ‘Put’ uses an auxiliary function ‘into’ that checks first to see if the branch already contains the negation of the formula to be inserted into the branch and, if so, inserts a formula containing contradiction “F” instead. This approach provides us with a very compact and natural definition for the tableau reduction rules, as illustrated below.

```
reduce :: Formula -> Tableau ()
reduce (n, F)          = close n
reduce (n, T)          = return ()
reduce (n, And x y)    = put (n,x) >> put(n,y)
reduce (n, Or x y)     = put (n, x) 'mplus' put (n,y)
reduce (n, Not (Necessary x)) = fresh >>= \n -> put (n', neg x)
```

```

>> claim (n, "[ ]", n')
reduce (n, Necessary x) = put (n, x) >> fromN w
>>= mapM_ (\(_, m) -> put (m, Necessary x))

```

...

The second to last case implements rule (\diamond) . The last case in the snippet implements rules (T_{\square}) and (4_{\square}) . It uses a function “fromN” that returns a list of verbs and world indexes accessible from a given world index, and the standard library “mapM” to map each of these to actions that insert each resulting formula into the branch. A program for the abstract tableau machine for completely expanding a branch is also very simple.

```

expand :: Tableau ()
expand = isComplete >>= \c -> if c then return ()
      else get >>= reduce >> reduction

```

Given an initial branch, the tableau can be applied to an initial branch, say b , with `runTab expand b`. We also have a number of convenience functions for producing an initial branch data structure from a list of formulas and dependency relation instances, and for extracting the results from the resulting tree of closed and saturated branches. Their implementation is straightforward though somewhat tedious.

6. Conclusions

In this paper, we have presented sophisticated cognitive and meta-cognitive supervisor models for joint swarms of robotic aerial and ground vehicles. Based on the research of the recent Nobel prize in Physiology on path integration and navigation in the mammalian and human hippocampus (briefly reviewed in the appendix), this paper develops a Hamiltonian path integral cognitive supervisor model. This model emulates an ∞ -dimensional neural recurrent neural network, yielding attractor fields for robotic swarms. While direct simulation of this Hamiltonian path integral can be done using IBMs’s TrueNorth chip, for the purpose of its immediate evaluation on common hardware, we have transformed this into a coupled pair of NLS equations and simulated this in *Mathematica*.

The central point of our meta-cognitive model is that we are not utilising inference in modal logic systems to determine low-level movement, but rather to simulate a kind of high-level awareness in light of descriptive goals, general conditions and broad actions to make sense of sensor data and guide overall vehicle behaviour. Our representation of meta-cognitive communication using simple atoms and verbs reflects this choice. By effectively delegating details, the relatively simple models are supported by the propositional multi-modal logic \mathcal{LAPD} suffice, with decidability of inference and, in practice, reasonable computation costs and fairly compact meta-cognitive behavioural definitions. We plan to supplant our current use of a tableau method for meta-cognitive inference using the multimodal logic \mathcal{LAPD} with a new path integral representation; that is, to eventually fully integrate meta-cognition and affine Hamiltonian control into what would amount to a single coherent generalized Hopfield

type recurrent neural network.

Acknowledgements

The authors are grateful to Dr Yi Yue and Dr Martin Oxenham, Decision Sciences, Joint and Operations Analysis Division, DST Group, Australia—for their constructive comments which have improved the quality of this paper. This work is a part of the DSTG TAS SRI (Tyche) project.

References

- [1] Ivancevic, V. and Yue, Y. (2016) Hamiltonian Dynamics and Control of a Joint Autonomous Land-Air Operation. *Nonlinear Dynamics*, **84**, 1853-1865. <https://doi.org/10.1007/s11071-016-2610-y>
- [2] McNaughton, B.L., Battaglia, F.P., Jensen, O., Moser, E.I. and Moser, M.-B. (2006) Path Integration and the Neural Basis of the Cognitive Map. *Nature Reviews Neuroscience*, **7**, 663-678. <https://doi.org/10.1038/nrn1932>
- [3] Moser, E.I., Kropff, E. and Moser, M.-B. (2008) Place Cells, Grid Cells, and the Brain's Spatial Representation System. *Annual Review of Neuroscience*, **31**, 69-89. <https://doi.org/10.1146/annurev.neuro.31.061307.090723>
- [4] O'Keefe, J., Burgess, N., Donnett, J.G., Jeffery, K.J. and Maguire, E.A. (1998) Place Cells, Navigational Accuracy, and the Human Hippocampus. *Philosophical Transactions of the Royal Society of London A*, **353**, 1333-1340. <https://doi.org/10.1098/rstb.1998.0287>
- [5] Monteiro, S., Vaz, M. and Bicho, E. (2004) Attractor Dynamics Generates Robot Formation, from Theory to Implementation. *International Conference on Robotics and Automation*, Vol. 3, New Orleans, 26 April-1 May 2004, 2582-2586. <https://doi.org/10.1109/robot.2004.1307450>
- [6] O'Keefe, J. (1976) Place Units in the Hippocampus of the Freely Moving Rat. *Experimental Neurology*, **51**, 78-109. [https://doi.org/10.1016/0014-4886\(76\)90055-8](https://doi.org/10.1016/0014-4886(76)90055-8)
- [7] O'Keefe, J. and Nadel, L. (1978) *The Hippocampus as a Cognitive Map*. Clarendon Press, Oxford.
- [8] Ivancevic, V. and Ivancevic, T. (2008) *Complex Nonlinearity, Chaos, Phase Transitions, Topology Change and Path Integrals*. Springer, New York.
- [9] Ivancevic, V. and Reid, D. (2015) *Complexity and Control, towards a Rigorous Behavioral Theory of Complex Dynamical Systems*. World Scientific, Singapore.
- [10] Feynman, R.P. (1951) An Operator Calculus Having Applications in Quantum Electrodynamics. *Physical Review*, **84**, 108-128. <https://doi.org/10.1103/PhysRev.84.108>
- [11] Feynman, R.P. (1948) Space-Time Approach to Non-Relativistic Quantum Mechanics. *Reviews of Modern Physics*, **20**, 267. <https://doi.org/10.1103/RevModPhys.20.367>
- [12] Ivancevic, V., Reid, D. and Scholz, J. (2014) Action-Amplitude Approach to Controlled Entropic Self-Organization. *Entropy*, **16**, 2699-2712. <https://doi.org/10.3390/e16052699>
- [13] Hopfield, J.J. (1982) Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proceedings of the National Academy of Sciences*, **79**, 2554-2558. <https://doi.org/10.1073/pnas.79.8.2554>
- [14] Hopfield, J.J. (1984) Neurons with Graded Response Have Collective Computations

- al Properties Like Those of Two-State Neurons. *Proceedings of the National Academy of Sciences*, **81**, 3088-3092. <https://doi.org/10.1073/pnas.81.10.3088>
- [15] Service, R.F. (2014) The Brain Chip. *Science*, **345**, 614-668. <https://doi.org/10.1126/science.345.6197.614>
- [16] Ivancevic, V. and Ivancevic, T. (2009) Quantum Neural Computation. Springer, Berlin.
- [17] Ivancevic, V. and Reid, D. (2012) Turbulence and Shock-Waves in Crowd Dynamics. *Nonlinear Dynamics*, **68**, 285-304. <https://doi.org/10.1007/s11071-011-0227-8>
- [18] Ivancevic, V. (2010) Adaptive-Wave Alternative for the Black-Scholes Option Pricing Model. *Cognitive Computation*, **2**, 17-30. <https://doi.org/10.1007/s12559-009-9031-x>
- [19] Ivancevic, V. (2011) Adaptive Wave Models for Sophisticated Option Pricing. *Journal of Mathematical Finance*, **1**, 41-49. <https://doi.org/10.4236/jmf.2011.13006>
- [20] Manakov, S.V. (1974) On the Theory of Two-Dimensional Stationary Self-Focusing of Electromagnetic Waves. *Soviet Physics*, **38**, 248-253. (In Russian)
- [21] Lax, P. (1968) Integrals of Nonlinear Equations of Evolution and Solitary Waves. *Communications on Pure and Applied Mathematics*, **21**, 467-490. <https://doi.org/10.1002/cpa.3160210503>
- [22] Haelterman, M. and Sheppard, A.P. (1994) Bifurcation Phenomena and Multiple Soliton Bound States in Isotropic Kerr Media. *Physical Review E*, **49**, 3376-3381. <https://doi.org/10.1103/PhysRevE.49.3376>
- [23] Yang, J. (1997) Classification of the Solitary Wave in Coupled Nonlinear Schrödinger Equations. *Physica D*, **108**, 92-112. [https://doi.org/10.1016/S0167-2789\(97\)82007-6](https://doi.org/10.1016/S0167-2789(97)82007-6)
- [24] Hanm, S.-H. and Koh, I.G. (1999) Stability of Neural Networks and Solitons of Field Theory. *Physical Review E*, **60**, 7608-7611. <https://doi.org/10.1103/PhysRevE.60.7608>
- [25] Fisher, M.J. and Ladner, R.E. (1979) Propositional Dynamic Logic of Regular Programs. *Journal of Computer and System Sciences*, **18**, 194-211. [https://doi.org/10.1016/0022-0000\(79\)90046-1](https://doi.org/10.1016/0022-0000(79)90046-1)
- [26] Zhang, D. and Foo, N.Y. (2001) EPDL, a Logic for Causal Reasoning. *Proceedings of the IJCAI 2001*, Seattle, 4-10 August 2001, 131-138.
- [27] Hughes, G.E. and Cresswell, M.J. (1996) A New Introduction to Modal Logic. Routledge, Abingdon-on-Thames. <https://doi.org/10.4324/9780203290644>
- [28] Castilho, M.A., Herzig, A. and Varzinczak, I. (2002) It Depends on the Context! A Decidable Logic of Actions and Plans Based on a Ternary Dependence Relation. *NMR02*, Toulouse, 19-21 April 2002, 343-348.
- [29] Castilho, M.A., Gasquet, O. and Herzig, A. (1999) Formalizing Action and Change in Modal Logic 1, the Frame Problem. *Journal of Logic and Computation*, **9**, 701-735. <https://doi.org/10.1093/logcom/9.5.701>
- [30] Hebb, D.O. (1949) The Organization of Behavior. Wiley, New York.
- [31] O'Keefe, J. and Dostrovsky, J. (1971) The Hippocampus as a Spatial Map, Preliminary Evidence from Unit Activity in the Freely Moving Rat. *Brain Research*, **34**, 171-175. [https://doi.org/10.1016/0006-8993\(71\)90358-1](https://doi.org/10.1016/0006-8993(71)90358-1)
- [32] Mittelstaedt, M.L. and Mittelstaedt, H. (1980) Homing by Path Integration in a Mammal. *Naturwissenschaften*, **67**, 566-567. (In German) <https://doi.org/10.1007/BF00450672>
- [33] Ranck, J.B. (1985) Electrical Activity of the Archicortex. Akademiai Kiado, Budapest.

est, 217-220.

- [34] Bostock, E., Muller, R.U. and Kubie, J.L. (1991) Experience-Dependent Modifications of Hippocampal Place Cell Firing. *Hippocampus*, **1**, 193-205. <https://doi.org/10.1002/hipo.450010207>
- [35] McNaughton, B.L., Chen, L.L. and Markus, E.J. (1991) "Dead Reckoning", Landmark Learning, and the Sense of Direction, a Neurophysiological and Computational Hypothesis. *Journal of Cognitive Neuroscience*, **3**, 190-202. <https://doi.org/10.1162/jocn.1991.3.2.190>
- [36] Wilson, M.A. and McNaughton, B.L. (1993) Dynamics of the Hippocampal Ensemble Code for Space. *Science*, **261**, 1055-1058. <https://doi.org/10.1126/science.8351520>
- [37] O'Keefe, J. and Recce, M.L. (1993) Phase Relationship between Hippocampal Place Units and the EEG Theta Rhythm. *Hippocampus*, **3**, 317-330. <https://doi.org/10.1002/hipo.450030307>
- [38] McNaughton, B.L., *et al.* (1996) Deciphering the Hippocampal Polyglot, the Hippocampus as a Path Integration System. *Journal of Experimental Biology*, **199**, 173-185.
- [39] Tsodyks, M. and Sejnowski, T. (1995) Associative Memory and Hippocampal Place Cells. *International Journal of Neural Systems*, **6**, S81-S86.
- [40] Zhang, K. (1996) Representation of Spatial Orientation by the Intrinsic Dynamics of the Head-Direction Cell Ensemble, a Theory. *Journal of Neuroscience*, **16**, 2112-2126.
- [41] Samsonovich, A. and McNaughton, B.L. (1997) Path Integration and Cognitive Mapping in a Continuous Attractor Neural Network Model. *Journal of Neuroscience*, **17**, 5900-5920.
- [42] O'Keefe, J. (1999) Do Hippocampal Pyramidal Cells Signal Non-Spatial as Well as Spatial Information? *Hippocampus*, **9**, 352-364. [https://doi.org/10.1002/\(SICI\)1098-1063\(1999\)9:4<352::AID-HIPO3>3.0.CO;2-1](https://doi.org/10.1002/(SICI)1098-1063(1999)9:4<352::AID-HIPO3>3.0.CO;2-1)
- [43] Fyhn, M., Molden, S., Witter, M.P., Moser, E.I. and Moser, M.-B. (2004) Spatial Representation in the Entorhinal Cortex. *Science*, **305**, 1258-1264. <https://doi.org/10.1126/science.1099901>
- [44] Hafting, T., Fyhn, M., Molden, S., Moser, M.-B. and Moser, E.I. (2005) Microstructure of a Spatial Map in the Entorhinal Cortex. *Nature*, **436**, 801-806. <https://doi.org/10.1038/nature03721>
- [45] Witter, M.P. and Moser, E.I. (2006) Spatial Representation and the Architecture of the Entorhinal Cortex. *Trends in Neurosciences*, **29**, 671-678. <https://doi.org/10.1016/j.tins.2006.10.003>
- [46] O'Keefe, J. and Burgess, N. (2005) Dual Phase and Rate Coding in Hippocampal Place Cells, Theoretical Significance and Relationship to Entorhinal Grid Cells. *Hippocampus*, **15**, 853-866. <https://doi.org/10.1002/hipo.20115>
- [47] Solstad, T., Moser, E.I. and Einevoll, G.T. (2006) From Grid Cells to Place Cells, a Mathematical Model. *Hippocampus*, **16**, 1026-1031. <https://doi.org/10.1002/hipo.20244>
- [48] Burgess, N., Barry, C. and O'Keefe, J. (2007) An Oscillatory Interference Model of Grid Cell Firing. *Hippocampus*, **17**, 801-812. <https://doi.org/10.1002/hipo.20327>
- [49] Yartsev, M.M. and Ulanovsky, N. (2013) Representation of Three-Dimensional Space in the Hippocampus of Flying Bats. *Science*, **340**, 367-372. <https://doi.org/10.1126/science.1235338>

- [50] Yartsev, M.M. (2013) Space Bats, Multidimensional Spatial Representation in the Bat. *Science*, **342**, 573-574. <https://doi.org/10.1126/science.1245809>
- [51] Turing A.M. (1990) The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society B*, **237**, 37-72. <https://doi.org/10.1007/BF02459572>

Appendix

Hippocampal Path Integration and Navigation in Mammals and Humans

We start with a brief history of *hippocampal navigation*, from O'Keefe's pioneering work to the discovery of grid cells by Mosers.

While most of neural network theory (including the concepts of associative synaptic plasticity, cell assemblies and phase sequences) is founded on Hebb's seminal work [30], and hippocampus as a spatial cognitive map was proposed in [7] [31], the pioneering paper on hippocampal navigation was [6], in which O'Keefe proposed a theoretical suggestion of a landmark-independent navigational system upstream of the hippocampus. A few years later, path integration in mammals was reported in [32], followed by a quantitative description of head direction-sensitive cells in the brain by [33], a report of remapping in *hippocampal place cells* in [34], and an early version of the *head-direction path-integrator model* in [35], which formed the conceptual basis of subsequent continuous attractor models for path integration.

A landmark paper [36] introduced empirical understanding of *hippocampal neurodynamics*, by the ability to record simultaneously from many neurons in the freely behaving animal. The phase relationship between hippocampal place units and the EEG theta rhythm was shown in [37], and the *hippocampus as a path-integration system* was proposed in [38].

A series of continuous attractor papers started with [39], followed by an attractor model of head direction cell by angular velocity integration in [40] and the introduction of the concept of *periodic boundaries* and an early introduction of *medial entorhinal grid cells* in mammals by [41].

Next two papers by O'Keefe, [4] [42], consider human hippocampus place cells, which are signaling both spatial and non-spatial information.

The pioneering study [43] reports that spatial position is represented accurately among ensembles of principal neurons in superficial layers of the *medial entorhinal cortex* (MEC), while the scale of representation increases along the MEC's dorsoventral axis. It is followed by [44] that reports the discovery of *grid cells*, which are suggested as a foundation for a universal path integration-based neuronal map of the spatial environment. Spatial representation and the architecture of the entorhinal cortex was presented in [45].

Next, we give a current brief overview of *hippocampal formation: place cells and grid cells*.

The review paper [2] shows that the hippocampal formation is able to encode relative spatial location of mammals and humans (without any reference to external cues) by the integration of translational and rotational self-motion, which is called the *path integration*.

Both theoretical and empirical studies show that the synaptic matrix of the MEC-grid cells of young mammals perform heavy self-organizing path-integration computations, similar to Turing's symmetry-breaking operation⁵, while the

⁵A landmark Turing's paper [51] demonstrating that symmetry breaking can occur in the simple reaction-diffusion system, that results in spatially periodic structures can account for pattern formation in nature.

scale at which space is represented increases systematically along the dorsoventral axis in both the hippocampus and the MEC. Spatially periodic inputs (at multiple scales) converging from the MEC-grid cells, result in non-periodic spatial firing of the *hippocampal place cells*.

The paper [3] reviews how place cells and grid cells form the *entorhinal-hippocampal representations*, initially observed in [46] and mathematically modeled in [47] [48], for quantitative spatio-temporal representation of places, routes, and associated experiences during behavior and in memory.

It has been observed that place cells perform both *pattern completion* and *pattern separation*, while hippocampal representations cannot always be discontinuous as in a sequential Hopfield network [13], but rather similar to the graded-response Hopfield network [14].

Finally, while all the research mentioned so far was dealing with 2D hippocampal path integration and navigation, which is relevant for our UGVs, in recent years this research has been generalized to 3D navigation of bats in [49] [50].



Submit or recommend next manuscript to SCIRP and we will provide best service for you:

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact ica@scirp.org