Awareness without Neural Networks: Achieving Self-Aware AI via Evolutionary and Adversarial Processes

Nigel Greenwood, Brruntha Sundaram, Alex Muirhead and James Copperthwaite

October 4, 2020

Awareness without Neural Networks: Achieving Self-Aware AI via Evolutionary and Adversarial Processes

Nigel Greenwood
Evolving Machine Intelligence Pty Ltd
Brisbane, Australia
nigel@machinegenes.com

Brruntha Sundaram
Evolving Machine Intelligence Pty Ltd
Brisbane, Australia
brruntha@machinegenes.com

Alexander Muirhead
Evolving Machine Intelligence Pty Ltd
Brisbane, Australia
alex@machinegenes.com

James Copperthwaite
Brisbane, Australia
j.copperthwaite@uq.net.au

***Abstract*— A key difficulty in achieving self-aware artificial intelligence (AI) is the achievement of epistemological knowledge, i.e. a machine that "knows what it knows" and "knows what it does not know" with respect to some model of itself or its surroundings. Given a nonlinear dynamical system with known algebraic structure expressible as differential equations, with sensors able to create a time-series of measurements of sufficient variables to create a suitable partial state vector, then novel forms of evolutionary machine learning and adversarial processes are sufficient to create a form of AI that is "aware" of its knowledge set regarding this system, and can use a form of differential Game Theory and adversarial processes to "think" about its knowledge set to address ambiguities and achieve objectives, including moving beyond its original training data. This may itself constitute a form of "self-awareness". Results from successful use of these techniques in medical and engineering problems are outlined. This AI architecture does not involve neural networks or their derivative architectures, but instead is inspired by evolutionary ecosystems. Implications for self-aware operating systems are discussed.**

Keywords*— *evolutionary, epistemological, learning, adversarial, differential, game, nonlinear, dynamics, ecosystem, operating

I. INTRODUCTION

In discussing achievement of "self-aware" artificial intelligence (AI), the first challenge is deciding what the minimal criterion for self-awareness actually requires. One can propose a spectrum exists, between mere sentience (the ability to feel, perceive or experience subjectively) at one end, and at the other the capacity to assert, in the words of Descartes, *Cogito ergo sum* ("I think, therefore I am"). For the purposes of the present paper, perhaps the variant provided by Antoine Léonard Thomas, *Dubito, ergo cogito, ergo sum* ("I doubt, therefore I think, therefore I am") is more germane.

This paper argues that the crucial ingredient to achieve self-aware AI is what might be described as epistemological knowledge, the ability of a machine "to know what it knows" and perhaps more importantly, "to know what it doesn't know" in the sense of being aware of ambiguity within its knowledge set. If epistemological knowledge is achieved and structured appropriately, then the machine can perform further mathematical processes upon its knowledge set to refine that set, such that the refined set better describes the system to be known.

Given the difficulties artificial neural networks (ANN) have in achieving epistemological knowledge, an alternative basis for machine learning has been adopted, namely evolutionary algorithms, with ambiguities in chromosome-based knowledge then interrogated by adversarial processes. Dynamical systems are used as the underlying object for self-awareness, rather than attempting awareness of static patterns or rule recognition.

Criteria for self-awareness are provided, and preliminary results outlined from work done building a machine-intelligent artificial pancreas in diabetes and evolving explicit digital twins for aviation engines, showing that each of the building-blocks for self-awareness have been successfully deployed. If instead of studying an external dynamical system, the dynamics being studied were associated with the cognitive or computational processes of the AI itself or its supporting infrastructure, then the AI could be said to be genuinely self-aware.

This would appear to form the basis of a new form of self-aware operating system for enhanced management of complex "systems of systems", such as modern aircraft and shipping and medical critical-care systems.

A. Criteria for Self-Awareness

If a particular limited system exists for the machine to understand, then describe as the knowledge set the set of all data and consequent models possessed by the machine about that system.

Here the key criteria for self-aware AI are taken to be as follows. If the machine has the abilities:

The case studies described in this paper were a result of over a decade of work, partly funded by the Queensland Government (Medical Devices Financial Incentive Program 2009; Proof of Concept Grants 2010, 2012; Knowledge Transfer Partnerships Program 2016, Ignite 2017), JDRF International Innovative grant 2011; Australian Government 2013 T1D Clinical Research Network Pilot and Feasibility grant ACRNPF_FY13_47-2013-626; plus funding from Rolls-Royce plc, the shareholders of NeuroTech Research, Diabetes Neuromathix and Turbine MachineGenes.

- To measure, via sensors, time-series data of at least part of the state vector of the system to be studied, and “know” it possesses these data;
- To perform mathematical processes upon these data to extract one or more descriptions of the system, in this case mathematical models of the system;
- To be able to test the predicted behavior of these models against the original sensor data and quantify the errors between the predictions and the measurements;
- To be able, upon observing a sufficiently large error, to perform mathematical processes upon the models to reduce this error; and be able to observe the consequent change in error resulting from these processes;
- To be “aware” of persistent ambiguities within its knowledge set (typically, a combination of persistent irreducible errors within its knowledge set associated with models describing the system, and/or a plurality of distinct models within its knowledge set that describe the system equally well);
- To be able, if confronted by such “known” persistent ambiguities, autonomously to manipulate its knowledge set and/or the external system to endeavor to reduce these ambiguities;
- To be able, by virtue of its ability to quantify the errors between its models and the system, to “know” whether it has improved its knowledge set and the processes that have achieved this, and can reproduce this behavior in the future when needed;
- To be able to synthesize a novel strategy, despite that strategy appearing nowhere in prior data, enhance the quality of that strategy via internal processes, and then deploy that strategy to manipulate the system, to achieve a new specified outcome previously unachieved,

then it can be said, in sum of all this, to think, and know that it thinks.

These building-blocks have all been successfully achieved, as performed in two real-world projects outlined in this paper.

B. *Defining Synthesizing Artificial Intelligence (sAI)*

When discussing “self-aware” AI the second challenge resides in the usual taxonomy employed for AI, whereby machine learning is a subset of AI and is itself often assumed to be predicated on architectures using ANN. Although this may be logical from a Computer Science perspective, it is problematical from a mathematical viewpoint. The terminology of classical Control Theory provides a different way to view AI: “identification” is the process analogous to machine learning, whereby the machine uses time-series sensor data to reconstruct appropriate parameter values for a model to track a dynamical system, and “control synthesis” is the process whereby a strategy is then generated, enabling the model to be manipulated in a desired way, and hence the real system itself to be controlled to a desired outcome.

When discussing self-awareness in a machine, this paper follows classical Control Theory in suggesting there are two distinct processes (possibly repeated): first, machine learning, the creation of models whereby the machine comes to “understand” the system, and second, the synthesis of strategies, whereby the machine employs that understanding to modify either its own state or that of the system. It is in the manifestation of this second process that the “awareness” of the machine of its own state (such as its knowledge of the system) can be tested.

Consequently, we refer to the second process as *synthesizing AI* (sAI). In this paper sAI is a process distinct from the machine learning process.

II. Design of Components of the AI Architecture

A. *The Dynamical System*

The dynamical system studied by the AI is assumed nonlinear, typically high-dimensional and not amenable to linearization. Note that this system is defined over a continuum of states rather than a countable set of finite states.

Its dynamics are described by ordinary differential equations (ODEs) continuously differentiable over the relevant domain, obeying

$$\dot{x} = f(x, u, \lambda, t). \tag{1}$$

In (1) the state variable is denoted $x \in \Delta \subset \mathbb{R}^N$, the control variable is $u \in U \subset \mathbb{R}^R$ and the parameter vector is $\lambda \in \Lambda \subset \mathbb{R}^P$, for some non-trivial compact and convex set Λ. These ODEs possess control variables which operate over compact intervals. Each chromosome $\bar{c}_i$ carries estimates of values for the initial conditions $x(t_0)$ and parameter vector λ, so (denoting vector concatenation by the symbol $|$) we write the evolutionary approximation

$$\bar{c}_i \approx \lambda | x(t_0). \tag{2}$$

Filipov’s Theorem, [1], is obeyed, so non-trivial control strategies $p(\bar{c}_i) : \Delta \to U$ generated by the chromosome exist, generating solution trajectories

$$x(t) = \varphi\big(x(t_0), p(\bar{c}_i), \lambda, t\big). \tag{3}$$

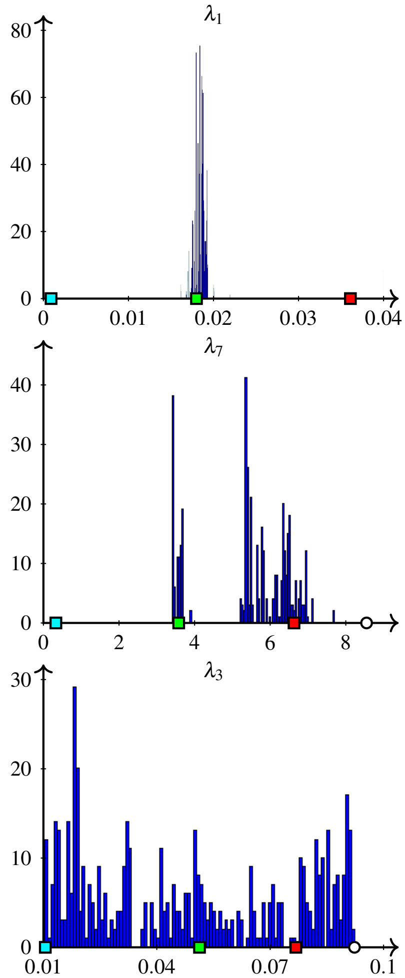
Such control strategies are called *permissible*.

Sensors generate time-series measurements (either noise-polluted or noiseless) of some state variables of the system, provided to the machine. These sensors are assumed inadequate to enable unique identification of the system (and hence the system is formally under-determined). In the real-world case-studies presented below, the systems are severely under-determined. Sensor noise, where present, is assumed symmetrically distributed but might not be Gaussian.

B. *Phi-Textured Evolutionary Algorithms (φ-TEA)*

Epistemological knowledge requires complete “explainability” (explicability), not from an AI to a human but from the AI to itself. This requirement appears to exclude the use of current forms of ANN and their numerical patterning, so a new form of evolutionary machine learning was developed.

Fig. 1. Plot of gene histograms indicating “certainty” for a parameter value (*top*), “ambiguity” (*middle*) and known “ignorance” (*bottom*). These plots are from the initial simulation study for the diabetes study, hence show the hidden “correct” parameter value on the x axis (green dot), minimum plausible value (*blue dot*) and maximum plausible value (*red dot*). The upper bound of the evolutionary search interval is also shown when distinct (*white dot*).



Genetic algorithms (GAs) are a form of heuristic optimization (first introduced by Holland, [2]; see also [3], [4]) for solution-finding in complex problems. By expressing the problem in the form of “chromosomes” evolving to increase “fitness” they mimic evolutionary processes of crossover, mutation and inheritance to search for approximate optima. In the 1970s–1990s they were proposed as a form of explicit model-based machine learning, modeling nonlinear deterministic dynamical systems. The chromosome formulation can encode aspects of explicit mathematical models to force convergence between model output and observed system output and hence (provided the model equations are sufficiently accurate) force convergence between the model and the system. This has been successfully achieved with moderately nonlinear systems, such as reconstructing aspects of stellar dynamics, [5].

Unfortunately, when applied in this way GAs suffer from two key deficiencies. First is the so-called *stagnation problem*: when analyzing sufficiently high-dimensional nonlinear systems under partial information, the evolutionary process stagnates: that is, it plateaus and further convergence fails. Various efforts (e.g. [6], [7], [8]) have been made to understand and overcome this problem using techniques such as niching, but it persists, an overarching constraint on the use of GAs as a mechanism for machine learning.

By re-structuring the axioms of evolution and by adding a further component described as trajectory-based “texturing” of fitness, we appear to have overcome the stagnation problem of evolutionary machine learning, at least for a large class of real-world problems. (For a deeper discussion of φ-TEA, see Greenwood, [9]; also [10], [11], reporting the results from [12], [13], [14].)

The second deficiency could be called the *interpretability problem*: candidate gene values for a specific parameter form histogram distributions. In the case these have multiple peaks (see Figure 1, middle and bottom), these have the following properties:

- Every candidate parameter value is valid for the specified sensor data and model equations, inasmuch as the associated chromosome generates model behavior that tracks the sensor data within the specified accuracy.
- For histograms with multiple peaks, the relative distribution of values that cause these peaks is at least partially an artifact of the evolutionary heuristic process, hence conventional statistical analysis is not useful in deciding a “best” candidate value.

As outlined in greater depth in [9], [11], we have resolved this problem using two steps: first, by structuring the information within the machine’s knowledge set using 3-tuples combining the chromosomes with their associated algebraic and geometric properties; and second, by using a novel form of adversarial sAI that employs differential Game Theory to explore the behavior of these trinities.

C. Structuring of Information within the Knowledge Set

As introduced in [9], given a set of known ODEs that describe the system, for which the associated parameters and initial conditions are unknown but may be found within a (typically, extremely large) compact parametric hypercube, the associated data for the AI’s knowledge set are notionally structured as *trinities*, defined as 3-tuples comprising:

- The algebraic model M, namely the set of ODEs of (1) in algebraic form assumed to be describing the system.
- A chromosome $\bar{c}$, a vector, each of the components (the genes) of which represents a candidate value for one of the parameters or state initial conditions associated with the algebraic equations, as in (2). Each chromosome also has additional genes appended, for metrics describing the behavior of the trajectories, below.
- Trajectories φ in state-space obeying (3), generated by substituting the information carried by a chromosome $\bar{c}$ into the model M. Depending on the structure of M there may be more than one trajectory thus generated per chromosome.

The information carried in each trinity is completely explainable, to the machine itself or to a human user.

D. Differential Game Theory

The preferred form of sAI is chosen to be a process that could employ differential Game Theory, the mathematical analysis of dynamical systems and the generation of strategies to achieve specific objectives. This would provide a valid algorithmic basis for the machine to "think", once it was furnished with a suitable knowledge set about some specified dynamical system.

It is sometimes assumed that the use of adversarial processes for machine intelligence began with Generative Adversarial Networks (GAN), first demonstrated in December 2014, [15]. However differential Game Theory, the mathematical theory of employing adversarial processes for navigating and controlling continuous dynamical processes under conditions of conflict or uncertainty, is much older.

First introduced by Isaacs, [16], who employed low-dimensional geometrical techniques, differential Game Theory subsequently underwent a transformation (see e.g. [17], [18], [19], [20]) when Lyapunov functions were adopted to model conflict and generate control strategies for high-dimensional nonlinear systems, particularly for multi-player conflict describable using zero-sum games. These are adversarial techniques designed fundamentally differently from GAN.

An important example of this is the so-called "Game against Nature". Instead of describing conflict between two or more players. the Game against Nature is usually intended to model a single player exploring the implications of poorly-understood externalities. In this formulation these externalities (parameters or variables) are manipulated as control variables by an entity, Nature, and the other player has to design strategies robust against these hostile actions. One version of this has Nature's choice of control values being random, analogous to a form of Monte Carlo simulation (e.g. [21]). However an alternative formulation (e.g. [18], [19], [20]) has Nature actively hostile and intelligent, itself using Lyapunov control theory to generate strategies algorithmically.

The form of adversarial AI used in φ-TEA on trinities was successfully demonstrated in 2011–12, in simulation-based trials of φ-TEA as the basis for a machine-intelligent artificial pancreas. This work was published as Greenwood and Gunton, [22] in July 2014, once the provisional patent for [9] had been drafted and lodged, hence also pre-dates GAN.

III. Demonstration Objectives

What will be shown is the synthesis of results from two very different highly-nonlinear, high-dimensionality dynamical systems, to provide a contribution to the field of self-aware AI:

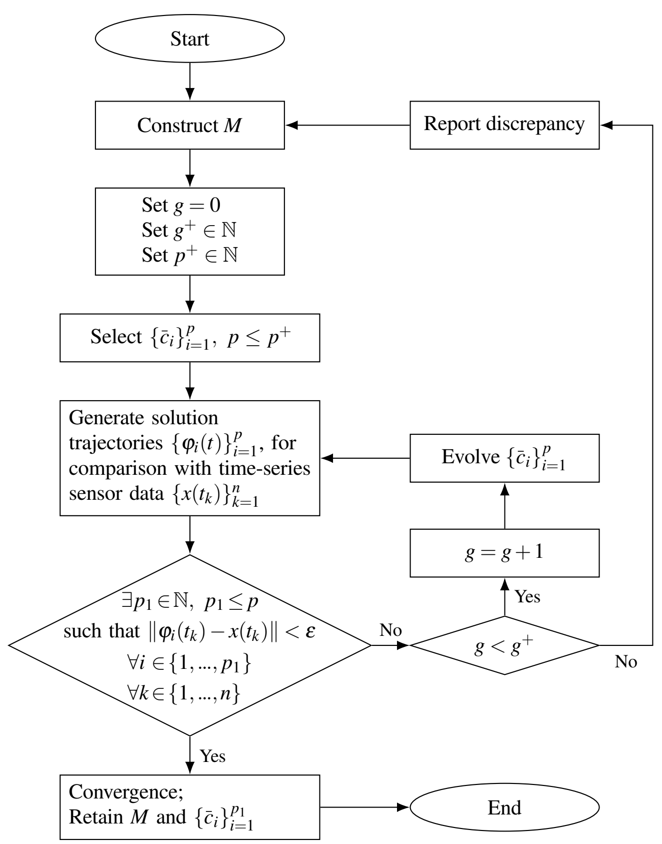
- A time-series vector of noise-polluted, formally incomplete sensor data, possibly with significant gaps for various components, is provided to the machine.
- Given known algebraic structure of the system, computational models are built via evolutionary processes using φ-TEA. Missing structure is reconstructed and stability emerges, until stable candidate models exist.
- *Awareness* of incomplete learning exists: by overlaying candidate trajectories onto the equivalent sensor data, the machine can determine whether its model behavior matches the observed system behavior to within a specified threshold.
- If "yes", then the corresponding chromosome and model within that trinity represent a valid model, within the constraints of available information;

  If "no", then the machine can continue to run evolution across further generations, to test whether further convergence occurs.

  - If "yes", then the knowledge set was simply immature.
  - If "no", then a structural flaw is "known" to exist within the assumed algebraic knowledge of the system. Depending on the size of the discrepancy it can be reported to a human user, the AI explores alternative algebraic models or the discrepancy can be ignored.

The process is outlined in Figure 2, while actual examples of evolved trajectories on sensor data are shown in Figure 3.

Fig. 2. Flowchart of the evolutionary machine learning process, including algorithmic awareness of incomplete learning. The symbol g denotes evolutionary generation, with maximum value g^+; p denotes chromosome population, with maximum value p^+.The solution trajectory $\varphi_i(t)$ and chromosome $\bar{c}_i$ belong to the i^{th} trinity, assuming all trinities here share the same model M. "Report discrepancy" pertains to an executive process of the AI, whereby alternative algebraic structure for M is explored if the initial equations prove unsatisfactory.



- Assume the machine is eventually "satisfied" that evolutionary machine learning has been completed with respect to available information. This happens when, superimposed over relevant sensor data, a sufficient number of trajectories track the sensor data within an acceptable tolerance (see Figure 3). The corresponding evolved trinities are retained (this is the "knowledge set"), while all others are deleted. The machine then examines the state of its knowledge:
  - By aligning the chromosomes of its surviving trinities, it constructs histograms for each gene. Example histograms are shown in Figure 1.
  - By interrogating these histograms, for a sufficiently large set of trinities the machine becomes "aware", for each gene value, of:

    Certainty: a histogram with a unique cluster of candidate values forming a single interval within an ε-neighborhood of some mean value;

    Ambiguity: a histogram with either multiple clusters of candidate values, or a single broad interval of values, where the total union of intervals of candidate values is nonetheless small relative to the total spectrum width of possible values;

    Ignorance: a histogram exhibits noise across most or all of its spectrum. Machine knowledge of its own ignorance is actually a valuable outcome, as it can then engage in further activities to modify its knowledge set, e.g. by perturbing the system.
- Confronted by ambiguity among the genes, the machine then "doubts". Specifically, it establishes two subordinate AIs in conflict. Both have access to the remaining trinities; hence both have full knowledge of the histograms. Assuming that the system is to be steered to a desirable target state T in state space,
  - The first AI, the *Prime*, employs differential Game Theory to generate a control strategy relevant to a particular chromosome $\bar{c}_i$:

    $$p_1(\bar{c}_i) : \Delta \to U, \tag{4}$$

    such that for some $t_1 > t_0$,

    $$\varphi\big(x(t_0), p_1(\bar{c}_i), \lambda, t_1\big) \in T. \tag{5}$$

  - The second AI, the *Adversary*, seeks to find another chromosome $\bar{c}_j \neq \bar{c}_i$ among the surviving trinities such that it achieves

    $$\varphi\big(x(t_0), p_1(\bar{c}_j), \lambda, t\big) \notin T \quad \forall t > t_0 \tag{6}$$

    and indeed, preferably, achieve for some $t_2 > t_0$

    $$\varphi\big(x(t_0), p_1(\bar{c}_j), \lambda, t_2\big) \in T' \tag{7}$$

    where T' is some highly undesirable *anti-target* set in state space
- This conflict typically results in a differential game between the Prime and Adversary, across the set of evolved trinities and admissible control strategies. As demonstrated for the type-1 diabetes case study below, Figure 5, this resulted in the synthesis of a new insulin strategy that was not only superior to anything in the training data of the medical history, but was dosed in a completely different way from the insulin doses in the medical history: see [11].

IV. Case Studies

Results have been generated for two case studies of severely-underdetermined nonlinear dynamical systems requiring evolutionary reconstruction and machine interpretation, namely:

1) A data-mining-based demonstration of organ-scale medicine for type-1 diabetes was performed for the IBM Watson AI XPRIZE (2016–2020). This involved evolving explicit models using actual (de-identified) medical histories of people with highly unstable glucose-insulin dynamics provided by Westmead Hospital, followed by successfully demonstrating the simulated use of our new form of adversarial AI, generating novel insulin strategies on an Edge device (in this case, an isolated laptop) in a clinically appropriate timescale. The adversarial sAI generated a radically superior and novel insulin strategy over the training data. This was originally reported in [12], [13], [14]; results to be published as [10], [11].

The dynamics were based on those of [22], with corrections for the meal modeling and glycogen release appropriate for unstable type-1 diabetes. The system had at least 16 state variables, of which only five could be measured, including blood glucose (BG) via fingerstick, interstitial fluid glucose (ISFG) via continuous glucose monitor (CGM) every five minutes, and fasting plasma insulin measured in a single time-series. The system had at least 36 associated parameters, none of which could be directly measured, forming a compact hypercube.

2) Demonstrating evolutionary reconstruction of the thermodynamics and physical dynamics of the main gas path of a single-spool aviation jet turbine engine from simulated partial time-series sensor data (2015–2018). Although the computationally arduous nature of evolving engine models across generations meant that the project had to be halted before convergence had been completed, the process of evolutionary convergence had been successfully demonstrated to work, as well as key features relevant to the present discussion of machine awareness. This study was reported in [23], [24].

The was modeled using a similar technique to that of the NASA Numerical Propulsion System Simulation (NPSS) as per [25]. A key difference was that the algebraic engine model used by φ-TEA avoided linearization, retaining the underlying highly nonlinear dynamics. The simulated system had 121 state variables, of which at most 33 were measurable by sensors, and over 630 parameters, almost none of which were measurable but lay within an extremely large closed, bounded hypercube.

V. Illustration of Results

For brevity the results are presented in graphical form. Readers desiring more detail are directed to [9], [10] and [11].

Fig. 3. Plot of evolved trajectories overlaid on simulated engine sensor data, (*top*) showing trajectory components (*colored lines*) "known" by the AI to be tracking sensor data (*black*) accurately subject to noise, and (*bottom*) trajectory components exhibiting both immature tracking (*needing further evolution to improve tracking*) and a sensor transient being ignored by the evolving trinities.

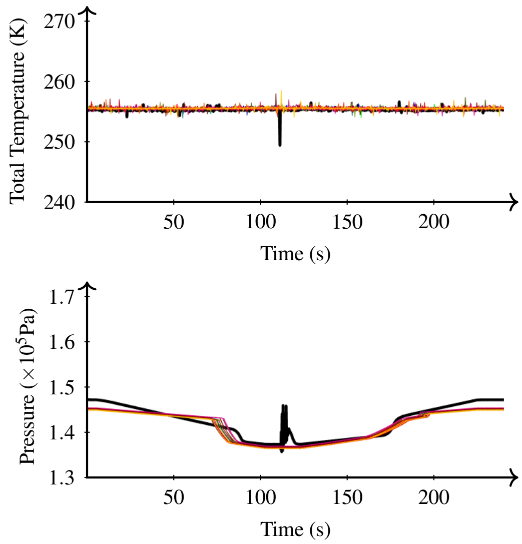
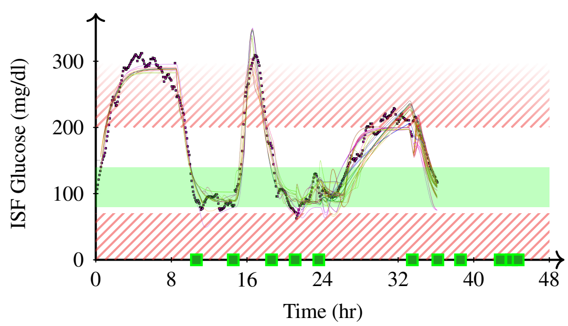
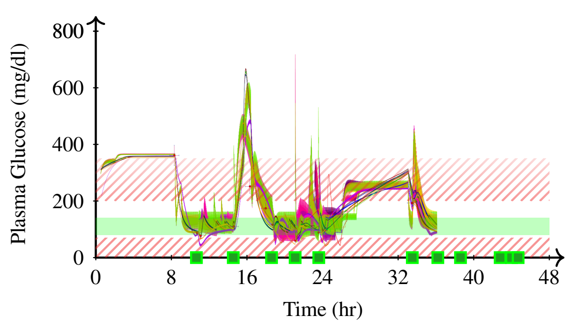


Fig. 4. Plot (*top*) of actual CGM data from patient at Westmead Hospital (*pink and black squares*) measuring ISFG levels. Meals denoted on x-axis (*green squares*). (*Bottom*) Plot of actual BG data, same subject and 30-hour interval, showing evolution of unified personalized models was successful (*colored trajectories*). Fingerstick measurements shown with $\pm 10\%$ errorbars (*red and black squares*). Desirable ranges for BG and ISFG are shown (*green rectangle*) as are undesirable/dangerous zones (*crosshatched pink*). Narrow spikes in the BG trajectories are machine-generated possible events that would not be detected by the CGM; they may be filtered out if desired.



(a) ISF Glucose Measurements



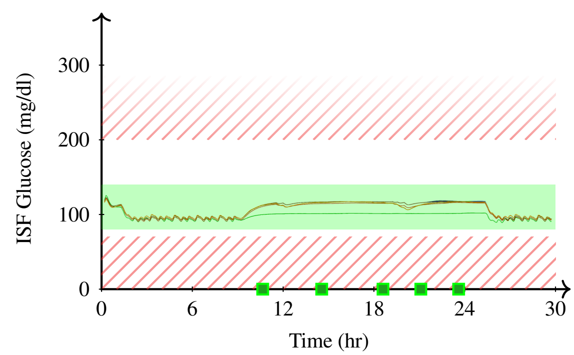
(b) Blood Glucose Measurements

Figure 3 shows segments of engine sensor data, illustrating the mechanism whereby the AI can be "aware" of whether satisfactory evolutionary machine learning is completed.

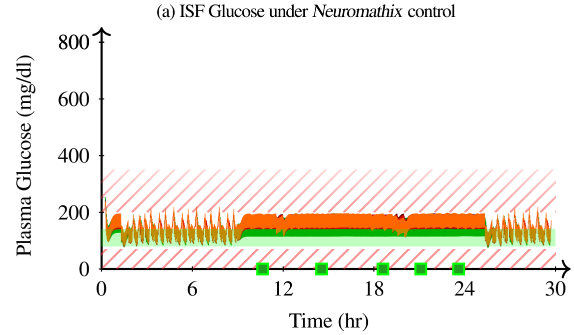
Figure 4 shows evolutionary fitting of an actual medical history of a subject suffering highly unstable ("brittle") type-1 diabetes, provided by Westmead Hospital. Unified personalized organ-scale models have been evolved, with trajectories overlaying data (each color corresponds with a distinct trinity). As is evident from Figure 4, the subject has extreme difficulty achieving stable BG and ISFG levels.

Figure 5 shows the results of an adversarial process, whereby the AI sets the Prime and Adversary in conflict. The Prime uses trinities generated from the evolutionary modeling process of Figure 4 to design an insulin control strategy to steer BG and ISFG levels to a desired target interval (80–140 mg/dl) while avoiding dangerous values (below 70 mg/dl or above 200 mg/dl). The Adversary attempts to play alternative trinities to achieve adverse outcomes, which the Prime counters in a differential game. The result is synthesis of a novel, effective insulin strategy that existed nowhere in the training data.

Fig. 5. Plots of the same 30-hour interval, same trinities as in Figure 4, but simulated using an insulin strategy generated by the adversarial sAI playing differential games after the data in Figure 4. Plots of predicted ISFG (*top*) and predicted BG levels (*bottom*) are shown. Note the avoidance of unsafe zones and the enhanced time of trajectories within the desired range.



(a) ISF Glucose under *Neuromathix* control



(b) Blood Glucose under *Neuromathix* control

VI. Conclusion

Evolutionary generation of computational models from noisy incomplete data enables epistemological knowledge of evolved trinities, including certainties, ambiguities and ignorance. The application of adversarial AI using differential Game Theory across these trinities, playing to collide with or avoid designated sets in state space, then enables the generation of new knowledge not present in the training data. And then the machine can further build on its results.

The two case studies presented represented what might be called an external locus: the AI is trying to understand some external dynamical system. Already, based on the arguments outlined, one might say that the relevant ingredients for self-awareness are present. If, instead of this external locus, the relevant dynamical system pertained to the dynamics of the computational infrastructure and local environment of the AI itself, then a deeper argument for self-awareness could be made.

It should be clear that this work offers intriguing possibilities for self-aware operating systems controlling processes or machines involving nonlinear dynamics and partial information, for mission-critical "systems of systems" with known or guessable algebraic structure.

It also offers possible avenues for AI creativity: the machine synthesis of new strategies not present in the training data.

Acknowledgment

We would like to thank Professor Jenny Gunton and Associate Professor Jane Holmes-Walker of Westmead Hospital, Sydney for their kind assistance in organizing de-identified data of patients with brittle type-1 diabetes at Westmead and advice thereafter. We would also like to thank Dr. Ingo Jahn of the University of Queensland, for his kind help and advice with the computational model of the aviation engine.

References

[1] A. F. Filippov, "The existence of solutions of generalized differential equations," *Mathematical notes of the Academy of Sciences of the USSR*, vol. 10, no. 3, pp. 608–611, Sep. 1971.

[2] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Michigan Press, 1975.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. New York: Addison-Wesley, 1989.

[4] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts: MIT Press, 1998.

[5] T. S. Metcalfe and P. Charbonneau, "Stellar structure modeling using a parallel genetic algorithm for objective global optimization," *Journal of Computational Physics*, vol. 185, pp. 176–193, 2003.

[6] J. Lampinen and I. Zelinka, in *On Stagnation Of The Differential Evolution Algorithm*, (Brno, Czech Republic, Jun. 2000), Aug. 2000.

[7] Z. Zhou and K. D. M. Harris, "Counteracting stagnation in genetic algorithm calculations by implementation of a micro genetic algorithm strategy," *Phys. Chem. Chem. Phys.*, vol. 10, pp. 7262–7269, 48 2008.

[8] F. Ye, W. Qi, and J. Xiao, "Research of Niching Genetic Algorithms for Optimization in Electromagnetics," *Procedia Engineering*, vol. 16, pp. 383–389, 2011.

[9] N. J. C. Greenwood, "System and method for modelling system behaviour," U.S. Patent No. 10,565,329, Feb. 2020.

[10] N. J. C. Greenwood, "Achieving personalized therapies for type-1 diabetes via explainable AI. Part 1: Machine learning inspired by ecosystems not neurons."

[11] N. J. C. Greenwood, "Achieving personalized therapies for type-1 diabetes via explainable AI. Part 2: Adversarial AI using edge computing in near-real time."

[12] N. J. C. Greenwood, "Evolutionary artificial intelligence: Phi-textured evolutionary algorithms reconstructing and controlling complex systems," Team MachineGenes report to IBM Watson AI XPRIZE, Sep. 2018, restricted circulation.

[13] N. J. C. Greenwood, "MachineGenes: Machine-intelligent artificial pancreas software suitable for people with 'brittle' type-1 diabetes," Team MachineGenes report to IBM Watson AI XPRIZE, Oct. 2019, restricted circulation.

[14] N. J. C. Greenwood, "Semifinal progress report: Team MachineGenes," semifinalist report to IBM Watson AI XPRIZE, Jan. 2020, restricted circulation.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., 2014, pp. 2672–2680.

[16] R. Isaacs, *Differential Games*. New York: Wiley, 1965.

[17] A. Blaquière, F. Gérard, and G. Leitmann, *Quantitative and Qualitative Games*. New York: Academic Press, 1969.

[18] J. M. Skowronski, *Nonlinear Liapunov Dynamics*. Singapore: World Scientific Publishing, 1990.

[19] J. M. Skowronski, *Control of Nonlinear Mechanical Systems*. New York: Plenum Press, 1991.

[20] N. J. C. Greenwood, "Applied Differential Games in the Aerial Scenario," PhD dissertation, University of Queensland, 1993.

[21] T. Basar and G. Olsder, *Dynamic noncooperative game theory*, 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 1999.

[22] N. J. C. Greenwood and J. E. Gunton, "A computational proof of concept of a machine-intelligent artificial pancreas using Lyapunov stability and differential game theory," *J. Diabetes Sci.Technol.*, vol. 8, pp. 791–806, 2014.

[23] B. Sundaram, N. J. C. Greenwood, and I. Jahn, "A report on the MachineGenes platform incorporating the TreeModel for jet engines. Part 1: Model construction and simplification," Report by EMI to Rolls-Royce plc and the Queensland Government, 2017, restricted circulation.

[24] N. J. C. Greenwood, B. Sundaram, J. Copperthwaite, A. Muirhead, and I. Jahn, "A report on the MachineGenes platform incorporating the TreeModel for jet engines. Part 2: Engine evolution and reconstruction with φ-textured evolutionary algorithms," Report by EMI to Rolls-Royce plc and the Queensland Government, 2018, restricted circulation.

[25] M. Wolff, *Aerothermal design of an engine/vehicle thermal management system*, RTO-EN-AVT-195. NATO, 2011.