Multiobjective Neuromodulated Controllers for Efficient Autonomous Vehicles with Mass and Drag in the Pursuit-Evasion Game

Ian Showalter
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada
ianshowalter@cmail.carleton.ca

Howard Schwartz
Department of Systems and Computer Engineering
Carleton University, Ottawa, Canada
Howard.Schwartz@sce.carleton.ca

***Abstract*—Autonomous vehicles in the pursuit-evasion game, subject to the effects of mass and drag, are controlled using an evolutionary multiobjective neuromodulated controller with unsupervised learning. Multiobjective evolution of network weights and topologies (NEAT-MODS) is extended with Lamarckian-inherited neuromodulated learning. NEAT-MODS is an NSGA-II augmented multiobjective neurocontroller that uses two conflicting objectives. By evolving pursuit agents optimized with the separate and conflicting objectives of 'capturing evaders' and 'minimizing energy consumption', efficient neurocontrollers can be evolved. NEAT-MODS uses a selection process that aims to ensure Pareto-optimal genotypic diversity and elitism. Neuromodulation is a biologically-inspired technique that can adapt the per-connection learning rates of synaptic plasticity. Lamarckian inheritance allows behaviours learned during parent generations to be passed on to their offspring. The capability of the design is demonstrated in a series of experiments with a simulated evolved vehicle pursuing a basic evader vehicle. It is shown that compact and efficient neurocontrollers for pursuer agents with nonzero mass and drag, capable of capturing an optimal evader while simultaneously minimizing energy consumption, are evolved.**

***Index Terms*—Artificial Neural Network, Autonomous Vehicle, Hebbian Learning, Lamarckian Inheritance, Multiobjective, NEAT-MODS, Neuromodulation, Pursuit-Evasion, Unsupervised Learning.**

I. Introduction

Pursuit and evasion contests are some of the most important, challenging, and common problems that ambulatory animals encounter [1]. Similarly, pursuit-evasion games can be used to simulate many challenges that mobile agents face. The pursuit–evasion differential game can be considered a generalized exercise for many other robotic problems such as autonomous vehicle navigation, platooning, obstacle avoidance, leader following, homing, docking, path planning and wall following. Additionally, it models many real-world applications, including surveillance and tracking, search and rescue, location and capture of hostile forces, and localizing and neutralizing environmental threats [2].

Agents in pursuit-evasion games are typically modelled as mass-less. Classical optimal solutions generally do not consider agents with mass, inertia or drag or other forces that may act upon the vehicle. Real-world vehicles are subject to the effects of mass and drag. In [3], simple coevolved predatory and prey neurocontrollers using two Khepera robots (both simulated and real) are demonstrated. Mass is not simulated, and is unlikely to be a factor on 5.5cm diameter robots operating in a 47cm square arena, as maximum speed could not be great considering that each robot occupies more than 10% of the distance from each wall to that opposite.

Many real-world problems involve multiple competing objectives. In pursuit-evasion games, the primary objective of a pursuer is to capture or facilitate the capture of evaders. However, as real-world resources are finite and costly, a secondary objective of minimizing energy consumption exists. Using multiobjective optimization, it is possible to maximize the capture of evaders while simultaneously minimizing energy usage.

In [4] we proposed a novel architecture where Lamarckian inheritance is applied to neuromodulated multiobjective evolutionary neural controllers that are trained in real time during each evaluation tournament. The neural controllers are assembled based on NEAT-MODS [5], and augmented with neuromodulation and Lamarckian inheritance. Each candidate neurocontroller's fitness is determined by tournament, where each candidate neurocontroller is judged based on its performance controlling a pursuer or evader agent vehicle. The candidate neurocontroller's weights are modified during each tournament using neuromodulated Hebbian learning. This architecture alternatively tests the learning space with a genetic operator, and then attempts to improve upon these results using neuromodulated learning to adapt the network between each time step during operation. This approach also allows exploration of the entire learning space, and fine tuning to find each local error minimum, until a solution with the global minimum error is found. Using NEAT-MODS alone, fine-grain adjustment of the connection weights requires mutation, which only occurs between generations, when offspring are produced. By including neuromodulation, the weights can be adjusted continuously during the lifetime of each generation [6]. Implementing Lamarckian inheritance by re-encoding the neuromodulated adjusted weights back into the parents' genes allows these adjustments to be passed on to the offspring generations, so that they are effectively pretrained with their

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

parent's learned behaviour.

Here, these Lamarckian-inherited neuromodulated evolutionary controllers are applied to agents in a basic pursuit-evasion game. A population of pursuers is evolved to catch a simple evader while simultaneously minimizing energy use. Additionally, the pursuer and evader agents are modelled with inertia in the form of mass and aerodynamic drag. We believe that this is the first time agents with explicitly modelled mass and drag have been applied to the pursuit-evasion game. It is postulated in [1] that "better understanding of pursuit and evasion would extend to game theory, animal biology, evolutionary psychology, and neuroethology."

II. Background

Fully autonomous robotic vehicles are needed to aid humans in many fields. Robots can go places that biological lifeforms cannot, and willingly perform tasks that humans find monotonous. When communication between robots and human controllers is difficult due to distance or interference, some degree of autonomy is required. Autonomy requires robots able to adapt to changing environments. Evolution and unsupervised learning are both mechanisms that can provide autonomous adaptation with respect to a changing environment.

Pursuit-evasion games can simulate multiple robotic vehicles working together and against one another. Pursuit-evasion games offer a simplified but sufficiently complex problem to test and demonstrate the effectiveness of experimental fully autonomous algorithms. The goal of pursuit-evasion games is to find the best strategy for one or many pursuers to catch one or many evaders, while concurrently finding the best strategy for the evader(s) to avoid capture by the pursuer(s).

Lifeforms face competing problems. For example, plants typically face competing objectives such as finding water and obtaining sunlight. These objectives compete for the same resources, yet the plant cannot survive without the resources provided by both. Similarly, the vehicle designer faces competing objectives such as minimizing energy consumption by reducing mass, and maximizing vehicle range requiring energy storage (which increases mass). Autonomous robots face many different objectives such as completing missions in the minimum amount of time while simultaneously minimizing power consumption. Multiobjective optimization is an area of research allowing several objective functions to be maximized simultaneously without the use of an auxiliary function. Using an auxiliary function requires that separate objectives be weighted and combined into a single function. Auxiliary functions require assumptions about the Pareto front, whereas multiobjective solutions aim to search for the entire Pareto front simultaneously. Multiobjective evolutionary neurocontrollers have been shown to successfully adapt neural network topology and weights [5], incrementally growing from a basal initial structure, and evolved to a minimal topological solution [7].

Artificial neural networks (ANN) have been successfully used to operate robotic systems over the last few decades. They are an effective tool for robotic control, and promise many advantages over conventional control, such as the ability to learn, and to adapt unsupervised to changing environments. Determining the smallest size network topology is desirable to minimize computational cost, latency, and power consumption. Many different techniques have been applied to the training and topology of ANNs, including gradient descent methods with grown or pruned topologies [8], evolutionary methods, and biologically plausible methods such as Hebbian learning and neuromodulation [9]. NeuroEvolution of Augmented Topologies (NEAT) was successfully demonstrated for function approximation and the double pole balancing problem in the original publication [7], and subsequently for other problems. NEAT-MODs has adapted NEAT for multiobjective problems, and demonstrated the evolution of robot neurocontrollers [5]. Similarly, NEAT has been adapted to evolve neurocontrollers in a distributed on-line manner in odNEAT [10]. The odNEAT method has been augmented with Hebbian neuromodulation to further reduce convergence times [6].

Combining evolution and learning can provide a powerful synergy between complementary search algorithms. Networks with evolved initial weights can be trained faster and to a higher degree of accuracy than networks with random initial weights [11]. According to Hebbian theory, synaptic plasticity is the mechanism by which, when an axon of cell A repeatedly excites cell B, a change takes place in one or both cells such that A's efficiency in firing B is increased [12]. Hebbian learning is therefore an unsupervised method of training where the connection weights (strengths) are updated as a function of pre- and post- synaptic activity [13]. Neuromodulation is considered to be a major mechanism producing memory and learning in biological nervous systems [13]. Specialized neuromodulatory neurons control the amount of plasticity of other neurons in biological organisms by using neurotransmitters such as dopamine and serotonin [13]. Neuromodulation of the synaptic plasticity augments the Hebbian learning rule by providing gating of the plasticity of a synapse between two other neurons, by updating the synapse after the neuron has fired [9], [14].

A novel architecture for a neuromodulated multiobjective topology and weight evolution of artificial neural networks with Lamarckian inheritance is proposed in [4]. Combining neuromodulation with evolution provides a powerful tool for exploring the search space. This combination gives the unique ability to test the search space with a genetic operator, and then improve upon these results using neuromodulated learning to adapt the network during operation. This approach allows exploration of the entire search space, and fine tuning to find each local maximum, until a solution with the global (or at least a more global) maximum is found. Including Lamarckian inheritance allows the transfer of learned behaviour from parent to offspring populations, reducing convergence times and improving performance. The proposed architecture is demonstrated by simulation of a differential wheeled robot applied to an autonomous foraging task in a maze. The results demonstrate that the augmentation of neuromodulated NEAT-

MODS with Lamarckian inheritance gives an effective and efficient tool for generating neurocontrollers by allowing pre-training of offspring generations based on parent generations' unsupervised learning obtained while the neurocontrollers are evolving.

A. Pursuit-Evasion Differential Games

Pursuit-evasion games are a type of differential game [15] where the Pursuer group aims to track down and capture the Evader group in an environment. Figure 1 shows the mechanics of the pursuit-evasion game when mass and drag are not modelled.

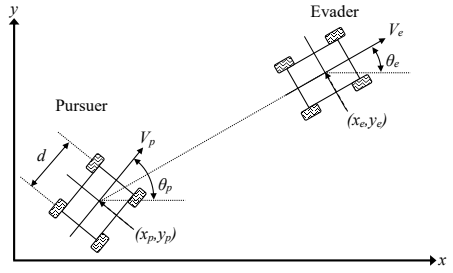

Fig. 1. Pursuit-Evasion Mechanics

The equations of motion for the pursuers and the evaders [16], [17] when mass and drag are not modelled are presented in Equation 1 .

$$\begin{aligned} \dot{x}_i &= V_i \cos(\theta_i) \\ \dot{y}_i &= V_i \sin(\theta_i) \\ \dot{\theta}_i &= \tfrac{V_i}{L_i} \tan(u_i) \end{aligned} \tag{1}$$

Here, i is a placeholder for p to represent the pursuer, or e for the evader, x_i, and y_i are the Cartesian coordinates of the robot, V_i the robot velocity, θ_i the orientation, L_i the wheelbase (distance between front and rear axles), and u_i the steering angle.

B. Dynamic Model Including Mass and Drag

The pursuer and evader agents presented here are modelled using a simple dynamic model that includes mass and drag. The dynamic model is presented in Equation 2, where T is the agent's thrust, k_d is the drag coefficient, m the agent's mass, and v and a are the agent's velocity and acceleration respectively.

$$T - k_d v = ma \tag{2}$$

Adding the effects of mass to agents in the pursuit evasion game used here means that pursuer agents must evolve to follow the evader closely, as any deviations from their optimal paths require large energy expenditures to correct. Changing direction is more costly to agents modelled with mass, as not only does more ground have to be covered to regain the optimal path, but energy must also be expended to change direction.

C. Neuroevolution

NEAT is a direct representational method for genetically encoding and evolving the weights and architecture of ANNs [7]. NEAT uses a unique innovation number associated with each gene to track the history of the genetic markers, facilitating crossover without suffering from the "competing convention" where computation is wasted when duplicates of the same or virtually identical structure compete against each other. NEAT also uses speciation: the total population of individuals is divided into species to preserve innovation. Inter-species differences are determined using a compatibility distance function based on the number of excess and disjoint genes (those that are not common to both sequences), and average weight differences. Species are weighted using a sharing function based on the compatibility distance function, such that organisms in the same species share their fitness. Offspring populations are evaluated using a fitness function; the result is weighted using the sharing function. The next offspring population is populated based on the weighted fitness. The new population is then randomly mutated by any of: perturbation of weights, replacement of weights, addition of a new node, addition of a new connection, disabling a connection, intraspecies crossover, or interspecies crossover. NEAT has been applied to many problems, including the pole balancing problem [7], computer games [18], [19], and robot control [20].

D. NEAT-MODS

NEAT-MODS is a NEAT-based multiobjective evolutionary algorithm that aims to maximize two (or more) objectives without the use of an auxiliary function. In [21] it is argued that it is more efficient to approach objectives in a simultaneous manner than sequentially in the search for the Pareto-optimal solution, as multiobjective evolutionary algorithms are more easily parallelizable, and conflicting objectives ensure good diversity in the search space [22]. In NEAT-MODS, the basic genotype, species diversification and steps of NEAT are followed, but with the substitution of the nondominated sorting of NSGA-II [23] being used, allowing Pareto-optimal controllers to be evolved simultaneously for problems with conflicting objectives. For the problem of robot navigation, the conflicting objectives, as presented in [5], are *achieving goals* while *avoiding obstacles*. NEAT-MODS uses NEAT's speciation. The NEAT-MODS process implemented for the research presented in this paper is described in Algorithm 1.

E. Neuromodulation

According to Hebbian theory, synaptic plasticity is one of the mechanisms of learning in neural circuits. Synaptic plasticity, Δw, is the strengthening or weakening of synapse strength over time according to increases or decreases in their activity[12]. As in [9], the updating of synaptic weights is performed as per Equation 3 where η is the learning rate, x is the activation level of the pre-synaptic neuron, y that of the post-synaptic neuron, w the connection weight, and

Algorithm 1 NEAT-MODS

Initialization: A minimal topology network is defined with no hidden layer nodes. One edge connects each input directly to each output. An initial offspring population of individuals is generated with randomly assigned weights. An empty parent population is also defined.

While $gens < gens_{max}$, repeat the following until the generational count has reached the termination condition.

1) Tournament: The NEAT genes of each offspring individual are used to construct an ANN that is then used to control a simulated robot.
2) Evaluation: The fitness of each offspring individual's ANN is calculated for each objective.
3) Combine Populations: The parent and offspring populations are combined for selection.
4) Ranking: The combined population is ranked using the nondominated sorting algorithm of NSGA-II.
5) Species: The species affiliation of each individual in the combined population is calculated as in NEAT.
6) Sorting: Individuals are grouped into their species, and sorted within the species based on the nondominated ranking from Step 4.
7) Sorting of Species: The species are sorted based on their highest nondominated ranking individuals.
8) Selection: From top ranked species to lowest ranked, the top ranking individual of each species is selected, followed by the next top ranking individual of each species. The process continues down the ranking of each combined population species until the offspring population is filled.
9) Parent Population: The new offspring population is saved as the parent population.
10) Reproduction: As in NEAT, the mutation of ANN weights by uniform perturbation and random replacement, new node addition, new connection addition, connection disabling, crossover and inter-species crossover are performed on the offspring population in a probabilistic manner.
11) Stopping criteria: Steps 1 through 10 are repeated until the generational count has reached the termination condition.

A, B, C, D are the correlation term, pre-synaptic term, post-synaptic term, and constant weight increase or decay rate. These parameters are tuned to adapt the synaptic plasticity.

$$\Delta w = \eta \left(Axy + Bx + Cy + D \right) \tag{3}$$

In the brain, some specialized neurons release chemical transmitters to control the rate of learning of the connections between neurons [24]. This phenomenon is called neuromodulation and is considered to be a major mechanism producing memory and learning in biological nervous systems [13]. The neuromodulatory neurons control the amount of plasticity of other neurons in biological organisms by using neurotransmitters such as dopamine and serotonin [13]. The computational theory on the roles of neuromodulatory systems and how they mediate signals that regulate the learning mechanisms in the brain are presented in [25]. Based on a review of experimental data and theoretical models, a unified theory on the roles of neuromodulators is presented. In this model, dopamine controls the error in reward prediction, serotonin controls the time scale of reward prediction, noradrenaline controls the randomness in action selection, and acetylcholine controls the speed of memory update.

Neuromodulation of the synaptic plasticity augments the classic (Hebbian) learning rule by providing gating of the plasticity of a synapse between two other neurons, by updating the synapse after the neuron has fired [9], [14]. Increased performance in ANNs through simple Hebbian plasticity has previously been demonstrated, but shown to have limited learning and memory capabilities in more complex tasks [9]. Controlling Hebbian synaptic plasticity by neuromodulation has been presented as more powerful and biologically plausible than simple Hebbian plasticity in [14]. In [13], neural networks that employed neuromodulatory neurons were found to have a clear advantage over those with no neuromodulatory neurons based on experimental data.

A simplified version of neuromodulation is assumed in [6], and a similar approach is used in this research. Here the model of the neuromodulation activation for each neuromodulating neuron is calculated using Equation 4, where w_{ji} is the weight connection of the pre-synaptic neuron j and the post-synaptic neuron i, and o_j is the output of pre-synaptic neuron j.

$$m_i = \sum_j w_{ji} o_j \tag{4}$$

Applying neuromodulation from Equation 4 to the model of synaptic plasticity described in Equation 3, the weight between neuron j and neuromodulated neuron i is modified using Equation 5 (o_i is the output of the post-synaptic neuron i, and o_j is the output of pre-synaptic neuron j).

$$\Delta w_{ji} = \eta \tanh\left(\frac{m_i}{2}\right) \left(A o_j o_i + B o_j + C o_i + D \right) \tag{5}$$

Where m_i represents the amount of neuromodulator (such as dopamine) received and is the neuromodulation transmitted by the neuromodulating neuron and connections. The values A, B, C, and D can be determined in a variety of manners, including evolutionary methods. Figure 2 shows how neuromodulation is applied by a neuromodulating neuron to neuromodulated neurons. Here, each weight represents a synapse. The value of the weight represents the amount of signal transmitted from the pre-synaptic neuron, through the synapse, to the next neuron, the post-synaptic neuron.

Hebbian learning, and by extension neuromodulated learning, are unsupervised learning methods, as no desired value is necessary. Unlike the backpropagation algorithm, no error feedback is required in neuromodulated Hebbian learning, and thus it is fully unsupervised, fulfilling one of the requirements for fully autonomous robots. Evolutionary methods can be used to determine the parameters of the neural networks,

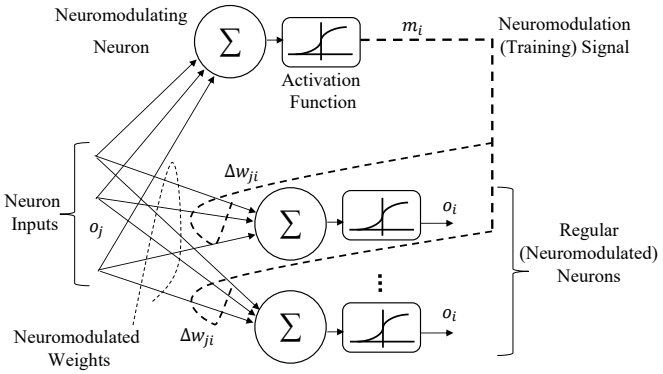

Fig. 2. Neuromodulation

including those of neuromodulation, and in these cases, the objectives used in the evolutionary optimization could be considered as a form of supervision.

III. NEAT-MODS with Neuromodulation

The combination of neuromodulation and NEAT-MODS alternately tests the learning space with a genetic operator, and then attempts to improve upon these results using neuromodulated Hebbian learning to adapt the network during operation. This approach allows exploration of the entire learning space, and fine tuning to find each local error minimum, until a solution with the global minimum error is found. The evolutionary neurocontroller is assembled based on NEAT-MODS [5], with network weights that are modified during each evaluation tournament using neuromodulated Hebbian learning, as applied to a single objective NEAT-based neurocontroller in [10]. Here, we define a tournament as the time period during which candidate controllers are *both* learning *and* evaluated to determine their individual fitness. The NEAT node (neuron) gene is augmented to include the synaptic plasticity terms A, B, C, D, and a flag to denote if the node was standard or neuromodulated. The NEAT connection (synapse) gene is similarly augmented to include neuromodulatory neurons. In the experiments presented here, the model of neuromodulation allows neuromodulating neurons to modulate any neurons, including themselves and other neuromodulating neurons.

A. Lamarckian Inheritance

Lamarckian inheritance is the term commonly used to describe the hypothesis that a parent organism can pass on characteristics acquired during its lifetime to its offspring [26]. Here, Lamarckian inheritance is implemented by saving the parent generation's adapted weights after completion of the neuromodulated learning period during each tournament. Lamarckian inheritance extends the neuromodulated evolved neurocontroller by allowing knowledge about the parent generation's environment learned through neuromodulation to be passed on to offspring populations. Thus offspring do not have to relearn behaviours that were acquired during parent generations, saving time and computational resources.

B. Neuromodulated NEAT-MODS with Lamarckian Inheritance

Lamarckian inheritance is implemented by inserting a new step (Step 2.5) into the procedure described in Algorithm 1. At this new step, we save the offspring generation's weights adapted during Step 2 by re-encoding them in their respective NEAT genes. Thus the weights that have been adapted by neuromodulated learning during each tournament are available for future populations. This architecture has previously been demonstrated with a simulated maze and foraging task in [4].

IV. Simulation

The proposed evolved multiobjective Lamarckian-inherited neuromodulated controllers are applied to simulated pursuer agents in the pursuit-evasion game. The evader agents use a simple controller that applies maximum thrust in the direction of the pursuer agent's original location.

A. The Neurocontrollers

The initial network topology is minimal. There are 3 input neurons directly connected to two output neurons through 6 initially randomly assigned weights as shown in Figure 3. The

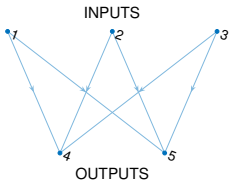

Fig. 3. Initial Neural Network Topology

neurocontroller's inputs are the evader range, evader bearing, and 'friend or foe' signals. The neurocontroller outputs are the robot's thrust and steering commands. The network topology is then augmented in a minimalist fashion, by a maximum of one node gene and one connection per individual per generation. The genotype is defined as in [7] with a list of connection genes and a list of node genes. The node genes have been augmented to include the synaptic plasticity parameters A, B, C, D. A learning factor of 0.05 was found to be effective in [4], and is used here. As in [5], no simple neuron biasing has been included in the neuron model. The process outlined in Algorithm 1 (with the Lamarckian inheritance step described in Section III-B, but without the speciation steps) is then used to evolve the candidate neurocontrollers, with neuromodulation being performed as described in Section II-E during the tournament step.

B. The Pursuit-Evasion Game

A basic 2D pursuit-evasion differential game is used to measure the performance of the proposed system. In each game, one pursuer attempts to capture one evader. There are no obstacles or walls. The pursuer is always placed at the origin. The evader is placed at a radial distance of 1m from the pursuer, but at a randomly determined angle. An evader is considered captured if the distance between pursuer and

evader is less than 0.1m. If the evader has not been captured with 600 time-steps of simulation, equivalent to 60s, the game is terminated, and considered a draw. Both agents have a mass of 1kg, and a drag coefficient of 0.295. This value is based on drag coefficients of similar vehicles. The pursuer can produce a thrust of 0.1N, the evader 0.09N. Both agents can produce thrust in any direction.

A population of 44 pursuer individuals is maintained. The population size of 44 is used in [5] and [4]. The populations are evolved for 150 generations as in [5] and [4].

C. Objective Functions

For each generation, the individual candidate pursuer neurocontrollers are evaluated based on their performance against a simple evader opponent in the pursuit-evasion game. The candidate pursuer neurocontrollers are each evaluated using two objective functions. The two objective functions F_1 and F_2 for the pursuer are:

$$F_1 = 2H + \frac{1}{1+R} \tag{6}$$

$$F_2 = \frac{1}{1+T} \tag{7}$$

Where H is the Boolean outcome of the game (1 for a capture, otherwise 0). The distance between pursuer and evader at the end is R. The average thrust of the pursuer is T. The purpose of F_1 is to reward the pursuer for capturing the evader, and achieving the minimum final distance from the evader. The purpose of F_2 is to minimize energy consumption, by minimizing the amount of thrust produced during the simulation.

The optimal F_2 agent is one that uses no energy, and remains stationary. It is impossible to maximize F_1 and capture the evader used here while remaining stationary. Therefore objective functions F_1 and F_2 are considered contradictory, and a Pareto-optimal set of neurocontrollers should exist.

V. Results

As evolutionary algorithms are stochastic in nature, repetitive runs (a run being a random seeded completion of Algorithm 1) are generally used to obtain statistically relevant results. Therefore, the simulation results are exhibited statistically as the standard deviation, and mean of 30 independent runs (as in both [5], and [6]) of 150 generations. Table I presents the performance of Lamarckian-inherited neuromodulated simulated pursuer controllers.

Objective Fitness	Mean	Max	Standard Deviation
F_1	2.8504	2.9279	0.37243
F_2	1	1	0

TABLE I
Neurocontroller Fitness

Figure 4 shows the mean and standard deviation of performance objective F_1 for each generation. After an initial period of approximately 50 generations, Lamarckian-inherited neurocontrollers capable of capturing the evader agents have been evolved.

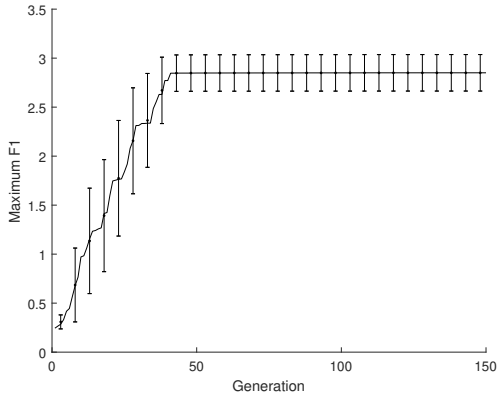

Fig. 4. Neurocontroller Objective F_1 Performance versus Generation.

Figure 5 shows the mean and standard deviation of performance objective F_2 of the neurocontrollers for each generation. After an initial period of approximately 10 generations, the neurocontrollers have evolved to maximize the F_2 objective. This is the trivial case, as an F_2 value of one indicates that the evolved pursuer agent has optimized objective F_2, and not moved at all. After the initial 10 generations there is always at least one agent in every population that exhibits this behaviour, as the mean and maximum over the generations are one (with standard deviation zero) in Table I.

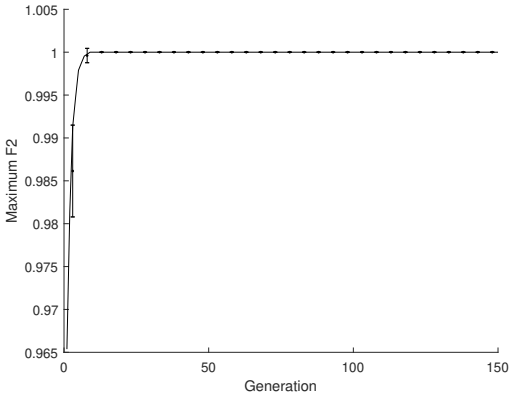

Fig. 5. Neurocontroller Objective F_2 Performance versus Generation.

Figures 6 and 7 show the pursuer and evader agent's paths. Initial positions are marked with an 'x'. The evader path is a red solid line with dot markers. The pursuer path is shown as black dots. For clarity, the path data points have been down-sampled by a factor of five (only one in 5 position points is displayed). A black circle indicating the capture radius surrounds the pursuer agent's final position. The game outcome and game number are displayed in the figure title area. Figures for objective function F_2 are not shown, as individuals that dominate objective F_2 are the trivial case

where the pursuing agent does not move, as discussed in Section IV-C.

Figure 6 shows the paths of the evader agent and pursuer agent with the best performance objective F_1 over all 30 runs. This pursuer's neurocontroller has captured the evader, but does not follow the most direct, efficient path.

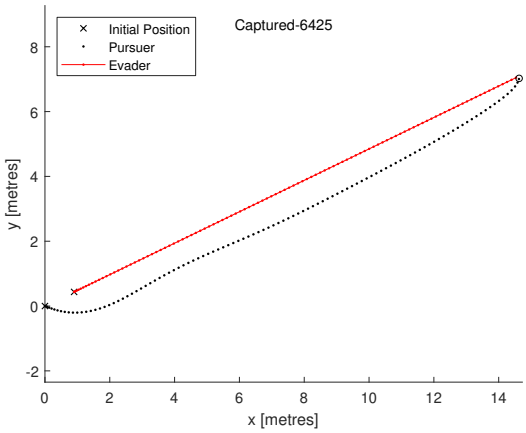

Fig. 6. Robot Path for Neurocontroller with Best $F_1 = 2.9279$ Performance Objective ($F_2 = 0.9095$).

As the objective of this research is to synthesize pursuer neurocontrollers that are not only capable of capturing the evader, but doing so using the minimum amount of energy, the best neurocontroller is not the one with the best F_1 value, but some compromise between best F_1 and best F_2 values. The top ranked neurocontroller as defined by the nondominating algorithm (as used in NSGA-II) is one such compromise. This neurocontroller is considered 'top ranked' in that it is not dominated by any other neurocontroller. Figure 7 shows the paths for the evader and pursuer agent with the top ranked (NSGA-II dominating) performance objective F_1 over all 30 runs. This pursuer neurocontroller has captured the evader, and follows a more direct, efficient path than that shown in Figure 6. This neurocontroller has a slightly less fit F_1 value, but has still managed to capture the evader. However, it has a fitter F_2 value, and hence it has consumed less energy than the neurocontroller used in Figure 6. Thus efficient and compact neurocontrollers with unsupervised learning can be evolved that are capable of capturing evaders while minimizing energy consumption.

Figure 8 shows the evolved neural network structure with the highest ranked neurocontroller over all runs and generations. Non-neuromodulated neurons are shown in blue, with regular connections being blue lines. Connections that have been disabled by the NEAT mutation algorithm are shown in grey. Neuromodulating neurons and connections are magenta, and neuromodulated neurons are red. The values displayed for each edge are the evolved neuromodulated weight values for the synapse associated with their respective edge. The inputs are numbered one through three, the outputs four and five. The neurocontroller's inputs are the evader range, evader bearing, and 'friend or foe' signals. The neurocontroller outputs are

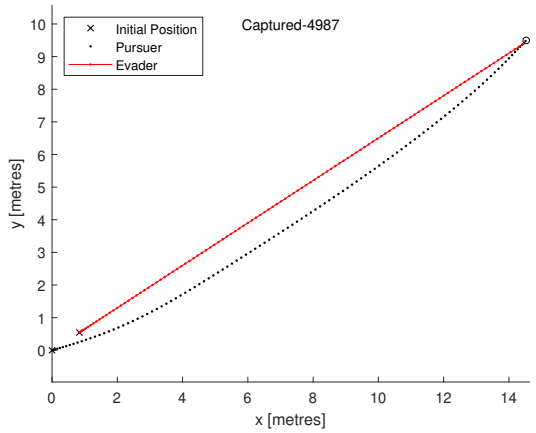

Fig. 7. Robot Path of Top Ranked Neurocontroller ($F_1 = 2.9199$, $F_2 = 0.9150$)

the robot's thrust and steering commands. This neurocontroller has two neuromodulating neurons (nodes 6 and 9), and one neuromodulated neuron (node 4). The synapse (connection) between nodes 5 and 7 has been disabled, and as a result, there is no input signal to node 7, and therefore no output signal from node 8. It is likely that eventually, this entire sub-circuit would be disabled by the evolutionary algorithm; however, it could also be removed with a simple pruning algorithm, yielding a compact, efficient neurocontroller that is also effective in capturing evaders and energy efficient.

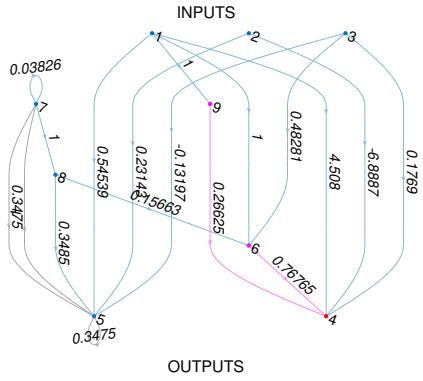

Fig. 8. Highest Ranked Evolved Lamarckian-Inherited Neuromodulated Neural Network Structure.

On average, after 150 generations, the Lamarckian-inherited neuromodulated neurocontrollers used 9.6 neurons and 23.2 connections. The average number (over all 30 runs) of nodes (neurons, all types), connections, and neuromodulated nodes per generation is shown in Figure 9. The bars display the standard deviation from the mean.

While, as shown in Figures 4 and 5, after 150 generations the objective fitnesses have converged to final values, the size of the neural networks has not yet converged to final values as shown in Figure 9. In other experiments involving thousands of generations, we have observed that the size of Lamarckian-inherited networks does converge to a final size in terms of

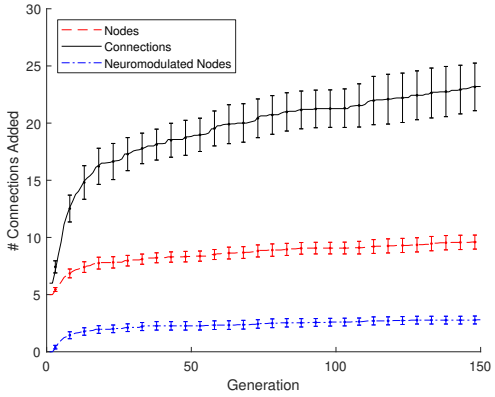

Fig. 9. Average Number of Nodes/Neurons per Generation

number of both neuromodulating and non-neuromodulating nodes and connections. In future publications we intend to show the common factors among these long term evolved neural networks.

VI. Conclusions

An evolutionary multiobjective neuromodulated controller with unsupervised learning, capable of controlling autonomous vehicles subject to the effects of mass and drag, is demonstrated. By optimizing the conflicting objectives of 'capturing evaders' and 'minimizing energy consumption', efficient neurocontrollers are evolved. Combining neuromodulation with evolution provides a powerful tool for exploring the search space. This combination gives the unique ability to test the search space with a genetic operator, and then improve upon these results using neuromodulated learning to adapt the network during operation. This approach allows exploration of the entire search space, and fine tuning to find each local maximum, until a solution with the global maximum is found. Including Lamarckian inheritance allows transfer of learned behaviour from parent to offspring populations, reducing convergence times and improving performance.

The ability of the proposed architecture to control vehicles subject to mass and drag is demonstrated in a series of experiments with a simulated evolved vehicle pursuing an evader vehicle. The results demonstrate that the augmentation of neuromodulated NEAT-MODS with Lamarckian inheritance gives an effective and efficient tool for generating neurocontrollers by allowing pretraining of offspring generations based on parent generations' unsupervised learning obtained while the neurocontrollers are evolving. Compact and efficient neurocontrollers for pursuer agents with nonzero mass and drag, capable of capturing an evader while simultaneously minimizing energy consumption, are consistently evolved.

References

[1] G. F. Miller and D. Cliff, "Co-Evolution of Pursuit and Evasion I: Biological and Game-Theoretic Foundations," School of Cognitive and Computing Sciences, University of Sussex, Brighton, Tech. Rep., 1994.

[2] S. F. Desouky and H. M. Schwartz, "Self-learning fuzzy logic controllers for pursuit evasion differential games," *Robotics and Autonomous Systems*, vol. 59, no. 1, pp. 22–33, 2011.

[3] D. Floreano, S. Nol, and F. Mondada, "Competitive Co-Evolutionary Robotics : From Theory to Practice," in *Proc. of The Fifth International Conference on Simulation of Adaptive Behavior (SAB), From Animals to Animats*, 1998.

[4] I. Showalter and H. Schwartz, "Lamarckian inheritance in neuromodulated multiobjective evolutionary neurocontrollers," in *27th Mediterranean Conference on Control and Automation, MED 2019 - Proceedings*. IEEE, 2019, pp. 63–68.

[5] O. Abramovich and A. Moshaiov, "Multi-objective topology and weight evolution of neuro-controllers," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, 2016, pp. 670–677.

[6] F. Silva, P. Urbano, and A. L. Christensen, "Online evolution of adaptive robot behaviour," *International Journal of Natural Computing Research (IJNCR)*, vol. 4, no. 2, pp. 59–77, 2014.

[7] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[8] I. Showalter and H. M. Schwartz, "A growing and pruning method for a history stack neural network based adaptive controller," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 5, 2004, pp. 4946–4951.

[9] Y. Niv, D. Joel, I. Meilijson, and E. Ruppin, "Evolution of Reinforcement Learning in Uncertain Environments: A Simple Explanation for Complex Foraging Behaviors," *Adaptive Behavior*, vol. 10, no. 1, pp. 5–24, 2002.

[10] F. Silva, P. Urbano, S. Oliveira, and A. L. Christensen, "odNEAT: An Algorithm for Distributed Online, Onboard Evolution of Robot Behaviours," *Artificial Life 13*, pp. 251–258, 2012.

[11] Richard K. Belew, J. McInerney, and N. N. Schraudolph, "Evolving Networks: Using the Genetic Algorithm with Connectionist Learning," in *Proceedings of the Second Artificial Life Conference*, 1991, pp. 511—-547.

[12] D. O. Hebb, *The Organization of Behavior, a Neuropsychological Theory*. New York: Wiley, 1949.

[13] A. Soltoggio, J. a. Bullinaria, C. Mattiussi, P. Dürr, and D. Floreano, "Evolutionary Advantages of Neuromodulated Plasticity in Dynamic, Reward-based Scenarios," in *Artificial Life XI: Proceedings of the 11th International Conference on Simulation and Synthesis of Living Systems (ALIFE 2008)*, vol. 2, 2008, pp. 569–576.

[14] P. S. Katz and R. M. Harris-Warrick, "The evolution of neuronal circuits underlying species-specific behavior," *Current Opinion in Neurobiology*, vol. 9, no. 5, pp. 628–633, 1999.

[15] R. Isaacs, *Differential games*. Dover, Mineola, NY: Wiley, 1965.

[16] Shen Hin Lim, T. Furukawa, G. Dissanayake, and H. Durrant-Whyte, "A time-optimal control strategy for pursuit-evasion games problems," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*. IEEE, 2004, pp. 3962–3967 Vol.4.

[17] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[18] J. Reisinger, E. Bahceci, I. Karpov, and R. Miikkulainen, "Coevolving Strategies for General Game Playing," in *2007 IEEE Symposium on Computational Intelligence and Games*, 2007, pp. 320–327.

[19] K. O. Stanley, B. D. Bryant, and R. Miikkulainen, "Real-time Learning in the NERO Video Game," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 6, pp. 653–668, 2005.

[20] K. O. Stanley and R. Miikkulainen, "Competitive coevolution through evolutionary complexification," *Journal of Artificial Intelligence Research*, vol. 21, pp. 63–100, 2004.

[21] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[22] J. D. Knowles, R. A. Watson, and D. W. Corne, "Reducing local optima in single-objective problems by multi-objectivization," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2001, pp. 269–283.

[23] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[24] M. S. Norouzzadeh and J. Clune, "Neuromodulation Improves the Evolution of Forward Models," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference - GECCO '16*, 2016, pp. 157–164.

[25] K. Doya, "Metalearning and neuromodulation," *Neural Networks*, vol. 15, no. 4-6, pp. 495–506, 2002.

[26] J.-B. Lamarck, *Philosophie Zoologique*. Paris: Germer Baillière, 1830.