

Article

Neuroevolution Application to Collaborative and Heuristics-Based Connected and Autonomous Vehicle Cohort Simulation at Uncontrolled Intersection

Frederic Jacquelin ^{1,*}, Jungyun Bae ^{1,2}, Bo Chen ^{1,3} and Darrell Robinette ¹

¹ Department of Mechanical Engineering-Engineering Mechanics, Michigan Technological University, Houghton, MI 49931, USA

² Department of Applied Computing, Michigan Technological University, Houghton, MI 49931, USA

³ Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI 49931, USA

* Correspondence: fjacque@mtu.edu

Abstract: Artificial intelligence is gaining tremendous attractiveness and showing great success in solving various problems, such as simplifying optimal control derivation. This work focuses on the application of Neuroevolution to the control of Connected and Autonomous Vehicle (CAV) cohorts operating at uncontrolled intersections. The proposed method implementation's simplicity, thanks to the inclusion of heuristics and effective real-time performance are demonstrated. The resulting architecture achieves nearly ideal operating conditions in keeping the average speeds close to the speed limit. It achieves twice as high mean speed throughput as a controlled intersection, hence enabling lower travel time and mitigating energy inefficiencies from stop-and-go vehicle dynamics. Low deviation from the road speed limit is hence continuously sustained for cohorts of at most 50 m long. This limitation can be mitigated with additional lanes that the cohorts can split into. The concept also allows the testing and implementation of fast-turning lanes by simply replicating and reconnecting the control architecture at each new road crossing, enabling high scalability for complex road network analysis. The controller is also successfully validated within a high-fidelity vehicle dynamic environment, showing its potential for driverless vehicle control in addition to offering a new traffic control simulation model for future autonomous operation studies.

Keywords: optimal control; intelligent systems; artificial intelligence; Neuroevolution; uncontrolled intersection; minimum energy control



Citation: Jacquelin, F.; Bae, J.; Chen, B.; Robinette, D. Neuroevolution Application to Collaborative and Heuristics-Based Connected and Autonomous Vehicle Cohort Simulation at Uncontrolled Intersection. *Eng* **2023**, *4*, 1320–1336. <https://doi.org/10.3390/eng4020077>

Academic Editor: Goncalo Jesus

Received: 16 February 2023

Revised: 27 April 2023

Accepted: 28 April 2023

Published: 1 May 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Thanks to the increased interest in driverless and connected vehicle technology, new traffic control concepts are being investigated. This paper looks at the use case of uncontrolled intersection simulation where connectivity provides the opportunity to further reduce traffic congestion and energy usage. The increased complexity in controlling the flow at an intersection in real time is a challenge that Neuroevolution, the process of transforming streaming information into an optimal control strategy via evolutionary optimization, is proposed to address here. This process is explained and described in the technical section. Both CAV cohorts and single-vehicle operations are considered. Two control concepts are proposed and compared. Firstly, a collaborative NeuroController-based architecture is developed. This concept relies purely on Neuroevolution to control the speed and the priority of vehicles entering an intersection. Secondly, a rule-based architecture is proposed. This architecture uses heuristics for vehicle prioritization, leaving the vehicle speed control to a set of simpler NeuroControllers. The simplicity, reusability and scalability of the second concept are demonstrated and retained as feasible architecture. While trained in a simple traffic environment, the resulting controller is then tested within

a high-fidelity vehicle dynamic simulation environment, using detailed powertrain and chassis physics-based models. The heuristics and its NeuroController set are then used to study the impact on traffic flow when compared to controlled intersections and a risk severity reference scenario where the vehicles stay at the speed limit (but include a collision count). The concept is then applied without any necessary modifications to the turning lanes, demonstrating scalability and flexibility as a driverless simulation component for transportation studies.

The interest in autonomously operating uncontrolled intersections stems from the need to improve safety and congestion within urban and extra-urban environments. The large number of accidents occurring at intersections prompted the proposal of specific traffic rules [1] as well the development of classification methodologies to understand and potentially mitigate safety critical occurrences. For example, safety heatmaps [2] have been proposed to better characterize the probability and the severity of uncontrolled intersection operation. Other classification methods via clustering use traffic flow and the geometry of uncontrolled intersections to enable the definition of different delay regimes [3]. With the lack of measured data for CAV vehicles, such assessments can be difficult to make and validate. The control of autonomous vehicles within traffic flow simulations is a strong factor in the results, hence requiring scrutiny. This requires the development of fast and very flexible traffic flow simulations to overcome the shortcomings of legacy tools in this domain. Crossing regimes within uncontrolled intersections have been studied using a MATLAB simulation and a Behavior Tree Controller, which resulted in three emergent regimes across low to high congestion scenarios [4]. The challenge of creating collision-free controllers was concluded. The Ideal Driving Model (IDM) [5] has been a very popular car-following model for micro-traffic simulation and was repurposed into a Generalized Intelligent Driver Model (GIDM) [6] to enable basic uncontrolled intersection flow simulation for comfort and safety studies. More complex methodologies make use of the Markov decision process [7], as well as Game theory which was demonstrated to simulate realistic behavior at uncontrolled intersections within manageable complexity [8,9]. More conventional optimal control methodologies using Model Predictive Control [10–12] making use of ADAS sensors have been developed. Such an approach has been successful in the vehicle and powertrain control engineering community. However, conventional optimal control methods, including dynamic programming (DP), require longer development time and higher compute power and are limited to the use of simplified plant models. Recently, these methods have increasingly been integrated with Machine Learning algorithms such as support vector machines [13] to enhance their inference capabilities. Mixed Integer Linear Programming (MILP) has been recently used to optimize vehicle scheduling for complex, uncontrolled intersections with multiple turning lanes [14]. In order to reduce the MILP computation time, which increases with the number of vehicles, Monte Carlo Tree Search (MCTS) and heuristics, which include combining vehicles into groups when in close proximity, have been proposed instead [15].

We propose that running an optimization during operation be replaced by inferring the optimal strategies based on prior training. This provides the advantage of low computing requirements with faster than real-time capabilities. The use of Neuroevolution enables new learning features to be added organically as needed within the neural network architecture and reduces the need for methodologies to change when complexity and dimensionality increase. Hence, only the objective function, input signals and topology changes may be required in these cases. We also demonstrate that control design changes can be mitigated by decomposing the problem into solving for non-turning lanes first. The resulting concept can then be used to integrate new turning lane concepts which enable higher speed throughput than previously achieved while maintaining low lateral accelerations. Indeed, mixing turning and non-turning operations inherently constrains the speed achievable for all vehicles and forces sub-optimal operation. These limitations prompted the authors to investigate the use of Artificial Intelligence, which demonstrated high energy savings with large heterogeneous cohorts operating at controlled

intersection [16] and for local powertrain adaptive control [17]. Artificial intelligence offers a simple and fast development framework, low compute power requirements and high reusability thanks to its adaptiveness to a wide range of dynamics. This is a critical factor in enabling our large heterogeneous CAV cohorts operation demonstrator [16] to be adapted to the uncontrolled intersection use case while retaining its inherent capabilities to deal with noise and dynamics perturbation as demonstrated when simulated within a high-fidelity chassis simulation in CarSim.

2. Materials and Methods

In this specific engineering application, Neuroevolution is the process of interlinking a system's sensors and actuators via a neural network architecture and iteratively optimizing its topology and performance within the operating environment. Here the environment is chosen as an uncontrolled intersection where large CAV cohorts interact, as shown in Figure 1. The NeuroController performance is driven by evolutionary optimization around an objective function. Various applications of Neuroevolution demonstrated its fast and effective learning capability [18], including when dealing with multi-objective functions [19].

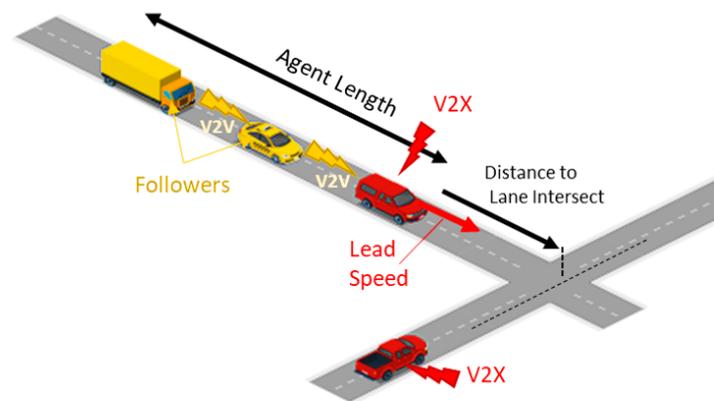


Figure 1. Agent definition as a cohort of CAV vehicles within the uncontrolled intersection.

Video game playing [20] has been popular in demonstrating this technique's ability to beat human players in real time [21,22]. Engineering applications have recently been published around antenna beam forming control [23], UAV control [24] and swarm robotics [20,25] using this method. While not yet as popular as conventional optimum techniques and deep learning, Neuroevolution can easily integrate within control architectures and heuristics to solve complex and system of systems control problems in a simpler manner while enabling adaptive and real-time performance with low compute power, as demonstrated in previous research [16]. Therefore, the Neuroevolution process proposed here is based on previous experience in developing predictive energy management functions for CAV cohorts and vehicles. Previous applications to vehicle speed optimization at controlled intersections and local powertrain control optimization form the basis of this work. In doing so, a base neural network architecture is chosen to consist of two hidden layers. The hidden layers contain one less neuron than the input layer. For this use case, the neural network has one output neuron, namely the speed target of the vehicle. In the first step, the weights of the neurons are used as the optimization variables. The activation functions are fixed and set with a Rectified Linear Unit (**ReLU**) function:

$$\text{ReLU}(\mathbf{a}) = \max(\mathbf{0}, \mathbf{a}). \quad (1)$$

when \mathbf{a} is defined as the sum of the weights \mathbf{W} and bias \mathbf{Bias} at a node:

$$\mathbf{a} = \sum \mathbf{IN} \cdot \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} \\ W_{41} & W_{42} & W_{43} & W_{44} & W_{45} \end{bmatrix} + Bias_{IH_1}, \quad (2)$$

when, in this example, the computation is performed at the first hidden layer \mathbf{H}_1 (here with four neurons), which receives the values from the input vector \mathbf{IN} of size five. \mathbf{IN} is defined later in this section.

The **ReLU** function has low computing requirements and has been demonstrated to be feasible for real-time control on standard onboard vehicle controllers in previous research. In the second step, the direct optimization of the activation functions is also performed. Additionally, a Weight Agnostic Neural Net (WANN [26]) process is demonstrated, which also provides unique insights on choosing an objective function. The topography of the neural network is, however, left unchanged to minimize optimization time within the current computing environment. Such optimization is extremely long and still prohibitively computing-intensive. Past experience and engineering judgment are preferred at this stage of applied research. Per the System of Systems (SoS) definition, each vehicle is considered autonomous and capable of transmitting and receiving information from and to the environment, including other connected vehicles, hence permitting emergent behavior to form. It is assumed that all vehicles are connected and that no interaction with unconnected vehicles is included. Therefore, this use case targets SAE level 5 driverless operation at a 100 % penetration within the traffic flow. While futuristic, the goal is to provide the ability to study traffic control in future environments and enable innovative traffic infrastructure to be identified early on. Two Neuroevolution architecture concepts are proposed as shown in Figure 2. From this point on, both single CAVs and cohorts of CAVs will be referred to as “agents” for clarity.

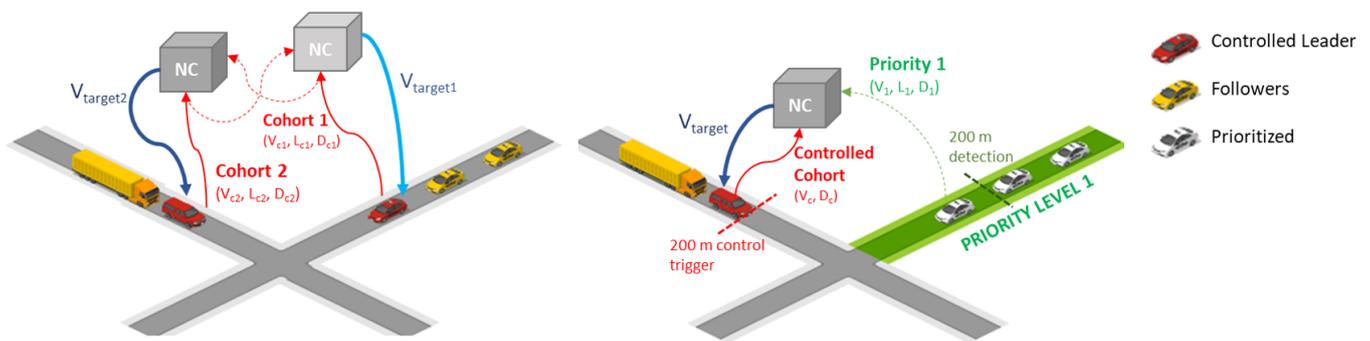


Figure 2. Two-interaction training example. **Left:** Concept 1 with two collaborative NeuroControllers (NCs). **Right:** Concept 2 with heuristics prioritization and one NeuroController.

Concept 1: Each agent uses its own neural network. For a two-agent interaction (cross-traffic interaction), two neural networks are necessary and the following inputs are chosen.

- Current speed of the agent V_c ;
- Current agent distance from the lane intersect D_c ;
- Current agent length L_c ;
- Current speed of the second agent (cross traffic agent) V_1 ;
- Current distance of the second agent to the lane intersect D_1 ;
- Second agent length L_1 .

Therefore,

$$\mathbf{IN} = \left[\frac{V_c}{V_{max}} \quad \frac{D_c}{D_{max}} \quad \frac{L_c}{L_{max}} \quad \frac{V_1}{V_{max}} \quad \frac{D_1}{D_{max}} \quad \frac{L_1}{L_{max}} \right]. \quad (3)$$

Note that each input is normalized by its maximum value. Restricting inputs to vary from a maximum range of 0 to 1 prevents creating a bias for larger value input parameters and enables the use of a smaller range of weights during optimization, hence saving optimization time. The second agent neural network is identical, with the proper input connection, such that the first three inputs always relate to the agent's own states. In this collaborative setup, two NeuroControllers are trained at once. Therefore, the number of optimization variables doubles and accounts for 96 optimization parameters. Initially, the training environment is set for two agent interaction scenarios. The immediate downside of this is that as the number of vehicle interactions increases, the number of neural networks and inputs increases, thus hastening forward unfeasible optimization time within the current compute resource limitations. Therefore, this solution is not currently scalable. In this paper, the training is thus limited to a maximum of three vehicle interactions, accounting for 174 optimization parameters. Increasing computing resources may, in the future, allow for this important ability to be more effectively utilized and scaled.

Concept 2: Heuristics is used to iteratively define the priority between agents coming to the intersection. This heuristics creates pairs and trios of interacting agents depending on their location, allowing reduced neural network sizes. The neural networks trained remained the same in both cases. The speed control is only activated if an agent receives a lower priority and therefore needs to modulate its speed. The highest priority vehicle, in this case, retains the right to operate at the speed limit. The heuristics are defined as:

- The first agent within 200 m of the intersection gets to level 1 priority and can remain at the speed limit.
- A second agent entering the 200 m threshold on a cross-traffic road is then set to level 2 priority and is thus requested to modulate its speed to avoid conflict with the level 1 priority agent.
- The third agent entering the 200 m threshold would then be at the lowest priority level 3—requiring it to modulate its speed to avoid conflict with the first and/or second agent depending on the road and lane it is occupying.
- As agents clear the intersection and arrive, the priorities are updated, always combining up to three agents at a time. By design, any lower-priority agent cannot pass the intersection until it becomes a priority level 1 agent. The rolling combination of agents upstream of the intersection permits high traffic density to be simulated.

The 200 m threshold was chosen so as to enable the controller to achieve low comfortable deceleration, especially when considering the edge case where a prioritized cohort is 100 m long and has not yet entered the intersection. Additionally, the priority to the right was not retained in order not to over-constrain the speed optimization and cause higher deceleration rates when a cohort, already within the detection zone, loses priority due to this rule. The two- or three-agent interaction NeuroControllers in this architecture can be trained separately within a simpler environment since collaboration is not required. Therefore, the heuristics enable simplification of the training, high scalability, as well as reusability, which will be demonstrated with multi-vehicle traffic flow and turning lanes in the later sections. In the training environment, the inputs to the neural networks are activated when the vehicle reaches the 200 m mark from the intersection and at least one cross-traffic agent with priority is already within 200 m from the intersection. The set of input can be simplified to:

- Current agent speed V_c ;
- Current agent length L_c ;
- Priority 1 agent speed V_1 ;
- Priority 1 distance from the intersection D_1 ;
- Priority 1 agent length L_1 .

Therefore,

$$\text{IN} = \left[\frac{V_c}{V_{\max}} \frac{L_c}{L_{\max}} \frac{V_1}{V_{\max}} \frac{D_1}{D_{\max}} \frac{L_1}{L_{\max}} \right], \quad (4)$$

when D_{max} is set to 200 m per the heuristic rule. Note that as the control starts when the controlled agent crosses the 200 m line, its distance to the intersection is known and it is not necessary to carry this into the input function as the neural network can then infer the remaining distance from the speed. For a two-agent interaction, this network requires one less input than Concept 1. It can be easily extended to a three-agent interaction by adding 3 additional inputs for the priority level 2 agent’s speed, distance and length if present:

$$IN = \left[\frac{V_c}{V_{max}} \frac{L_c}{L_{max}} \frac{V_1}{V_{max}} \frac{D_1}{D_{max}} \frac{L_1}{L_{max}} \frac{V_2}{V_{max}} \frac{D_2}{D_{max}} \frac{L_2}{L_{max}} \right]. \tag{5}$$

The heuristics prevent the need for any further interactions to be accounted for as its rules roll upstream the traffic flow forming pairs and/or trios. Respectively, 43 and 115 weights and biases will be optimized for the two- and three-level interaction neural networks proposed. The training environment is created in Matlab. Two perpendicular roads with one lane each define the intersection. The vehicle trajectory intersection point is at the origin of the x - and y -axes. The simulation timestep is set at 0.1 s. The maximum speed of the vehicle entering the intersection is limited to 55 mph.

A design of experiment (DOE) matrix is generated using a modified Latin Hypercube design. This allows the creation of a fine space-filling design defining the operation for two or three agent interactions, depending on which neural network is being trained. The goal is to cover as many agent interaction scenarios to train the NeuroController to generate a robust and adaptive optimal strategy with zero crashes. For Concept 1, the DOE variables cover the initial speed, length and distance from the intersection for each collaborative agent. For Concept 2, the DOE variables define the prioritized (level 1 or levels 1 and 2) agent speed and length and the trained agent’s initial speed and distance from the intersection. The trained agent starts upstream of the 200 m threshold, while the priority agent starts at the threshold. The variable range used is shown in Tables 1 and 2. To create a realistic combination of conditions, the Latin Hypercube is used to create one hundred base scenarios.

Table 1. Variable range used in the Base Design for concept 1.

Variable	Min	Max
Agents Initial Speed (V_c, V_1)	10 m/s	25 m/s
Agents Length (L_c, L_1)	15 m	100 m
Agents Distance (D_c, D_1)	50 m	500 m

Table 2. Variable range used in the Base Design for concept 2.

Variable	Min	Max
Agents Initial Speed (V_c, V_1)	10 m/s	25 m/s
Agents Length (L_1)	15 m	100 m
Agent Distance (D_1)	201 m	250 m

These scenarios assume that the max allowed road speed can be reached. However, when traffic density is high, agents may be operating at lower speeds, including the prioritized agents. Therefore, the 100 scenarios are replicated but with a maximum speed limit lower than the current maximum allowed speed on the road, down to 5 m/s. These two sets are further modified by creating 40 additional scenarios where all agents have the same length and speed limits to force unique or edge cases (Figure 3).

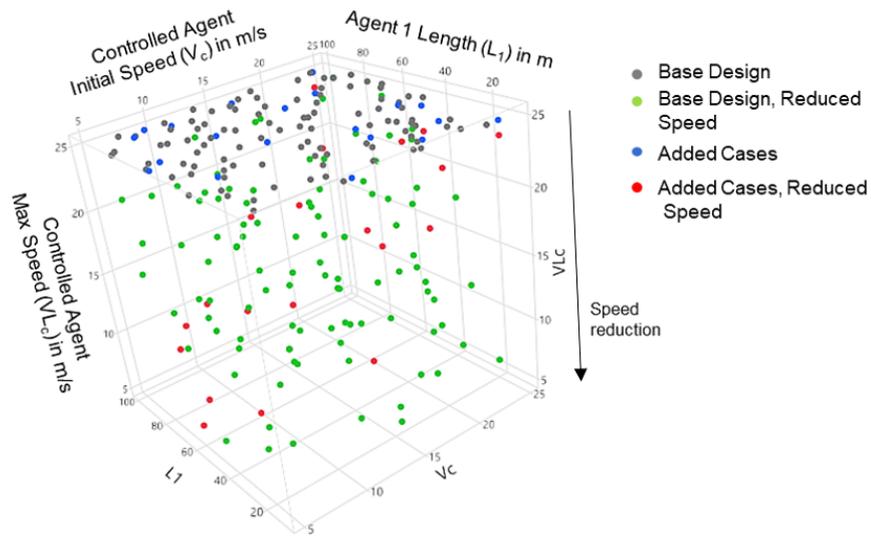


Figure 3. DOE matrix representation for 3 variables including the controlled agent’s initial speed, maximum allowed speed, and the cross-traffic agent length. The max allowed speed axis provides a better view of the reduced speed DOE scenarios.

The minimum agent length encompasses a two-vehicle cohort scenario based on the median US vehicle length with an equivalent gap length. The distance from the intersection range allows cohorts to enter the detection zone as randomly as possible based on their initial speeds so as to improve the training comprehensiveness.

For each optimization iteration, 100 combinations of neural network weights and biases are created, meaning that the Particle Swarm Optimization (PSO) uses 100 particles at once. Each particle, defining the NeuroController’s weights and biases, is simulated across the 240 DOE scenarios (see Figure 4). After each iteration, the PSO updates the particle values based on the objective function results. The training objective function is defined as:

$$ObjF1 = CrashPenalty - \frac{\int_{t=0}^T V_c dt}{T} \tag{6}$$

when T is total simulation time. This objective function is meant to minimize time at the intersection without the occurrence of crashes. The crash penalty is set to a high number, making any occurrence within the 240-scenario set penalizing enough to significantly impact the objective function. Note that an energy reduction (minimizing acceleration and deceleration events only) is introduced in the WANN section for discussion so as to demonstrate the importance of the objective function choices being made when developing these controllers:

$$ObjF2 = CrashPenalty + \int_{t=0}^T V_c \times |A_c| dt, \tag{7}$$

when A_c is the acceleration of the controlled agent. This objective function is of interest as real-world driving conditions have a significant impact on the energy consumption of electrified powertrain [27] and autonomous vehicles [28]. This element may need to be taken into account along with the main safety improvement and congestion reduction metrics, especially if the vehicle motion aggressiveness is exacerbated [29].

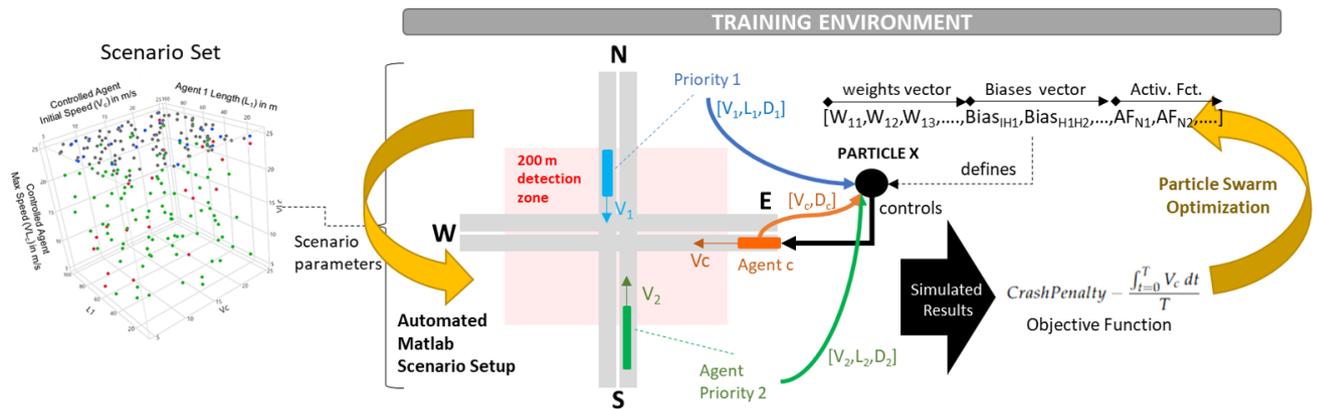


Figure 4. Training environment setup: for all Design of Experiment scenarios, the agent-based simulation is run and the objective function output. The PSO algorithm adjusts each particle’s information defining the controlled agent NeuroController to minimize the objective function. Note that the activation function vector was added in a later step.

3. Results

3.1. Training Results

A 52-core dual-Xeon desktop computer is used for the training. This maximizes the use of parallel processing, which is compatible with the PSO algorithm process. The set of 240 simulations takes approximately 10 s to simulate. More complicated tools, while compatible with the process of Neuroevolution, would heavily affect runtime. Convergence was twice as fast with concept 2, at around five hours. Both concepts were successful in resulting in zero crashes across the DOE scenarios. While having similar behavior, the main difference in the results stems from the objective function choices.

As Figure 5 shows, the Objective Function 1 solution shifts the controlled agent’s position from the prioritized vehicle early on and avoids any speed fluctuation within the intersection area, which enables a more deterministic approach to the uncontrolled intersection operation. The other solutions would bring uncertainty from any system trying to classify safety at the intersection. They also come closer to the prioritized agent and hence can be qualified as the more aggressive solutions. The energy objective function, however, uses 2% and 4% less energy than the Mean V and Reduce D Mean V functions operate, respectively, when using a 15,000 lbs delivery electric vehicle model as the lead vehicle. Further investigation into the tuning and blending of these objective functions is described in the next section.

3.2. WANN and Activation Function Optimization Discussion

As the ReLu activation function is uniquely used, additional optimization was performed, adding four activation functions to the PSO particle vector:

- ReLu with encoding as 1;
- Linear with encoding as 2;
- Sigmoid with encoding as 3;
- Radial Basis with encoding as 4.

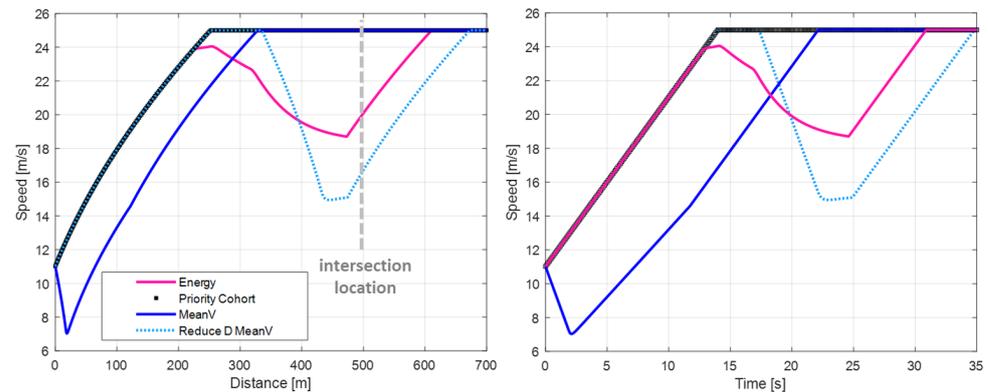


Figure 5. Three-speed profiles with different objective functions are shown. The mean speed objective function (Mean V) forces an early deceleration and crosses the intersection at top speed. If the mean speed value is computed only before the intersection (Reduce D Mean V), it delays its deceleration and only accelerates again within close distance to the intersection. The energy-based objective function acts similarly to the latter but reduces the speed fluctuation range to minimize losses.

The resulting optimization showed a 3% improvement in the objective function when using a mix of Linear and ReLu activation functions. The ReLu was therefore retained as the benefit of a more complex neural network was insignificant to the results. However, a WANN approach [30] was also performed. This approach permits the permutation of the four activation functions across all the neurons while keeping all the weights fixed to a unique value across the nodes. Iteratively changing the fixed weight value is also performed. The number of simulation runs reached 262,144 in order to cover the network design space finely. The resulting simulation information not only allows us to understand if an activation function is more adequate than another at a given location but also how different objective functions are impacted, as shown in Figure 6. In agreement with the prior optimization step, the result shows that both the ReLu and the Linear activations are performing equally well. More importantly, as this process is not driven by an optimization around a specific objective function, the raw simulation results can be used to investigate various objective functions, and identify the limitation and robustness of each.

Figure 6 provides a comprehensive view of various performance parameters. The columns represent the various nodes belonging to the output and hidden layers. Each node has four activation choices which are listed as sub-indexes at the bottom of the graph (activation function encoding from 1 to 4). For each sub-index, a three-dimensional scoring is provided based on the following attributes. First, the y -axis is chosen to represent the value of the Objective Function ObjF1 to be minimized as described in Equation (6). For every 2500 increments in ObjF1, a box is provided for the two additional attributes. Its color represents the number of crashes encountered across the training DOE runs (green is 0%). Its size represents the energy used across the DOE runs as defined by ObjF2 (see Equation (7)). In this way, it is clear that ObjF1 can be minimized and achieve 0% crash. Additionally, we see that the ReLu function, if used for ObjF1, also reduces energy usage. However, the minimum energy is achieved at higher ObjF1 values (between -5000 to $-10,000$), where ObjF2 is indeed minimized. The blending of the two objectives is, however, challenging as the probability of crash occurrence increases, as shown by the yellow boxes separating ObjF1 and ObjF2 minimums. Therefore, this result shows that trying to find an equilibrium between the time at the intersection and acceleration-based energy consumption is unfeasible or at least leads to non-robust controllers—and that a two-level optimization would be required. This is compatible with the idea that local vehicle energy reduction can be achieved by using a local predictive energy management powertrain strategy that relies on the speed profile of the vehicle [17]—as opposed to an all-in-one solution purely relying on cohort speed control.

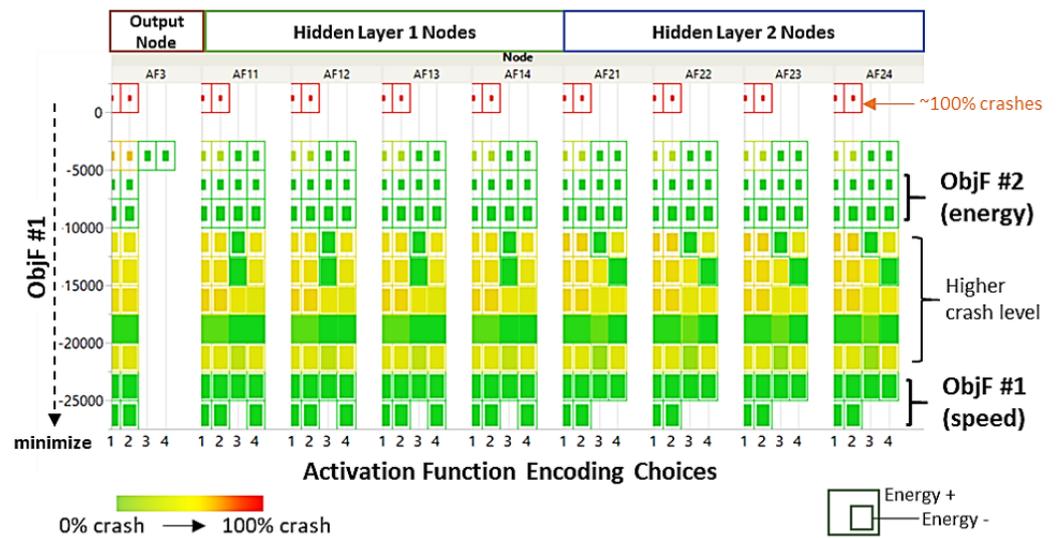


Figure 6. Results from the WANN process showing the two distinct objective function areas where no crash occurs. The size of each box is proportional to the energy-based objective function. The y-axis represents the mean speed-based objective function. The color indicates the probability of a crash (dark green is 0%). Nodes and activation encoding options are shown, identifying that both the ReLu (1) and the Linear (2) functions are the best choices.

3.3. Full Vehicle Dynamic Validation

A Simulink-CarSim co-simulation is created to validate the controllers. The NeuroController output is passed as an input to an Adaptive Cruise Control (ACC) model governing the speed of a cohort’s lead vehicle. The NeuroController is set up in Simulink and received the necessary input information from the ego vehicle (CarSim model) and other agents, as shown in Figure 7. While the vehicle does not instantly achieve the NeuroController speed target due to the ACC internal control and the vehicle and powertrain dynamic effects, the NeuroController accordingly adapts to keep the scenario free of crash events. This is due to the inherent capability of a NeuroController to adapt to new input signal values, thanks to its training across a wide range of dynamic scenarios. Moreover, note that as the NeuroController abides by the comfortable acceleration and deceleration rates, its modulation is well within the ability of the vehicle’s dynamic performance. Outside a certain range, it would be likely that the controller fails due to increasing deviation between the target and achieved states. At comfortable acceleration and deceleration rates, no such occurrences were reported.

While the NeuroControllers are designed to help study the potential of driverless technology on traffic congestion, the results of the high-fidelity simulation highlight a major challenge in operating uncontrolled intersections at high speed. Indeed, an Automated Emergency Braking (AEB) system would likely activate during close cross-traffic interaction. Therefore, the proper integration of this function in the AV stack is paramount and noted but is out of scope within the premise of this research paper. The important point is that the training of the NeuroController could be directly performed using CarSim if not for the increased simulation and optimization runtime limitations. Compared to the runtime of the Matlab training environment, the optimization time would increase from hours to weeks on the training desktop. However, unlike other optimal control methods, such as dynamic programming and Model Predictive Control, the current process does not require the plant model simplification or translation into a specific optimization language. Neuroevolution can use CarSim directly as a black box if a cloud-based computing resource is available.

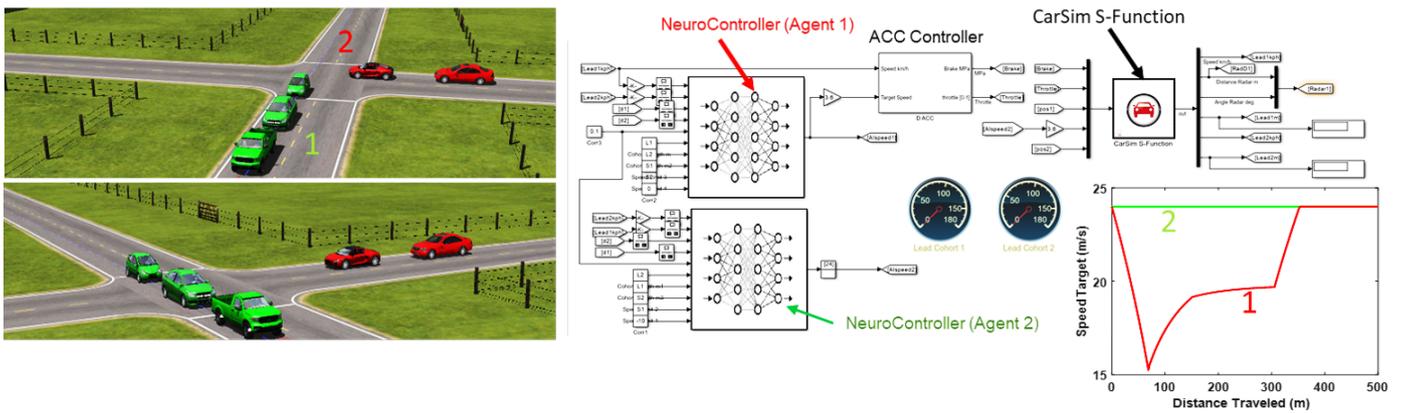


Figure 7. Concept 1 Collaborative agent setup in Matlab Simulink–CarSim co-simulation. Both NeuroControllers collaborate to safely operate three- and two-car CAV cohorts at an uncontrolled intersection. In this case, cohort 1 gets priority and cohort 2 modulates its speed accordingly. At 300 s, cohort 2 clears the intersection and goes back to the speed limit.

3.4. Traffic Flow Simulation and Results

3.4.1. Simple Intersection

A two-way intersection is chosen as a simple use case as shown in Figure 8. Traffic is generated stochastically at each road and lane leading to the intersection. The intensity of traffic is generated using a Monte Carlo-based agent generator, with a probability function deciding when and if an agent is created while assigning it an initial speed and length. Each agent is created 500 m from the intersection and assumes the cohort is fully formed (see [31] for reference on formation strategies). Once entering the intersection 200 m detection zone, the heuristics assign a priority. Per concept 2, once assigned a priority level 2 or above, an agent would receive one to two prioritized (level 1 or levels 1 and 2) agent information to modulate its speed. Additionally, when on the same road and lane, an agent uses the Gipps driver model [32] to maintain a safe gap to any other agent in front of it. This limits its maximum speed and rear-ending events. The Gipps model uses the same comfortable acceleration and deceleration as the NeuroController. The Gipps model relies on a safe distance and speed computation:

$$d_s \geq s_o + v\Delta t + \frac{v^2}{2b} - \frac{v_{lead}^2}{2b} \tag{8}$$

$$v_{safe} = -b\Delta t + \sqrt{b^2\Delta t^2 + v_{lead}^2 + 2b(d_s - s_o)} \tag{9}$$

when the vehicle speed is v and s_o is the minimum safety distance between agents. These are used to compute the safe distance d_s . Gaps and safe velocity v_{safe} are then updated based on the cohort “comfortable” deceleration b . Δt is the simulation time step. Note that the NeuroController is only allowed deceleration that is less aggressive than -3 m/s^2 as we are interested in realistically identifying its performance in real-world situations. Therefore, in the absence of AEB, crashes will likely occur when traffic density is very high and the objective of optimizing mean vehicle speed is retained. This limitation is useful when realistically comparing this concept to the operation of a controlled intersection with the same traffic flow boundary conditions. The controlled intersection, while requiring agents to stop, is ideally crash-free. Furthermore, a reference scenario is added to measure each scenario run’s risk severity by counting collisions when agents only maximize their speed within the speed limit. In this reference case, the agents can target the speed limit or safe inter-agent speed depending on congestion, without any additional constraints. These two bounding cases are used to demonstrate concept 2 advantages in preventing crashes while improving traffic flow. Hence this experiment aims to highlight

unsafe conditions for driverless operations at uncontrolled intersections. A snapshot of the simulation is shown in Figure 8.

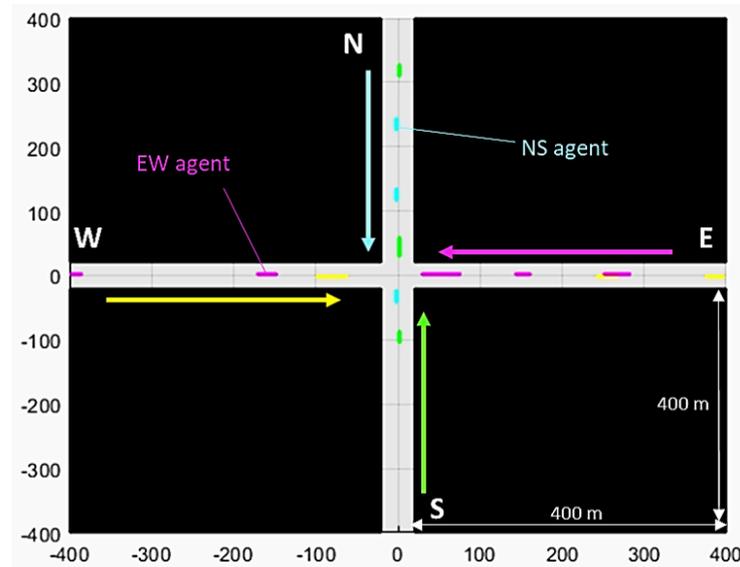


Figure 8. Snapshot from a scenario using raw simulation data. Note that the widths of the lane are not at scale for clarity. Maximum allowable agent length is 50 m in this specific scenario.

A DOE is created and applied to the three intersection control options. The DOE sets the agent generator frequency, different randomization seeds and the maximum length allowed for an agent when created [20–100 m]. The controlled intersection is set with a period of one minute with a 50% time split for the red and green timing. A total of 4760 scenarios are created and the simulation results are shown in Figure 9.

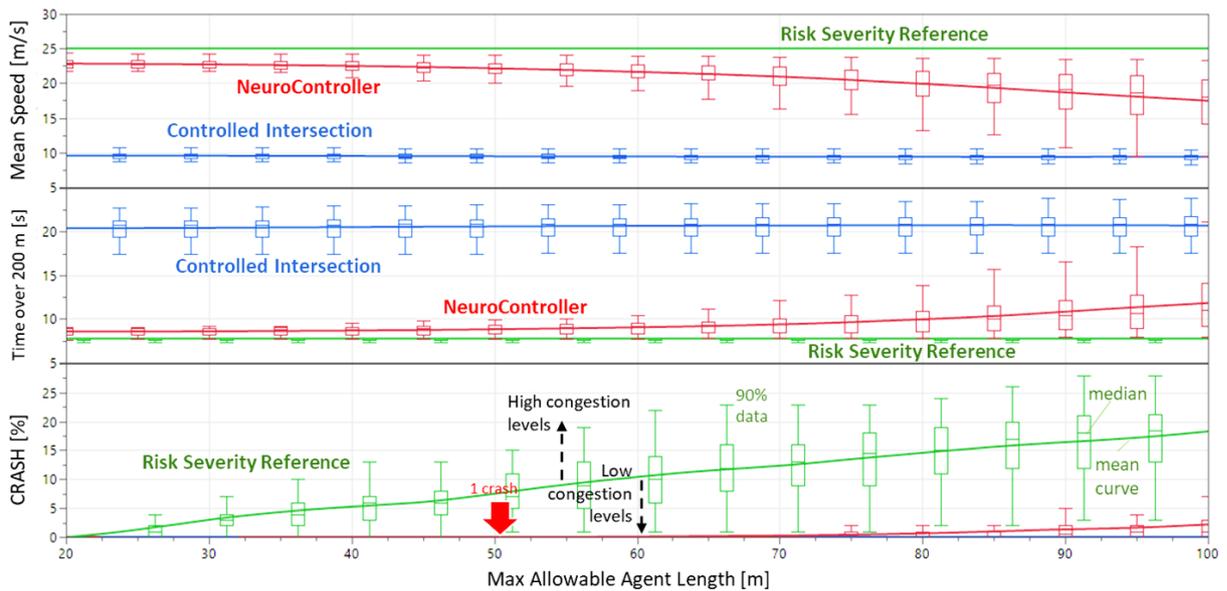


Figure 9. Results from 4760 DOE runs with the three intersection options: no control (risk severity) in green, uncontrolled (NeuroController) in red and controlled intersection in blue. The top graph compares the mean speed of the agents coming into the intersection. The middle graph compares the time spent within the 200 m zone upstream of the intersection. Lastly, the bottom graph shows the % crash. The agent length was chosen as the *x*-axis as it has the largest impact on the results.

The controlled intersection forces a mean speed of around 10 m/s at the intersection due to the stop-and-go behavior of the flow. Mean speed drops by one second across the max allowed agent length scale. The average time for vehicles within the 200 m zone upstream of the traffic light is 20 s. For the risk severity reference scenarios, the mean speed stays at 25 m/s with a residence time of 5 s. Under the allowed maximum agent length of 50 m, the optimal gap between the NeuroController and this reference condition is around 2 m/s for mean speed and 1 s residence time. At these conditions, the NeuroController achieves near-optimal performance while controlling complex conditions. Based on traffic congestion, a maximum of 2 m/s in mean speed drop is shown. This still enables twice the mean speed compared to the controlled intersection. Above the maximum allowable agent length of 60 m, the NeuroController starts showing crash outliers, with a mean % crash of up to 2.5% when 100 m agents are present in the traffic flow.

For agents under the 50 m length range, the NeuroController is stable and close to the optimal effectiveness without crash occurrences. For small agents, the NeuroController performance stays below the ideal performance, demonstrating that single vehicles would operate similarly to small cohorts. The cohort length limitation can be mitigated using additional lanes. This would require the cohort above this length to split and operate on parallel lanes so as to become one single agent with a lower length. Therefore, the concatenation of agents on lanes can make use of Concept 2 without the above limitation.

The NeuroController drastic reduction in speed fluctuation is shown in Figure 10. A maximum speed decrease of 10 m/s is required in contrast to the stop-and-go behavior of the controlled intersection.

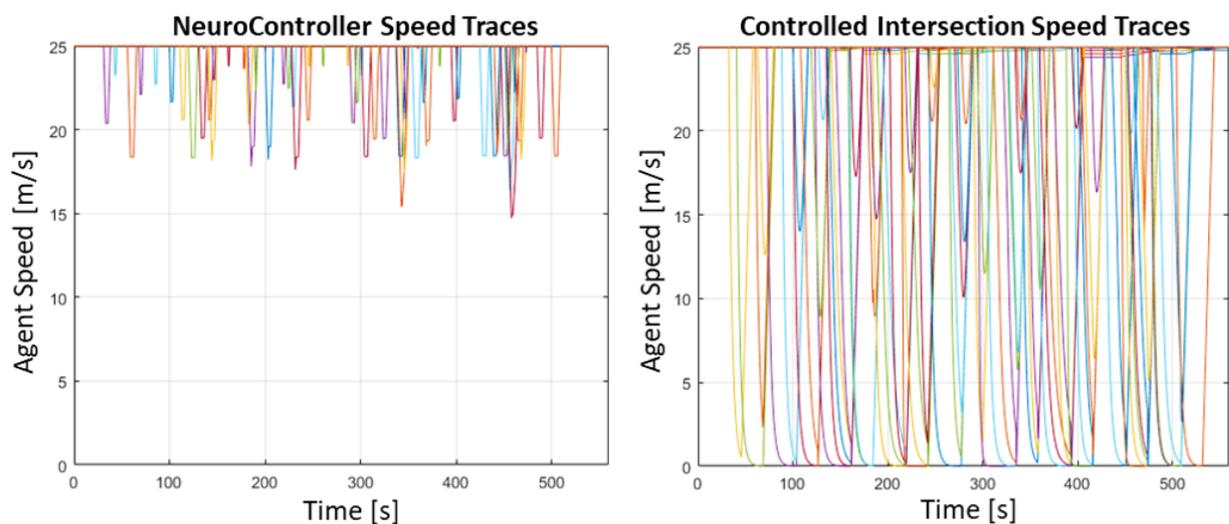


Figure 10. Comparison of one hundred agents' speeds under uncontrolled (**left side**) and controlled intersection (**right side**) operations. Note the simulation time for the uncontrolled scenario increases by one minute due to the slower throughput with the traffic light.

Assuming a 15,000 lbs vehicle, the increased propulsion energy required to achieve the speed traces above, for the uncontrolled and controlled intersections, compared to a steady 55 mph steady operation is shown in Figure 11. Note that depending on the level of powertrain electrification, the overall energy increase will be reduced thanks to regenerative braking.

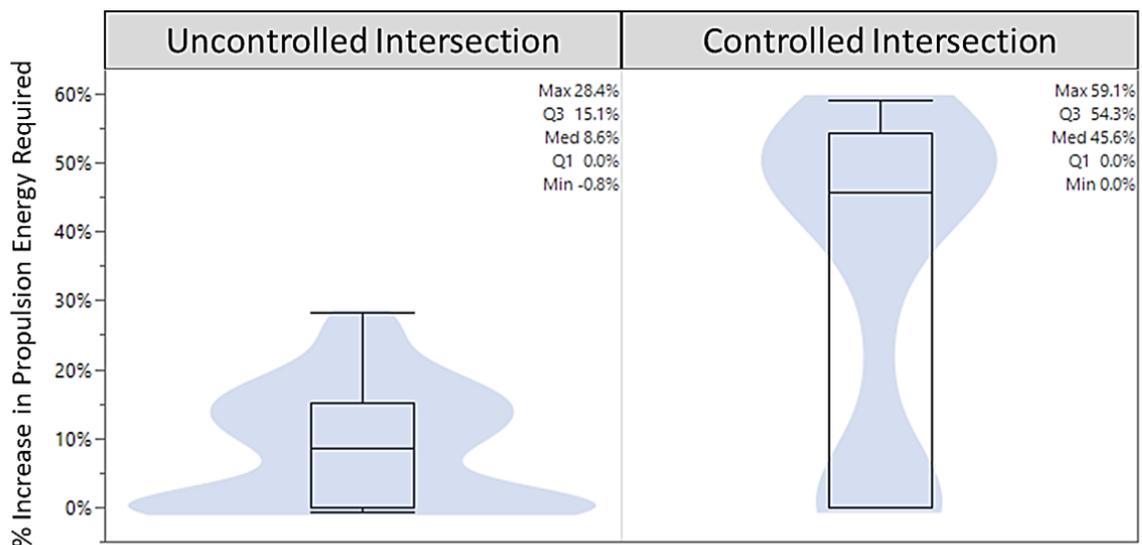


Figure 11. Increased propulsion energy required to operate a 15,000 lbs vehicle at the uncontrolled and controlled intersections when compared to a steady state operation at 55 mph across the DOE scenarios.

3.4.2. Adding Turning Lanes

Agent's turning from one road to the other has not been simulated so far due to the fact they would need to decelerate to a realistic and comfortable speed during the turning operation. Therefore, the optimization of their mean speed is baseless in the current intersection setup. They would indeed receive the lowest priorities and only perform the turn when safe. This would, however likely cause high congestion on the turning lanes when the traffic density is high. However, Concept 2 is compatible and can be reused as part of a new intersection design. A turning lane is designed to deviate from the main road at a constant curvature allowing safe and comfortable lateral acceleration levels without change in allowable speed, as shown in Figure 12. For the N-S right-hand turning lane, agents can free flow from the main road without a speed decrease. When reaching the E-W road, they will enter the flow using the same Concept 2 heuristics. If crossing the 200 m mark toward the E-W road first, the N-S agent will have priority. If an E-W agent is already within 200 m from this intersection, then the N-S agent will use the 1 vehicle interaction NeuroController to modulate its speed before merging on the E-W lane. In the case of the left-hand turn, the agent will need to use a one-vehicle interaction NeuroController when splitting from the N-S main road and crossing the S-N road. It will use either a 1 or 2 vehicle interaction NeuroController when crossing the E-W road to merge into the W-E road. In either case, the current Concept 2 does not need any modifications or retraining to achieve turning operation.

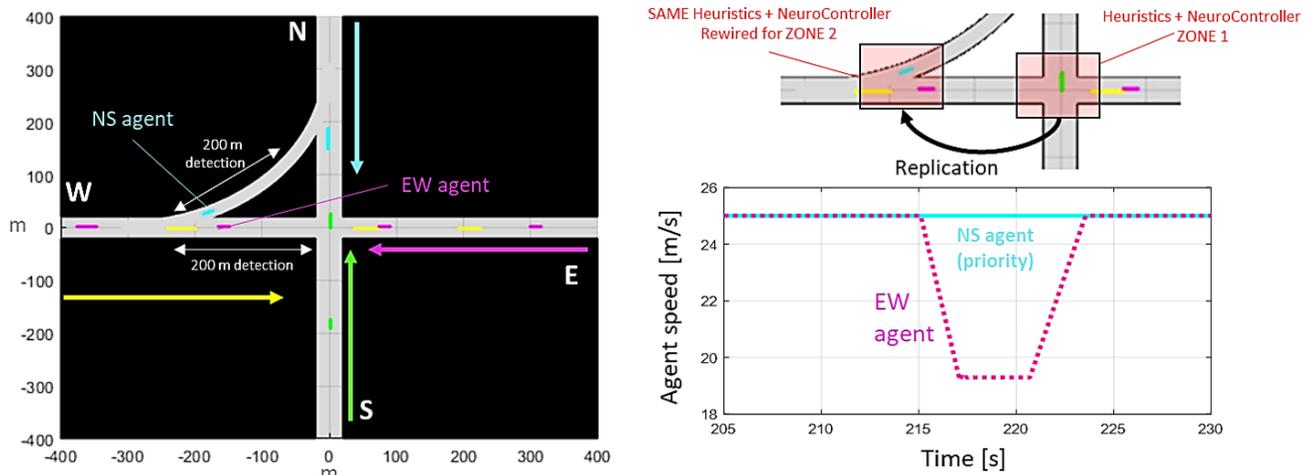


Figure 12. Replication and rewiring of Concept 2 NeuroController and heuristics at the N-S turning lane merging onto E-W road. In this example, the E-W agent arrives last within the lane merging detection zone and is required to control its speed accordingly. The controller target speed trace is shown on the right.

4. Discussion

Neuroevolution is a very flexible optimization technique that enables various optimal control architectures to be studied and compared rapidly. Two concepts are presented here, demonstrating the ability to generate a general collaborative control architecture or alternatively integrate it within a heuristic-based architecture. This work enables the inclusion of driverless technology in traffic simulation workflow and provides insight into how future vehicle technologies will impact transportation. While the heuristics approach proved more flexible in this application, it may not always be the right option. Collaborative NeuroControllers likely provide strong candidates for other problems seeking a real-time and adaptive solution and therefore should not be automatically discounted. The scalability of the solution is appealing as it only requires replication and rewiring of the neural network inputs to each new intersection boundary condition—between roads, merging lanes and other network design ideas. Additional heuristics can be added by concatenating cohorts' lengths when operating in different lanes to form one single agent. The current performance and extension potential are enabled by the use of a System of Systems architecture building, where the solution is autonomous and not dependent on any centralized control scheme. Creating road and intersection networks using this proposed fundamental replication technique would permit the study of emergent behaviors across large and uncontrolled networks for single vehicles and CAV cohort operation. This in turn may result in new requirements on information exchange required to further improve the traffic flow. However, the basic architecture should not change and still relies on localized learning and hence remains computationally inexpressive. Beyond computing requirements, the simplicity and ability to train controllers from any type of simulation, from low to high fidelity, are also very appealing, especially compared to conventional optimal control approaches requiring simplification and/or programmatic translation of the plant models. Based on these characteristics, this area of artificial intelligence is set to become extremely useful as systems and control complexity soars in the future while still keeping solutions simple and more explainable than, for example, deep learning techniques so far allow.

Author Contributions: Conceptualization, methodology, formal analysis, validation and original draft writing and editing by F.J.; Review and editing by all co-authors; Supervision by D.R., J.B. and B.C.; Project administration and funding acquisition by D.R. and F.J. All authors have read and agreed to the published version of the manuscript.

Funding: This material is partially based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Vehicle Technologies Office (VTO) award number DE-EE0009209.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Some of this material is based on work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Vehicle Technologies Office (VTO) award number DE-EE0009209. A special thanks to Erin Boyd, Daniel Nardozi and Danielle Chou at DOE for their support and guidance. We also thank Damon Becker at Applied Intuition for his support with CarSim.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Karimi, A.; Duggirala, P.S. Formalizing traffic rules for uncontrolled intersections. In Proceedings of the 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS), Sydney, Australia, 21–25 April 2020; pp. 41–50. [\[CrossRef\]](#)
2. Ma, Y.; Qin, X.; Grembek, O.; Chen, Z. Developing a safety heatmap of uncontrolled intersections using both conflict probability and severity. *Accid. Anal. Prev.* **2018**, *113*, 303–316. [\[CrossRef\]](#)
3. Datta, S.; Rokade, S.; Rajput, S.P.S. Classification of Uncontrolled Intersections Using Hierarchical Clustering. *Arab. J. Sci. Eng.* **2020**, *45*, 8591–8606. [\[CrossRef\]](#)
4. Safarov, K.; Kent, T.; Wilson, E.; Richards, A. Emergent Crossing Regimes of Identical Autonomous Vehicles at an Uncontrolled Intersection. *arXiv* **2021**, arXiv:2104.04150. [\[CrossRef\]](#)
5. Treiber, M.; Kesting, A. Traffic flow dynamics. In *Traffic Flow Dynamics: Data, Models and Simulation*; Springer: Berlin/Heidelberg, Germany, 2013.
6. Kreutz, K.; Eggert, J. Analysis of the Generalized Intelligent Driver Model (GIDM) for Uncontrolled Intersections. In Proceedings of the IEEE Conference on Intelligent Transportation Systems, Online, 16–17 June 2021; The Institute of Electrical and Electronics Engineers, Inc. (IEEE): Piscataway, NJ, USA, 2021; Volume 2021, pp. 3223–3230. [\[CrossRef\]](#)
7. Song, W.; Xiong, G.; Chen, H. Intention-Aware Autonomous Driving Decision-Making in an Uncontrolled Intersection. *Math. Probl. Eng.* **2016**, *2016*, 1–15. [\[CrossRef\]](#)
8. Li, N.; Yao, Y.; Kolmanovskiy, I.; Atkins, E.; Girard, A.R. Game-Theoretic Modeling of Multi-Vehicle Interactions at Uncontrolled Intersections. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 1428–1442. [\[CrossRef\]](#)
9. Nan, J.; Deng, W.; Zheng, B. Intention Prediction and Mixed Strategy Nash Equilibrium-Based Decision-Making Framework for Autonomous Driving in Uncontrolled Intersection. *IEEE Trans. Veh. Technol.* **2022**, *71*, 10316–10326. [\[CrossRef\]](#)
10. Jeong, Y. Stochastic model-predictive control with uncertainty estimation for autonomous driving at uncontrolled intersections. *Appl. Sci.* **2021**, *11*, 9397. [\[CrossRef\]](#)
11. Jeong, Y.; Yi, K. Target Vehicle Motion Prediction-Based Motion Planning Framework for Autonomous Driving in Uncontrolled Intersections. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 168–177. [\[CrossRef\]](#)
12. Park, S.; Jeong, Y. Proactive Motion Planning for Uncontrolled Blind Intersections to Improve the Safety and Traffic Efficiency of Autonomous Vehicles. *Appl. Sci.* **2022**, *12*, 11570. [\[CrossRef\]](#)
13. Jeong, Y.; Yi, K.; Park, S. SVM based intention inference and motion planning at uncontrolled intersection. *IFAC PapersOnLine* **2019**, *52*, 356–361 [\[CrossRef\]](#)
14. Yao, Z.; Jiang, H.; Jiang, Y.; Ran, B. A Two-Stage Optimization Method for Schedule and Trajectory of CAVs at an Isolated Autonomous Intersection. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3263–3281. [\[CrossRef\]](#)
15. Jiang, H.; Yao, Z.; Jiang, Y.; He, Z. Is All-Direction Turn Lane a Good Choice for Autonomous Intersections? A Study of Method Development and Comparisons. *IEEE Trans. Veh. Technol.* **2023**, 1–16. [\[CrossRef\]](#)
16. Jacquelin, F.; Bae, J.; Chen, B.; Robinette, D.; Santhosh, P.; Orlando, J.; Knopp, D. Connected and Autonomous Vehicle Cohort Speed Control Optimization via Neuroevolution. *IEEE Access* **2022**, *10*, 97794–97801. [\[CrossRef\]](#)
17. Jacquelin, F.; Bae, J.; Chen, B.; Robinette, D.; Santhosh, P.; Kraemer, T.; Henderson, B. Real Time Predictive and Adaptive Hybrid Powertrain Control Development via Neuroevolution. *Vehicles* **2022**, *4*, 942–956. [\[CrossRef\]](#)
18. Lehman, J.; Miikkulainen, R. Neuroevolution. *Scholarpedia* **2013**, *8*, 30977. [\[CrossRef\]](#)
19. Künzel, S.; Meyer-Nieberg, S. Evolving Artificial Neural Networks for Multi-objective Tasks. In *Applications of Evolutionary Computation: 21st International Conference, EvoApplications 2018, Parma, Italy, 4–6 April 2018*; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 671–686. [\[CrossRef\]](#)
20. Silva, F.; Correia, L.; Christensen, A. Dynamics of Neuronal Models in Online Neuroevolution of Robotic Controllers. In Proceedings of the Portuguese Conference on Artificial Intelligence, Azores, Portugal, 9–12 September 2013; pp. 90–101. [\[CrossRef\]](#)

21. Stanley, K.; Bryant, B.; Miikkulainen, R. Real-time neuroevolution in the NERO video game. *IEEE Trans. Evol. Comput.* **2005**, *9*, 653–668. [[CrossRef](#)]
22. Hausknecht, M.; Lehman, J.; Miikkulainen, R.; Stone, P. A Neuroevolution Approach to General Atari Game Playing. *IEEE Trans. Comput. Intell. Games* **2014**, *6*, 355–366. [[CrossRef](#)]
23. Kim, H.K.; Becerra, R.; Bolufé, S.; Azurdia-Meza, C.A.; Montejo-Sánchez, S.; Zabala-Blanco, D. Neuroevolution-based adaptive antenna array beamforming scheme to improve the v2v communication performance at intersections. *Sensors* **2021**, *21*, 2956. [[CrossRef](#)]
24. Kim, E.J.; Perez, R.E. Neuroevolutionary Control for Autonomous Soaring. *Aerospace* **2021**, *8*, 267. [[CrossRef](#)]
25. Galassi, M.; Capodici, N.; Cabri, G.; Leonardi, L. Evolutionary strategies for novelty-based online neuroevolution in swarm robotics. In Proceedings of the 2016 IEEE International Conference on Systems, Man and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016; pp. 002026–002032. [[CrossRef](#)]
26. Gadikar, P.U.; Ganesan, V.; Panda, P.K. Generalized Weight Agnostic Neural Networks for Configurable and Continual Autonomous Systems. In Proceedings of the ACM International Conference Proceeding Series, Online, 12–14 November 2021. [[CrossRef](#)]
27. Berry, I.M. The Effects of Driving Style and Vehicle Performance on the Real-World Fuel Consumption of US Light-Duty Vehicles. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2010.
28. Mersky, A.C.; Samaras, C. Fuel economy testing of autonomous vehicles. *Transp. Res. Part Emerg. Technol.* **2016**, *65*, 31–48. [[CrossRef](#)]
29. Sharer, P.; Leydier, R.; Rousseau, A. Impact of drive cycle aggressiveness and speed on HEVs fuel consumption sensitivity. In Proceedings of the SAE World Congress & Exhibition. SAE International, Sacramento, CA, USA, 9–12 July 2007. [[CrossRef](#)]
30. Gaier, A.; Ha, D. Weight Agnostic Neural Networks. *arXiv* **2019**, arXiv:1906.04358. [[CrossRef](#)]
31. Soni, A.; Hu, H. Formation Control for a Fleet of Autonomous Ground Vehicles: A Survey. *Robotics* **2018**, *7*, 67. [[CrossRef](#)]
32. Spyropoulou, I. Simulation using Gipps' car-following model—An in-depth analysis. *Transportmetrica* **2007**, *3*, 231–245. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.