applied sciences

MDPI

Article

Jam Mitigation for Autonomous Convoys via Behavior-Based Robotics

Calvin Cheung, Samir Rawashdeh and Alireza Mohammadi *

Department of Electrical and Computer Engineering, University of Michigan-Dearborn, Dearborn, MI 48128, USA
* Correspondence: amohmmad@umich.edu

Abstract: Autonomous ground vehicle convoys heavily rely on wireless communications to perform leader-follower operations, which make them particularly vulnerable to denial-of-service attacks such as jamming. To mitigate the effects of jamming on autonomous convoys, this paper proposes a behavior-based architecture, called the Behavior Manager, that utilizes layered costmaps and vector field histogram motion planning to implement motor schema behaviors. Using our proposed Behavior Manager, multiple behaviors can be created to form a convoy controller assemblage capable of continuing convoy operations while under a jamming attack. To measure the performance of our proposed solution to jammed autonomous convoying, simulated convoy runs are performed on multiple path plans under different types of jamming attacks, using both the assemblage and a basic delayed follower convoy controller. Extensive simulation results demonstrated that our proposed solution, the Behavior Manager, can be leveraged to dramatically improve the robustness of autonomous convoys when faced with jamming attacks and can be further extended due to its modular nature to combat other types of attacks through the development of additional behaviors and assemblages. When comparing the performance of the Behavior Manager convoy to that of the basic convoy controller, improvements were seen across all jammer types and path plans, ranging from 13.33% to 86.61% reductions in path error.

Keywords: autonomous convoy; behavior-based robotics; jamming

Citation: Cheung, C.; Rawashdeh, S.; Mohammadi, A. Jam Mitigation for Autonomous Convoys via Behavior-Based Robotics. *Appl. Sci.* **2022**, *12*, 9863. https://doi.org/10.3390/app12199863

Academic Editors: Jose Machado, Fabrizio Giulietti, Nadjim Horri and William Holderbaum

Received: 19 August 2022
Accepted: 27 September 2022
Published: 30 September 2022

1. Introduction

Ground vehicle convoys are utilized in both commercial and military applications in order to reduce costs, increase vehicle efficiency, and improve safety for the transport personnel. At the most basic level, a convoy is a group of two or more vehicles, traveling from a set origin to an objective destination under a single leader [1]. From a military perspective, the use of ground vehicle convoys to perform supply operations is an important part of an efficient logistics strategy. While several different transportation methods such as aircrafts, trains, and ships are available, ground vehicles still account for a significant portion of supply and equipment distribution due to the battlefield complexities and the need to embed protective measures such as gun trucks [2]. Similarly, the usage of ground vehicle convoys is vital in the commercial sector, due to the prevalence of paved roads and cost effectiveness when moving large or heavy materials [3].

With the advent of autonomous driving, there has been heavy focus on leveraging autonomous vehicles to improve convoys. Much of the research and development towards autonomous vehicle convoys focus around three iterative employment concepts: minimally manned (MM), partially manned (PM), and fully autonomous (FA) [4]. In a minimally manned system, the lead vehicle in the convoy has a human driver, while the follower vehicles operate autonomously with a safety rider monitoring autonomous performance, ready to take over in the event a fault occurs. In a partially manned system, the lead vehicle convoy has a human driver, but all the follower vehicles are autonomous and unmanned. In a fully autonomous system, all vehicles in the convoy, leader and followers, are unmanned

autonomous. In all cases, there are numerous benefits in applying autonomy to ground vehicle convoys. From both the commercial and military perspective, the utilization of autonomous vehicle convoys can reduce fuel consumption, reduce traffic congestion, and improve safety [2].

In all the autonomous vehicle convoy employment concepts, the utilization of wireless communications has been paramount in their development. Two important applications of wireless communications include inter-vehicle communications (IVC) and Global Positioning Systems (GPS). IVC is used to exchange vital information between vehicles in the convoy, including pose, heading, speed, acceleration, and maneuver intentions [5]. This information is used by follower vehicles in maintaining the desired convoy formation along the desired path. In addition, GPS is often used as a vital tool in providing position, navigation, and timing information for each vehicle [4]. In many cases, proper wireless communications are a necessity in the operation of an autonomous convoy, where disruption of the wireless communications system can cause the autonomous convoy to fail. The criticality of proper wireless communications extends beyond autonomous convoys and applies to various other autonomous applications such as surveillance [6], mining, and agriculture [4]. This outsized importance of wireless communications makes it a critical piece of the system to protect from potential cyber-physical attacks, such as jamming.

Jamming is a type of denial-of-service attack that attempts to disrupt or block wireless communications. Jammers interfere with a radio's ability to transmit or receive data, and are the most widely used denial-of-service attack for vehicular networks due to their effectiveness and simplicity [7–9]. The risk of jamming is especially prevalent in military applications, due to electronic warfare systems and strategies employed by adversaries to gain tactical advantages [9]. Given the importance of IVC in autonomous convoys, disruptions caused by jamming could cause instability in the following functionality of the autonomous convoys, or even potentially cause a complete breakdown of the convoy [4]. Since jamming is one of the most simple and effective denial-of-service attacks [10], it is important to understand the attack and the proper ways to defend against it. There have been some limited efforts focused more narrowly on the topic of jamming as it relates to convoys, with a greater body of work focused on investigating the effects of jamming on vehicular ad-hoc networks (VANET) in general [11–13].

On the topic of jamming and its effects of vehicular convoys, there has been some work in developing a framework to measure the impact of the attack and a convoy's resilience. Van der Heijden et al. examined three Cooperative Adaptive Cruise Control convoy controllers and how various attacks such as jamming and data injection impacted their respective performance [7]. Both an attacker model and an evaluation framework were developed for a thorough examination of what sort of attack detection and attack resilience algorithms should be used to resist the effects of attacks on the VANETs. In another convoy specific effort, Hu et al. investigated how stealthy jamming attacks impacted convoy stability [14]. The stealthy jammer performed jamming against a simulated convoy's basic safety message (BSM), which contains information such as vehicle speed, position, and acceleration, over the control channel of dedicated short-range communications (DSRC), but only when the probability of being detected was low. The probability of being detected was determined based on how significantly jamming at a particular time would affect the packet loss ratio of the communications. To detect the stealthy jams, they analyzed received power and transmission delays of messages to differentiate between jamming attacks and normal interference. The effects of jamming on VANETs in general is a more thoroughly explored topic. Various efforts have focused on how jamming affects the transmission of BSMs over DSRC in relation to Forward Collision Warning capabilities [11–13]. These efforts create attack models to quantify the effects of the jamming attacks and come up with different mechanisms to mitigate their effects. Alturkostani et al. looked at vehicle distance with packet delivery ratio to detect jams, causing the system to enter a fail-safe mode [11]. Serageldin et al. utilized redundant channels and data through dissimilar message types to reduce the effectiveness of jams [12]. In a different effort, Serageldin et al.

looked to develop architectures resistant to jamming by using dual and triple-redundant channels at higher power ratings to enable BSM transmission [13]. Other efforts have leveraged more of the Wireless Access in Vehicular Environment standards in trying to detect jams, such as Malebary's work in looking at beacon frequency from roadside units to detect jams and broadcast warnings to other network members in its vicinity [8]. As a whole, prior efforts related to jamming of convoys and VANETS have had a heavy focus on detection, with a response of either broadcasting warnings to convoy vehicles or entering fail-safe modes. Rather than investigating the issue of jam detection, which has been the focus of the previous literature, our proposed solution instead focuses on control-oriented countermeasures when jamming is unavoidable. We aim to develop methods and techniques to allow for continued convoy operation in the face of these denial-of-service attacks, mitigating the undesired outcome of jamming attacks, rather than the jamming itself.

There are a variety of control approaches and architectures utilized in multirobot autonomous convoying. More broadly speaking, multirobot convoying is a subset of formation control, and the proper choice for formation control design is dependent on which factors need to be prioritized, such as formation shape generation time [15], formation robustness during network congestion [16], and flexible prioritization between system performance and control effort [17]. One particularly robust formation control approach for dynamic environments is behavior-based robotics. In behavior-based robotics, robotic control is built as a collection of basic behaviors. These behaviors are modular and run in a concurrent and distributed manner to achieve desired system functionality. The utilization of behavior-based robotics architectures in multirobot teaming has been previously explored in various research efforts seeking to take advantage of its benefits in dynamic environments. Early efforts include DARPA studies on the integration of navigational behaviors and formation behaviors to create human-led robotic teams to be used in different types of task environments [18]. This effort looked to take four robot teams and set them in a line, column, diamond, or wedge using a motor schema behavior approach. The behaviors used were move-to-goal, avoid-static-obstacle, avoid-robot, maintain-formation, and noise. Member vehicles in the robotic team transmitted position information to properly space out appropriately given the formation and method of formation centering. The path error and duration out of formation of the robots for each formation was compared to one another to determine which one had the best performance in different experimental scenarios. Another effort proposed a decentralized method utilizing a formation matrix [19]. In this effort, each robot leveraged a formation matrix, which defined leader follower pairs. This formation matrix allocated robots to specific positions in a formation before the system began operation by using onboard sensors to detect the location of surrounding robots. Once the proper position of each member robot was allocated, the system was able to traverse to a destination avoiding obstacles, while maintaining the desired formation based on what was defined in the matrix. In this approach, a behavior network architecture was used, in which basic behaviors were linked together and either stimulated or suppressed. This stimulation/suppression was based on input vector values such as detected object location. The output vectors of the network defined the robot's path. While this effort did not use communications between robotics in motion, initial allocation of vehicles in the specific starting positions was required for formation matrix setup. In addition, this effort focused on static robotic formations that maintained their shape, making this method less suitable for applications such as a convoy following defined roads. More recent efforts in behavior-based multi-robot teaming have sought to improve performance by optimizing inter-robot communications, rather than eliminating them. One such effort created a distributed framework for multirobot behavior sequencing that allowed teams of robots to adjust to their configuration to meet communication requirements for the different tasks [20]. The effort focused on the idea that coordinated behaviors between multiple robots must be sequenced together in a way that takes inter-robot distances into account to meet information flow constraints due to communications. The constraints translate

into specific robot position configurations that need to be met within a finite time for the behaviors to be performed. The framework leveraged finite-time convergence control barrier functions to adjust configurations to meet the communication requirements for different sequences of behaviors. While these efforts focused on utilizing behavior-based control architectures for multi-vehicle teaming, they either heavily rely on IVC throughout their operations or require a manually arranged initial state. Furthermore, the focus on traveling in formation shapes makes the multi-vehicle movement less suitable for convoy path following operations, which is the outcome being sought by industry and military from autonomous convoying. Our efforts focus on the development of a more flexible framework towards convoy path following that utilizes on IVC when available and having the additional capability to activate more robust behaviors not relying on communications when faced with jamming to continue convoy operations. Refer to Table 1 for a summary of the control approaches referenced in this section.

Table 1. Prior work in convoying and formation control.

Title	Authors	Summary
Path Planning for Multiple Unmanned Vehicles (MUVs) Formation Shape Generation Based on Dual RRT Optimization [15]	Gong, Tianhao; Yang, Yu; Song, Jianhui	Perform shape formation generation with multiple unmanned vehicles using rapidly-exploring random trees to plan vehicle paths during formation.
Research on Self-Organizing Behavior-Based UAV Formation Based on Distributed Control [16]	Yunhe, Li; Bo, Li; Xiaowei, Niu	Optimize unmanned aerial vehicle formation through use of detected information with self-organizing behaviors with a quasi-static communication topological structure.
Autonomous Vehicle Convoy Formation Control with Size/Shape Switching for Automated Highways [17]	Jond, Hossein Barghi; Platoš, J; Sadreddini, Zhaleh	Development of a convoy formation control architecture consisting of four closed-form control law algorithms to handle formation size and shape switching.
Behavior-Based Formation Control for Multirobot Teams [18]	Balch, Tucker; Arkin, Ronald	Integrated navigational and formation behaviors to create human-led robotic teams to be used in different types of task environments.
Decentralized behavior-based formation control of multiple robots considering obstacle avoidance [19]	Lee, Giroung; Chwa, Dongkyoung	Proposed a decentralized formation control method utilizing a formation matrix and behavior network architecture.
A Sequential Composition Framework for Coordinating Multirobot Behaviors [20]	Pierpaoli, Pietro; Li, Anqi; Srinivasan, Mohit; Cai, Xiaoyi; Coogan, Samuel; Egerstedt, Magnus	Created a framework for behavior sequencing that allowed teams of robots to adjust their configuration to meet communication requirements for the different tasks.

This paper investigates developing an autonomous convoy system that mitigates the effects of jamming attacks. We aim to create an autonomous convoy that does not inherently rely on any Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I) network communications by design, creating a highly scalable system that is robust to system error and loss of communications by using only the on-board sensors of each vehicle. A behavior-based robotics architecture is used to facilitate the employment of simpler sensors and reduce the reliance on complex planning with world models.

The contributions of this paper are as follows:

- This paper develops control-oriented countermeasures against jamming attacks on autonomous vehicle convoys that are independent of traditional radio anti-jamming techniques. This allows for layers of protection against jamming at both the network and convoy controller level, improving the robustness and performance of an overall convoy system when confronted with jamming.
- This paper creates an architecture for behavior-based robotics that is uniquely integrated with the Robot Operating System (ROS) framework. This architecture combined a Motor Schema behavior-based robotics with the ROS navigation stack to a depth that had not previously been seen, paving the path for the greater ROS community to leverage behavior-based robotic architectures.
- The jamming-proof motion planning in this paper improves upon the basic Motor Schema approach by integrating it with vector field histogram motion planning, allowing us to avoid the pitfalls of potential field motion planning.

The remainder of the paper is organized as follows: Section 2 provides background on autonomous ground vehicle convoys, jammers, and behavior based robotics; Sections 3 and 4 discusses the materials and methods used, including the behavior-based robotics architecture, algorithms, and design of experiment; Section 4 shows the results to quantify the performance differences between autonomous ground convoy systems when jam mitigation is present; and Section 5 contains the final discussions and conclusions.

2. Background

The purpose of this effort is to develop an autonomous ground vehicle convoy system that mitigates the effects of jamming by utilizing a behavior-based robotics approach. To enable proper exploration of the effort, the following sections provide a brief background on three topics fundamental to our work: autonomous ground vehicle convoys, jammers, and behavior-based robotics.

2.1. Autonomous Ground Vehicle Convoys

Autonomous ground vehicle convoys are composed of a leader vehicle and follower vehicles. Follower vehicles follow the path of the leader at a given offset distance. While there are various levels of autonomy employed by the autonomous ground vehicle convoys, such as MM, PU, and FA [4], the fundamental design of autonomous convoys remains consistent. The leader vehicle sets the path, either through human driving or autonomous navigation, and utilizes V2V communications to send information to the follower vehicles. The information sent varies depending on system design, but typically includes path points, vehicle pose, vehicle speed, and other sensor information to help follower vehicles track their leader. The follower vehicle uses onboard sensors to estimate its own state, and produces a state estimate of its leader using the information received from V2V and various onboard sensors. A path planning system uses the state information and leader path points to generate a motion plan, and the vehicle is actuated accordingly [3]. Figure 1 lays out the high-level architecture as described.

The V2V communications sent between vehicles can contain a broad range of information from a wide suite of sensors. Information relevant to the vehicle's speed and position, such as sensor data (cameras, GPS, Light Detection and Ranging [LiDAR], wheel encoders, etc.), vehicle kinematics and intended maneuvers, would all need to go through the V2V communications to allow follower vehicles to reach the desired speed and formation for leader following [2]. Various V2V communication topologies can be utilized in autonomous ground vehicle convoys, depending on the needs of the system. Figure 2 shows some common leader-follower topologies used, including predecessor following, predecessor-leader following, bidirectional, and bidirectional-leader [21].

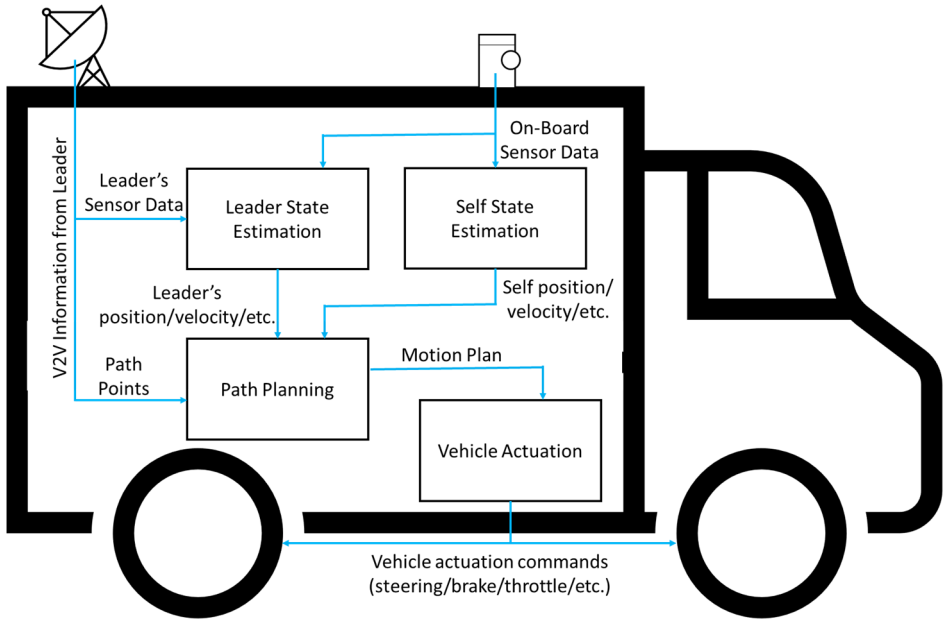

Figure 1. High-level architecture for a follower vehicle in an autonomous ground vehicle convoy.

Figure 2. Common leader-follower communication topologies, adapted from [21]: (**a**) predecessor following; (**b**) predecessor-leader following; (**c**) bidirectional; (**d**) bidirectional-leader.

2.2. Jammers

Jamming is an effective and simple radio interference attack that can be used against wireless networks. In its most basic form, jammers emit radio frequency signals to fill a

wireless channel to interfere with the sending and receiving of wireless communications. The jammer can work to either prevent the source from transmitting packets or disrupt the receiver from properly recognizing and processing legitimate packets. The end result in either case is the interference of the physical transmission and reception of wireless communications [10]. Different classifications of jammers utilize different techniques in jamming wireless networks. The five most commonly described techniques in literature are as follows in Table 2.

Table 2. Common Jamming Techniques.

Jammer Type	Description
Constant [8,10–12,22]	Continuously emits random noise over a wireless medium to interfere with legitimate communications.
Deceptive [10,12,22]	Periodically inject valid packets in their transmissions to deceive receivers into believing that legitimate messages are being sent.
Random [8,10–12,22]	Operates on jam and sleep periods. They will jam for a time duration of t_J, and sleep for a duration t_S before jamming again. Both t_J and t_S can be either fixed or values random.
Reactive [8,10,12,22]	Continuously monitor a communications channel and only jams when it senses activity.
Intelligent [11,22]	Uses knowledge of the communications protocols they are seeking to jam and analyze the transmissions to target specific messages or message types.

Despite the relative simplicity of basic jammers, such as constant jamming and random jamming, jamming attacks are difficult to defend against. Conventional security mechanisms are ineffective, because attackers can disrupt the entire medium of communications itself rather than having to exploit the traditional areas of confidentiality, authentication, and integrity [10]. Additionally, the deployment of jammers has a low barrier to entry. All radio operations on a wireless communications channel can be completely disrupted by a single device emitting noise at high power, regardless of what security measures are built into the underlying communications protocols. In addition, Software Defined Radios that can be used to launch jamming attacks are becoming increasingly affordable and easy to use [9]. In response to the vulnerabilities presented by jamming, several efforts throughout the years have proposed different anti-jamming methods. While many novel anti-jamming methods exist, the primary mechanisms to mitigate jamming can be broadly classified into larger groupings, with two prominent groupings being filtering and Spread Spectrum (SS) techniques. In filtering techniques, the radio receiver attempts to filter out the jammed signal by various means so that the intended signal can be properly parsed. One method of doing this used beamforming, in which a beamformer performed spatial filtering of spatial samples collected from propagating wave fields [23]. It then separated signals with overlapping frequency content that were delivered from different spatial locations. In a different filtering approach, an adaptive Gaussian filter was used to combat jamming caused by narrowband interference with gaussian white noise and pulsed noise [24]. The adaptive filter used optimal time-frequency localization and variable notch depth in conjunction with a fast Fourier-transform-based correlation to filter both continuous and time-varying narrowband interface. In addition to filtering techniques, SS techniques are often used to combat jamming. With SS techniques, a narrowband information signal uses data-independent, random sequences to spread the signal over a wide band of frequencies, in hopes that the jammer is unable to disrupt the entire frequency band [25]. The receiver can then correlate the signal it received with a copy of the random sequence to decode the data that was meant to be delivered. The two most commonly used SS techniques are direct sequence SS (DSSS) and frequency hopping SS (FHSS) [26]. In DSSS systems, the information signal is spread throughout the entire available bandwidth at the same time. This spreading of the signal causes the energy present at each particular frequency of the bandwidth to be very low and mistaken as noise by jammers that are actively listening for signals to jam. The intended receiver can then reconstruct the signal that it received from

all the frequencies in the bandwidth. In FHSS systems, data is sent through a narrowband signal, but the frequency and channel being used to transmit signal rapidly changes. The frequency hopping pattern is known by the sender and intended receiver, allowing for messages to successfully be sent between them. In addition to filtering and spread spectrum techniques, various unique anti-jamming methods have been proposed to protect against the attacks, including exploiting reactive jammer reaction times [27], jammed node isolation [28], and directional antennas [29]. While all these methods can be effective at countering jamming attacks, no solution can offer absolute protection in all scenarios. For this reason, creating robust autonomous convoy solutions that can maintain operation in the face of a jamming attack is important for widespread commercial and military adoption.

2.3. Behavior-Based Robotics

The development of behavior-based robotics was a response to the prevalent control paradigms of the time, which were rooted in function-based, deliberative models [30], as shown in Figure 3a. Under the *deliberative approach*, robotic control relied heavily on world models. Updates to the world model would be determined in a sensing phase, plans based on the model were calculated in a planning phase, and action would then be taken in an acting phase. This approach responded poorly in dynamic environments due to the slow speed of model updates coupled with the need for re-planning [31].

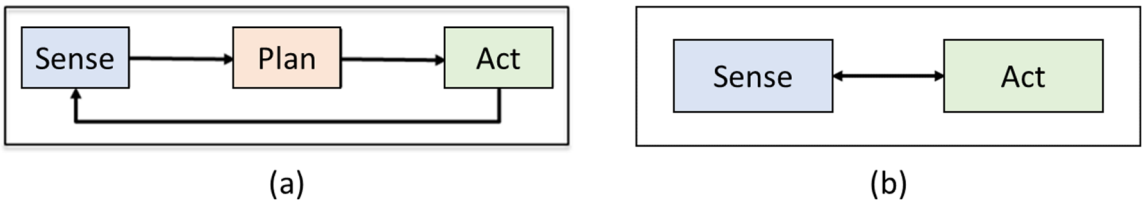

Figure 3. Robotic control approaches: (**a**) deliberative approach; and (**b**) reactive approach [31].

To address the issues of deliberative control, researchers began to develop more *reactive approaches* with faster response times that would be more suited to the dynamic, real-world environments. In these reactive control approaches, there is a tight coupling between sensors and actuators. Control is performed via concurrent condition-action rules so that no complex computations or world models are needed, as shown in Figure 3b. A sensor reads the environment and acts upon the information given the condition-action rules in place. The tight coupling minimizes the need for heavy computation and complex world model [31]. Despite the benefits afforded by reactive control approaches, a purely reactive approach limits an autonomous system's ability to perform complex tasks. While there are various approaches that leverage reactive control with internal states and models, one approach that is particularly well suited to dynamic environments and multi-vehicle control is behavior-based robotics.

In a *behavior-based robotics approach*, robotic control is built as a collection of basic behaviors, with behaviors being defined as something that generates a motor response given some sensory stimulus. These behaviors are modular, and run in a concurrent and distributed manner to achieve desired system functionality. For instance, an autonomous vehicle that moves from its origin to some target location may be concurrently running behaviors such as Move-to-Goal, Avoid-Obstacles, and Stay-on-Path at all times in order to meet the system objective. These different behaviors may be purely reactive, in which they take sensor readings and use rule-based logic to immediately generate some desired motor response, or they can maintain their own state and memory to allow for more complex behaviors. With behaviors being run concurrently, an arbitration mechanism is used to decide on what actions to take, whether it be a single behavior's output or a fusion of multiple behaviors [32]. Arbitration techniques vary based on the specific behavior-based robotics architecture that is being used. Some examples include subsumption architectures [30],

motor schemas [33], and circuit architectures [34]. Figure 4 shows an example of how a generalized behavioral-based robotics architecture can be laid out.

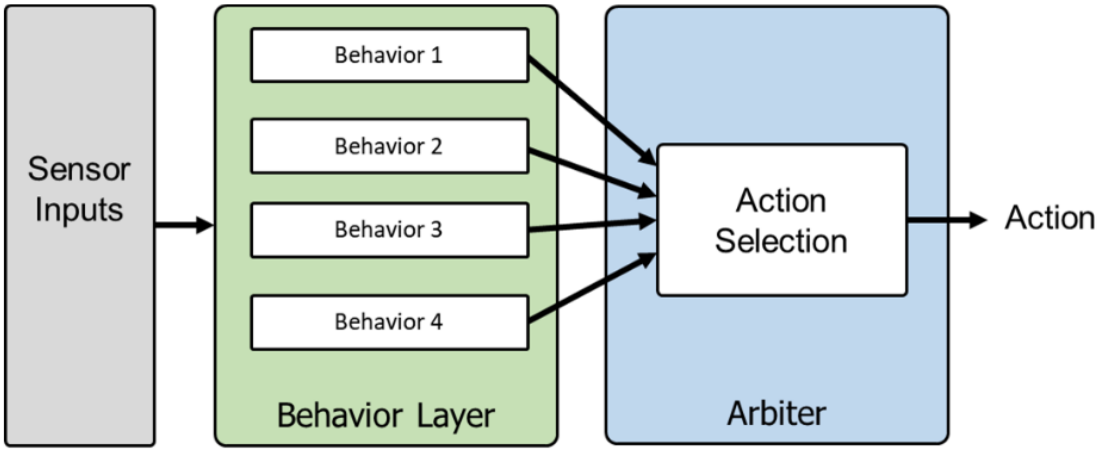

Figure 4. A generalized behavioral-based robotics architecture.

The behavior-based robotics architecture developed for this effort was based on Motor Schemas, a neurological-based approach towards behavioral robotics inspired by schema theory [33]. A schema is a unit of behavior that describes how an agent should react in a given situation and how the reaction can be performed [32]. The concurrent control of many different schemas is used to explain motor behavior. In the Motor Schema approach, each motor schema has an associated perceptual schema that provides the sensor information, which the motor schema uses to generate motor responses per its defined behavior. This is traditionally conducted through potential field response vectors around the robot. Coordination of motor schemas is achieved cooperatively, with no hierarchy of behaviors. Instead, weightings for the motor schemas are set based on a robot's current needs and combined through vector addition. The coordination between different schemas allows for the execution of complex, emergent actions with only a small number of primitive behaviors.

3. Materials and Methods

This following section details the development and testing of the proposed behavior-based autonomous ground vehicle convoy system. A description of the tools leveraged, jamming attacker model, behavior-based robotics architecture, experimental design, and data analysis methods ensue for a thorough discussion on the quantifiable benefits of the proposed system.

3.1. Tools

Development of our autonomous ground vehicle convoy system leveraged the ROS. ROS is a flexible framework for the development of autonomous robotic software, leveraging a collection of libraries, tools, and conventions to facilitate and encourage collaborative software development and reuse [35]. In addition, Gazebo, an open-source 3D robotics simulator, was used in development, due to its integration with ROS and overall community support [36]. Utilizing ROS and Gazebo in our effort allowed us to build upon pre-existing models for our own developmental and simulation environment. To that end, the Clearpath Husky model was chosen as the robotic platform for development.

The Clearpath Husky is a robotic platform that is specialized for research and rapid prototyping [37]. The ROS libraries and Gazebo models are well supported with a large user base, making it an ideal platform to leverage for development [38]. The sensor suite used in this effort includes a GPS, inertial measurement unit, and a forward facing 270° field-of-view LiDAR system. Figure 5 shows a convoy of Clearpath Husky models in a

column convoy formation, along with ROS sensor visualization tools showing the 270° field-of-view LMS1xx LiDAR for each vehicle.

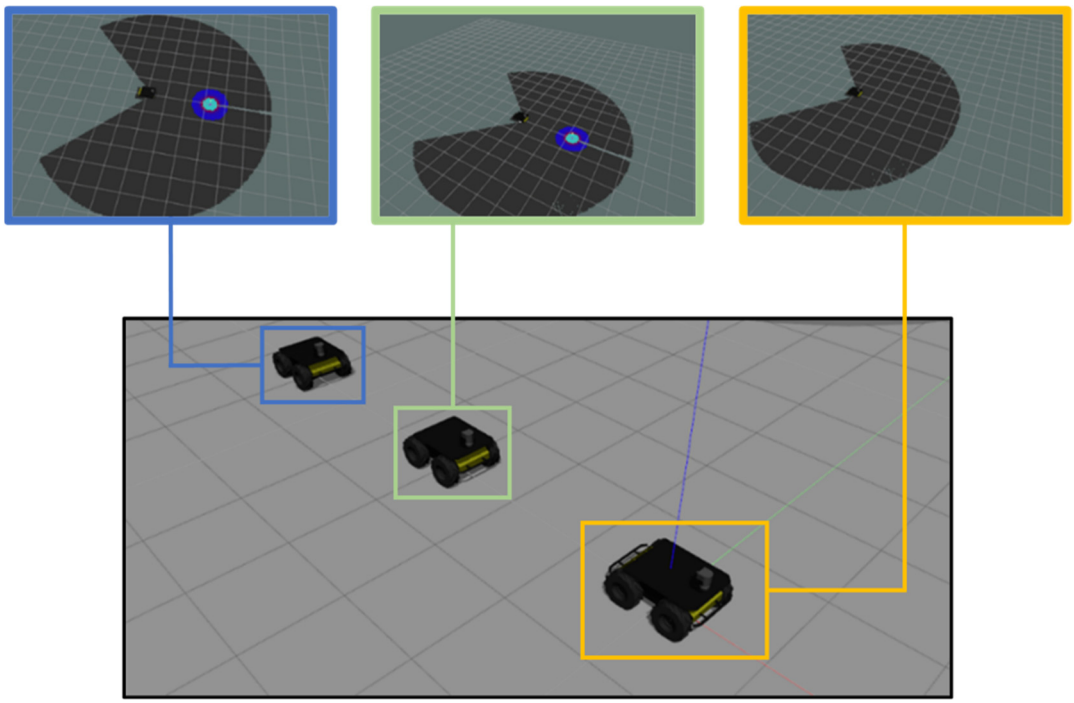

Figure 5. Convoy of Clearpath Husky robots in the Gazebo robotics simulator with each robot's sensor visualization in ROS tools.

3.2. Attacker Model

To develop a jamming attacker model, certain assumptions regarding the network configuration of the robots had to be established beforehand. We assumed that the three Clearpath Husky robots were equipped with wireless radios, allowing them to become mobile nodes. Each node would be able to transmit and receive data packets, forming a generic autonomous vehicular ad-hoc network, similar to what is described in an IEEE 802.11bd network. In addition, each node would be able to leave and rejoin the network without causing the ad-hoc network to fail. Due to the proximity of the robots, we simplified the overall model by assuming that no signal loss or degradation occurs from separation distances between the robots.

The jamming attacker model developed for this effort allows for a choice between simulating a constant jammer and a random jammer. The constant jammer type was chosen due to it being the worst-case jamming attack [12], while the random jammer type allows us to test a greater range of unique jamming scenarios in which follower vehicles end up being partially in the jamming zone. The other types of jammers described are focused on disguising jamming attacks. Since detecting disguised jamming attacks was not a focus of this effort, we focused on constant and random attacker models.

In developing the attacker model, a few key assumptions were made regarding its configuration and capabilities. Firstly, the jammer would be a single node, acting as a single point of jamming in the environment. The jammer was assumed to not be restricted by its power source, meaning it would run continuously for indefinite periods of time without any issue. In addition, it would be frequency unbounded, meaning that the jammer would successfully jam on all frequencies the network communicated on. This maximized the impact of the jamming attacks, making it so we considered packets to be dropped completely when going from the sender to receiver. Implementing jamming in this fashion allowed us to simulate an upper bound on successful denial-of-service attacks [7]. To

simplify the jamming model, we assumed that all areas of the jamming zone were equally strong, with no wireless communication being possible anywhere within the jammed area.

The jammer developed for this effort is configurable with the following inputs:

- lat, long—latitudinal and longitudinal coordinates for the center of the jammer;
- r_J—radius of the jamming area, centered at lat, long;
- type of jammer;
- t_J—jamming time for random jammer;
- t_S—sleep time for random jammer.

The latitudinal and longitudinal coordinates for the jammers center and the jamming radius create a cylindrical jamming zone. Figure 6 illustrates the jamming attack model targeted at a robotic convoy.

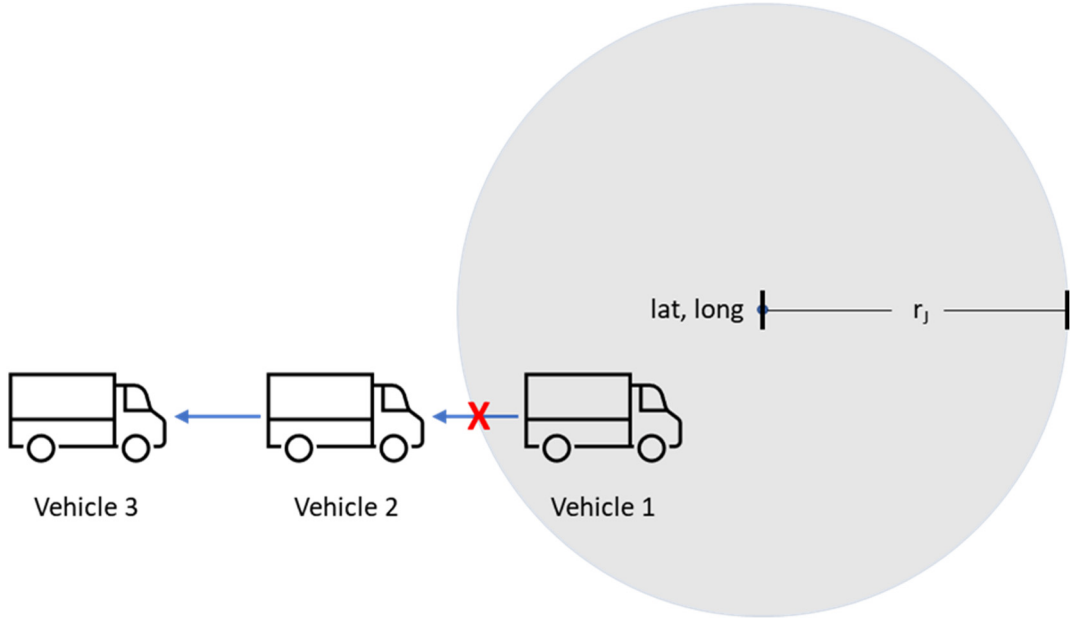

Figure 6. A convoy entering a jamming zone of radius r_J centered at lat, long.

3.3. Behavior Manager

As previously described, a behavior-based robotics architecture uses a collection of basic behaviors to perform robotic control, where behaviors are modular components that generate motor responses given some sensory stimulus which are run in a concurrent and distributed manner. Behavior-based robotic architectures are particularly well-suited to multi-vehicle control in dynamic environments due to scalability, decentralization of control, and tight coupling of rapid sensor readings to real-time path planning [31,39]. The advantages inherent to a behavior-based robotics architecture make it highly well-suited to provide robust protections against jamming for an autonomous convoy. Using the Motor Schema approach as a starting point, we developed a novel behavior-based robotics architecture that uniquely takes advantage of the layered costmap system implicit in the ROS navigation stack. Our behavior-based architecture, henceforth referred to as the Behavior Manager, encompasses behavioral costmaps, behaviors, assemblages, and how they holistically interact. The following sections provide details on the general design of the Behavior Manager, along with the specific implemented system we are proposing for jamming mitigation of autonomous convoys.

3.3.1. Behavioral Costmaps

A core component of the Behavior Manager are the behavioral costmaps. Traditionally, costmaps are two-dimensional grids in which every cell represents a traversibility cost around a robot, based on sensor readings. These costs are used in calculating the optimal

path when traversing to a goal by a path planner [31]. ROS utilizes a layered costmap system, in which multiple, separate costmaps are defined for different contexts. Each costmap is considered a layer that is combined to create a master costmap, which is utilized by the path planner [40].

The Behavior Manager leverages ROS' layered costmap system by having each layer represent the costs as defined by a perceptual schema behavior. In essence, every costmap layer exists due to a behavior dictating that a costmap is needed to perform some behavior. These costmap layers, which we call behavioral layers, are utilized both individually and as a combined master costmap by the different behaviors and by the Behavior Manager's path planning system. In our implementation of the behavioral layers, the costmaps are 200 × 200 grids, with every cell containing a value between 0 and 254. The autonomous vehicle is centered at the approximate center of the costmap, at the grid index (100,100). Figure 7 shows a simplified example of the behavioral costmaps.

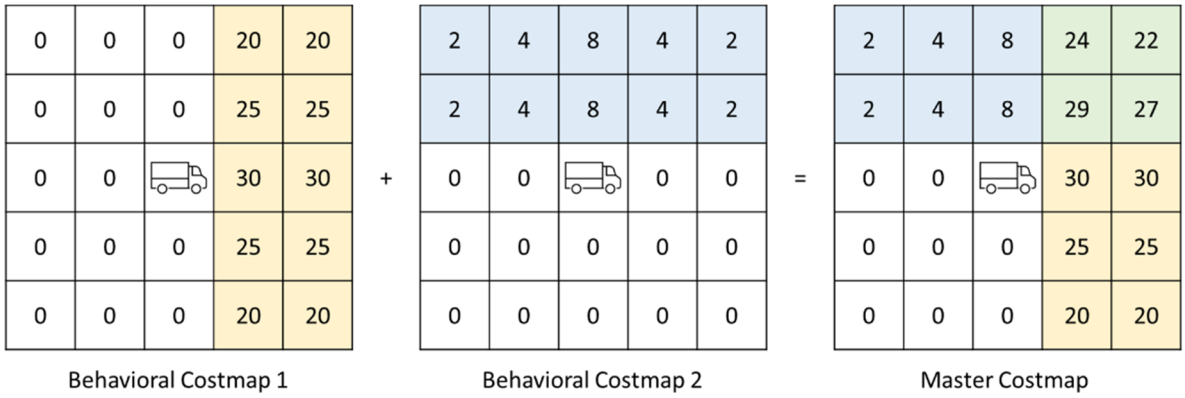

Figure 7. Simplified example of 5 × 5 behavioral costmaps, combined into a master costmap via addition.

3.3.2. Behaviors

In general, behaviors are modular components that generate motor responses given some sensory stimulus which are run in a concurrent and distributed manner [32]. In the Motor Schema approach, the definition of behaviors is expanded, and they are categorized as either motor schemas, which generate motor responses per its defined behavior, or perceptual schemas, which provide sensor information to the other behaviors [33]. The Behavior Manager is based on the Motor Schema approach, and accordingly, categorizes its behaviors as either perceptual schemas or motor schemas.

Behavior Manager perceptual schema behaviors utilize sensor data and generate behavioral costmaps based on the requirements of the behavior. For example, Figure 8 shows how a Keep-Standoff-Distance perceptual schema behavior would work. Figure 8a shows the world state, with the autonomous vehicle in the center, and an obstacle being detected in the upper right-hand corner. Figure 8b shows what the LiDAR readings would be, with a value of 253 being associated with the detected obstacle. Figure 8c shows the final behavioral costmap, with the perceptual schema increasing the costs in the area directly around the obstacle to meet the behavior's goal of keeping a standoff distance between the autonomous vehicle and objects detected by the sensors. Each perceptual schema has a gain that it assigns to its behavioral costmap to be used when combining the layers into a master costmap. Multiple perceptual schemas are run concurrently, creating multiple behavioral costmaps that are used by the greater Behavior Manager in individual behaviors and in overall path planning.

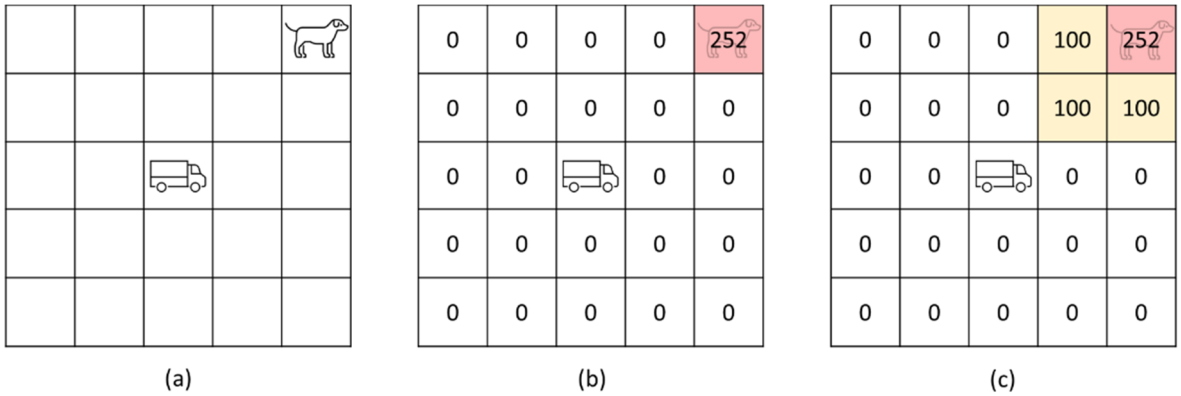

Figure 8. A simplified example of a Keep-Standoff-Distance perceptual schema behavior: (**a**) the world state, with the autonomous vehicle in the center, and an obstacle being detected in the upper right-hand corner; (**b**) costmap with LiDAR readings; (**c**) final behavioral costmap, with the perceptual schema increasing the costs in the area directly around the obstacle.

Behavior Manager motor schemas produce target goals, expressed as two-dimensional Cartesian coordinates **T**, based on the requirements of the behavior. For example, Figure 9 shows two different notional motor schema behaviors. Figure 9a represents a Stay-on-Road behavior. The target goal produced, represented by the star, causes the vehicle to stay on the road. Figure 9b on the other hand, represents a Follow-Leader behavior, with the leader being represented by a dog on the grass. The target goal produced directs the vehicle towards the leader. Each motor schema has a gain that it assigns to the goal that it produces. The goals and associated gains are utilized by the Behavior Manager to determine the final goal point for path planning.

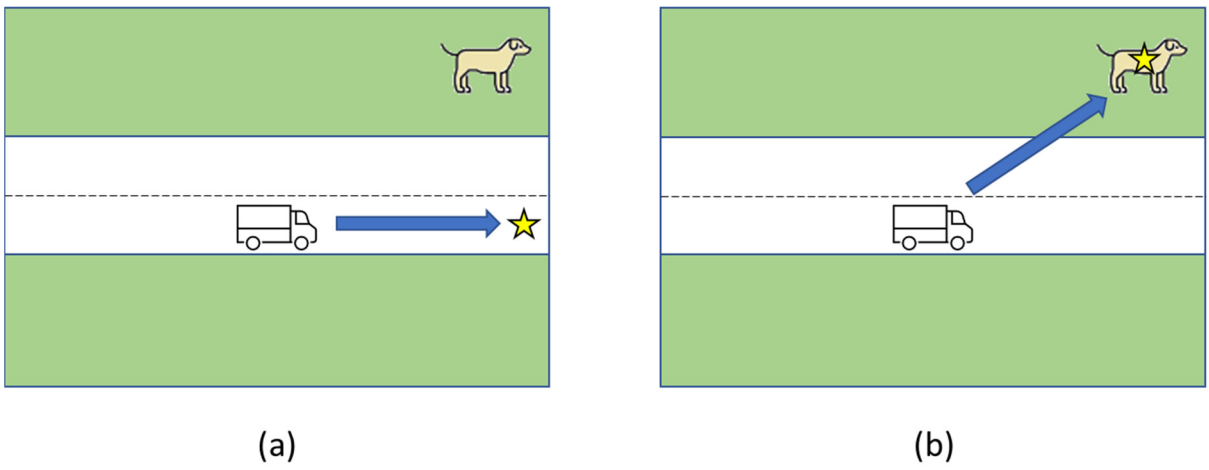

Figure 9. Two examples of motor schema behaviors: (**a**) Stay-of-Path; (**b**) Follow-Leader.

In the development of our autonomous convoy system, we created four behaviors to allow for leader following in the presence of jams: two motor schemas and two perceptual schemas. The behaviors are Move-to-Goal, Maintain-Formation, Avoid-Obstacle-Proximity, and Avoid-Leader-Zone, respectively.

Move-to-Goal

The Move-to-Goal motor schema behavior enables basic goal following. A high-level planner provides a coordinate **B** as an input to the Move-to-Goal behavior. The behavior then publishes the coordinates as the target goal. For our autonomous convoy following

system, Move-to-Goal is used by the vehicles to move towards GPS breadcrumbs provided by their respective leaders for a PF convoy approach as follows:

$$\mathbf{T} = \mathbf{B}, \tag{1}$$

where new GPS breadcrumb **B** coordinates are provided periodically.

Maintain-Formation

The Maintain-Formation motor schema behavior mitigates the effects of jamming attacks on autonomous convoy following by generating target goals **T** that are not dependent on a leader's GPS coordinates. Instead, the behavior uses the density-based spatial clustering of applications with noise (DBSCAN) algorithm to find a target goal nearest to the last known valid goal and to continue convoy operations during jamming attacks.

DBSCAN is a non-parametric clustering algorithm that, when given a set of points, groups together points that are closely packed together into different clusters [41]. In the Maintain-Formation behavior, the points provided are from the behavioral costmaps generated by the Avoid-Obstacle-Proximity perceptual schema. For every cluster identified, the center of mass for each cluster is calculated for comparison to the last known valid **T** coordinates. To find the center of mass $\mathbf{C}_i$, where **C** are the coordinates of the center of mass for cluster I, and $i = 1, \ldots, n$ clusters, we considered each cluster as a system of weighted particles P_j, where $j = 1, \ldots, n$. The coordinates for each particle were given by the costmap cell coordinates c_j, with the mass m_j for each particle being defined as the cost for each given coordinate. The formula for $\mathbf{C}_i$ is as follows:

$$\mathbf{C}_i = \frac{1}{M_i} \sum_{j=1}^{n} \quad m_j c_j, \tag{2}$$

where M_i is the total cost of all the cluster points in cluster i, given by:

$$M_i = \sum_{j=1}^{n} \quad m_j. \tag{3}$$

When the autonomous vehicle fails to receive a GPS breadcrumb coordinate **B**, the Behavior Manager determines that a denial-of-service attack is occurring and that communications with the rest of the convoy have been disrupted. The Maintain-Formation behavior will then find the $\mathbf{C}_i$ nearest to the last valid target goal **T**, and set that as **T**. This process is repeated until a new valid **B** is provided, as shown in Algorithm 1.

Algorithm 1 Maintain-Formation

```
if no B received from leader
    C_i = center of mass of clusters 1, ... , n from DBSCAN(costmap)
    temp_distance = high value placeholder
    for i = 1, ... , n
        cluster_distance = distance between T and C_i
        if cluster_distance < temp_distance
            T = C_i
            temp_distance = cluster_distance
        end if
    next i
end if
```

Avoid-Obstacle-Proximity

The Avoid-Obstacle-Proximity perceptual schema behavior produces a behavioral costmap that includes costs for objects detected by LiDAR, along with a buffer zone of costs around the objects, in which the cost of the cell decreases as the distance from the object increases, up to a preset radius. The implementation of the buffer zone leverages

the inflation layer provided by the ROS Navigation stack [42], which assigns a cost around objects with the following formula:

$$cost = e^{(-1 \times s \times (d-r))}(o-1), \tag{4}$$

where s is a cost scaling factor, d is the distance from the obstacle, r is the robot's radius, and o is the cost to assign a cell that falls within the robot's radius. Refer to Figure 10 for an example of how the buffer zone inflation is represented on a costmap. The buffer zone provided by the Avoid-Obstacle-Proximity behavior helps to prevent the vehicle from getting too close to obstacles, decreasing the chances of collisions.

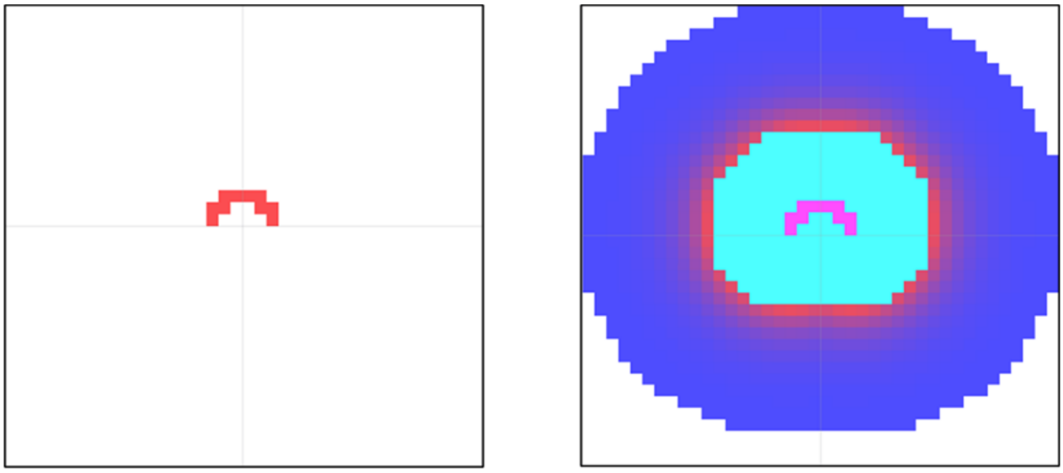

Figure 10. Costmap inflation in the Avoid-Obstacle-Proximity behavior: (**a**) the costmap produced from LiDAR sensing a construction cone; (**b**) the costmap produced from creating a buffer zone of inflated costs around the construction cone.

Avoid-Leader-Zone

The Avoid-Leader-Zone perceptual schema behavior produces a behavioral costmap that inserts a high-cost ring of a preset radius around a robot's leader. This ring creates a zone around the leader that the follower vehicles will not enter, in effect enforcing a following distance between the vehicles. To do this, the behavior takes the position of its leader as an input. If network communications are available, the position is provided wirelessly by the lead vehicle. If the convoy is under a jamming attack, the target goal **T**, as provided by the Maintain-Formation behavior is used instead. The costmap coordinates that comprise the ring are calculated using basic trigonometric functions to obtaining coordinates of a scaled unit circle with an offset center, as follows:

$$x_i = floor\left(\frac{a}{f} coscos\left(\frac{i}{n}2\pi\right)\right) + l_x, \tag{5}$$

$$y_i = floor\left(\frac{a}{f} sinsin\left(\frac{i}{n}2\pi\right)\right) + l_y, \tag{6}$$

where $[x_i, y_i]$ are costmap indices for each point in the ring; $i = 1, \ldots, n$ costmap cells, with n representing the number of total number costmap cells the ring should contain; a is the radius of the leader's following distance zone; f is the costmap cell resolution; and $[l_x, l_y]$ is the costmap index of the leader. Figure 11 shows a robot and the leader zone produced by the Avoid-Leader-Zone behavior.

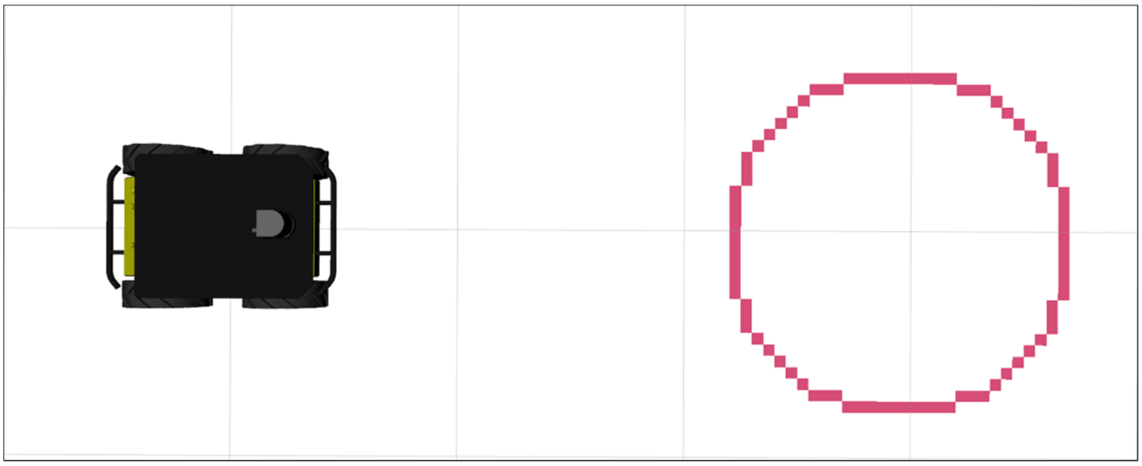

Figure 11. Behavioral costmap created by the Avoid-Leader-Zone perceptual schema behavior.

3.3.3. Assemblages

At a basic level, assemblages are the complex tasks that the robot is trying to accomplish. The assemblages are composed of multiple concurrently running behaviors. The behaviors can be classified as motor schemas, which produce goals to generate motor responses, or perceptual schemas, which generate behavioral costmaps. The behavioral costmaps are used by the motor schema behaviors in goal generation and by the assemblage for navigation. The assemblage acts as a coordinator of the behaviors, weighing and combining them as appropriate for the task at hand. Figure 12 shows the architecture of an assemblage.

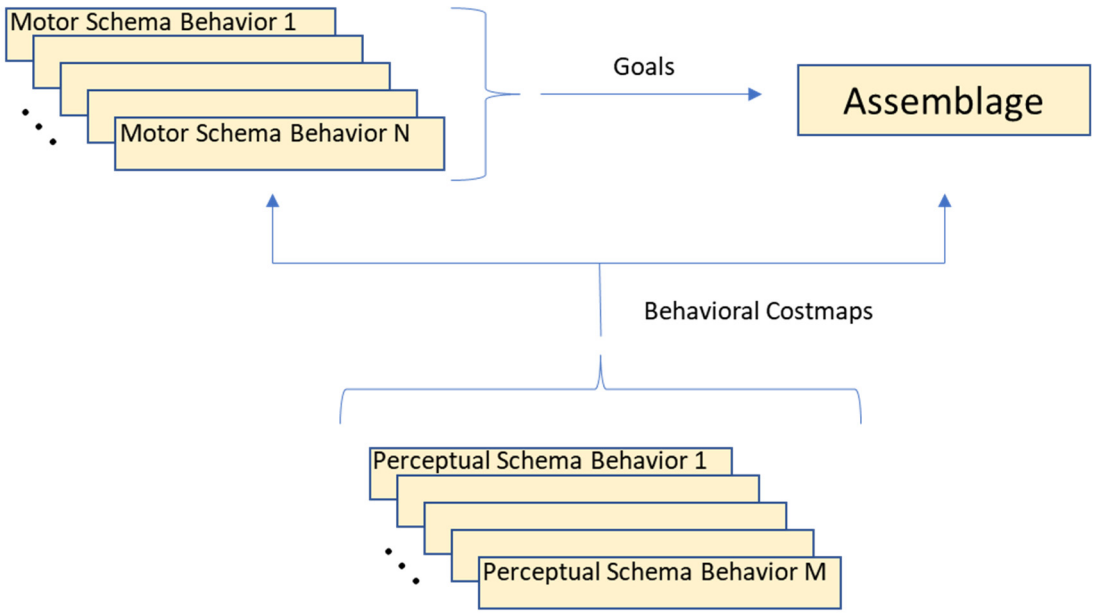

Figure 12. Architecture of an assemblage.

For our effort, we utilized the behaviors described above and developed a Follow-Leader-with-Jam-Mitigation (FLJM) assemblage that performs autonomous convoying with jam mitigation techniques. The architecture of the assemblage and the associated behaviors can be seen in Figure 13.

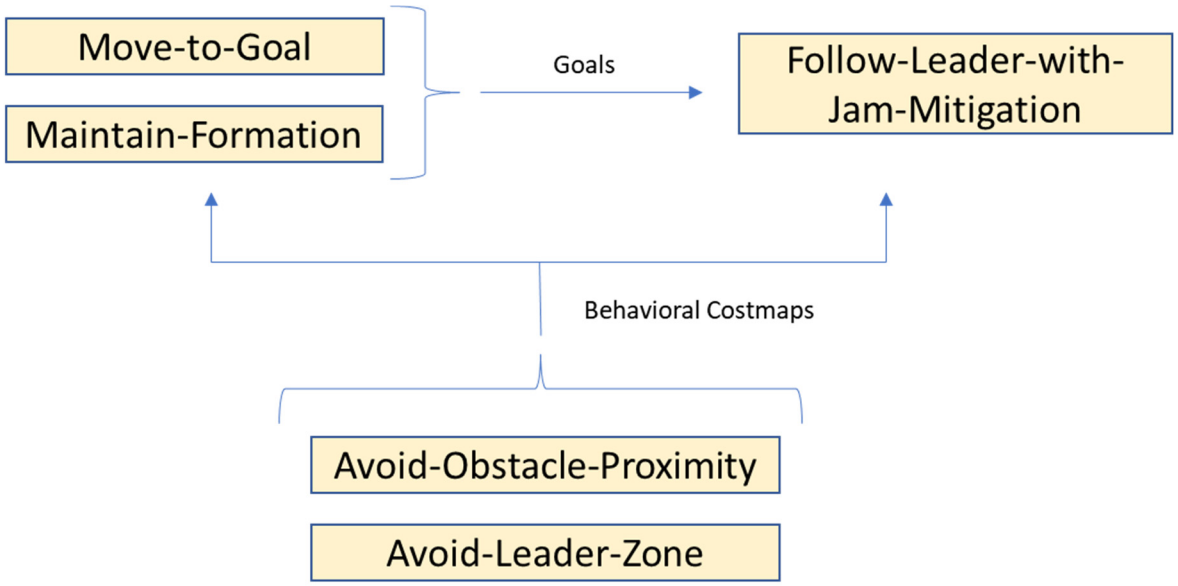

Figure 13. Architecture of the Follow-Leader-with-Jam-Mitigation assemblage.

The basic task of the FLJM assemblage is to perform leader following. An autonomous vehicle running the assemblage will take an assigned leader and await waypoints from them to follow. If the vehicle's network communications are unavailable, either due to a system failure or a denial-of-service attack, it will continue following the moving cluster of costmap points nearest to the last valid waypoint. Once network communications and waypoints are available again, the assemblage will revert to following waypoints. As with all assemblages, the FLJM assemblage is responsible for goal and behavioral costmap combination/selection for all the behaviors. The costmap gains for Avoid-Obstacle-Proximity and Avoid-Leader-Zone are set to be equal, meaning that combining their behavioral costmaps is conducted through addition. Essentially, the costmaps are stacked on top of each other and the sum cost of every cell forms the master costmap. For goal combination, a selection method is utilized. If the system detects valid waypoints from network communications, the **T** provided by Move-to-Goal is used, allowing for waypoint following. If no network communications are received, the **T** provided by Maintain-Formation is used until network communications are restored and waypoints are provided again. Figure 14 characterizes the interactions between behaviors and waypoint transmissions through the network.

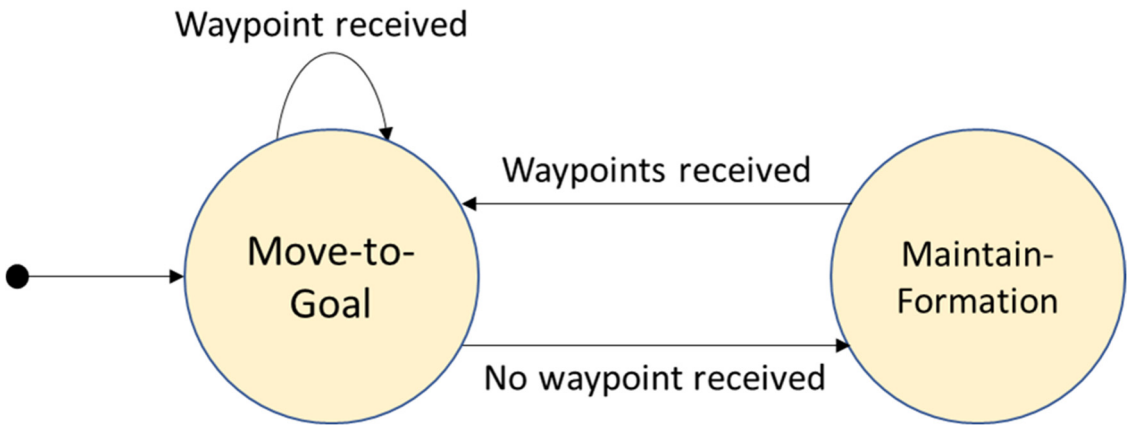

Figure 14. Architecture of the Follow-Leader-with-Jam-Mitigation assemblage.

3.3.4. Path Planning

A key difference between the Behavior Manager and the traditional Motor Schema approach is how motor responses are generated and acted upon. While the Motor Schema approach uses potential field response vectors and navigation, the Behavior Manager leverages layered costmaps with Vector Field Histogram (VFH) path planning. VFH is a path planning technique that was developed specifically to address the inherent limitations of potential field methods, such as U-shaped obstacle traps, oscillation in narrow corridors, or the inability to pass between closely spaced objects [43]. The VFH method represents the obstacles around a robot with certainty values in a Cartesian histogram grid. Data reduction over the histogram grid reduces it to a polar histogram of angular sectors around the robot with associated polar obstacle densities. Based on the desired goal and polar obstacle densities, drive and steering commands are given that best avoid obstacles while making progress towards the goal [44]. VFH was chosen as our path planning technique because of the aforementioned histogram grid. By using the combined layered behavioral costmap as the histogram grid, we naturally integrated costmap-based perceptual schemas with path planning, thereby avoiding potential field methods and their pitfalls.

3.4. Experimental Setup

To test the efficacy of the FLJM assemblage, we compared its performance to that of a basic convoy controller in a Gazebo simulation environment. The Gazebo world used in the experiment contained a flat ground pane without terrain or obstacles and a three-vehicle convoy of Cleapath Husky robots, a four-wheel skid-steer robotic vehicle. The setup was kept minimal to reduce the variables present when testing convoy performance. For the basic convoy controller, we implemented a leader-follower controller that performed "delayed following" with a predecessor-leader following network topology. In a "delayed following" approach, a lead vehicle records the path it drove and relays that information to followers to repeat after some time delay [45]. This approach is recognized as a common leader-follower convoying method [46], making it the ideal baseline for comparison.

The high-level autonomous convoy setup was kept consistent between the basic convoy controller and the FLJM assemblage for an accurate comparison. In both cases, the autonomous convoy was FA and consisted of three vehicles: one convoy leader that drove along preset path plans, and two autonomous followers. The convoy leader precisely followed through given path points, allowing for repeatability in test runs, resulting in better comparisons between the basic convoy controller and the FLJM assemblage. Both unmanned followers would be running either the basic convoy controller or FLJM assemblage to perform autonomous following, depending on which system was being tested at the time. Each vehicle recorded GPS breadcrumbs of its position that it transmitted to its direct follower when wireless network communications were available.

The two preset path plans created for the test runs were a square loop and a roundabout turn, as seen in Figure 15. These paths were chosen to mimic common road formations and driving maneuvers. The square loop tests basic turns, while the roundabout tests the intersections of the same name. In addition, jamming zones were established along the paths to simulate the effects of denial-of-service attacks on the autonomous convoy. Figure 16 overlays the jamming zones along the paths.

To comparatively test the performance of the convoy controllers, multiple test runs with the three vehicle convoy were performed for each jammed path shown in Figure 16. Ahead of each test run, the autonomous followers were configured to use either the basic convoy controller or the FLJM assemblage. Five runs were performed under constant jamming, and an additional five runs were performed under random jamming, with t_J, and t_S set to 10 s and 2 s, respectively. Both the basic convoy controller and the FLJM assemblage were tested in this fashion for each path plan.

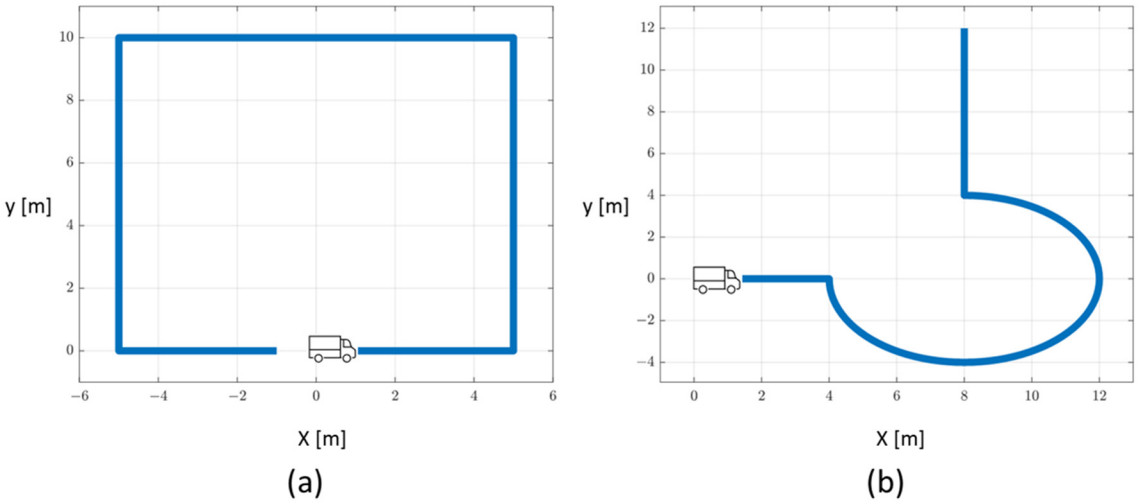

Figure 15. The preset path plans for the lead vehicle: (**a**) square loop and (**b**) roundabout.

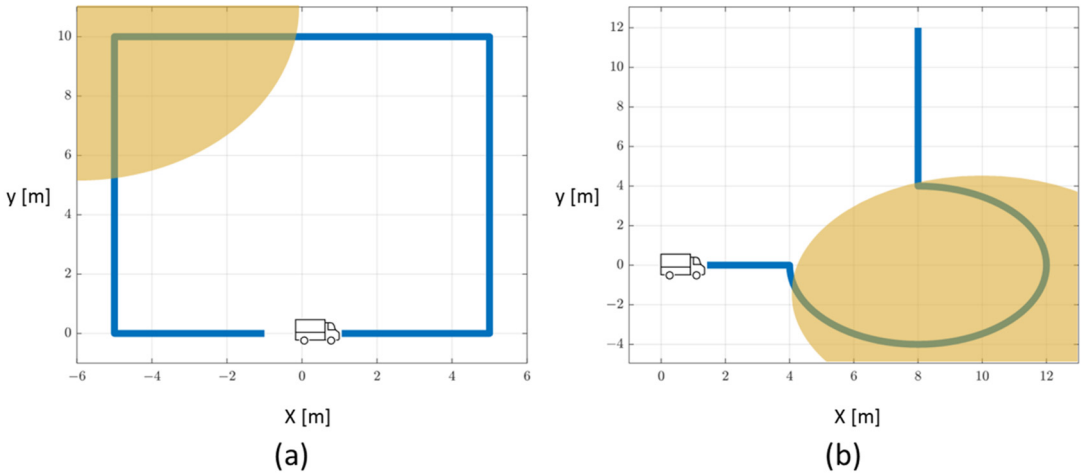

Figure 16. Jamming zones overlaid onto the preset plans for the lead vehicle: (**a**) square loop and (**b**) roundabout.

4. Results

To measure the performance of the convoy controllers, we calculated the mean absolute error (MAE) between the follower vehicles' path and the convoy leader's path. The MAE was chosen as the measurement for evaluation as it has a direct interpretation to the real-world quantities being compared, namely the Euclidian distances between follower points and convoy leader points, in addition to not inflating the penalty for larger errors via squaring, which occurs in other commonly used performance measurements such as root mean squared error [47]. The paths of both the followers and the convoy leader were sampled at a rate of 1000 Hz, respectively creating arrays of position points for each vehicle. The Euclidean distance between the follower position points and corresponding convoy leader positions points were then used to calculate the MAE with the following formula:

$$\text{MAE} = \frac{\sum_{i=1}^{n} |e_i|}{n}, \tag{7}$$

where e is the Euclidean distance between the follower and convoy leader position points, and $i = 1, \ldots, n$, where n is the size of the array e. Table 3 shows the average MAE over five runs for each convoy controller for both the square path and roundabout path when under a constant jamming attack. Figure 17 shows the paths of the five runs taken by the followers, overlaid on the convoy leader's goal points for a visual comparison of performance.

Table 3. Average MAE under a constant jamming attack.

Convoy Configuration	Square (m)	Roundabout (m)
Follower 1 Using Basic Convoy Controller	1.2665	3.1334
Follower 2 Using Basic Convoy Controller	1.9226	3.5205
Follower 1 Using FLJM assemblage	0.4942	0.4197
Follower 2 Using FLJM assemblage	0.7321	0.8452

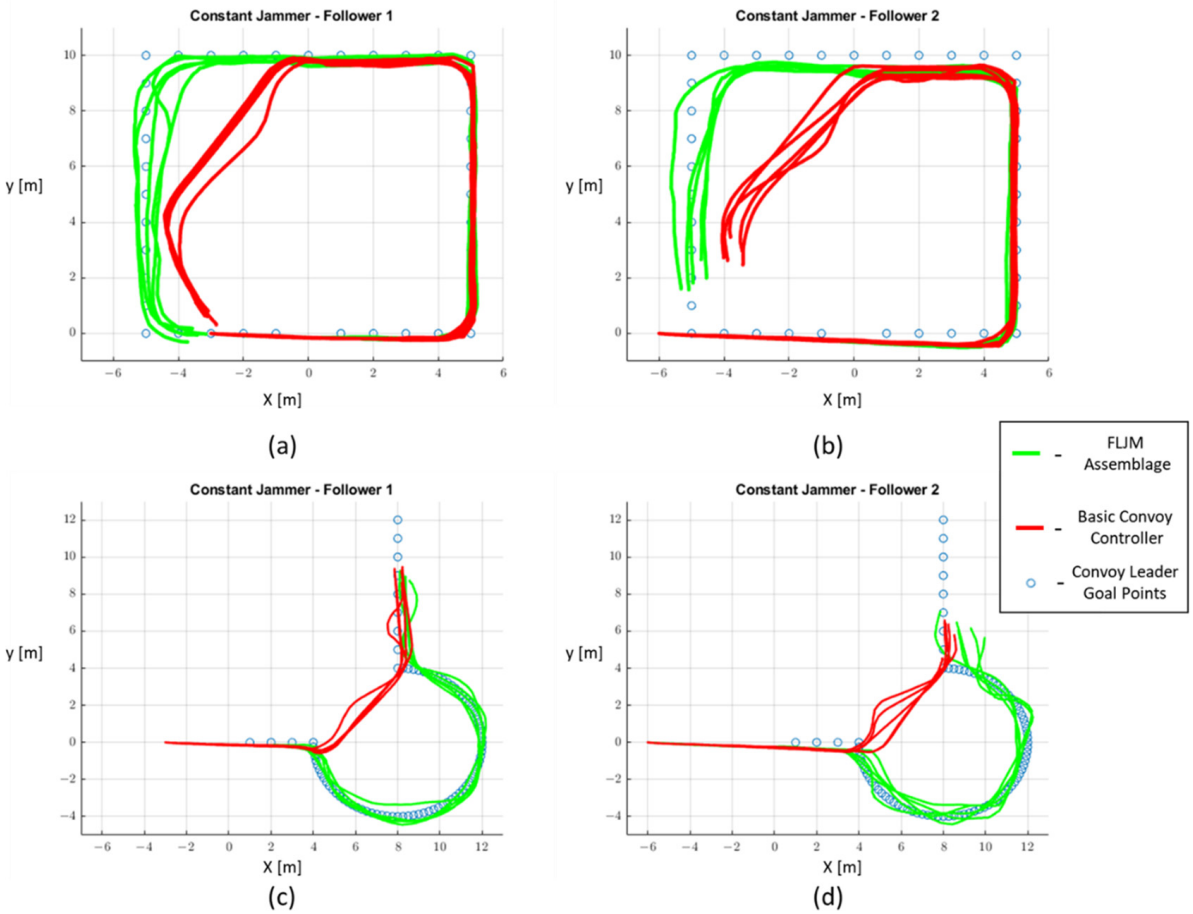

Figure 17. Plot of the paths taken by the followers under constant jamming on the square path for (**a**) follower 1 and (**b**) follower 2; and on the roundabout path for (**c**) follower 1 and (**d**) follower 2.

Likewise, Table 4 shows the average MAE while under a random jamming attack with t_J, and t_S set to 10 s and 2 s, while Figure 18 shows the paths taken for the five runs by the followers, overlaid on the convoy leader's goal points for a visual comparison of performance.

Table 4. Average MAE under a random jamming attack.

Convoy Configuration	Square (m)	Roundabout (m)
Follower 1 Using Basic Convoy Controller	0.6844	1.2637
Follower 2 Using Basic Convoy Controller	1.0028	1.0697
Follower 1 Using FLJM assemblage	0.4680	0.4821
Follower 2 Using FLJM assemblage	0.7654	0.9271

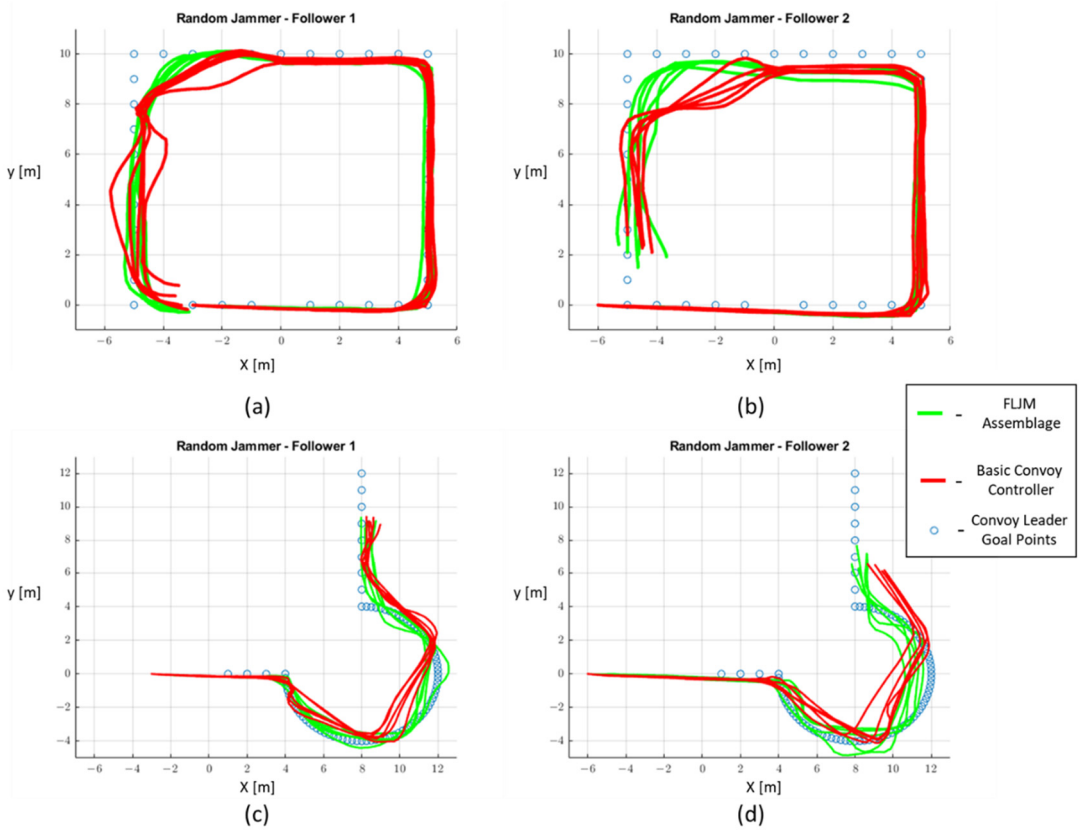

Figure 18. Plot of the paths taken by the followers under random jamming on the square path for (**a**) follower 1 and (**b**) follower 2; and on the roundabout path for (**c**) follower 1 and (**d**) follower 2.

As seen in Table 3, the utilization of the FLJM assemblage significantly improved convoy performance across both path plans when compared to the basic convoy controller under constant jamming. When following the squared loop path, using the FLJM assemblage decreased the average MAE by 60.98% for follower 1 and 61.92% for follower 2. The improvement was even more drastic with the roundabout path, where the average MAE was decreased by 86.61% and 75.99% for follower 1 and follower 2, respectively. These significant improvements were due to the basic convoy controller missing turns that the convoy leader performed in the jamming zones laid out in yellow in Figure 16. As seen in Figure 17, the FLJM assemblage allowed the followers to successfully continue leader-following inside the constant jamming zones, whereas the stoppage of communications prevented the followers from getting proper waypoints when relying solely on basic delayed following.

As seen in Table 4, the utilization of the FLJM assemblage also improved convoy performance across both path plans when compared to the basic convoy controller under random jamming. When following the squared loop path, using the FLJM assemblage decreased the average MAE by 31.62% for follower 1 and 23.67% for follower 2. Improvement was exhibited with the roundabout path as well, where the average MAE was decreased by 61.85% and 13.33% for follower 1 and follower 2, respectively.

While the utilization of the FLJM assemblage yielded similar performance across both the constant and random jammers, the performance improvements when compared to the basic convoy controller were less significant under random jamming attacks. This is due to the nature of random jammers, which alternate between a jamming and sleeping state. Rather than missing all the convoy leader's waypoints in a jamming zone, as in the case of constant jamming, the follower vehicles would periodically receive network transmissions from the jamming zone under random jamming, allowing for occasional transmission of waypoints inside the jammed area. This can be seen when comparing the

basic convoy controller paths in Figures 17 and 18. In Figure 17, the basic convoy controller followers stop outside of the jamming zones and only restart following when the convoy leader exits the constant jamming zone. In Figure 18 however, the followers are able to break into the random jamming zone and follow the occasional waypoints they receive when the jammer is sleeping. These results demonstrate that the value of using jamming mitigation techniques increases accordingly with the severity of the jamming attack. As previously stated, constant jamming is considered the worst-case jamming scenario with the most damaging impact, so the potential benefits afforded by using the FLJM assemblage is greater when compared to a random jamming attack, as seen in the experimental results described here.

While the results described here were focused on the constant and random jammer attack model developed for this effort, the performance benefits provided by the FLJM assemblage would be realized even when applied to other attacker models. Various other attacker models have focused on reactive jammers with assumptions of global eavesdropping, zero-delay jamming startup, and unbounded frequency jamming [9,48]. With these attacker models, the FLJM assemblage performance would match that of our constant jammer, given that the reactive jamming occurs instantaneously when entering the jammed areas and disrupts all communications, which closely aligns with our model for constant jamming. Other attacker models have additional parameters for their reactive jammers, such as jamming range, triggering range, and multiple jammer nodes, but they make the same uniform power and omnidirectionality of jamming assumptions we made in our model, resulting in the same cylindrical jamming zones [28]. While the notion of multiple jamming zones introduces coverage areas of different shapes due to overlapping cylinders, the performance improvements from using the FLJM assemblage will remain due to the reactive jammer closely aligning with our constant jamming model. In addition, other attacker models have focused on more dynamic jamming scenarios, such as the jammer being embedded in one of the convoy's follower vehicles [7]. In this attacker model, a follower vehicle in a convoy would switch from a standard network node into a uniform power mobile jammer that prevented messages from arriving at their destination. Assuming the mobile jammer continues following its leader to disguise its adversarial nature, the FLJM assemblage would once again allow for continued convoy operations. In this scenario, the jammed vehicles would remain in the Maintain-Formation behavior for as long as the mobile jammer was activated and switch back to Move-to-Goal upon the deactivation of the jammer. Convoy operations would continue for all vehicles in the convoy. As described in these examples, the benefits of utilizing the FLJM assemblage are generally applicable and has the potential to improve the performance and robustness of autonomous vehicle convoys against a wide range of jamming attacks.

5. Discussion

Jamming attacks are a simple, yet effective type of denial-of-service attack that have the potential to adversely degrade the performance of autonomous convoys. While many efforts have been undertaken to detect the presence of jams, we focused our efforts on finding ways to mitigate a jammer's effects to allow for continued convoy operation. The prevalence of behavior-based robotics approaches in multi-vehicle teaming made it a prime starting point for developing an approach to allow convoy following when confronted with jamming, and lead to the development of the Behavior Manager.

By utilizing the Behavior Manager and creating behaviors and assemblages to mitigate the effects of jamming, we were able to show improved convoy performance with up to an 86.61% reduction in average MAE. This demonstrates that a behavior-based robotics approach can be leveraged to dramatically improve the robustness of autonomous convoys when faced with jamming attacks. Furthermore, the modular nature of the Behavior Manager means that the capabilities can be extended to mitigate the effects of other types of attacks, such as LiDAR spoofing, replay attacks, and RADAR absorption [49], through the development of additional behaviors and assemblages.

A potential future area of research on the usage of behavior-based robotics towards mitigating the effects of attacks is dynamic behavioral weighting. While the usage of Q-learning towards behavior selection has been researched in the past [50], the application of modern learning techniques, such as deep reinforcement learning, towards multi-robot attack mitigation is a novel area that warrants further investigation. By training on datasets that demonstrate human reactions to attacks on manned convoys, the robustness of autonomous convoy systems utilizing behavior-based robotics could be greatly improved on, creating even more stable systems able to handle a broad spectrum of attacks.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, visualization, project administration: C.C.; writing—review and editing, supervision, funding acquisition: A.M. and S.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Murphy, T.J. *Convoy Operations in Afghanistan Handbook*; U. S. Army Combined Arms Center: Fort Leavenworth, KS, USA, 2010.
2. Cheung, C.; Mohammadi, A.; Rawashdeh, S.; Baek, S. Delivery of Healthcare Resources Using Autonomous Ground Vehicle Convoy Systems: An Overview. *Front. Robot. AI* **2021**, *8*, 250. [CrossRef]
3. Nahavandi, S.; Mohamed, S.; Hossain, I.; Nahavandi, D.; Salaken, S.M.; Rokonuzzaman, M.; Ayoub, R.; Smith, R. Autonomous Convoying: A Survey on Current Research and Development. *IEEE Access* **2022**, *10*, 13663–13683. [CrossRef]
4. McKay, S.; Boyer, M.E.; Beyene, N.M. *Automating Army Convoys: Technical and Tactical Risks and Opportunities*; RAND: Santa Monica, CA, USA, 2020.
5. Llatser, I.; Festag, A.; Fettweis, G. Vehicular Communication Performance in Convoys of Automated Vehicles. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016.
6. Lopes, H.J.; Lima, D.A. Cellular Automata in path planning navigation control applied in surveillance task using the e-Puck architecture. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020.
7. van der Heijden, R.; Lukaseder, T.; Kargl, F. Analyzing Attacks on Cooperative Adaptive Cruise Control (CACC). In Proceedings of the 2017 IEEE Vehicular Networking Conference (VNC), Turin, Italy, 27–29 November 2017.
8. Malebary, S. Real-Time Jamming Detection in Vehicular Network. *Int. J. Sci. Res. Innov. Technol.* **2016**, *3*, 159–166.
9. Sciancalepore, S.; Di Pietro, R. Bittransfer: Mitigating Reactive Jamming in Electronic Warfare Scenarios. *IEEE Access* **2019**, *7*, 156175–156190. [CrossRef]
10. Xu, W.; Trappe, W.; Zhang, Y.; Wood, T. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In Proceedings of the 6th ACM International Symposium on Mobile ad Hoc Networking and Computing-MobiHoc '05, Champaign, IL, USA, 25–27 May 2005.
11. Alturkostani, H.; Chitrakar, A.; Rinker, R.; Krings, A. On the Design of Jamming-Aware Safety Applications in VANETs. In Proceedings of the 10th Annual Cyber and Information Security Research Conference, Oak Ridge, TN, USA, 7–9 April 2015.
12. Serageldin, A.; Alturkostani, H.; Krings, A. On the Reliability of DSRC Safety Applications: A Case of Jamming. In Proceedings of the 2013 International Conference on Connected Vehicles and Expo (ICCVE), Las Vegas, NV, USA, 2–6 December 2013.
13. Serageldin, A.; Krings, A. The Impact of Dissimilarity and Redundancy on the Reliability of DSRC Safety Applications. In Proceedings of the 2014 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 13–16 May 2014.
14. Hu, Y.; Shan, H.; Dutta, R.G.; Jin, Y. Protecting Platoons from Stealthy Jamming Attack. In Proceedings of the 2020 Asian Hardware Oriented Security and Trust Symposium (AsianHOST), Kolkata, India, 15–17 December 2020.
15. Gong, T.; Yu, Y.; Song, J. Path Planning for Multiple Unmanned Vehicles (Muvs) Formation Shape Generation Based on Dual RRT Optimization. *Actuators* **2022**, *11*, 190. [CrossRef]
16. Yunhe, L.; Bo, L.; Xiaowei, N. Research on Self-Organizing Behavior-Based UAV Formation Based on Distributed Control. In Proceedings of the 3rd International Conference on Vision, Image and Signal Processing, Vancouver, BC, Canada, 26–28 August 2019.

17. Jond, H.B.; Sadreddini, Z.; Platoš, J. Autonomous Vehicle Convoy Formation Control with Size/Shape Switching for Automated Highways. *Int. J. Eng.* **2020**, *33*, 2174–2180.
18. Balch, T.; Arkin, R.C. Behavior-Based Formation Control for Multirobot Teams. *IEEE Trans. Robot. Autom.* **1998**, *14*, 926–939. [CrossRef]
19. Lee, G.; Chwa, D. Decentralized Behavior-Based Formation Control of Multiple Robots Considering Obstacle Avoidance. *Intell. Serv. Robot.* **2017**, *11*, 127–138. [CrossRef]
20. Pierpaoli, P.; Li, A.; Srinivasan, M.; Cai, X.; Coogan, S.; Egerstedt, M. A Sequential Composition Framework for Coordinating Multirobot Behaviors. *IEEE Trans. Robot.* **2021**, *37*, 864–876. [CrossRef]
21. Zheng, Y.; Eben Li, S.; Wang, J.; Cao, D.; Li, K. Stability and Scalability of Homogeneous Vehicular Platoon: Study on the Influence of Information Flow Topologies. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 14–26. [CrossRef]
22. Pelechrinis, K.; Iliofotou, M.; Krishnamurthy, S.V. Denial of Service Attacks in Wireless Networks: The Case of Jammers. *IEEE Commun. Surv. Tutor.* **2011**, *13*, 245–257. [CrossRef]
23. Van Veen, B.D.; Buckley, K.M. Beamforming: A Versatile Approach to Spatial Filtering. *IEEE ASSP Mag.* **1988**, *5*, 4–24. [CrossRef]
24. Luo, G.Y. On-Line Wavelet Filtering of Narrowband Noise in Signal Detection of Spread Spectrum System for Location Tracking. *Int. J. Commun. Syst.* **2011**, *25*, 598–615. [CrossRef]
25. Popper, C.; Strasser, M.; Capkun, S. Anti-Jamming Broadcast Communication Using Uncoordinated Spread Spectrum Techniques. *IEEE J. Sel. Areas Commun.* **2010**, *28*, 703–715. [CrossRef]
26. Poisel, R. *Modern Communications Jamming Principles and Techniques*; Artech House: Boston, MA, USA, 2011.
27. Liu, Y.; Ning, P. Bittrickle: Defending against Broadband and High-Power Reactive Jamming Attacks. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012.
28. Xuan, Y.; Shen, Y.; Nguyen, N.P.; Thai, M.T. A Trigger Identification Service for Defending Reactive Jammers in WSN. *IEEE Trans. Mob. Comput.* **2012**, *11*, 793–806. [CrossRef]
29. Noubir, G. On Connectivity in Ad Hoc Networks under Jamming Using Directional Antennas and Mobility. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 186–200.
30. Brooks, R. A Robust Layered Control System for a Mobile Robot. *IEEE J. Robot. Autom.* **1986**, *2*, 14–23. [CrossRef]
31. Siciliano, B.; Khatib, O. (Eds.) *Springer Handbook of Robotics*; Springer International Publishing: Cham, Switzerland, 2016.
32. Arkin, R.C. *Behavior-Based Robotics*; MIT Press: Cambridge, MA, USA, 2000.
33. Arkin, R.C. Motor Schema—Based Mobile Robot Navigation. *Int. J. Robot. Res.* **1989**, *8*, 92–112. [CrossRef]
34. Kaelbling, L.P. An Architecture for Intelligent Reactive Systems. *Reason. About Actions Plans* **1987**, 395–410. [CrossRef]
35. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A. ROS: An open-source Robot Operating System. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) Workshop on Open Source Robotics, Kobe, Japan, 12–17 May 2009.
36. Koenig, N.; Howard, A. Design and Use Paradigms for Gazebo, an Open-Source Multi-Robot Simulator. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, Japan, 28 September–2 October 2004.
37. Clearpath Robotics. Robots/Husky-ROS. Available online: http://wiki.ros.org/Robots/Husky (accessed on 10 June 2022).
38. Clearpath Robotics. Husky/husky: Common Packages for the Clearpath Husky. Available online: https://github.com/husky/husky (accessed on 10 June 2022).
39. Soni, A.; Hu, H. Formation Control for a Fleet of Autonomous Ground Vehicles: A Survey. *Robotics* **2018**, *7*, 67. [CrossRef]
40. Lu, D.V.; Hershberger, D.; Smart, W.D. Layered Costmaps for Context-Sensitive Navigation. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014.
41. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
42. Macenski, S. costmap_2d/hydro/inflation. Available online: http://wiki.ros.org/costmap_2d/hydro/inflation (accessed on 12 July 2022).
43. Koren, Y.; Borenstein, J. Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991.
44. Borenstein, J.; Koren, Y. The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots. *IEEE Trans. Robot. Autom.* **1991**, *7*, 278–288. [CrossRef]
45. Vasseur, L.; Lecointe, O.; Dento, J.; Cherfaoui, N.; Marion, V.; Morillon, J.G. Leader-Follower Function for Autonomous Military Convoys. *SPIE Proc.* **2004**, *5422*, 326–337.
46. Zhao, X.; Yao, W.; Li, N.; Wang, Y. Design of Leader's Path Following System for Multi-Vehicle Autonomous Convoy. In Proceedings of the 2017 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 27–29 October 2017.
47. Li, X.R.; Zhao, Z. Measures of Performance for Evaluation of Estimators and Filters. *SPIE Proc.* **2001**, *4473*, 530–541.
48. Pietro, R.D.; Oligeri, G. Silence Is Golden: Exploiting Jamming and Radio Silence to Communicate. *ACM Trans. Inf. Syst. Secur.* **2015**, *17*, 1–24. [CrossRef]

49. Chowdhury, A.; Karmakar, G.; Kamruzzaman, J.; Jolfaei, A.; Das, R. Attacks on Self-Driving Cars and Their Countermeasures: A Survey. *IEEE Access* **2020**, *8*, 207308–207342. [CrossRef]
50. Martinson, E.; Stoytchev, A.; Arkin, R. Robot Behavioral Selection Using Q-Learning. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System, Lausanne, Switzerland, 30 September–4 October 2002.