2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Decentralized Linear Convoying for Underactuated Surface Craft with Partial State Coupling

Raymond Turrisi[1,2] and Michael Benjamin[1]

***Abstract*— This work introduces a novel decentralized algorithm and control law for stable linear convoying using a layered control approach. This algorithm was implemented in MOOS-IvP, using an abstraction layer that models a virtual system decoupled from the system's actual dynamics. This makes the algorithm platform-agnostic and able to be combined with other behaviors such as collision avoidance and operating region behaviors. A trajectory defined by a lead agent is discretized and embedded with the leader's dynamics and propagated to all following agents. We first demonstrate that this approach when paired with a simple PD controller prevents accumulated errors and improves trajectory tracking for follower agents. Thereafter we demonstrate how virtually coupling a subset of agent states improves the overall cohesiveness of the convoy. Improvements are demonstrated in both simulations and field trials using five autonomous surface vehicles.**

I. Introduction

Convoying is a specialized multi-agent formation that provides a safe and orderly method for transporting a group of agents. In recent years, there has been growing interest in developing robot swarms for large-scale field operations, and autonomous convoying is a rapidly growing research topic in working towards these capabilities [1]. In these scenarios, the number of human operators monitoring or controlling these swarms is often limited, creating a need for efficient organization and transportation of agents through guided control or high-level objectives. Convoying is particularly well-suited for these purposes, offering flexibility in convoy leadership.

In this paper, we present a novel decentralized algorithm and control law within a behavior-based framework for stable linear convoying with discretized trajectory state embeddings and virtual partial state coupling. The algorithm presented is platform agnostic and developed within the MOOS-IvP [2]. A linear convoy, as referred to in this paper, is a type of formation consisting of N agents in series, consisting of a lead agent and trailing agents, and generally evenly separated along a trajectory. In this formation, agents can maintain a relative 180-degree bearing with their target, or follow a trajectory set by their target. The distance metric defined for each of these may maintain a constant distance between agents, or a varying distance based on a desired *time* between agents along a trajectory given a desired cruising speed.

This work emphasizes advantages to layers of autonomy and controls, with success demonstrated both in simulation and field trials. For field trials, tests were conducted on five Clearpath M300 Autonomous Surface Vehicles (ASV's), which can be seen in Fig. 1. We focus on serialized linear convoying due to the unique challenges with propagated errors and disturbances throughout a trajectory for agents in series, with additional challenges for underactuated systems, while making innovations with a behavior-based and decentralized approach.

Fig. 1. Five Clearpath M300 vehicles used in testing on dock and in a convoy on the Charles River in Boston, Massachusetts.

II. Related Work

This work extends the approach presented in [3], originally motivated by the DARPA Sea Train program. The algorithms presented in [3] discuss advancements in the MOOS-IvP Swarm Toolbox, and demonstrates successful linear convoying developed within the framework. The algorithm and speed policy for linear convoying did have limitations, particularly for an increasing number of agents; where issues with propagated errors and unstable trajectory tracking arise. The main issue addressed from this work is the propagation of tracking errors for an increasing number of agents and improvements for both maintaining a steady speed and ranging along a trajectory.

Similar challenges have been realized in developing Adaptive Cruise Control (ACC). ACC systems are primarily focused on adjusting a vehicle's cruising velocity to maintain a distance which varies based on headway between vehicles [4]. To address this problem a cooperative approach was developed which improved stability in maintaining a constant speed for a collection of autonomous vehicles [5]. The work in [5] however only addresses vehicles which are following a straight line trajectory, and does not propose solutions for handling noise which is introduced by the environment or the trajectory itself.

Earlier work in trajectory-based formation control with a behavior-based perspective is discussed in [6]. This approach

[1] Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA rturrisi@mit.edu, mikerb@mit.edu, [2] Woods Hole Oceanographic Institute, Woods Hole, MA 02543, USA

demonstrates both leader-referenced and unit-centered formation algorithms on holonomic and non-holonomic ground robots within the robot architectural suite AuRA [7]. This work successfully demonstrates a variety of types of formations, discussing a line formation which is analogous to a convoy within the context of this paper. The approach presented however limits agents to a constant geometric formation rather than a common trajectory defined by the leader. Over time and when following a more complex trajectory, this introduces a variety of complications. Geometric challenges occur as the length of the spread of agents increase, where if the leader takes a sharp 90-degree turn, the turning radius for all trailing vehicles will continue to increase with respect to their order, which was observed in the presented results. In the presence of obstacles in the workspace where a leading agent may have chosen this path deliberately, any number of or all of the following agents would have to break formation and rely more heavily on additional obstacle avoidance behaviors.

A single cohesive framework for synchronizing motion tracking of underactuated systems with collision avoidance has been presented in [8], although only simulated results are presented. While this work lends itself towards a rigorous mathematical approach, there are additional considerations which must be made, especially in the realm of collision avoidance which is of the utmost priority for field operations with combinations of crewed and uncrewed systems. Consider the international rules for Collision Regulation (COLREGS) [9], where the means by which a vessel must avoid a collision has a discrete hierarchy of rules and procedures. Successful implementations have been developed such that crewed and uncrewed systems can be compatible in a common working environment [10], where each agent is individually responsible for avoiding collisions. This approach is simpler to generalize to a larger variety of conditions; such as limited communication bandwidth and range, a variety of perceptive capabilities, and can be combined support existing infrastructure such as AIS [11]. While this approach is simpler to generalize to heterogeneous and independent systems, it does not claim to provide an optimal solution for multi-agent coordination, which may otherwise be obtained if communication bandwidth is not an issue, or working conditions permit deviation from COLREGS.

Recent convoying work, like [12], integrates waypoint following and obstacle avoidance using **AprilTag** tracking for leader-follower dynamics and individual trajectory planners. Tested with only two vehicles, this approach may face scaling challenges similar to those in [3] and [5], leading to errors and instability in larger convoys. In contrast, this work proposes a decentralized algorithm for multiple vehicles, mitigating error propagation with partial state coupling and discretized trajectory embeddings. It introduces a convoying policy with relaxation conditions that interact with other MOOS-IvP behaviors, such as collision avoidance and operating region control.

III. Problem Formulation and Approach

The algorithm presented in this work consists of three key ideas: the high-level frontseat-backseat dichotomy and the behavioral approach, the initialization of the convoy, and then the maintenance of the convoy for trailing agents. We start with the discussion and perspective for the frontseat and backseat approach which motivates particular details for the algorithms presented. We then discuss the initialization of the convoy which enables the autonomous construction of the convoy, which is a feature but not the core innovation of this work. We then present the core **ConvoyPD** algorithms which control agents within the convoy.

A. Control Schemes - Frontseat-Backseat Dichotomy

Users of MOOS-IvP typically employ a frontseat-backseat paradigm. This means that the core control system resides in isolation in a 'frontseat', which may be on a separate computer, separate middleware such as ROS, or all be implemented within MOOS. A 'backseat' is typically an autonomy or decision making framework, such as MOOS-IvP, which in turn provides a control input to the frontseat.

MOOS-IvP is a behavior-based autonomy framework for robotic systems. Users design a behavior which are typically either objective behaviors or guardrail behaviors. In general, one objective behavior is running at a time, and any number of guardrail behaviors may run passively until needed. For example, **ConvoyPD** is an objective behavior providing the high-level desired states and maintaining the control law presented herein, whereas the collision avoidance and operating region behaviors are additional safeguards which are passive until needed. Guardrail behaviors provide necessary safety benefits for operators and assets, and is also convenient for field work. The backseat, may be concerned with any number of environmental or situational variables, however it typically provides only decision variables such as desired heading, speed, or depth, which become control inputs to the low-level controller [2].

A 'behavior' is a function h which maps information, $\bar{v}$, and decisions available to an agent, $\bar{x}$, to a scalar utility value U. Information determines how an algorithm should distribute utility values across all decisions in the domain(s). k behavior functions are combined by weighted sum, where we are then concerned with which specific combination within the domain of decision variables, $\bar{x}$ provides the greatest utility U^* among decision variable domains. An example of a utility distribution across the domain for heading and speed is depicted in Fig. 2. In MOOS-IvP, the mechanism for obtaining the maximum is interval programming, which is described in Eq. 1.

$$\bar{x}^* = argmax_{\bar{x}}(\sum_{i=0}^{i=k} w_i \cdot h_i(\bar{v_i}, \bar{x})) \tag{1}$$

B. Initialization of the Convoy

First, a waypoint is defined as a **TaskBehavior** which contains a set of instructions for recursive task propagation.

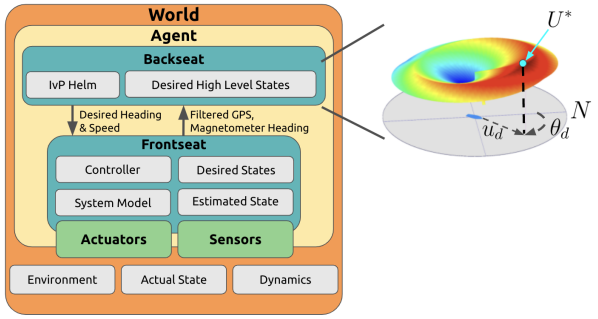

Fig. 2. A diagram presenting the interaction of the world and agent relation, in addition to the dichotomy of the frontseat-backseat paradigm. At the top right, we provide an example IvP behavior function.

Specifically, two **TaskBehaviors** are used; a **WaypointTask** and a **ConvoyTask**. The only difference between the two, is a flag which gets raised upon a bid being won which informs an agent on whether or not it won the bid to be the leader and activate a driving behavior (i.e. a waypoint behavior), or a follower and who to follow. With this, a **WaypointTask** occurs once, and then initiates the recursive bids on the **ConvoyTask** behavior. A bid is a scalar cost function where an agent estimates its capability to handle a task. These nested instructions contain variables which capture the location of the agent that won this latest auction and its name, so the agent's location is the new reference for the next bid, and its name is appended to the exclusion list so that no agents which are already included in the convoy, participate in future auctions. This process occurs recursively, until all agents have been assigned an ordering.

The scalar bidding function used for these task behaviors are based on the estimated odometry from a Dubins turn approximation [13] rather than the Euclidean distance, as illustrated in Fig. 3. Here we can see that agent a is closer to the waypoint than agent b ($r_a < r_b$), the Dubins turn approximation captures the additional cost of changing direction from an agents original orientation such that $d_b < d_a$. In the second round of bidding, agent b would have won the first bid on the **WaypointTask** and becomes the leader, spawning a **ConvoyTask**, where all agents then bid on the new reference which is the current location of the agent that won the previous auction.

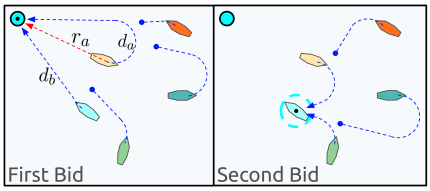

Fig. 3. This demonstrates the bidding scheme for at least the first two auctions, in addition to the motivation behind the Dubins turn approximation.

C. ConvoyPD Algorithm

ConvoyPD is a new behavior with a simple PD control law which uses optional partial state coupling between agents. At a high level, the process involves capturing the trajectory defined by a lead agent, discretizing it into a collection of points which become control inputs and references for follower agents, and reducing the trajectory into a one dimension linear form. Agents are connected to each other with virtual springs to maintain an ideal follow range, while a virtual damper is connecting an agent's current speed with the leader's speed at the time in which the point was instantiated. When extended to include partial state coupling, the errors of the neighbors are also coupled to an agent.

All agents run the **ConvoyPD** behavior, however it runs in different modes depending on whether or not the agent is the leader or a follower. On the leader, **ConvoyPD** does not contribute to the decision function - it runs passively by capturing position of the agent, and embedding the states of the leader agent and propagating it to the first follower. This was a design choice, where the lead agent could have been running another application rather than a behavior within the helm, however there are additional considerations in implementation which made this convenient. The point embedding contains the time in which it was instantiated, the speed of the vehicle at that point, in addition to the turning rate of the leader at that time. The rate in which points are seeded and sent to the first follower agent is determined by a user parameter defining the minimum Euclidean distance between two points which is denoted as Δr, which should be only as small as necessary to achieve desirable results to preserve communication bandwidth.

A point and its embedding is represented by p_n. The linearization of the trajectory is depicted in Fig. 4, while the encoding, propagation, and virtual coupling is presented in Fig. 5

$$p_n =< (x, y) : t, u_t, \dot{\psi}_t >$$

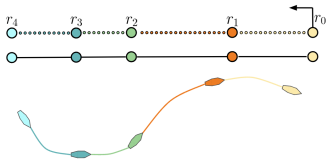

Fig. 4. Linearizing and discretizing of the trajectory set by the lead agent, where an ideal follow range metric is measured with respect to the distance between agents on this trajectory. r_0 is the start of the convoy and the datum in the convoy.

A follower running **ConvoyPD** is receiving points from whom they are following and accumulating them in a convoy point queue. The distance between a follower and its target is not measured by the euclidean distance between two points, rather the sum of all the distances between points in the convoy point queue plus residuals, which simplifies to a scalar quantity where when all agents are on-trajectory, they are ideally uniformly distributed along a bounded number line. If no points are present in the queue, which only occurs briefly during initialization, then the residuals are equal to the total distance between the targets.

When a vehicle is on-trajectory, they are pursuing the next point in their queue, which contains embeddings for the control law. An agent removes this point from its queue and moves onto the next point once it is within its capture range or slip range. The capture range is the minimum distance between an agent and a point before it is popped from the queue and propagated to an agent's follower in perfect conditions. The slip range is a practical consideration to handle environmental and situational disturbances, a poor entry onto the trajectory, and serves as a relaxation condition complementing the interplay with other behaviors running simultaneously such as collision avoidance, thus allowing an agent to deviate from the original trajectory. A point which is within the slip range and is beyond $\pm$ 90-degrees off of the agents current heading the vehicle will be propagated to the follower agents even if it is not within capture range. In practice, the capture range was set to 2 meters and the slip range to 10 meters, prioritizing trajectory alignment over precise point capture.

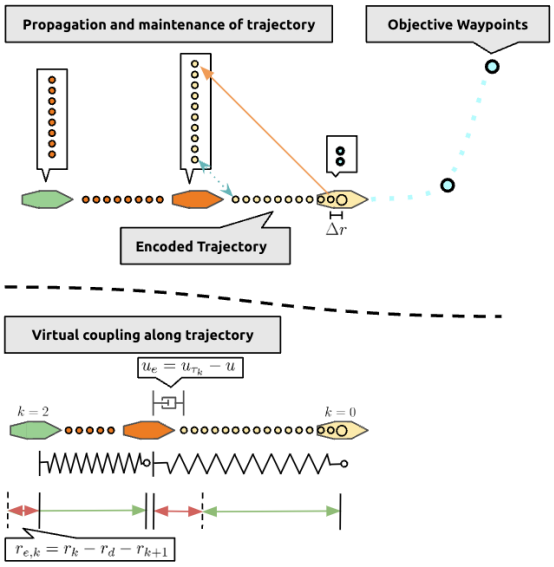

Fig. 5. A leader follows a prescribed objective, while encoded points are propagated to follower agents. The latest point set by the lead agent is the last point in the first follower agents point queue. Agents are virtually coupled to each other with virtual stiffness and damping to connect the agent's speed to the desired speed along the trajectory contained within the point embeddings.

IV. Control Laws

A. PD Control Law

The core speed controller for **ConvoyPD** which provides the control inputs for an agent without any coupling, is essentially a PD controller which penalizes larger errors and is centered around the nominal desired cruising speed of the convoy, u_n.

$$\begin{aligned} u_d = u_n &+ (K_{p1} + K_{p2} \cdot |r_e|) \cdot r_e) \\ &+ (K_{d1} + K_{d2} \cdot |u_e|) \cdot u_e) \end{aligned} \tag{2}$$

Where r_e is the range error defined $r_e = r_k - r_d - r_{k-1}$, and r_k is the position of the k^{th} agent on the trajectory, r_{k-1} is the position of the agent whom it is following along the trajectory, and the ideal follow range is defined by r_d. The speed error u_e, is defined by $u_d = u_{\tau_k} - u$, where u is the agent's current speed, and u_{τ_k} is the leaders speed embedded in the agent's next point at the time it was created. In implementation, $r_k - r_{k-1}$ is the distance along the total trajectory within the convoy point queue, plus residuals.

A new control law for heading was considered, but proved not necessary to address the fundamental issues which this work sought to address. With this, the behavior used typical waypoint-behavior functionality, in order to achieve an on-trajectory state. The desired heading, θ_d, is simply equal to the angle of the vector between the ownship and the next intermediate waypoint in the world frame.

The speed control law has a tendency to drive the ideal follow ranging error to zero, and when the range error is zero, follow the trajectory at the same speed as the lead agent at the time in which the leader was at that point in the trajectory. When the vehicle is not within the ideal follow range, then the desired speed term typically slows an agent during turns to prevent an agent from overshooting and moving off-trajectory. The computation of θ_d simply steers the agent along the trajectory at this desired speed. With this, a follower agent manages its range to whom it is following while taking turns at the speed the leader took them to prevent overshoot. Neither u_d or θ_d are conventional control effort variables, however are inputs to the frontseat control system, which varies from platform to platform and is responsible for interfacing with the vehicle's driving actuators. These variables are a choice to be compatible with the other behaviors used within the IvP-Helm simultaneously.

B. PD Controller with Partial State Coupling

For Partial State Coupling (PSC), we use the same core control law found in Eq. 2, however we then add local state coupling terms reflective of case II in Fig. 6, forming Eq. 3. The general form for this control law with PSC drew inspiration from [14]. The lead of the current agent $(k-1)$, and the trailing follower of the current agent $(k+1)$, each denoted with subscripts l and t respectively. Similarly, we have coupling PD terms on the errors of the local agents, such as $K_{p,c}$ and $K_{d,c}$.

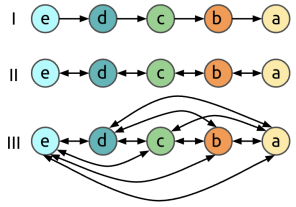

Fig. 6. Graphical representation of the different types of agent coupling.

$$\begin{aligned} u_d = u_n &+ (K_{p1} + K_{p2} \cdot |r_e|) \cdot r_e \\ &+ (K_{d1} + K_{d2} \cdot |u_e|) \cdot u_e \\ &+ K_{p,cl} \cdot r_{e,k-1} + K_{d,cl} \cdot u_{e,k-1} \\ &- (K_{p,ct} \cdot r_{e,k+1} + K_{d,ct} \cdot u_{e,k+1}) \end{aligned} \tag{3}$$

For case III coupling, this expression for serial convoying generalizes to:

$$\begin{aligned} u_d = u_n &+ (K_{p1} + K_{p2} \cdot |r_e|) \cdot r_e \\ &+ (K_{d1} + K_{d2} \cdot |u_e|) \cdot u_e \\ &+ \sum_{q=0}^{k-1} (K_{p,cl} \cdot r_{e,q} + K_{d,cl} \cdot u_{e,q}) \\ &- \sum_{q=k+1}^{N} (K_{p,ct} \cdot r_{e,q} + K_{d,ct} \cdot u_{e,q}) \end{aligned} \tag{4}$$

Coupling gains for ranging errors and speed errors are represented by $K_{p,c}$ and $K_{d,c}$, with subscripts l and t for all leading and trailing agents respectively. The asymmetry of the coupling terms means that if leading agents are lagging from their target, the current agent should speed up as well, and that if trailing agents are lagging, then the current agent should slow down. In both case II and case III the effect is propagated throughout the convoy but in varying strengths. In practice, leading gains became redundant. While it provided tighter formation control when the agents were already in a convoy with small errors, it rendered undesireable experimental results, so they were set to zero.

V. Results

First, we present simulation results, which isolate and showcase differences between **ConvoyBaseline** and **ConvoyPD**, which demonstrates improved error tracking and stability. Thereafter we demonstrate advantages which are realized with partial state coupling for improving the quality of the formation of the convoy. Results for a single run are communicated through two figures; one which provides insights on the trajectory tracking for all the agents, and one which provides detailed insights on summarized mission metrics and time series data throughout the convoy. Metrics include: the nominal range error, nominal speed error, indicator variables over timeseries data for when agents are in range to their target and when a full convoy is formed. A single table for the three algorithms and the same circuit presents results for additional metrics, such as the percent time in which each agent is in range of their target, the percent time in which the convoy is in full form, and the straightness and turning index for a mission.

A straightness index, I_s, was discussed in [15], which is used to communicate the effectiveness in which one follows a trajectory. We use it similarly here as the ratio of the path the agent needed to follow, which is the total length of the trajectory set leader minus it's desired ranging from the leader in ideal circumstances, $L_0 - k \cdot r_d$, and the length of the path an agent did follow, L_k, where the optimal value is 1, which can be found in Eq. 5.

$$I_{s,k} = \frac{L_0 - k \cdot r_d}{L_k} \tag{5}$$

We also extend this and add another metric which we refer to as a turning index, I_t, which is a metric for communicating how much does an agent make unnecessary turns, which can be found in Eq. 6. Similarly, we take the cumulative sum of all the absolute values for the changes in heading the leader made in the trajectory, divided by all the changes in heading an agent made when following the trajectory, for all the samples S available for an agent.

$$I_{t,k} = \frac{\sum_{i=0}^{i=S_0} |\delta\psi_{i,0}|}{\sum_{i=0}^{i=S_k} |\delta\psi_{i,k}|} \tag{6}$$

The turning index was introduced since it is more sensitive to errors throughout an entire mission regardless of path length. Both the turning index and straightness index are proxies for how well an agent follows a trajectory and their stability, they capture different subtleties.

A. Simulation Results

Each simulation was conducted from the same initial conditions, with the same desired speed, and equivalent overlapping parameters (i.e. ideal follow range, desired speed, collision avoidance configuration, etc..). The results from simulation studies support the claim of improved performance. The circuit selected was chosen to provide a challenging trajectory, which need to be completed twice before the mission was terminated.

Results are presented in Figures 7, 8, and Table II. The remainder of this section discusses these results. **ConvoyBaseline**, **ConvoyPD**, and **ConvoyPD** with partial state coupling are labeled *a*, *b*, and *c* respectively in all the following figures.

First we highlight the highly improved trajectory tracking performance which is most clearly observed from Fig. 7 by comparing parts *a* and *b*. This highlights the performance benefits of the PD speed policy with embedding the leader's dynamics into the trajectory points, over the previous speed policy presented in [3]. **ConvoyBaseline** did contain a feature for propagating the leader's waypoint throughout the convoy, which partially addressed fish tailing, but the results for a challenging trajectory were similar which means the largest differences and benefits are due to the speed policy. The results presented in Fig. 7 are with this mode turned off since it was implemented after the date of publication, and we compare directly with the algorithm presented in [3], however the authors wanted to make this note for the reader.

Next, we discuss specific metrics presented in Fig. 8 and Table II. We point out that through the evolution of these algorithms, the overall percentage of the time in which all the agents are in range of their target, and are simultaneously in range is improved for the same circuit and mission duration. For the full convoy, where all agents are +/- 4 meters from their ideal follow range, this is boosted from 7% to 24%. with more significant gains for each individual agent. The percent time in which agents are in a convoy is presented as a strict metric and the difference may not be obvious from this alone. Consider the subplots within Fig. 8, where the spread of the time series indicator variables in the third row on the right hand side and the synchronized blue-on-red dot plots seen in the second row of the left column.

These two plots illustrate where in the trajectory and in time were all agents simultaneously in range. We can see that for **ConvoyBaseline**, this only occurs at around 500 and 1000 seconds, which is also only towards the end of the zig-zag circuit on the long recovering leg, which is depicted with the blue-over-red markers. Comparatively, for **ConvoyPD** with and without partial state coupling this spread of indicator variables is more consistent, and the mean nominal follow range error is improved. The best straightness and turning indices are exhibited in **ConvoyPD** without partial state coupling, which is likely due to allowing followers to copy the leader's dynamics at a particular point while being more forgiving in the overall structure of the convoy.

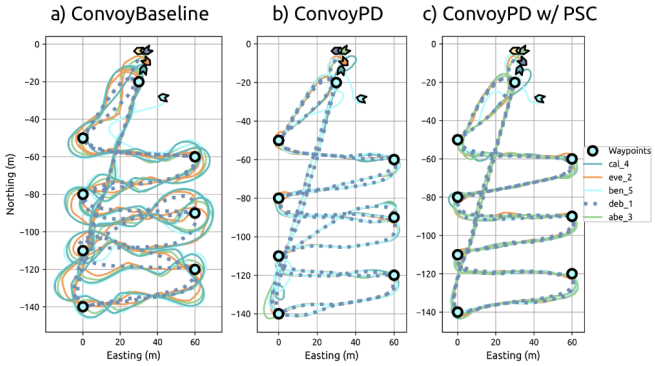

Fig. 7. Stress test trajectory tracking simulation results for a) Convoy Baseline, b) ConvoyPD, and c) ConvoyPD with partial state coupling. The non-smooth trajectory found in part *b* around (0,-120), was due to the intervention of the collision avoidance behavior.

-	**ConvoyBaseline**				
-	All	$k=1$	$k=2$	$k=3$	$k=4$
I_s	0.83	0.89	0.83	0.80	0.81
I_t	0.64	0.62	0.64	0.63	0.65
PTIC	7%	56%	49%	46%	36%
-	**ConvoyPD**				
-	All	$k=1$	$k=2$	$k=3$	$k=4$
I_s	0.96	**0.97**	0.96	0.96	0.95
I_t	**0.92**	**0.92**	**0.98**	**0.89**	**0.87**
PTIC	11%	52%	67%	64%	45%
-	**ConvoyPD w/ PSC**				
-	All	$k=1$	$k=2$	$k=3$	$k=4$
I_s	**0.97**	0.96	**0.97**	**0.97**	**0.96**
I_t	0.80	0.73	0.80	0.85	0.84
PTIC	**24%**	**66%**	**86%**	**72%**	**63%**

TABLE I

SIMULATION RESULT METRICS PRESENTING STRAIGHTNESS AND TURNING INDICES, AND THE PERCENT TIME IN CONVOY (PTIC).

B. Field Trials

For our presentation of field trial results, we also provide an example abridgment for how the high-level decision variables output by the IvP-Helm, are mapped to a control effort on the vehicles used in testing. We introduce the Heron M300 vehicles, a system model, and control effort mapping.

1) Hardware and System Model: For field trials we used five Clearpath M300 surface vehicles which can be seen in Fig. 1. A simplified 3-DoF dynamic model is presented in Fig. 9 and Eq. 7. While this is a common model, it was originally referenced from [16].

$$\mathbf{H}\dot{\mathbf{v}} = \tau - \mathbf{C}(\mathbf{v})\mathbf{v} - \mathbf{D}(\mathbf{v})\mathbf{v} \tag{7}$$

$$\dot{\mathbf{x}} = \mathbf{R}_{\mathbf{1}}^{\mathbf{0}}(\theta)\dot{\mathbf{v}} \tag{8}$$

Herein the center of mass is described in the world frame S_0 with $\mathbf{x} = [x, y, \theta]^T$, while state variables $\mathbf{v} = [u, v, \dot{\psi}]^T$ describe the vehicles velocity in the vehicle's body-fixed frame S_1. The transformation of velocities only from the vehicle frame to the world frame is given by a rotation about the Z axis, with rotation matrix R_1^0 Here we assume the inertia matrix H is diagonal with parameters $\texttt{diag}([m_1, m_2, I_z])$, and that both linear and quadratic drag, $\mathbf{D_L}$ and $\mathbf{D_q}$, are also diagonal with $\texttt{diag}([b_1, b_2, b_3])$ and $\texttt{diag}([b_{q1}, b_{q2}, b_{q3}])$ respectively, while $\mathbf{C}$ is described below. The thrust model for τ is described in the next section in Eq. 11.

$$\mathbf{D}(\mathbf{v}) = \mathbf{D_q}(\mathbf{v}) + \mathbf{D_L} \tag{9}$$

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & -m_2 v \\ 0 & 0 & m_1 u \\ m_2 v & -m_1 u & m_1 u \end{bmatrix}$$

2) Clearpath Frontseat Layer: The Clearpath M300's accept one input for each thruster, which is a scalar value ranging from -100 to 100 which are mapped to PWM values supplied to thruster's brushless motors. For the following control laws, the IvP-Helm outputs a desired heading and speed, however there are two interfacing processes. First, an application **pMarinePID** is an PID controller for achieving the desired heading and speed for the vehicle. It takes the output from IvP-Helm and produces a thruster magnitude ζ_d and a mixing term m. m is dependent on how aggressively the vehicle should turn to the left or the right relative to the difference in the desired heading and actual heading, and is normalized between -1 and 1. Thereafter, a MOOS application, in this case called **iM300**, uses this mixing value to generate final inputs the frontseat computer for both desired left and right thruster inputs, ζ_L and ζ_R respectively. The default mixing function in **iM300** for the Heron ASV is not configured to reverse, and thruster inputs are only between 0 and 100.

$$\zeta_{L,R} = \zeta_d \pm \frac{1}{2}\zeta_d m \tag{10}$$

This PWM mixing function is illustrative, and in practice is frequently adjusted depending on mission requirements and desired turning performance. Lastly, these high-level thruster terms provided by the backseat interface to the frontseat controller, are abridged to the following thrust model for the Clearpath Heron M300's is included below in Eq. 11, from [17].

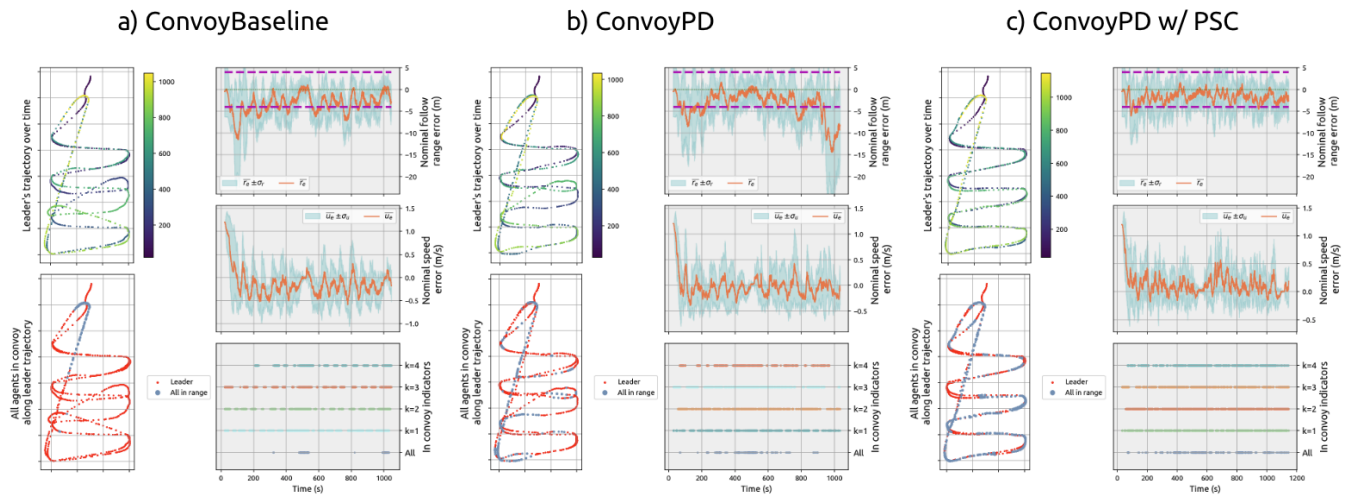

Fig. 8. Stress test trajectory tracking simulation results for a) Convoy Baseline, b) ConvoyPD, and c) ConvoyPD with partial state coupling.

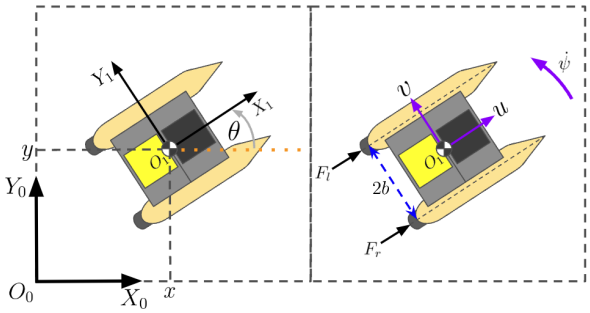

Fig. 9. Diagram for 3-DoF simplified dynamic model of a Heron ASV.

$$\tau(\zeta_L, \zeta_R) = \begin{bmatrix} \zeta_L & \zeta_L^2 & \zeta_R & \zeta_R^2 \\ 0 & 0 & 0 & 0 \\ p\zeta_L & p\zeta_L^2 & -p\zeta_R & -p\zeta_R^2 \end{bmatrix} \begin{bmatrix} \theta_{a1} \\ \theta_{a2} \\ \theta_{a3} \\ \theta_{a4} \end{bmatrix} \quad (11)$$

3) Field Trial Results: Field trials were conducted over several days with winds between 4-7 m/s to validate key simulation results. Practical considerations necessitated some modifications to the test setup after testing **ConvoyBaseline**, including adjusting waypoints to maintain reliable Wi-Fi communication, while roughly preserving the same mission duration. Results are presented in Fig. 10 and Table II.

The field trials largely corroborated simulation findings, with some notable observations. For one, when there were longer distances between turns, the nominal follow range error is the most stable among the different policies despite propagated errors which is depicted in figures presenting the trajectories. This is likely since the speed policy essentially has a deadzone, and if all agents are in range, they mirror the same speed as their target, resulting in less noise or variations in control effort. Similar results may have been realized if the control gains were set higher for **ConvoyPD**. We also point out that while the percent time in which all the agents are in range is best for **ConvoyPD** *without* partial state coupling, individually the agents were in range of their target for a longer duration of the mission. The authors would defer the readers to reference once more the spread of the indicator variables over the time series data, to observe that the spread is more consistent throughout the mission. These results validate the performance improvements observed in simulations and highlight the trade-offs between different convoying approaches in real-world conditions.

VI. Conclusions

This paper presents a novel decentralized algorithm for linear convoying, with improvements or unique advantages over current state of the art methods. The algorithm and implementation is developed within a layer-based approach through behavioral autonomy within MOOS-IvP, to allow the use of guardrail behaviors such as collision avoidance, operating region, or other behaviors. Through simulation, we demonstrate superior tracking performance and linear convoying for underactuated systems. We first eliminate propagated errors with a new algorithm, and we then propose a control law which enables excellent trajectory tracking for serialized agents in a convoy. We then demonstrate, how through partial state coupling, overall formation structure is improved.

Future work includes implementing and testing additional formations with these approaches, testing with heterogeneous systems, exploring new approaches for declustering and transitioning into a convoy, and developing and investigating transitioning control modes between off-trajectory and on-trajectory states.

Acknowledgment

The authors would like to thank Filip Strømstad, Tyler Paine, Mikala Molina, and Sam Huang for supporting field tests at the MIT Sailing Pavilion.

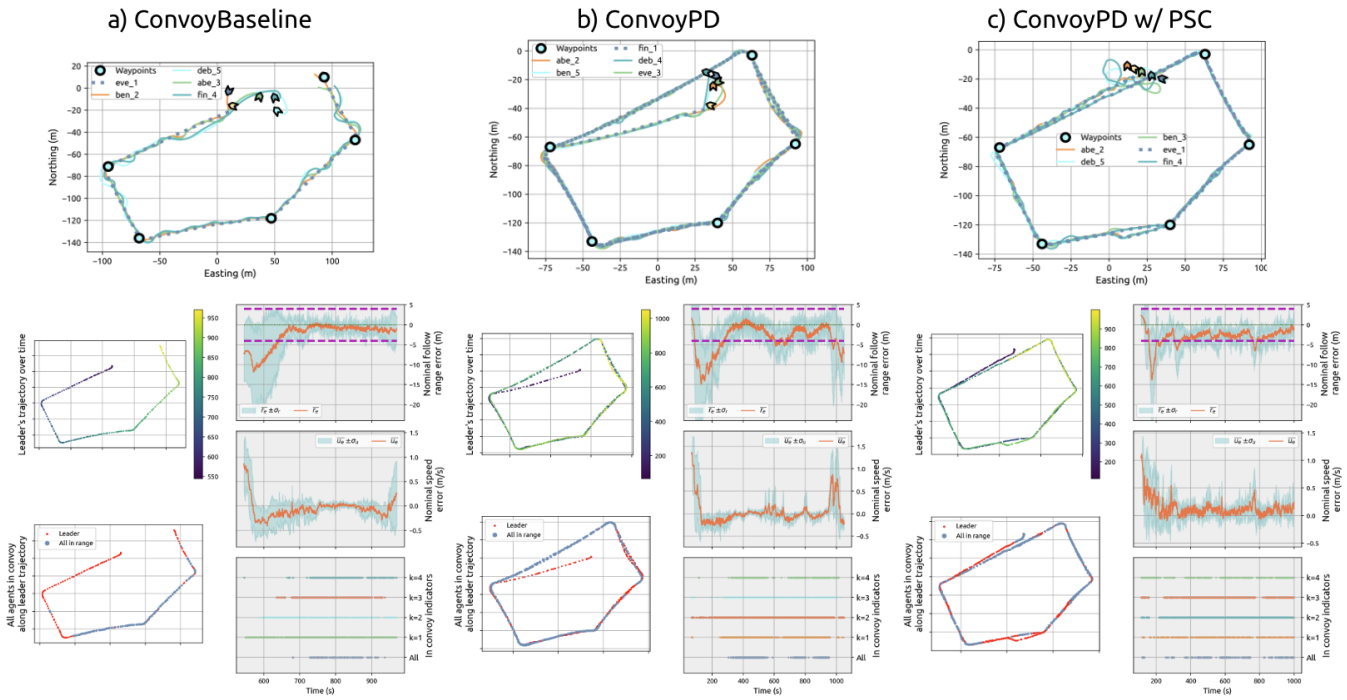

Fig. 10. Field trial results for a) Convoy Baseline, b) ConvoyPD, and c) ConvoyPD with partial state coupling.

-	**ConvoyBaseline**				
-	All	$k=1$	$k=2$	$k=3$	$k=4$
I_s	0.87	0.95	0.90	0.83	0.80
I_t	0.36	0.42	0.38	0.29	0.33
PTIC	16%	**91%**	66%	55%	48%
-	**ConvoyPD**				
-	All	$k=1$	$k=2$	$k=3$	$k=4$
I_s	**0.99**	**0.98**	**0.99**	**0.99**	**1.00**
I_t	0.60	0.65	0.59	0.60	0.56
PTIC	**30%**	64%	88%	59%	46%
-	**ConvoyPD w/ PSC**				
-	All	$k=1$	$k=2$	$k=3$	$k=4$
I_s	0.96	**0.98**	0.96	0.95	0.95
I_t	**0.66**	**0.67**	**0.66**	**0.65**	**0.67**
PTIC	25%	61%	**90%**	**74%**	**54%**

TABLE II

FIELD TRIAL METRICS PRESENTING STRAIGHTNESS AND TURNING INDICES, AND THE PERCENT TIME IN CONVOY (PTIC).

REFERENCES

[1] S. Nahavandi, S. Mohamed, I. Hossain, D. Nahavandi, S. M. Salaken, M. Rokonuzzaman, R. Ayoub, and R. Smith, "Autonomous convoying: A survey on current research and development," *IEEE Access*, vol. 10, pp. 13663–13683, 2022.

[2] M. R. Benjamin, H. Schmidt, P. M. Newman, and J. J. Leonard, "Nested autonomy for unmanned marine vehicles with moos-ivp," *Journal of Field Robotics*, vol. 27, no. 6, pp. 834–875, 2010.

[3] M. R. Benjamin, T. Paine, and S. Randeni, "Autonomy algorithms for stable dynamic linear convoying of autonomous marine vehicles," in *OCEANS 2021: San Diego – Porto*, pp. 1–10, 2021.

[4] L. Xiao and F. Gao, "A comprehensive review of the development of adaptive cruise control systems," *Vehicle System Dynamics - VEH SYST DYN*, vol. 48, pp. 1167–1192, 10 2010.

[5] J. Ploeg, B. Scheepers, E. Nunen, N. Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 10 2011.

[6] T. Balch and R. Arkin, "Behavior-based formation control for multi-robot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 926–939, 1998.

[7] R. C. Arkin and T. Balch, "Aura: principles and practice in review," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 9, no. 2-3, pp. 175–189, 1997.

[8] S. He, M. Wang, S.-L. Dai, and F. Luo, "Leader–follower formation control of usvs with prescribed performance and collision avoidance," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 572–581, 2019.

[9] U.S. Department of Homeland Security, United States Coast Guard, Washington, DC, *Navigation Rules and Regulations Handbook*, 2021. Available: U.S. Coast Guard Navigation Center.

[10] M. R. Benjamin, "Autonomous colregs modes and velocity functions," Technical Report MIT-CSAIL-TR-2017-009, Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory, Cambridge, MA, 2017.

[11] B. Cole, M. R. Benjamin, and S. Randeni, "Ais-based collision avoidance in moos-ivp using a geodetic unscented kalman filter," in *OCEANS 2021: San Diego – Porto*, pp. 1–10, 2021.

[12] A. Albrecht, N. F. Heide, C. Frese, and A. Zube, "Generic convoying functionality for autonomous vehicles in unstructured outdoor environments," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1949–1955, IEEE, 2020.

[13] C. Y. Kaya, "Markov-dubins interpolating curves," 2019.

[14] S.-J. Chung and J.-J. E. Slotine, "Cooperative robot control and concurrent synchronization of lagrangian systems," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 686–700, 2009.

[15] S. Benhamou, "How to reliably estimate the tortuosity of an animal's path:: straightness, sinuosity, or fractal dimension?," *Journal of Theoretical Biology*, vol. 229, no. 2, pp. 209–220, 2004.

[16] T. Huang, Z. Xue, Z. Chen, and Y. Liu, "Efficient trajectory planning and control for usv with vessel dynamics and differential flatness," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1273–1280, 2023.

[17] T. M. Paine, "Notes on heron usv model identification." unpublished, 2023.