Special Issue: Unmanned Marine Systems

Regular Article

Swarms of Pirates: Red Team Exercises Using Autonomous High-Speed Maneuvering Surface Vessels

Joshua Vander Hook[1], William Seto[1], Viet Nguyen[1], Zaki Hasnain[1], Carlyn-Ann Lee[1], Liam Gallagher[1], Tyler Halpin-Chan[2], Varun Varahamurthy[2] and Moises Angulo[2]
[1]Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA
[2]Surface Targets Engineering Branch, Port Hueneme, CA, USA

Abstract: We present a final overview of the efforts by the Naval Air Warfare Center Weapons Division (NAWCWD) and the Jet Propulsion Laboratory to automate the operation of the largest fleet of autonomous maritime vehicles. The vehicles are intended for large-scale demonstrations and tests of US Navy systems or tactics. This review covers a preexisting distributed architecture for human-in-the-loop control of several autonomous high-speed boats with a main focus on the planning, formation assignment, and formation switching pipeline. Algorithm capabilities are described and validated in simulation and field tests using real-world vehicles. Theoretical lower bounds on the time required to change formations are also derived and used to bound our experimental performance. We are able to present data from the 2019 "final exam" that pitted this architecture in a head-to-head competition, which was won after a perfect run with no hazardous maneuvers recorded under autonomous control. The effort concluded successfully on December 31, 2020, with the delivery of code and documentation after successful integration and testing exercises in 2019 and 2020. We conclude the paper with discussions of limitations, extensions, and suggestions for future work.

Keywords: agriculture, obstacle avoidance, navigation, wheeled robots

Nomenclature

Acronyms

CAN	Controller Area Network
CARACaS	Control Architecture for Robotic Agent Command and Sensing
COTS	Commercial off-the-shelf
HSMST	High-Speed Maneuvering Surface Targets

Received: 01 February 2021; revised: 01 August 2021; accepted: 22 September 2021; published: 20 May 2022.

Correspondence: Joshua Vander Hook, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, USA, Email: hook@jpl.nasa.gov

DOI: https://doi.org/10.55417/fr.2022029

INS	Inertial navigation system
NAVAIR	Naval Air Systems Command
PCCU	Portable Command and Control Unit
RHIB	Rigid hull inflatable boat
SeaCAN	Network of all nodes on boats, all operator stations
STEB	Seaborne Targets Engineering branch of NAVAIR
USV	Unmanned surface vehicles

Symbols

$\mathbb{U}$	Bounds on control
A	Mapping of vehicles from current position to desired formation points
$c(\cdot)$	Cost of formation switch
M	Makespan, or total time of convergence for a formation switch
m_v	Model of virtual leader dynamics
N	Number of boats or vehicles in autonomy group
o_i	Vehicle i's spatial offset from virtual leader position
P_i	Path of vehicle i given by set of positions indexed by time
$r_{i,s}$	Stage s safety radius for vehicle i
t_i	time required for boat i to reach its goal formation trajectory
t_{LB}	lower bound on time required for a boat to reach goal formation trajectory
U_i	Vehicle i's control sequence indexed by time
u_v	Virtual leader's speed and heading
X	Set of vehicle positions
$X_i^\star$	Moving formation position of point i given by model
x_i	Position of vehicle i
x_v	Position of leader, may be virtual
Y	Set of formation positions
y_j	Position of formation point j

1. Introduction and Objectives

Two decades ago, in the year 2000, a small boat laden with explosives collided with the USS *Cole*, killing or wounding dozens and critically damaging the ship. Since then, there has been increasing use of small, high-speed boats, especially in large numbers, against commercial or military ships, either as bombs or to intercept commercial vessels such as oil tankers or harass military escorts. This mirrors the threat predicted by the 2002 "Millennium Challenge," a large-scale combat exercise that, somewhat controversially, highlighted the threat of large numbers of small, highly maneuverable vehicles.

In response, the need for intelligent simulation of small, maneuverable, and swarming threats has never been greater. The Seaborne Targets Engineering Branch (STEB), as part of Threat/Target Systems Division of the Naval Air Warfare Center (NAVAIR), has been converting small vehicles into remotely controlled targets to support US Navy operations testing for the past two decades. As part of their ongoing development in Port Hueneme, California, STEB maintains the largest fleet of remote-controlled aquatic vehicles in the world. To accommodate new and developing threat models, reduce operator requirements, and modernize for robotics research, STEB partnered with NASA Jet Propulsion Laboratory to build autonomous and semi-autonomous capabilities into the vessels deployable in low-cost, off-the-shelf computing platforms.

The primary goal of this collaboration is to increase the number of vehicles simultaneously commanded by a single driver, with the eventual goal of fielding a fully-autonomous "red team" for test and evaluation of tactics and systems against a fast-moving, asymmetric opponent. After three years of rigorous, single-blind competed testing, the system was selected against other competitors

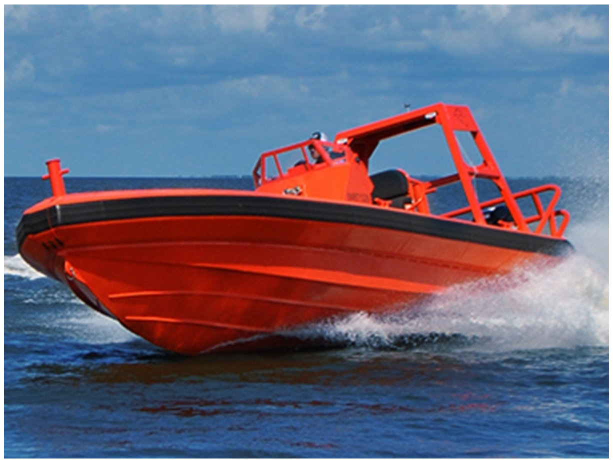


Figure 1. The High-Speed Maneuvering Surface Target (HSMST).

and is entering deployment. This report presents the high-level details of the program's output.[1] The paper is focused on the High-Speed Maneuvering Surface Targets (HSMST—Figure 1).[2]

We organize the paper as follows. In Section 3, we overview the existing systems employed by STEB. The system has been in operation for over 20 years and has a long legacy and robust modular architecture which is being modernized as part of this effort. We discuss the required autonomous capabilities and the design for human-in-the-loop operations in Section 4 and our implementation of those capabilities as a path-planning pipeline in Section 5. We describe our analysis techniques and analytical benchmarks in Section 6. In Section 7, we show the resulting capabilities in simulation tests. In Section 8, we present field testing results from 2019, which are also compared against the derived analytical results and simulation tests. Finally, in Section 9, we discuss limitations, our major lessons learned when designing and deploying the system, and preliminary ideas for more advanced capabilities that might be of interest to the community.

The goal of this paper is to (1) communicate to the community some of the challenges we encountered associated with developing an "autonomy controller" for an established system that sees regular use; (2) present the thoroughly vetted motion planning and formation control pipeline details that should be of general interest; and (3) provide insights into testing mechanisms and processes that helped us.

2. Related Work

Multi-agent unmanned surface vehicle (USV) studies have industrial, commercial, and military applications, and range in number of agents, scale of field experiments, choice of control algorithms, platform architecture, and degree of human interaction (Rowley, 2018; Zhang et al., 2015; Zereik et al., 2018). In the presented work, we focus on performing field experiments of various formations at high speed, with an explicit human-in-the-loop protocol to guide formations with a virtual leader. We use a deterministic approach for formation position allocation and employ model predictive control to determine formation-relative control sequences for fast convergence. We employ obstacle avoidance using optimal reciprocal collision avoidance (ORCA) to ensure that all controls are ultimately collision free. Therefore, in this section, we review the related efforts at automating collaborating maritime vehicles that span the three general application areas relevant to our work: formation assignment, convergence and control, and obstacle avoidance.

[1] Unfortunately, the competing team's data were not shared with us for this report.
[2] This paper is greatly expanded from its preliminary version, which appeared in Vander Hook et al. (2019).

2.1. Formation assignment and control

We describe here experimental and simulation-based studies that feature collective motion of multiple agents to perform formation maneuvers, patrolling actions, and other tasks using different formation assignment and control methods.

The generic problem of formation keeping is often focused on underactuated systems. For instance, simulations in Arrichiello et al. (2006) maintain formations of underactuated USVs in the presence of obstacles and sea currents by decomposing the control problem into coordination via a behavior-based controller that outputs desired velocities and maneuvering using a low-level control system that translates desired velocities to actuator inputs. Bio-inspired methods are also used; for example, Fu and Wang (2018) combine a bio-inspired speed controller with a sliding-mode controller for leader-follower formations of underactuated USVs to overcome the problem of speed jump in backstepping control methods.

Modeling external disturbances and realistic environments is a key challenge in planning for maritime applications. Peng et al. (2015) simulate cooperate dynamic positioning of a fleet of USVs connected by a network and subject to ocean disturbances using a predictor module for estimating disturbances and dynamic surface control. In an in-silico study (Dai et al., 2017), a decentralized adaptive formation control is achieved using a combination of prescribed performance control for transient performance specifications, neural network for model uncertainties, disturbance observers, dynamic surface control technique, and Lyapunov synthesis. Liu and Bucknall develop and simulate practical path-planning routines for maritime USV formations in realistic environments consisting of static and dynamic obstacles using variations of the fast marching method (Liu and Bucknall, 2015; Liu and Bucknall, 2016; Liu and Bucknall, 2018a) and employ a leader-follower paradigm such that the leader moves with constant velocity, and followers are allowed to vary speed to maintain formations.

The use of neural networks in control pipelines has been studied in Wang et al. (2020) where a leader-follower fleet is kept in formations defined by distance and angle constraints using an adaptive control strategy consisting of barrier Lyapunov functions and neural networks that account for model uncertainties. Similarly, in Peng et al. (2017), USVs get in formation around a virtual leader using a recurrent neural network to solve for optimal guidance signals and a fuzzy system to estimate unknown kinetics, thereby adhering to the velocity constraint and minimizing control torque during the transient phase.

In resource monitoring applications, the large scale of deployments poses special communication challenges and limited opportunities for human-in-the-loop operations. A low-cost, multi-robot platform for monitoring and sample collection over large areas with minimal human intervention is successfully designed and tested in Valada et al. (2014) where the use of smartphones, centralized control, and long-range communication yield robust performance in field tests where regions are assigned to vessels using random and highest uncertainty tracking sampling algorithms. In Duarte et al. (2016), a swarm of 8 agents is used for environmental monitoring where the decentralized control is formulated via a neuroevolutionary algorithm that trains an artificial neural network to transform sensor data to desired speed and heading. Often these applications lend to systems with a large number of agents; for example, in Zoss et al. (2018), a rule-based decentralized control algorithm is used to achieve collective behaviors, including flocking, navigation, and area coverage for swarms of up to 50 self-propelled buoys in open environments. At large scales, coordination between agents is focused on achieving connectivity in communication, such as Yu et al. (2019), which simulates minimally actuated multi-agent systems with synchronized trajectories to maximize communication and form connected sensor networks in gyre-like flows.

Threat monitoring and asset guarding applications often require unique objectives. For example, in Fan et al. (2020), a formation is required to move toward surrounding a target point simultaneously, and the tasks are assigned using a modified Dubins path method that incorporates rigid body constraints. Sensitive marine areas are patrolled and guarded using a fleet of USVs using a cluster-space control method that allows a human operator to monitor and control formation parameters in Mahacek et al. (2011) where the system is validated with a fleet of robotic kayaks. Note that this cluster-space approach does not require a formal assignment step. Harbor patroling experiments are performed by a three-boat platoon in Antonelli et al. (2014) and Marino et al. (2014) using a combination of Voronoi tessellations and Gaussian Processes in a decentralized strategy, where online optimization is used to assign the interception task to an agent with the shortest travel time to the threat. Raboin et al. (2013) develop a market-based approach for the task-planning problem of multiple USVs guarding a valuable asset from potential intruders by performing different actions collectively where tasks are assigned incrementally and a set of allocations are assessed using model-predictive simulation.

2.2. Obstacle avoidance

There are many approaches for avoiding obstacles within a motion-planning routine that generate safe trajectories or behavior patterns for multi-agent systems (Liu and Bucknall, 2018b; Rossi et al., 2018). A few of the techniques are velocity obstacle and its variations (Fiorini and Shiller, 1998; Van den Berg et al., 2008; Wilkie et al., 2009; Snape et al., 2011), model predictive control (Park et al., 2009; Ji et al., 2017), artificial potential fields (Ge and Cui, 2002; Warren, 1990), and roadmap-based approaches such as visibility graphs (Alt and Welzl, 1988; Kunchev et al., 2006), probabilistic roadmaps (Kavraki et al., 1994; Hsu et al., 2006), and Voronoi diagrams (Zhou et al., 2017; Breitenmoser et al., 2010). Here, we use optimal reciprocal collision avoidance (ORCA) (Van Den Berg et al., 2011a), which consists of improvements to reciprocal velocity obstacle (RVO) (Van den Berg et al., 2008) in the form of conditions that guarantee collision-free navigation in the presence of multiple robots. Of all the methods in literature, only velocity obstacles (and therefore ORCA) guarantee collision-free paths and will take immediate action to avoid collisions. The next-most-popular variant, Artificial Potential Fields, can significantly delay obstacle avoidance actions and does not guarantee that all agents can choose collision-free paths.

The next sections describe the system, its requirements, its design, and its performance.

3. System Architecture Overview

The goal of this section is to overview the existing system. For an overview of the requirements of the final design, see the next section. The planning and high-level control pipeline will be presented in Section 5.

The boat under test is the High-Speed Maneuvering Surface Target (HSMST, Figure 1). The HSMST is an 8-meter-long commercial rigid hull inflatable boat (RHIB). It has two outboard engines that enable it to achieve high speeds in fair and moderate seas. The HSMST is already used in a variety of systems testing or training scenarios to replicate various high-speed surface threats, especially multi-vehicle operations. Each HSMST has a local network of processors—the nodes—interconnected by a Controller Area Network (CAN) bus. The network of all nodes on each boat, across all boats, and all operator stations is called the SeaCAN network. A system diagram for the local SeaCAN network on each agent is shown in Figure 2. In general, each node is a single-board computer or microprocessor that has a single functional responsibility. For example, there are nodes to control the vehicles actuators (e.g., rudders and throttles), a node to run the engine interfaces, a node for instrumentation, one for radio communications, and a removable node for autonomous control (Autonomy). The Autonomy node can therefore be removed and is drop-in compatible with any vessel in the fleet. The modular design has been verified on various types of vessels from personal watercraft to large ships. STEB maintains a wide variety of vehicles types, but this work is focused on the HSMST.

The remote control system provides (via the onboard nodes) throttle, rudder, engine, and auxiliary devices control (if present). It also provides the operator with the following information via telemetry: speed, engine RPM, GPS location, and, heading. The command, control, and telemetry

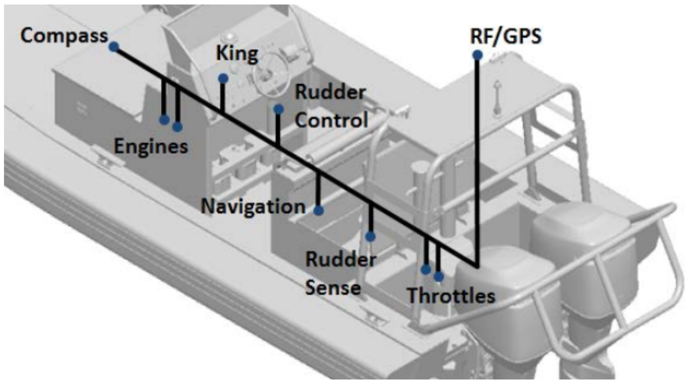


Figure 2. SeaCAN is the onboard remote control system for networks of seaborne targets. It uses the CAN bus to send a set of standardized messages between microcontrollers called nodes. Each node has a specific function such as control of an actuator or sensor management. The modular design provides the flexibility to be used in a variety of seaborne platforms from personal watercraft to ships. (Note: nodes shown are notional. "King" is the operator input station, and "Sense" is an optional sensor package).

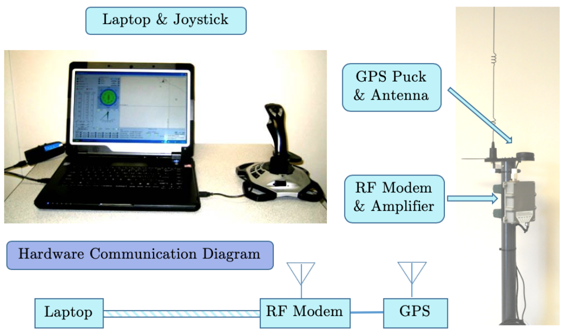


Figure 3. A block diagram of the PCCU (Portable Command and Control Unit). The unit connects to the SeaCAN network to transmit commands to each vehicle on the network, including remote-control or commands to Autonomy nodes.

exchange is accomplished via radio link. The Command interface is called PCCU (Figure 3). At minimum, it allows joystick control of any vehicle remotely. When an autonomy node is present on the vehicle, it also supports heading/course-keeping, drag-and-drop waypoint control for one or more vehicles, transits in formation, and several demonstration-specific commands such as following arcs or weaving paths, all discussed in the next section.

Each HSMST can be remote operated, directed by onboard autonomy or controlled by an onboard safety driver. Autonomous control is easily overridden by the onboard safety operator in the event of unsafe behavior. Safety drivers are always onboard except for the most aggressive test regimes that might endanger operator safety because of rough seas or during live fire exercises. Drivers are also not used when a human operator cannot react quickly enough because the boats are moving too fast, or are too close (for example, Section 8 and Figure 17). An example of the 2018 test as viewed from the control station is in Figure 4.

This architecture was fixed prior to the initiation of Autonomy development by operational constraints of operating a large number of vehicles over a long range. This had direct consequences on the Autonomy node development. Most importantly, it was a preexisting, immutable requirement that the only explicit exchange of information between the boats is a shared world model consisting of the boat's current operating mode, position, and velocity. This was shared at a varying rate from 3 to 10 Hz.

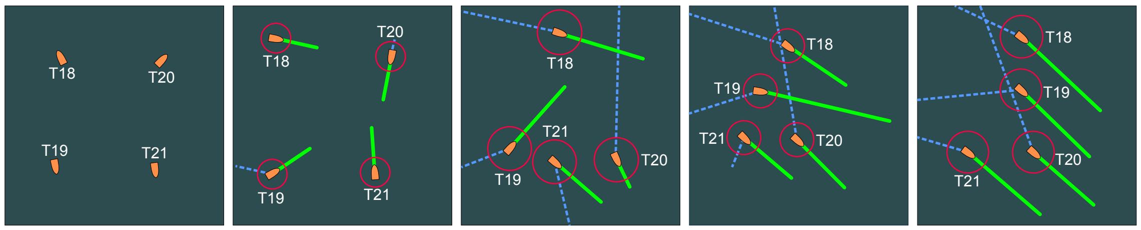


Figure 4. Playback of on-water data showing positions and velocities of the four HSMST test vehicles. This is the view from the PCCU control station during the 2018 tests. The green lines show velocity vectors of length proportional to velocity. The red circles around each vehicle are the standoff distances (overlapping circles is a failure). The boats were traveling at up to 25 knots during these tests.

This introduces a difficulty in designing Autonomy, however, since now vehicles cannot explicitly exchange information with each other. Therefore, all coordination must be *implicit* (i.e., control and decision processes are necessarily distributed).

4. Autonomy Overview and Requirements

The requirements described in this section were levied on the autonomy teams as part of the development program and were not subject to change. In addition, "blind" tests of formations, speeds, and maneuvers not included in the initial requirements were included in the test event. These were withheld from release to publication. The overall pipeline to realize these capabilities is discussed in the next section and is shown in Figure 10. The qualitative goal was to perform fast, agile maneuvers, in particular synchronized weaves and formation switches with minimal transition time.

The autonomy node designed for STEB at NASA JPL was built on top of the CARACaS autonomy framework for distributed robotic systems (Huntsberger et al., 2008; Huntsberger and Woodward, 2011; Woodward et al., 2017). When the vehicle is operating in autonomous formation drives, it is responsible for the safety of the vessel and operator (if present), as well as responding to operational commands such as the direction, speed, or type of the formation. Commands are sent in a shared bus architecture over the SeaCAN network, meaning all Autonomy nodes can read data from any other node.

HSMSTs can be issued commands individually or as a group. Groups are how autonomy is aware of which vessels it should cooperate with during commands. For example, software on one vehicle is aware that it should be in formation with other vehicles because it has received a formation transit command and knows which vehicles are in its group.

For example, during changes in formation, it is possible for two boats to have to swap positions. However, each boat is only given the set of parameters (e.g., formation positions) relevant for *its own* operation. To be effective with limited bandwidth and almost no ability to exchange information, autonomy is built to infer the intended actions of the other vessels.

4.1. Safety of navigation

To accommodate safe operations, STEB has set up a framework for determining safe operating speeds, minimum safe separation between vessels, and safe command sequences during testing. The safety requirement is expressed as a minimum approach distance and maximum speed, both measured by telemetry. Several factors play into the choice of safe speeds. The most relevant factors are proximity of operations, presence of humans, and sea state. Machines are given the same maximum speed constraints as human operators. Speed constraints are currently specified by human operators as part of the regular operation by observing current conditions and consulting safety protocols. (See discussion in Section 9.)

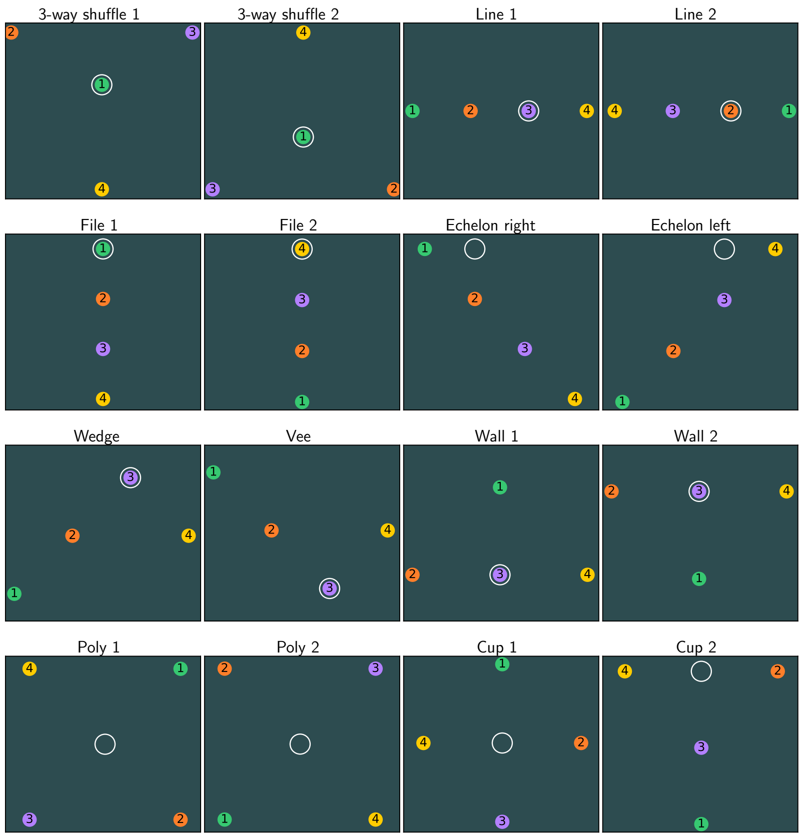


Figure 5. All formations presented in this paper, (virtual) leader positions x_v are shown as (virtual) white circles. Test results of transitions between these formations are presented in Section 8.

Whatever the safe speed, collisions during autonomous operations are a serious risk whenever human operators are onboard. Each vessel has a safe separation distance. As with safe speeds, standoff distances take into account several factors such as waves, winds displacement, speeds of operations, and types of maneuvers. At full speeds and in close proximity, improper maneuvers can cause a collision in less than a second. Thus human operators are not used, and autonomy is expected to intervene and avoid collisions by steering, reducing speed, or shutting down the vehicles and warning the operators.

4.2. Manual and mixed operation

Each HSMST can be remotely operated by an operator on the network or on the vehicle. The vehicles must allow seamless operation for the PCCU operator to switch between manual and autonomous control at any time. In addition to onboard override of autonomous operation, the PCCU allows joystick input, which is passed to the onboard navigation node. Autonomously maneuvering vehicles must give way to manually operated vehicles.

4.3. Waypoint navigation and velocity keeping

When commanded on course speed or toward a waypoint, onboard autonomy is responsible for setting a desired vessel velocity. To allow portability between different vehicle types, a lower-level, preexisting velocity controller is provided on each vessel via the navigation node.

Waypoint navigation consists simply of a line-following algorithm implemented as a feed-forward proportional controller. Velocity control is passed through to the vehicle's throttle and rudder controller. The primary role of autonomy is to keep the vehicle safe from collisions by steering around them while proceeding as commanded.

4.4. Formation control

The main ability required is the control of many vehicles by a single operator. The concept of operations is to allow a single operator to assign boats into formations and specify high-level commands to each formation. As such, the system allows a human operator to control one or more virtual boats that appear in the situational awareness picture of each autonomous vessel (i.e., its location and velocity is broadcast over the network). The boats then "form up" with the virtual boat. The virtual boat is the *leader* in what follows and can be steered through any maneuver or commanded in the modes above. The human operator can switch control between multiple virtual leaders and therefore control many formations simultaneously. A real vehicle, even a manually operated one, can be used; we will refer to any formation leader as a virtual leader going forward.

The formation for the autonomous vehicles can be preset so each vessel knows its assigned position relative to the leader. These are called *static formations*. In *shape formations*, parameterized shape is defined, but the location and assignment of formation points must be calculated by all boats simultaneously. Contrast with an *adaptive formation* in which the formation points are specified fully but not which agent should occupy which position.

Static and adaptive formations are specified as potentially time-varying offsets $o_i(t)$ for the ith boat, from the virtual leader position $x_v(t)$. Each boat receives a coordinate pair denoting the offset specified in a coordinate frame centered on the virtual leader and aligned with its direction of travel (Figure 8a). The PCCU interface allows preloading many formations so an operator can rapidly change the desired formation for one or more groups. By specifying the offset as a parameterized time-varying function, the formation can undergo smooth maneuvers such as weaving or gradual changes. The main requirement was for stepwise changes in formation so the onboard planner had to determine transition strategies.

4.4.1. Formation transitions

The key capability, and the one that underwent the most rigorous analysis and design, was optimal formation transitions. When the formation commands change, the boats are expected to expeditiously and safely assume the new formation without conducting maneuvers that wildly differ from what a human operator would expect to see, such as oscillations. The bulk of our experiments and analysis for this paper involved testing formation switches between a set of the formation types illustrated in Figure 5 and comparing the measured time with derived, analytical bounds as well as time from simulations. The formation list was levied as a requirement under the program, as were the rest of the capabilities described in this section. We were not privy to the selection process. See Section 9 for more discussion on this point.

4.4.2. Formation weaving

Typically, the goal of formation control is to converge to a fixed (in the local coordinate frame) formation point in the steady state. However, during formation weaves, the (local) desired formation point is time-varying, and the vessel will follow a sinusoidal pattern that oscillates around the nominal formation point (Figure 6). The bottom row of Figure 14 illustrates a simulated formation weaving scenario, specifically in a *line* formation.

4.4.3. Operational improvements

The above requirements support three key operational improvements enabled by quickly reassigning formation points. These are *arbitrary startup*, *dynamic groups*, and *reshuffle*. Arbitrary startup allows a formation to quickly form or re-form when starting (or continuing) a formation transit

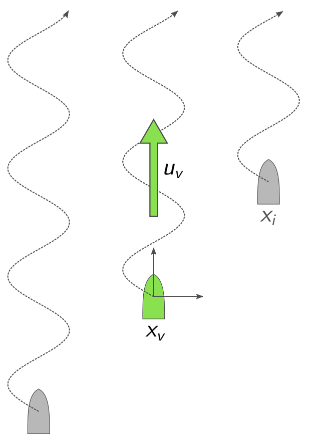


Figure 6. Weaving maneuver on a three-boat echelon left formation, where leader's position x_v and velocity u_v are shown in green. The weaving paths are in-phase and are designated by a wave amplitude and wavelength.

(Figure 4). Dynamic groups allow the formation to accommodate vessels joining or leaving formations safely and with minimal control effort. Reshuffle allows the formation to accommodate sharp turns with minimized effort by swapping formation points; for example, a line-abreast quickly reversing its course (Figure 7). Safe, reliable, intuitive execution of these capabilities builds trust in the system by untrained operators and allows streamlined test-day events, further decreasing training and operational overhead of the system.

Examples of each of these are shown in Figure 9. These capabilities went through single-blind, metric-validated tests starting in 2017, the winner selected in 2019, and culminating with a 2019 operational deployment. We present a subset of all the formations (Figure 5) and transitions between formations. We discuss these field tests in Section 8. Next, we discuss the implementation of these capabilities in detail.

5. Autonomy Design

We now define the motion-planning pipeline that results in the desired capabilities. Our goal was to define a single mode of operation that can provide all the capabilities described previously. The high-level design is shown in Figure 10. In summary, the main loop executed on the autonomy node is completely stateless, idempotent, and reentrant. The situational awareness information is sent into the node over the SeaCAN bus, as well as the command information (formation shape, control mode, etc.), and this is used to calculate the own-ship desired velocity for the current time step.

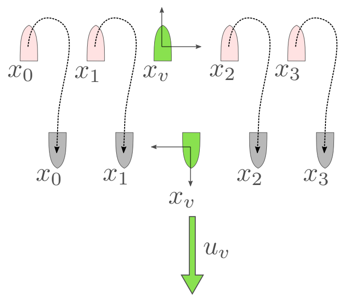


Figure 7. Example of an adaptive line formation where the virtual leader's heading is flipped from the initial direction of travel. The relative order and formation points of the agents change as they *reshuffle* to minimize distance traveled and get into formation faster.

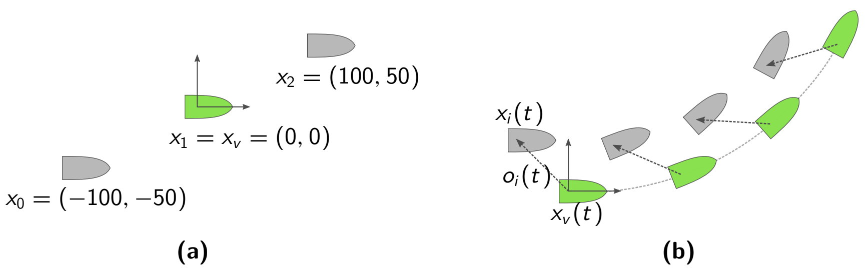


Figure 8. (a) Positions for each boat i are defined by offsets $o_i(t)$ from the leader position x_v (green) and are transmitted as part of the command from SeaCAN. (b) Problem 2 for a single vehicle. A real (or simulated) leader (green) is moving on a trajectory determined by a remote operator. The formation-keeping task is to control each HSMST so they stick to their offset (White), as defined by the leader position.

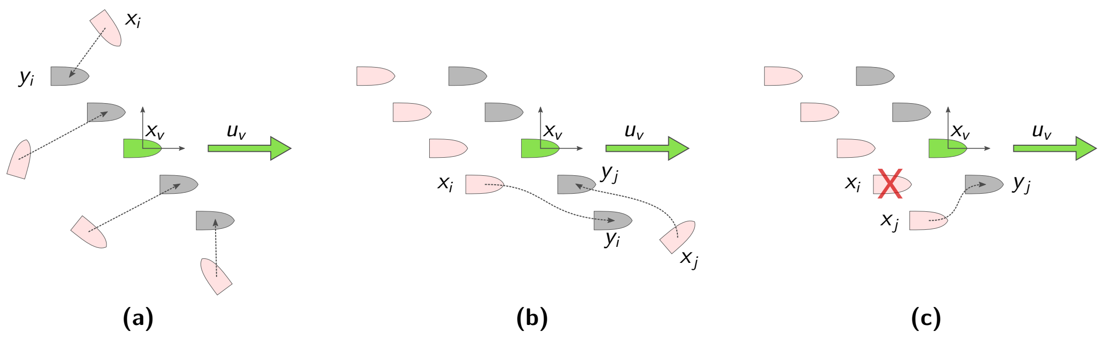


Figure 9. (a) From arbitrary starting positions (red) x_i, the boats use a deterministic assignment algorithm to ensure agreement. Formations can adapt (b) to a new boat at x_j joining at formation point y_j or (c) a boat at x_i leaving the formation. The leader is in green, and direction of travel is shown.

5.1. Prediction of formation path

When a real, possibly manned boat is not used as a leader, the virtual leader is actuated by a remote, human operator, and its position follows a kinematic model based on the joystick commands of the human operator. The virtual leader position and command is known to each boat, and autonomy predicts the path based on the current control input and simple kinematics to generate the formation path.

The kinematic model m_v for the virtual leader is a unicycle model with state: x and y for position, v for speed, and θ for heading. The PCCU captures the joystick motions of the human operator and outputs a commanded speed and heading c_v. The model puts limits on the acceleration a and turn rate δ. Operationally, δ was chosen to be 3 degrees per second.

$$x_v = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{v} \\ \theta \end{bmatrix}, \quad c_v = \begin{bmatrix} v_{\text{desired}} \\ \theta_{\text{desired}} \end{bmatrix}, \quad m_v(t) \quad = \begin{bmatrix} x(t) + v_{t+1}\cos t \\ y(t) + v_{t+1}\sin t \\ v(t) + a \\ \theta(t) + \delta \end{bmatrix}, \tag{1}$$

where $\begin{aligned} a &= 0.2(v_{\text{desired}} - v(t)), \\ \delta &= \max(-\delta_{\max}, \min(\theta_{\text{desired}} - \theta(t), \delta_{\max})). \end{aligned}$

5.2. Derivation of formation points

As stated in Section 4, there are three types of formation modes. *Static* specifies both formation points and assignment of boats to points. *adaptive* formations specify formation points but not

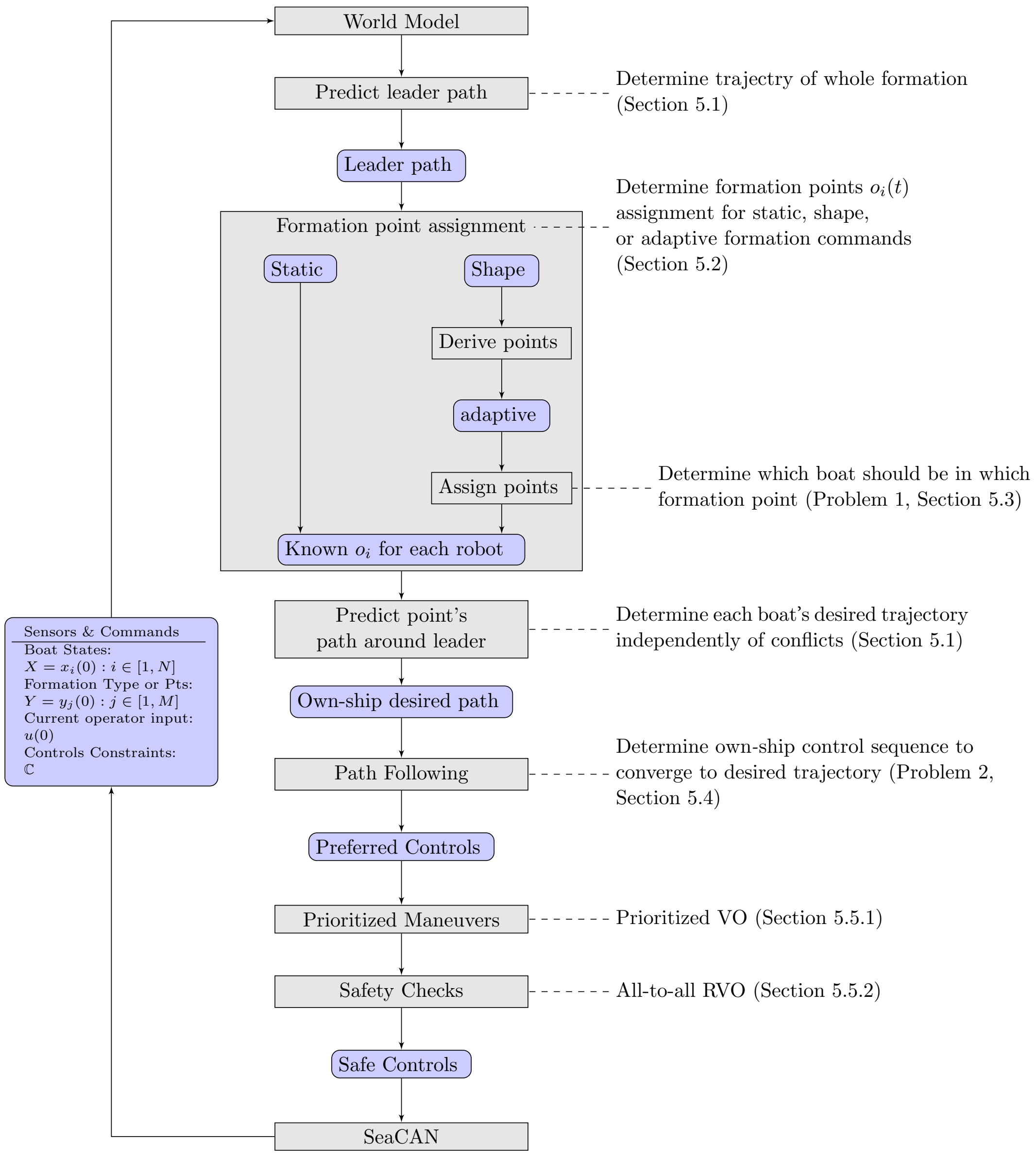

World Model
Predict leader path
Determine trajectry of whole formation (Section 5.1)
Leader path
Formation point assignment
Determine formation points $o_i(t)$ assignment for static, shape, or adaptive formation commands (Section 5.2)
Static
Shape
Derive points
adaptive
Assign points
Determine which boat should be in which formation point (Problem 1, Section 5.3)
Known o_i for each robot
Predict point's path around leader
Determine each boat's desired trajectory independently of conflicts (Section 5.1)
Sensors & Commands
Boat States: $X = x_i(0) : i \in [1, N]$
Formation Type or Pts: $Y = y_j(0) : j \in [1, M]$
Current operator input: $u(0)$
Controls Constraints: $\mathbb{C}$
Own-ship desired path
Path Following
Determine own-ship control sequence to converge to desired trajectory (Problem 2, Section 5.4)
Preferred Controls
Prioritized Maneuvers
Prioritized VO (Section 5.5.1)
Safety Checks
All-to-all RVO (Section 5.5.2)
Safe Controls
SeaCAN

Figure 10. Autonomy node planning pipeline.

assignment. *Shape* does not specify the formation points or the prior assignment of boats to points. By specifying formations with a shape, a small number of bytes broadcast to all boats allows them to use onboard calculations to avoid any ambiguity in the steady-state desired locations of the boats.

For *Static* formations, each boat simply reads its own offset from the PCCU command message and then passes it through as the output assignment.

We deal with *Shape* formations as follows. First, each boat receives a command specifying a shape formation. The types of shape formations are limited to a specific, parametrizable type ("Wedge," "Line," etc.). In our current system, the parameters of the formation are spacing and angle between boats but also includes a pattern of enumeration for adding subsequent boats that depends on the specific formation type. For example, the x and y offset points for the Wedge formation are calculated as follows, where d is the separation distance between the boats, i is the rank of the boat,

$i = 0, \dots, N-1$, and N is the total number of boats.

$$o_{ix} = \begin{cases} -di \cos 45^\circ & \text{if } i \bmod 2 = 0 \\ di \cos 45^\circ & \text{if } i \bmod 2 = 1 \end{cases}, \tag{2}$$

$$o_{iy} = -di \cos 45^\circ. \tag{3}$$

In general, the specific method used to determine formation points from the shape may vary, so long as each boat uses the same method in a deterministic way to arrive at the same results without communication. The output is then a set of points around the virtual leader, but not the assignments, reducing the problem to an *adaptive* formation.

5.3. Formation point assignment

To find its position around the virtual leader, a boat must know the available formation points and must also know which formation point it should assume. We use deterministic optimization algorithms to ensure that each boat can derive as much information as it can without explicit communication.

Given the derived formation points (or commanded, if already in *adaptive* mode), the boats must now determine which of the swarm should occupy each formation point. For this, we solve the following optimization problem.

Problem 1 (Formation assignment). *N vehicles at points $X = x_1, \dots, x_N$ are given a set of formation points $Y = y_j, \dots, y_M$ for $M \geq N$. Each point y is moving with the same course and speed u_v. All agents must choose a mapping $A : X \to Y$ such that the total intercept cost $c(X, A(X))$ is minimized.*

This assignment is the classic job assignment algorithm with an $N \times M$ matrix and was solved using Hungarian Assignment algorithm (Phillips and Garcia-Diaz, 1981) in polynomial time. The assignment regimes were free (boats take any point in the formation, e.g., $M > N$) or rank-based, such that N boats always occupy the "first" N points in the formation (truncate so $M = N$).

5.4. Path following

To converge to the desired formation point, which moves along with the virtual leader, each boat solves a trajectory optimization problem as follows. Let the virtual boat's state be $x_v(t)$ and the virtual boat's commanded speed and heading be $c_v(t)$ at the current time t. Using the model by which the virtual leader's state evolves, $x_v(t+1) = m_v(x_v(t), c_v(t))$, each boat i knows its desired trajectory for the next T seconds by $X_i^\star = m_v(x_v(t-1), c_v(t-1)) + o_i(t)$ for t up to T (Figure 8b).

Problem 2 (Formation point convergence). *Choose a command sequence for the next T steps, $C_i = c_i(t), \dots, c_i(t+T)$, to bring the vehicle's predicted position $X_i = m_i(x_i(t), C_i)$ to the path of the moving formation point, $X_i^\star = m_v(x_v(t-1), c_v(t-1)) + o_i(t)$ (Figure 8b). The vector $\mathbb{C}$ represents the bounds on the control allowable, such as maximum safe speed. The constraints are given as part of the command sequence and not known a priori.*

Our preferred control method to solve Problem 2 is a forward-shooting trajectory optimization algorithm. Our control architecture consists of a high-level model predictive controller (MPC) (Camacho and Alba, 2013) that feeds a desired velocity into a low-level throttle and steering. The MPC optimizes the trajectory and outputs speeds and headings that would achieve the trajectory. The desired speed and heading are passed as inputs to the vehicle controller. The vehicle controller, which we do not have direct access to, regulates the speed and heading of the vessel. It is developed and tuned by the sponsor, and the interface is accessed through the SeaCAN network where the desired setpoints for the speed and heading are specified in a CAN message.

To solve the trajectory optimization problem, the task is formulated as a discrete time optimal control problem where x contains the same state as the virtual leader and u is the control input for this specific problem formulation, which is acceleration and turn rate.

$$x = \begin{bmatrix} \mathrm{x} \\ \mathrm{y} \\ \mathrm{v} \\ \theta \end{bmatrix}, u = \begin{bmatrix} a \\ \delta \end{bmatrix},$$

$$\begin{aligned} \min_{X,U} \quad & \frac{1}{2}\sum_{t=0}^{T}\left\| \begin{bmatrix} x_t \\ u_t \end{bmatrix} - \begin{bmatrix} x_t^\star \\ u_t^\star \end{bmatrix} \right\|_W^2, \\ \text{s.t} \quad & x_0 = \bar{x_0}, \\ & 0 = Ax_t + Bu_t - x_{t+1}, \qquad \forall t = 0,\dots,T-1, \\ & 0 \le \mathrm{v}_t \le \mathrm{v}_{\max}, \qquad \forall t = 0,\dots,T, \\ & -a_{\max} \le a_t \le a_{\max}, \qquad \forall t = 0,\dots,T-1, \\ & -\delta_{\max} \le \delta_t \le \delta_{\max}, \qquad \forall t = 0,\dots,T-1. \end{aligned} \tag{4}$$

It is distinct from the command input c of the actual system, which is speed and heading. The desired speed and heading are obtained from the solution state of the optimization. $\bar{x_0}$ specifies the current state of the boat. W is a weighting matrix that corresponds to cost terms that penalize diverging from the trajectory. X and U are the full state and control trajectories that we optimize over. A time horizon of $T = 7s$ with a time step of 1 second was chosen for the experiments.

$$A_t = \begin{bmatrix} 1 & 0 & \cos\theta_t & -\mathrm{v}_t\sin\theta_t \\ 0 & 1 & \sin\theta_t & \mathrm{v}_t\cos\theta_t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_t = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5}$$

For the problem constraints, A and B represent the kinematic model of the boat. The unicycle model from the virtual leader is used but linearized about v and θ at each time step (Equation 5). A maximum speed was also imposed as a safety constraint for the experiment. Typically, $\mathrm{v}_{\max}$ was 25 or 35 knots. The maximum acceleration and turn rate were chosen to be 7m/s^2 and 25 deg/sec, respectively. These values were based on a rudimentary system identification process of the boat where measurements were taken at a number of steady state conditions, and then the maximum values observed were chosen. Future work would involve finely modeling the throttle and rudder to more accurately capture the control input, but for the purposes of this work, we were explicitly told to not make any assumptions about the physical characteristics of the boat, as the intent is to be able to deploy the autonomy system on any type of vessel. Having a separate high-level and low-level controller achieves a good balance for this requirement.

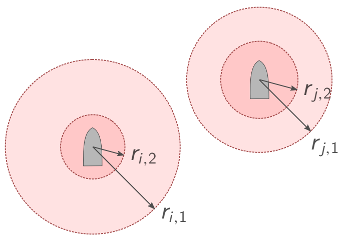
Finally, the control problem is transformed into a standard quadratic program in order for a solver to handle it. The open source solver, qpOASES (Ferreau et al., 2014), is used to solve the standard form, where the optimization variables are the entire trajectory states and controls stacked into a vector $w = [x_0, u_0, \dots, x_{T-1}, u_{T-1}, x_T]$. H and g represent the costs and the constraints to capture the dynamics.

$$\begin{aligned} \min_{w} \quad & \frac{1}{2}w^\top Hw + w\top g, \\ \text{s.t} \quad & lb \le w \le ub, \\ & lb_A \le Aw \le ub_A. \end{aligned} \tag{6}$$

5.4.1. Formation weave following

Building on the formulation for formation point convergence, the time-varying offset point that defines the desired weaving path of the vessel evolves according to the following, where o_{ix} and o_{iy} are the x and y coordinates of the assigned formation point in the local frame of the virtual leader.

$$X_i^\star = m_v\left(x_v(t-1), c_v(t-1)\right) + o_i(t), \tag{7}$$



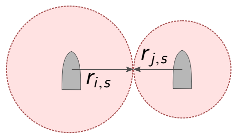


Figure 11. (a) The two-staged keep-out zones of vehicles i and j. Each radii can be changed and varies between vehicles or tests due to operational conditions. (b) Auto-shutdown occurs when safety buffers overlap and distance between vehicles i and j is less than $r_{i,s} + r_{j,s}$ for a given stage s.

$$o_i(t) = R_\theta(t)\left[\begin{pmatrix} o_{ix} \\ o_{iy} \end{pmatrix} + \begin{pmatrix} A\sin(\frac{2\pi vt}{\lambda} + \phi) \\ 0 \end{pmatrix}\right]. \tag{8}$$

$R_\theta(t)$ is a rotation matrix parameterized by the heading of the virtual leader, such that we are transforming our offset point into the current frame of the virtual leader at time t. v is the velocity of the virtual leader. A, λ, and ϕ are the amplitude, wavelength, and phase specified in the PCCU command message. The weaving amplitude is perpendicular to the direction of travel. The same model predictive control technique above is employed in order to track this path.

5.5. Safety checks

Regardless of the mode of operation, each vehicle is responsible for its own safety margins. The simplest of safe-keeping methods is to stop all vessels that get too close. In the event that any two boats violate a safety standoff distance as in Figure 11b, autonomy is designed to stop the vessel. However, if the vehicles detect an impending collision but have not yet reached the standoff distance, they can recover by coordinating their maneuvers to avoid each other.

For mutual avoidance maneuvers, we deploy velocity-obstacle-based collision avoidance because it caused fewer safety shutdowns during testing, was easier to tune for any situation, and had a minimal effect on the formation convergence times. NASA JPL has a long history of projects using velocity-obstacle-based collision avoidance (Kuwata et al., 2014). The collective decision-making implicit in ORCA has several tangible advantages on our formation control design. In traditional, single-agent VO, a boat in the middle of a tight formation would initiate a hazard avoidance maneuver but would be constrained by its neighbors, resulting in a solution that requires slowing down. However, using ORCA, all boats take partial actions to allow the middle boat to avoid collision (i.e., splitting the formation).

We use a combination of traditional velocity obstacles and ORCA in our hazard avoidance. Each agent has two sets of keep-out zones (Figure 11a). The first radius is responsible for higher-level heuristics to speed transitions while taking into account safety, and the second is responsible for avoiding unsafe approaches should the first fail.

5.5.1. Priorities

The first, larger stage was not set by operators but was instead set by our subsystem to be larger than the smaller radius for two reasons. First, it allowed a "buffer" between ORCA's assumption of instantaneous velocity changes. For this purpose it was often useful to ensure the first radius was at least 2x the smaller radius and at least as large as the assumed turning radius of the boats. Other tuning produced mostly cosmetic changes (earlier maneuvers versus later in terms of time to collision).

The more important use of the first stage was to inject higher-level priorities into the hazard avoidance *before* a hazard was encountered. This was accomplished as follows. Let c_i be the cost

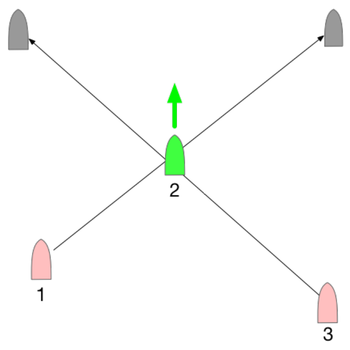


Figure 12. A scenario to illustrate the first-stage hazard avoidance with distance-based priorities. See text for description.

of agent i to reach its moving formation point, as calculated by the straight-line time to intercept (Section 6). We sort these costs and assign a priority according to this ordering so that a priority $p_i = j$ implies boat i has the jth longest to travel to reach its assigned formation point, with $p_i = 0$ implying it has the longest to travel. When checking the first stage of obstacle avoidance, each agent will *ignore* (not consider as part of the optimization) any other agent k with $p_i > p_k$.

This is conceptually illustrated in Figure 12. Agents 1 and 3 need to relocate, but their path would take them through agent 2's position. In the prioritized case, Agent 1 would "see" agent 3 as a noncooperative obstacle. Agent 2 would see both 1 and 3 as noncooperative velocity obstacles, and agent 3 would see no obstacles. This has the (perhaps surprising) effect of agent 2 accelerating to make way for 3 and 1.

The desirable effect we achieved by using a first-stage prioritization was improved maneuvering behavior during tight formation switches, such as better symmetry breaking when two boats are crossing or overtaking. We discuss further and compare to ORCA priorities in Section 9.

5.5.2. ORCA for safe-keeping

The second, smaller radius was set by operators as the auto-shutdown radius. Overlapping buffers of this radius was considered a failure during testing, and it was frequently changed from small (a few boat lengths) to large (100s of meters) to assess system performance. We employed ORCA as described in Van Den Berg et al. (2011b) for this layer. To be clear, in no cases was the second stage obstacle avoidance turned off for any vehicle.

5.6. Summary

We have presented in detail the planning and optimization pipeline for our vehicles to achieve safe and quick formation transitions. It is composed of well-known algorithms executed in sequence to steer a lower-level velocity controller. A more detailed discussion of the lessons learned and motivations for our decisions is deferred until Section 9. Before presenting simulated results we first derive some comparison metrics in the form of a useful lower bound.

6. Analytical Performance Lower Bound

To assess the experimental and simulated performance of formation switches, we compare total time required to complete a formation switch to an analytical lower bound. We calculate the lower bound to show how much *additional* cost is incurred by hazard avoidance and multi-agent path finding during transitions by comparing to simulated and on-water experimental timings. The difference in performance versus the lower bound tells us how well the path-planning pipeline is performing, as discussed in the remainder of the paper. Our derived lower bound will not be tight, but it will be

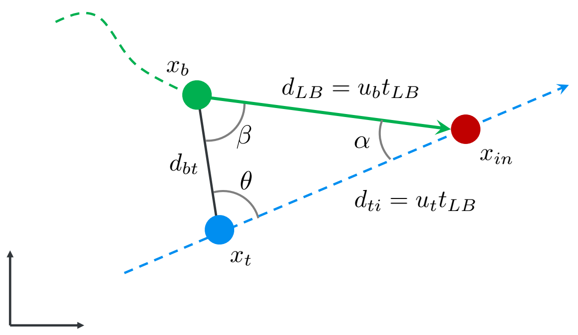


Figure 13. Notation used in lower bound calculations. The starting boat position x_b is shown in green, the starting position of the moving target x_t is blue and moves along the blue dashed line to the point of interception x_{in} (red). The lower bound interception path is shown in solid green.

sound, in that it is always less than the cost, but scales appropriately with the size of the formation. If done properly, a lower bound can provide a system-and-implementation-agnostic way to compare performance across different test regimes, as long as the lower bound is computable in those regimes.

In a formation of N boats, t_i is the time required for boat i to reach its goal trajectory, and the makespan M of the formation switch is the longest of the individual transition times (Eq. 9). The term makespan is used in job-shop literature, and we adopt it here as a performance metric.

$$M = \max_{i=1,..,N} t_i. \tag{9}$$

Therefore, we derive the lower-bound time t_{LB} required for in individual boat with speed u_b to converge to a target trajectory point whose speed is u_t. Consequently, the lower bound of the makespan M_{LB} will be the maximum of t_{LB} of N boats in a formation. We ignore acceleration constraints and assume a constant velocity for the boats and the target formation points. Further, if turning radius and heading are also ignored, the path from the starting position of a boat x_b, to the point where the moving target is reached x_{in}, is a straight line (Figure 13). To find the lower bound distance, direction, and time t_{LB} required for a boat to reach a moving target point, the law of cosines (Boyell, 1976) may be used on the triangle shown in Figure 13.

Assumption 1 **(Lower bound).** *The lower-bound time t_{LB}, required to intercept a target point moving at speed u_t by a boat with maximum speed u_b is given by*

$$t_{\text{LB}} = \begin{cases} \dfrac{d_{bt}}{2u_t \cos\theta} & \text{if } u_t = u_b \\ \dfrac{d_{bt}}{u_t \cos\theta + (u_b^2 - u_t^2 \sin^2\theta)^{1/2}} & \text{if } u_t \neq u_b \end{cases}, \tag{10}$$

where d_{bt} is the distance between the boat and the target point initially, and θ is the angle between the target point's heading and the vector from the initial target position to the initial boat position (Figure 13). This ignores heading and turning radius; therefore, the path from the starting position of a boat x_b to the point where the moving target is reached x_{in} is a straight line.

Proof: By the law of cosines,

$$u_b^2 t_{\text{LB}}^2 = u_t^2 t_{\text{LB}}^2 + u_{bt}^2 t_{\text{LB}}^2 - 2u_t u_{bt} t_{\text{LB}}^2 \cos(\theta), \tag{11}$$

which implies

$$0 = u_{bt}^2 - 2(u_t \cos\theta) u_{bt} - (u_b^2 - u_t^2), \tag{12}$$

$$u_{bt} = u_t \cos\theta + (u_b^2 - u_t^2 \sin^2\theta)^{1/2}, \tag{13}$$

$$d_{bt}/t_{\text{LB}} = u_t \cos\theta + (u_b^2 - u_t^2 \sin^2\theta)^{1/2}. \tag{14}$$

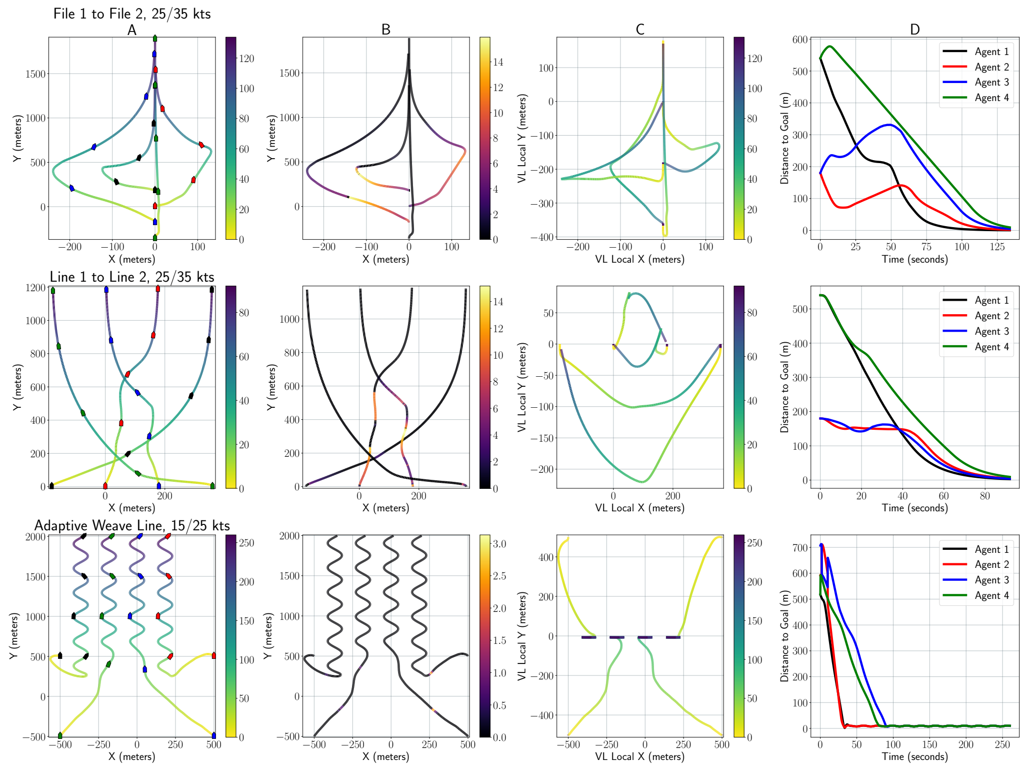


Figure 14. Analysis plots illustrating simulated formation switches. Each row represents a different scenario; the scenario name and nominal/max speed are listed above the leftmost plot in each row. Each column is a different type of analysis plot labeled **A-D**. **A**: the world path taken by each agent (colored by time and agent icon to visualize agent position at a particular time), **B**: world path colored by hazard avoidance effect (magnitude of change in input desired and output safe velocity), **C**: agent paths in the virtual leader (moving) local frame colored by time, and **D**: time versus distance to target formation point. Note that the virtual leader path is not plotted.

The result follows. Furthermore, a condition for the existence of an intercept point is found by the law of sines,

$$\frac{u_b t_{\mathrm{LB}}}{\sin\theta} = \frac{u_t t_{\mathrm{LB}}}{\sin\beta} \tag{15}$$

$$\Rightarrow \sin\beta = \frac{u_t}{u_b}\sin\theta \tag{16}$$

$$\Rightarrow u_b > u_t \sin\theta. \tag{17}$$

■

We report the field test makespan M_{test}, simulated makespan M_{sim}, as well as both these quantities normalized by the lower-bound makespan M_{LB}: $\hat{M}_{\mathrm{test}} = M_{\mathrm{test}}/M_{\mathrm{LB}}$ and $\hat{M}_{\mathrm{sim}} = M_{\mathrm{test}}/M_{\mathrm{LB}}$.

7. Simulation Results

In this section, we take advantage of simulation studies to present summary comparisons of the contribution of priorities to formation convergence and discuss example transitions. We also present insight into the field experiments in the next section using the same methodology. In Section 8, the simulated times to convergence are presented in Table 2 to compare to recorded times in our field experiments as well as the derived bounds.

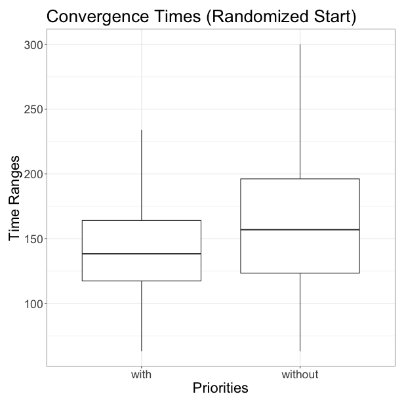


Figure 15. Convergence times.

Figure 14 illustrates a few simulated formation switches, along with a few of different types of plots we use for analysis of the agent behavior. Appendix A presents the entirety of our simulated formations, for reference. In these simulations, each row is a scenario, and four plots are shown. The leftmost is the paths of the vehicles. The second is the paths, overplayed with the "hazard-avoidance effort," which is measured by difference in preferred from actual velocity (yellow is highest). The third plot shows the "formation-local" coordinate frame, which is all the boats' positions over time relative to the virtual leader. The fourth column is the distance of each boat from its intended formation point.

The world path and virtual leader local plots together visualize the agent paths over time in a static plot (see Section 9 for more on the use of these plots in testing). As we run our hazard avoidance in a prioritized scheme, the plot illustrates how agents closer to their target point give priority to agents farther away to minimize time overall for the group. This is most notable in the first row, which we refer to as a "file shuffle" formation switch. The leading agents move aside for the one in the back to come all the way to the front. This is illustrated in plot Figure 14b where we see the magnitude of velocity change due to hazard avoidance greater (lighter colors) for the leading agents.

Note in column B, row 1 (file 1 to file 2) that the trailing boat in the first formation has to move to the lead boat in the second formation. To accommodate a fast transition, the rest of the boats "give way" by moving aside. This does a good job of illustrating the use of priorities discussed in Section 5.5.1. As another illustration, note that in row 2, the control "effort" (column B) was greatest for the middle two boats, which were forced to give way to the outer boats because they had longer to travel. Finally, an arbitrary start is shown in the last row. In this case, there was no need for prioritization or hazard avoidance, and all agents converge rapidly to their tracks.

To quantify some of the benefit of priorities-from-distance, we simulated a batch of 100 formation convergence times from an arbitrary startup. The simulation parameters are given as follows (N implies a normal distribution, and U a uniform one).

Parameter	**Value**
Stage 1 (VO) Time Horizon	30 s
Stage 1 (VO) Radius	60 m
Stage 2 (ORCA) Time Horizon	45 s
Stage 2 (ORCA) Radius	65 m
Max Speed	12 m/s
Formation Speed	8 m/s
Agent Turning Rate	20 d/s
Starting Points (x,y)	U(0,600), U(0,600)
Starting Velocity (x,y)	N(0,6), N(0,6)

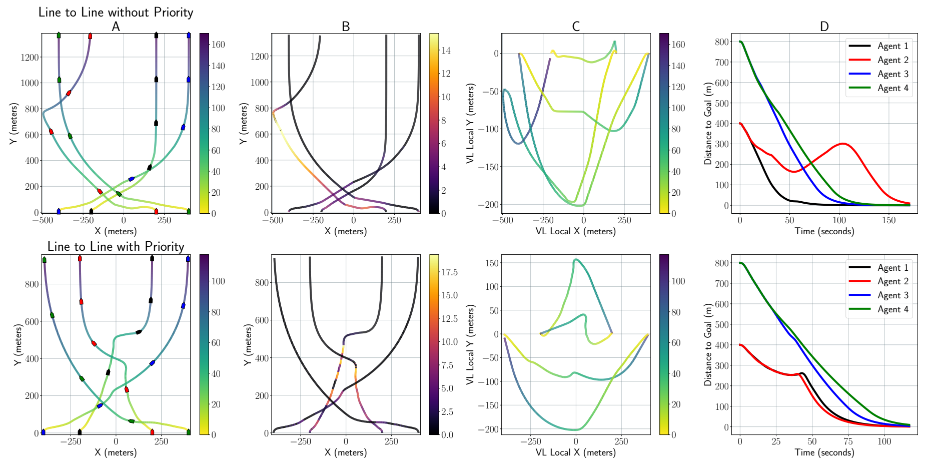


Figure 16. A "crowding out" example that shows the benefit of distance-based priorities. The top row does not use priorities (standard RVO/ORCA), and the bottom row uses the modified prioritization scheme introduced in Section 5.5.1.

We varied with and without prioritization (using priorities as discussed in this paper) but left all other parameters the same. The convergence time distributions are shown in Figure 15 on the facing page or Table 1 on page 905. Note that we capped the simulated time at 300s, and three runs timed out without priorities and none with priorities.

We note a 20% reduction in maximum convergence times. The true reduction is likely a little higher since we capped the simulated time at 300s. Figure 16 shows a particularly interesting case of how priorities help. In our experiences on water, we continually noticed that a "crowding out" occurs when one agent is stuck behind another, yet the minimum-deviation velocity correction does not cause the agent to "go behind." Note in Column B that the agent does not "go around" until its deviation (the color scale) gets very high. To contrast, in row 2, the same agent is allowed to move more directly to its formation point. This was the primary inspiration for our choice of first-stage hazard avoidance wherein agents are allowed to "ignore" possible collisions with other agents that are closer to their destination (and those agents consider the ignoring agent "noncooperative" as in Section 5.5.1). See Section 9 for more on his problem with nonprioritized RVO.

8. Field Experiments

From 2017 to 2020, the motion-planning pipeline was verified using up to four HSMSTs and human operators in Oxnard, California, and the Pacific Missile Range Facility, Hawaii, USA. Tests used the formations discussed in Section 4 (Figure 5).

The tests defined many scenarios, beginning with simple hazard avoidance verification, including head-on collisions (see Figure 17 on the following page), crossing left and right, and overtaking. We also tested up to four vehicles in a "pile-up" where they were all ordered to pass through the same point. These tests were used to validate hazard avoidance and replanning before beginning any formation transit tests. We omit the explicit details of these collision tests. However, additional tests and evaluation of the system described here are presented in Reitz and Wilkerson (2020). Formation weaving was also tested and validated, with Reitz and Wilkerson (2020) performing the bulk of the analysis in comparing the weaves with the desired amplitude, wavelength, and phase. However, for the purposes of this paper, we focus on analyzing the optimality of the formation switches, specifically using time-to-convergence metrics.

To show the scale of the STEB test area, consider Figure 18 that shows an example of a long, steady-state transit of three vehicles under a single operator control. Note the slight gap in the

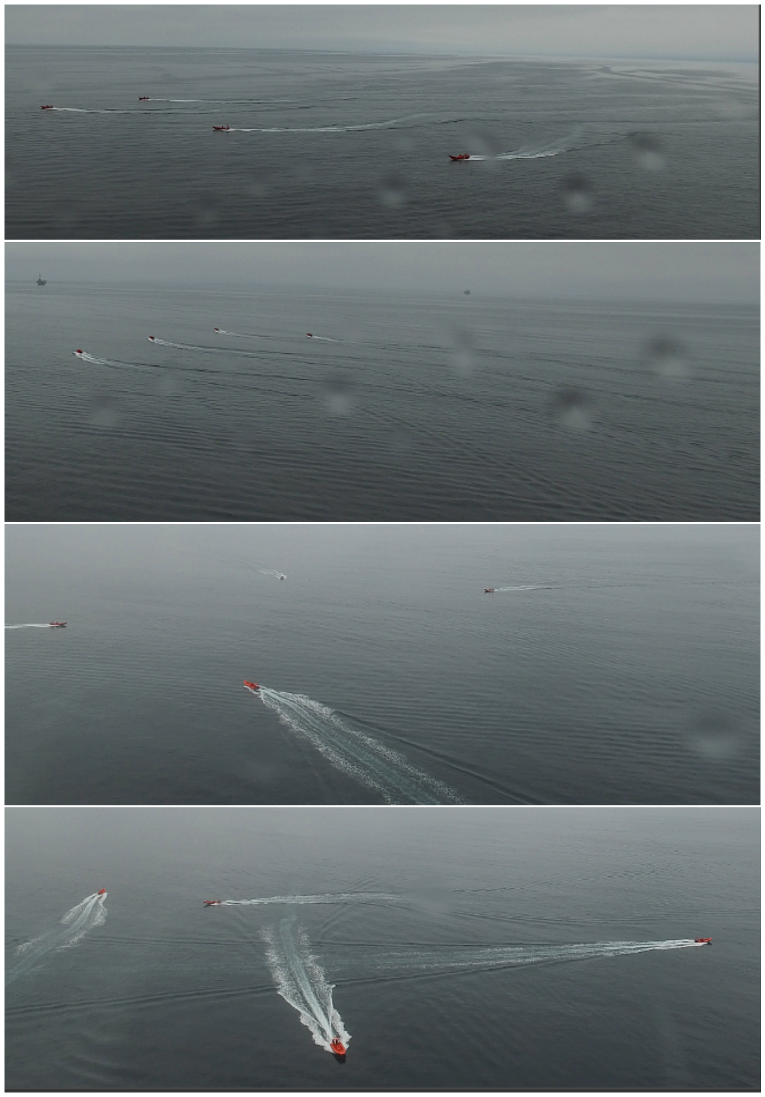


Figure 17. A 2019 test filmed from a remote-controlled quadrotor. The boats are conducting two tests at high speed, a weaving formation transit, and a four-way crossing. The boats are under completely autonomous control and are uncrewed.

tracks, which we attribute to the base station losing communication with the vehicles. Note that the test continued because the boats can communicate with each other still.

For completeness, we present the entirety of a 2018 validation test run that occurred over a week in July. The formations, convergence times, operating speeds, formation heading, and simulated and observed times are presented in Table 2 on page 906. The normalized test and simulated makespans are used to describe the differences in the theoretical and experimental results. Simulated makespan was smaller in all but 7 of the 37 total formation switches studied. The test makespans are 1.6–12.9 (median=4.0) times greater than the lower bound makespans, whereas the simulated makespans are 1.3–6.9 (median=2.6) times greater than the lower bound makespans. For the experimental tests, the normalized makespan was greatest for formation speed/max speed ratio of 15/25; however, the simulated normalized makespan grew with the magnitude of speeds of the boats and targets (see median values at the bottom of the table). Figure 24 plots these results in Appendix B.

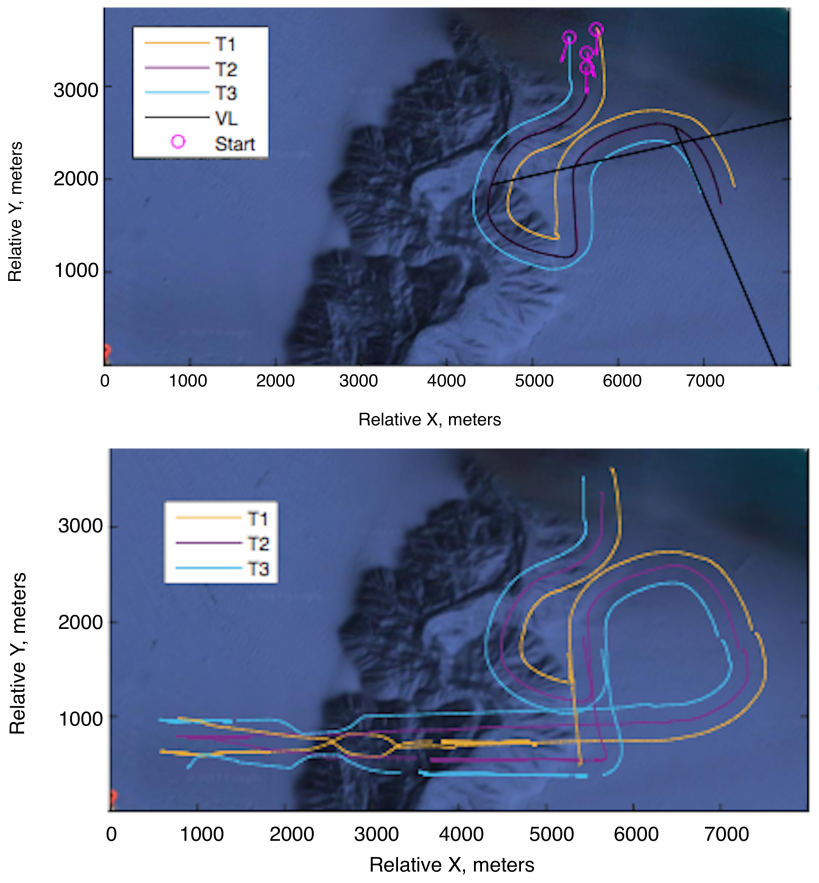


Figure 18. Two examples of long, steady-state formation transits under remote control. (North is up.) Note that in the second figure, the formation begins in the upper right, transitions along the paths shown to the bottom left, then turns 180° abruptly (reshuffling assignments), and then proceeds east through a pair of formation transitions.

We next plot the linear correlation between simulated and test formation switch makespans in Figure 19 and find that the coefficient of determination is $r^2 = 0.601$.

Note that if the field tests matched our expected simulations, we would expect that they would be perfectly correlated on a 1-1 slope (a unit slope is displayed for reference). However, the gap is not so large, but there are strong outliers in some of the higher speed tests (note the upper right quadrants). These "outlier" events *did* occur during a time when sea states were elevated, and it is true that we did not model sea state in our simulations. However, efforts were made by the test conductor to adjust operating procedures to minimize the effect of sea state. This was done to keep safety drivers safe, and its effect on the system was unclear. This qualitative assessment needs to be backed up with significant studies of the effect of sea state on the system. See Section 9 for more.

Note that the speed did not have a substantial effect on simulated and test formation switch makespans as shown in Figures 19–23 where formation switches performed at higher velocities do not necessarily result in faster convergence compared to the lower-bound makespan discussed in Section 6. This implies that for most cases, the makespan lower bound scales along with operating parameters.

Next, we present plots for three scenarios from an on-water validation exercise (an untested event). Specifically, these results depicted in Figure 20 on the following page are separate from the official test trial for which we have reported the timing results in Table 2. The scenarios chosen here

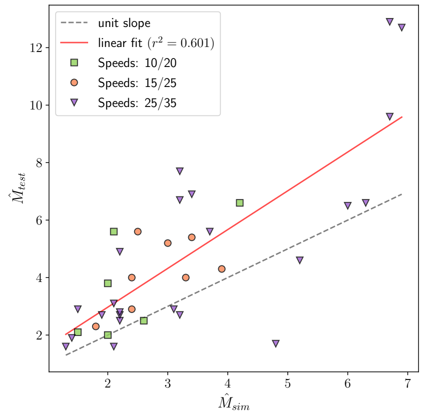


Figure 19. Simulated ($\hat{M}_{\text{sim}}/M_{\text{LB}}$) and test ($\hat{M}_{\text{test}}/M_{\text{LB}}$) ratios for 37 formation switches.

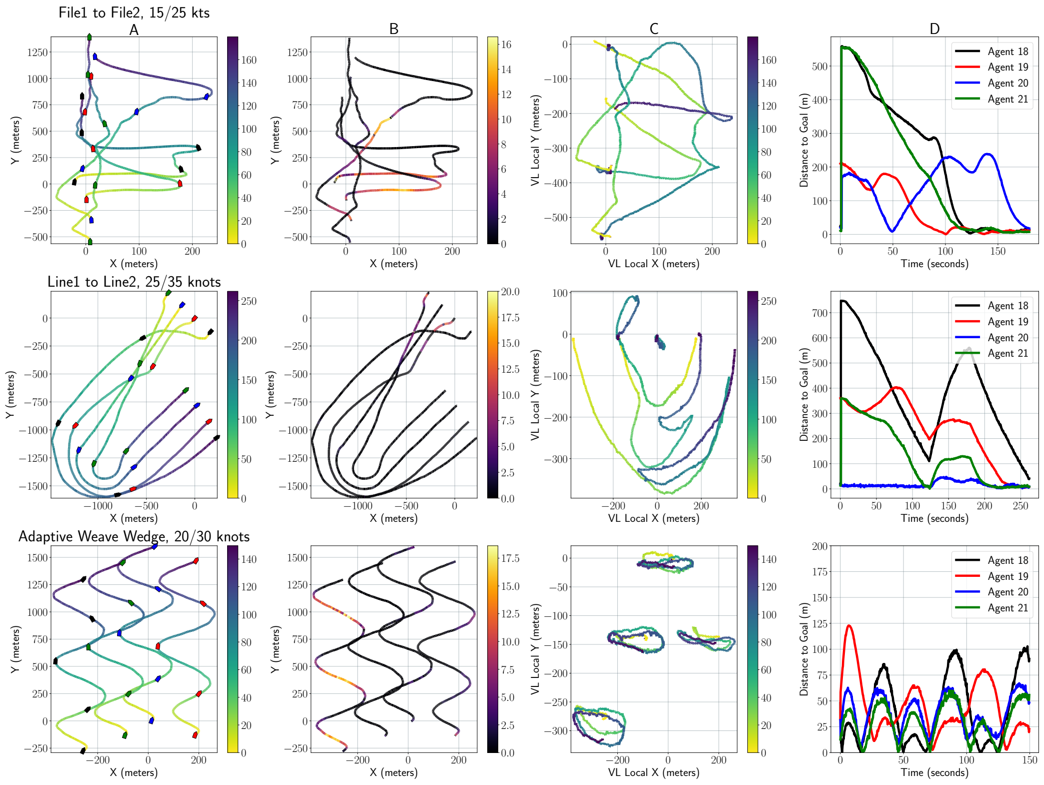


Figure 20. Formation switch scenarios from a live field test. These plots compare to simulations in Figure 14 but with slightly different parameters. Note that line 1 to line 2 switch demonstrates a formation turn as well, which causes convergence to take a little longer.

are to provide a rough analog with the simulated scenarios in Figure 14. The plots are generated by playing back recorded CAN data into our software pipeline and then using the same analysis routine as we did for our simulated data.

In the end, the pipeline described in Section 4 successfully provided hazard-free path path planning in a variety of real-world, high-speed, challenging multi-agent scenarios. It reliably performed close to analytical and simulated results, even with mildly increasing sea state or wind conditions. During all of our verification, no safety violations occurred that were not due to an occasional timeout of radio transmissions, which would cause a breakdown in situational awareness. The reliance on accurate situational awareness is not a problem for the development of on-range test vehicles, but see the next section for more discussion.

8.1. Results

In the final test event, conducted over a week-long period in 2019, the VO-based pipeline we have presented outperformed potential fields (the leading competing method), generating no safety buffer violations during test events and converging to final configurations faster in general (Halpin-Chan, Mr. Tyler, 2019). This team received the maximum possible score. The competing teams' data and implementation were not made available to us for this report for comparison.

9. Discussion

In this paper, we have discussed a distributed architecture for high-speed surface vessels to autonomously conduct formation maneuvers and transitions. The limited bandwidth available for communication necessitates that each vessel uses mostly onboard information to cooperate. Cooperative behaviors were designed and field tested using a live system, verifying the design against operational requirements. All software associated with these capabilities were successfully tested, delivered, and are in active deployment as of 2020 and the conclusion of the program.

In this section, we discuss the caveats, lessons learned, and future work, and suggest recommendations for the community based on our experiences.

Regarding Section 6, we note that the lower bound provided a good heuristic for our own analysis. It helped us debug the worst-running cases by showing us the "room for improvement." In our results, no simulation or test took *less* than the lower bound, implying it was sound. The relatively high ratios (some in Figure 19 were over 12) implies the bounds are loose, but these ratios are not unexpected given the gap between models and reality. A tighter lower bound to assess formation switching performance would be a good contribution for general literature. In particular, a lower bound that takes into account the potentials for collisions (perhaps by counting crossing paths) or changes in heading (perhaps by employing dynamics models or Dubins paths) would greatly assist with scoring activities for simulated and multi-vehicle real systems.

Our project concerned the automation of an existing process, that of remote operation of target vehicles. As such, we had many opportunities to interact with target boat operators, target range conductors, and stakeholders. Experts in naval and test range operations often have very different understandings of the expected behavior of a formation-keeping system than, say, a studied professional in control theory. Significant attention was paid to the maneuvers, not just for their analytical metrics such as time to converge, but also for their "intuitiveness." Unless a benefit could be shown from an autonomous maneuver that was "unexpected," it was often deemed "safer" to mimic a human's decisions. As robots proliferate through test ranges (and perhaps society at large), this will become an increasingly important mismatch. Especially at first, insufficient trust was placed in autonomous decision-making, and split-second deviations from what appears "obvious" would cause safety observers to assume the system has gone awry. For onboard human safety operators, it can be particularly unnerving when boats are allowed to perform close-in, high-speed passes or sudden maneuvers because the onboard planner decided to reassign formation points to speed convergence. The *time-optimal* formation switch can sometimes involve, for example, an individual boat moving "backwards" relative to the virtual leader. Figure 21 shows this case.

In the figure, a three-boat formation is assigned to transition while making a 90° right turn. In static mode, agent 3's new formation point is behind (below, on the page) and worse, across agent 2's path. In adaptive mode, agent 3 could turn left to take the rear formation point, but this makes the agent move the opposite direction in the formation before executing a hairpin turn at high

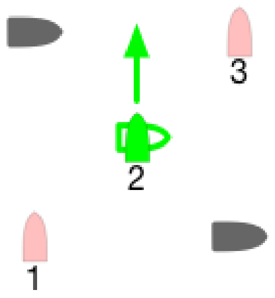


Figure 21. A challenging formation transition where time-optimality contradicts intuitive maneuvers.

speed—a maneuver lovingly termed the “Crazy Ivan” by boat operators. The solution condoned by onlookers is for agent 3 to reduce speed and let the point “catch up” while agent 1 assumes maximum effort to reach the point to its right. This is something like an optimization problem where each agent takes minimum effort actions so long as they are not the bottleneck to fast convergence, and demonstrates the conflict between mathematical/theoretical optimality that operators *say* they want and the reality of what they prefer. We anticipate that *ad hoc* solutions to this problem in systems that interact with each other or with crewed systems would make predictive control and joint optimization extremely challenging.

While we received the highest possible score in the final demonstration, we were all the while pushing against the boundaries of trust by the observers and operators. Fortunately, several volumes on small boat formation operations are available in the community, and we found these to be a useful resources during tuning of priorities for formation switches (Unknown, 1942).

Another caveat of the solution provided in this paper is that the test areas (though expansive) were closed to public traffic. In particular, the system does not provide true “obstacle avoidance” in the practical sense. It can only avoid cohort vehicles (those employing SeaCAN transmitters by design) because it has no way of detecting any other hazards.

A side effect of the closed-range operations was that minimizing the time spent *during testing* was paramount. A multi-vehicle deployment required base access, significant regulatory compliance, full-time safety operator oversight for each vehicle, a fully-manned operations center, and at least a skeleton crew of the autonomy-development team to be on site for each run. The cost to run such a demonstration meant it was cheaper and more effective to build a simulations pipeline that interfaced with actual boat hardware than it was to do multiple on-water test events. STEB prepared a portable hardware testbed that allowed us to test at three levels: simulations of our pipeline without hardware, our pipeline interfaced with their hardware in the loop for single agents, and the whole SeaCAN system running with real radios to emulate packet loss. This was accomplished by providing a simulator connected to real hardware interfaces that abstracted the rest of the nodes in the SeaCAN network. (See Figure 2). This portable and reconfigurable hardware was vital to ensure productive test events and ensure the program ended early and under budget. For example, it was trivial to try a new algorithm, validate on the whole software and hardware stack, and then ship code to STEB for initial checkout before we scheduled a test event. Automating this test process and generating batches of reports similar to Figure 14 were well worth the effort. GNU parallel (Tange, 2011) is used in executing the simulation code.

The discussion of priorities in Section 5.5. differs from the use of scalar priorities in important ways. The usual optimization is to minimize, across all vehicles, the weighted sum of the velocity deviations [$\min_d \sum w_i(v_i - d_i)^2$, where d_i is the hazard-avoidance maneuver for agent i]. ORCA was built for large numbers of vehicles and fast computation, so this is a computationally simple approach. However, Figure 16, shows a good example of a crowing behavior where the minimum sum of deviations *for all agents* overly penalizes one agent. A simplified illustration is shown in Figure 22. Here, conceptually, a *mutual* velocity obstacle defined by their desired velocities blocks

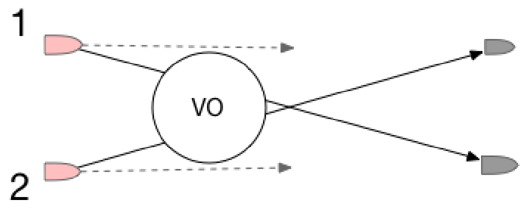


Figure 22. A conceptual diagram showing the problem with RVO for time-optimal transitions. The minimum deviation velocity for both agents is shown as a dotted line, which produces little progress toward the goal points.

any progress for both agents. The agents' attempt to minimize total control effort puts them into a local minimum where no progress is possible. Our particular use case is to minimize time, not control effort, so this is unacceptable. Instead, the formulation of priorities discussed in Section 5.5 produces a solution where one of the agents makes progress, forcing the other to avoid it. We attribute much of the success of the pipeline to our use of velocity obstacles and in particular prioritized hazard avoidance.

There is still the problem that in the real world the estimated distance to goal is essentially a random variable due to uncertain position estimates. If the two agents are essentially equidistant to their goal points when entering this situation, it is *theoretically* a toss-up who will gain priority. Quite remarkably, in testing, this was not a problem. We observed that (qualitatively) a priority assignment causes an immediate and decisive action thanks to the velocity obstacle, and this immediately reduces the lower-priority agent's progress toward goal, reinforcing the solution. We expect this was possible because of decent and reliable GPS solutions *but more so* because of "consensus of state." That is, agents received each other's position estimates by communication and acted on that information rather than onboard sensing. Although priority assignment is theoretically randomized when distances are nearly equidistant, agents agreed on the outcome because they all had identical data to calculate a solution.

This is an important point. In an off-range vehicle, it is quite common to assume only onboard sensing to estimate vehicle and obstacle positions. It is possible that if a sensing and estimation module onboard can provide situational awareness, our pipeline would "just work." At minimum, the presence of vessels would have to be hypothesized based on incomplete sensor information, and the path-planning pipeline would need to trade off risk versus cost in real time. This is outside the scope of the funded effort, but other programs are tackling this problem (Wolf et al., 2017). In particular, given recent, quite astounding advancements in vision-based recognition and tracking communities, we believe this problem to be approachable using the current state of the art. However, our experience suggests that the problem of close-in cooperative motion in an uncertain world may require communication in addition to robust sensing.

For this work, even if obstacles were somehow put into the situational awareness picture, the maneuvers would not be suitable for real-world use because they frequently violated the "rules-of-the-road" that other systems are built around [e.g., (Kuwata et al., 2014)]. Practically speaking, though, the initial adaptation of STEB vehicles for this research is straightforward. The SeaCAN data bus would simply need to be augmented with an additional node on each vehicle that provides situational awareness updates.

In Sections 8 and 5.4, note that the testing did not include slow speeds and that the MPC formulation and modeling was quite simple. Part of this was driven by application: during testing, faster is better to mimic targets' actions during engagements, and constant-speed operations while nearly hydroplaning were simpler to plan. Conversely, the transition between hydroplaning and low-hull control regimes is extremely challenging to model and plan against at a high level. For this reason, each STEB boat comes with an onboard engine and rudder controller that has been designed for a wide variety of sea and speed conditions.

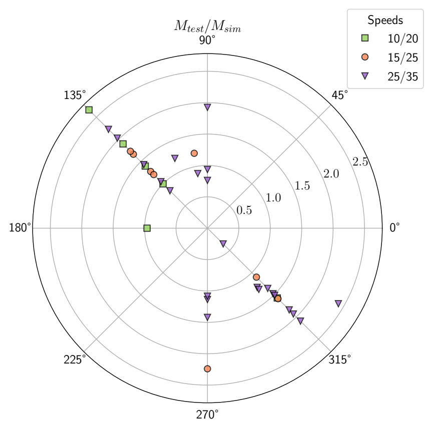


Figure 23. Test ratios versus heading. Note the preference for testing NW / SE trajectories during this event.

For this project, it was a *benefit* that the system was abstracted from this controller and could issue smoothly varying velocity commands to the lower-level, low-level controller, while the low-level controller navigated these different control regimes. While not discussed here, the "smoothly varying" aspect of the planner's output was an important design point so as not to cause problems with the lower-level set point. The details of this low-level controller were not available to us for publication. However, we expect that this "layered" approach is or will be common as it made our team very productive and established clear expectations between the fleet engineers and the autonomy designers.

For maritime operations, this abstraction breaks down after a point. An underappreciated difficulty is that of sea state. Human operators set the maximum safe speed based on observed sea state mostly according to their own preferences, and the trajectory optimization and control algorithms are otherwise agnostic of sea state. In particular, the maximum speed was updated according to an on-site operator's assessment of the prevailing wave height, and the direction of travel for a formation was chosen based on the prevailing wave direction. It was a proscribed feature of the system that these changes could be made in real time by the remote swarm operator. This is evident by inspection of Figure 23, which shows the heading and speeds of the formation for each of the tests that occurred during times where sea state was elevated. A strong preference for cross-shore travel (southeast or northwest) is obvious.

Hull slam or excessive pitch / roll can damage a vehicle or make it uncontrollable by conventional algorithms. A primary difficulty is that conditions vary based on the direction of waves, and so the safe heading and safe speed are tightly coupled. Prior work on sea-state-aware controllers should seamlessly integrate into Velocity Obstacle planners (Ono et al., 2014). Inclusion of sea-state awareness would allow vehicles to simultaneously choose collision-free velocities that are also safe for the vehicles. However, hydroplaning effects such as those experienced by fast-moving vehicles require special care since dynamics become very difficult to model in these regimes.

An even more challenging problem is *recognition* of sea state from onboard sensing. Precise weight, shape, flexibility, and hull composition information is required to be combined with well-calibrated inertial sensors from a variety of locations around the vehicle. For seas with a dominate wave direction and steady total wave energy, this may be possible in real time. However seas are often mixed, meaning model-based methods may fail in these regimes—precisely the ones requiring the most careful guidance, navigation, and control. One can imagine that an autonomous vehicle may employ the same tricks that humans do without fully modeling the sea state (e.g., turning and adjusting speed to "head into" waves), but methods for detecting or predicting safe speeds and directions from onboard sensor data do not appear to be well studied. It is our opinion that providing a solution for sea-state estimation onboard with the sea-state reactive planners receives far less funding than it deserves and would go a long way to improve the operational reliability of seaborne autonomous systems in littoral or stormy environments.

Without handling these above challenges, no system can provide true autonomous navigation and hazard avoidance in the real world, and ours is no exception. What we have provided should be considered an automated test range operation vehicle software suite with some promise for real-world use. We hope that the computational simplicity and clear composition of algorithms and discussion of gaps presented here inspires future work.

With the conclusion of this work, several application-specific problems remain. First, the goal of reducing the offsets between vehicles during transit poses a significant technical challenge. The close proximity leaves less room for control errors and requires much tighter synchronization of all the vehicles' motion plans. Without explicit communication, it might be required that each vehicle more accurately predict the motion of the others' to form a safe motion plan such as Alonso-Mora et al. (2013). The need for improved prediction and planning accuracy, including nonlinear effects from sea state, will at minimum require the use of more computationally expensive solvers.

Ideally, a single operator could control a nearly arbitrary amount of test vehicles. Computationally speaking, the architecture we have discussed scales easily with swarm size. In our implementation, the most computationally costly step was trajectory generation (Problem 2), but this does not change as the number of agents changes. We have verified ORCA /RVO for 40 agents and found no significant slowdown (nearly linear scaling). For larger swarms, the main bottleneck in this framework is the assignment step in Problem 1, which scales quadratically with n. Still, we expect good performance on modern hardware / software well beyond the current field test or demonstration sizes of dozens of simultaneous agents. Addressing scalability without requiring additional communication or exchange of information will remain a key challenge.

In closing, and based on the experiences in this program, we assert that the control of multiple vehicles using high-level commands (e.g., "Move in Echelon Formation to *this* Position") has been demonstrated at high enough technology readiness to be considered commoditizable. The remaining problems are adjacent to the formation control and planning problems, especially operational readiness in noncooperative environments (mostly a sensing problem), cohabitation with manned vehicles (rules-of-the-road and predictability), and extensions to high sea states (a sensing and planning problem). Additionally, there is nothing in our architecture or algorithms that prohibit use of 3D coordinates or alternate motion models, meaning that aircraft formations (or joint aircraft/ground teams) can be easily accommodated, though this is clearly not at a high technological readiness. Furthermore, the emphasis in this work on formation reshuffles, maneuvers, and crossing formations is applicable for semi-controlled environments at present, such as convoys, traffic highways, test ranges, and the like.

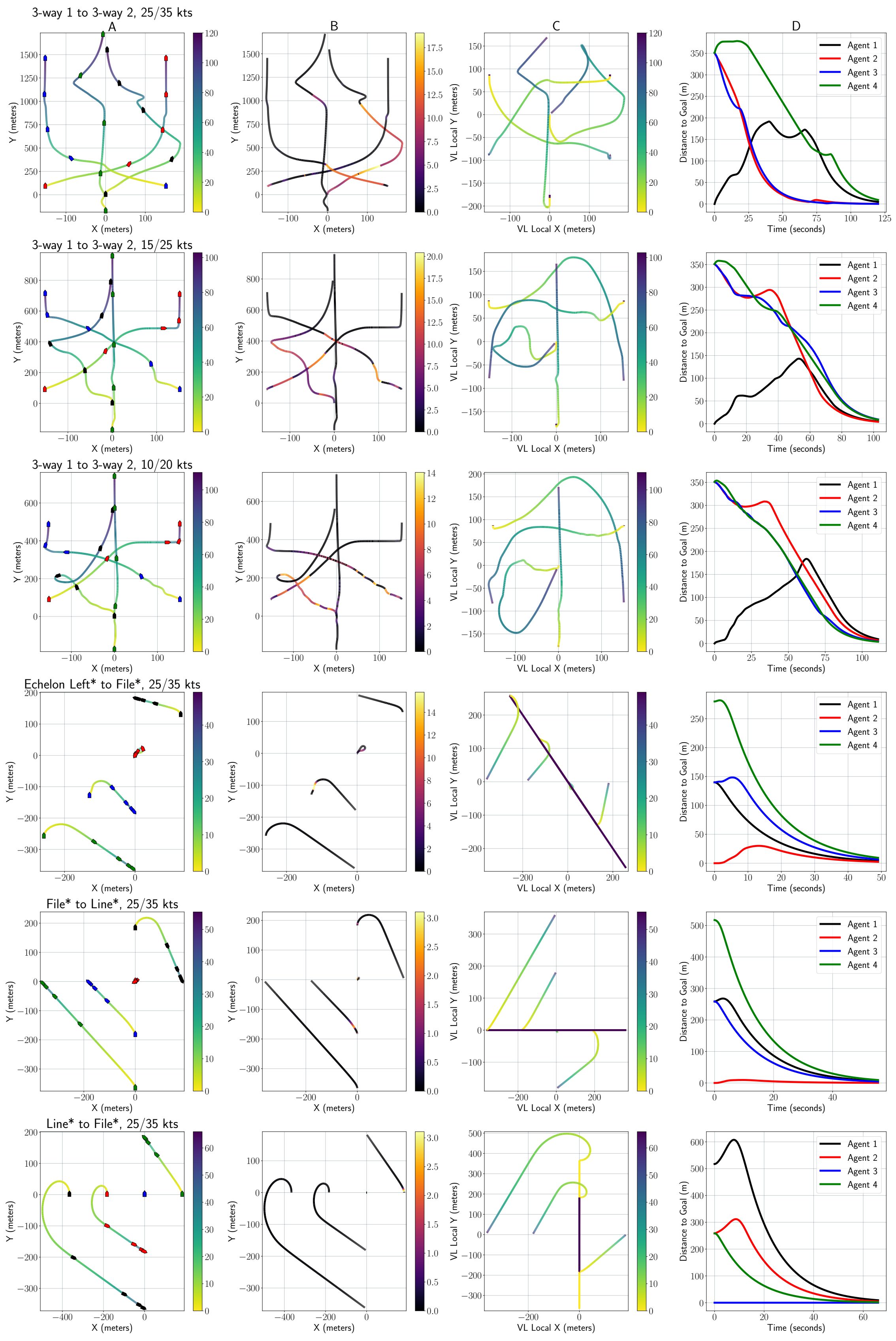
Acknowledgments

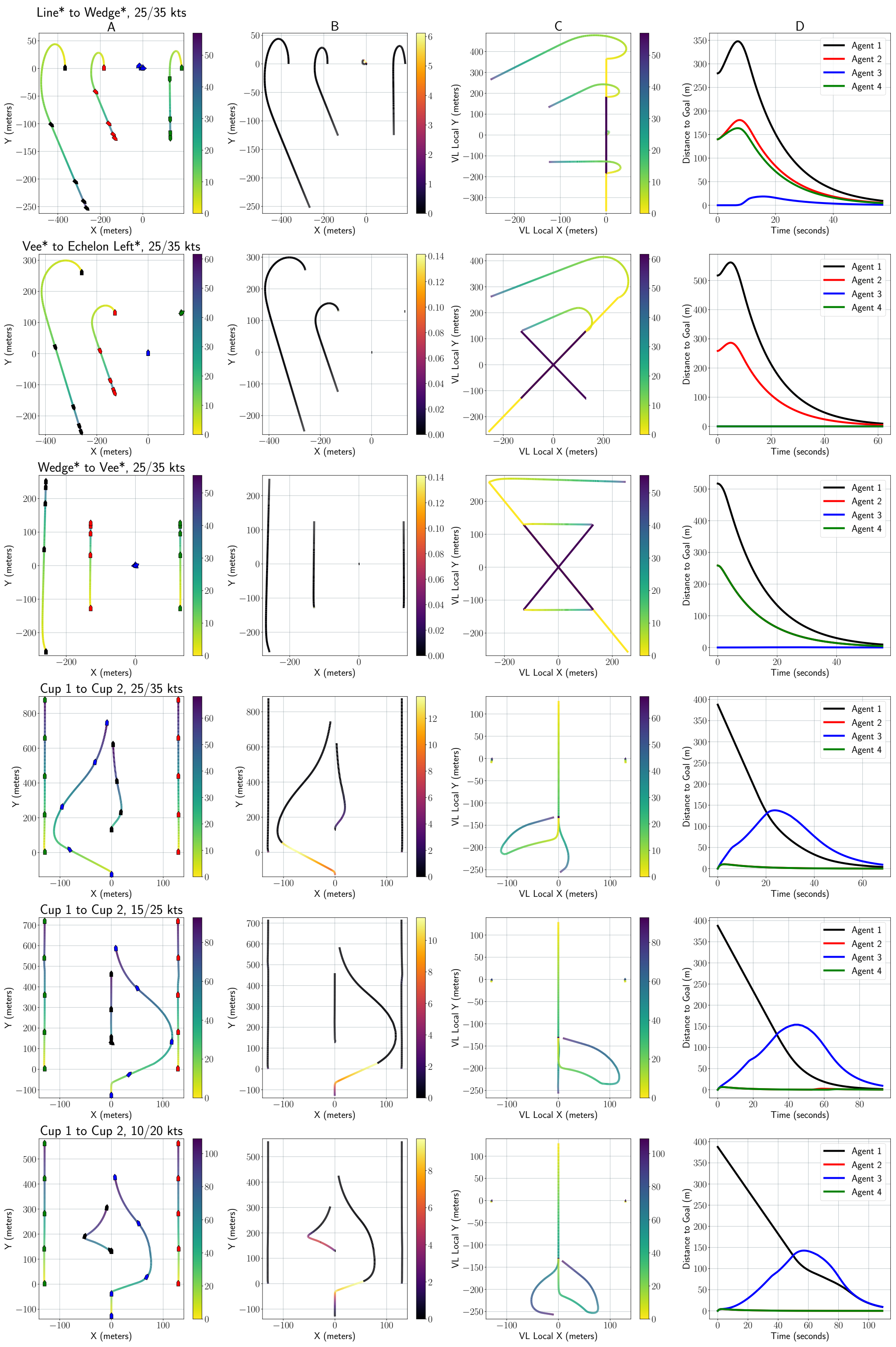
The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration (80NM0018D0004). This project was funded by the Central Test & Evaluation Investment Program (CTEIP). Copyright 2021, California Institute of Technology.

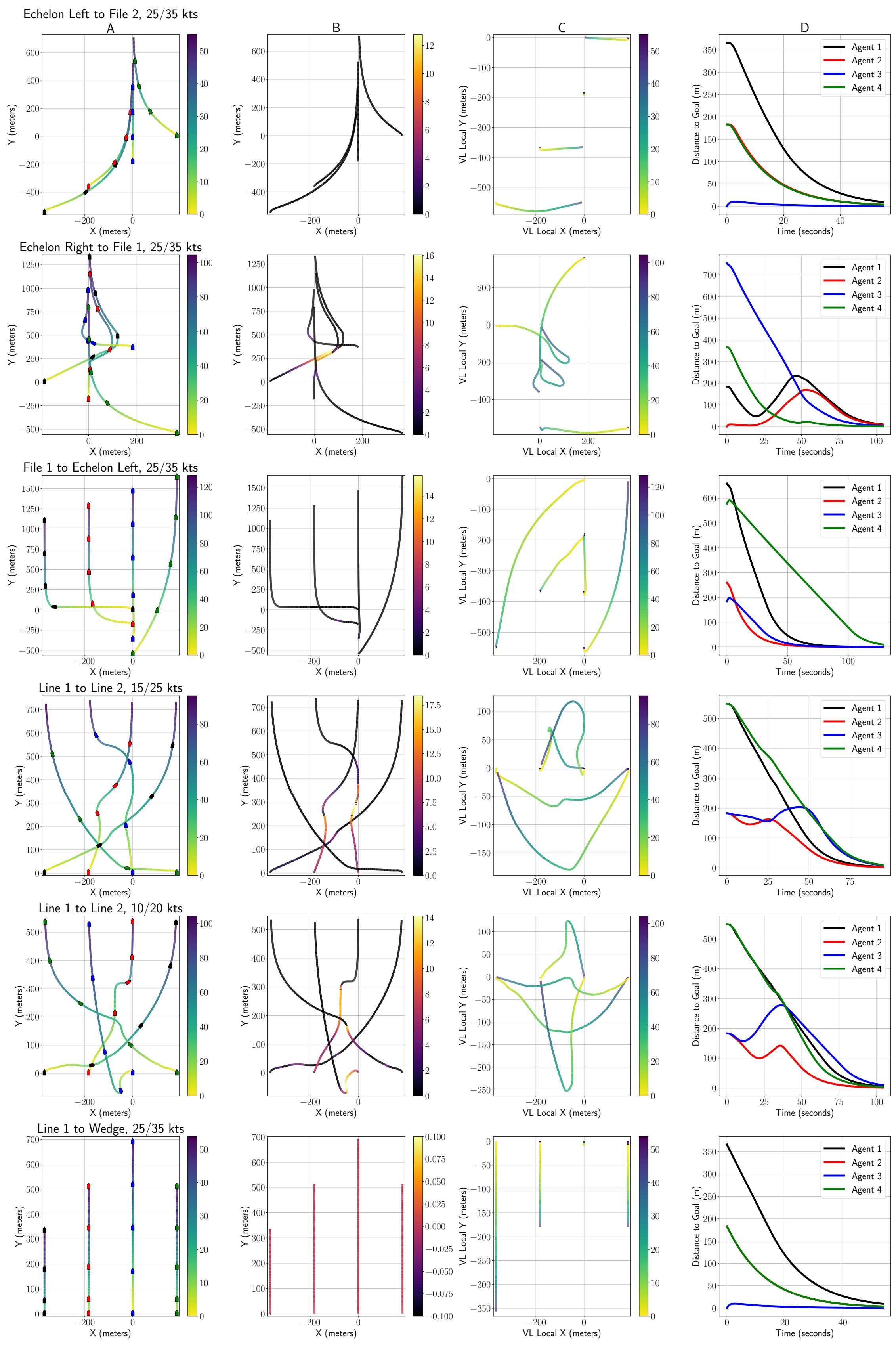
The authors would like to thank the technical staff of STEB Port Hueneme for their extreme professionalism, technical competence, and insights, without which this program would not have been successful. In particular, special thanks is due to the following drivers who willingly gave over control of their vessel (and safety) to a machine: Mike Phelps, Rich Thorp, Jim Montes, Moises Angulo, Carlos Cordero, Varun Varahamurthy, Dennis Metcalf, Robert Wright, Tony Tafarella, Jim Schontzler.

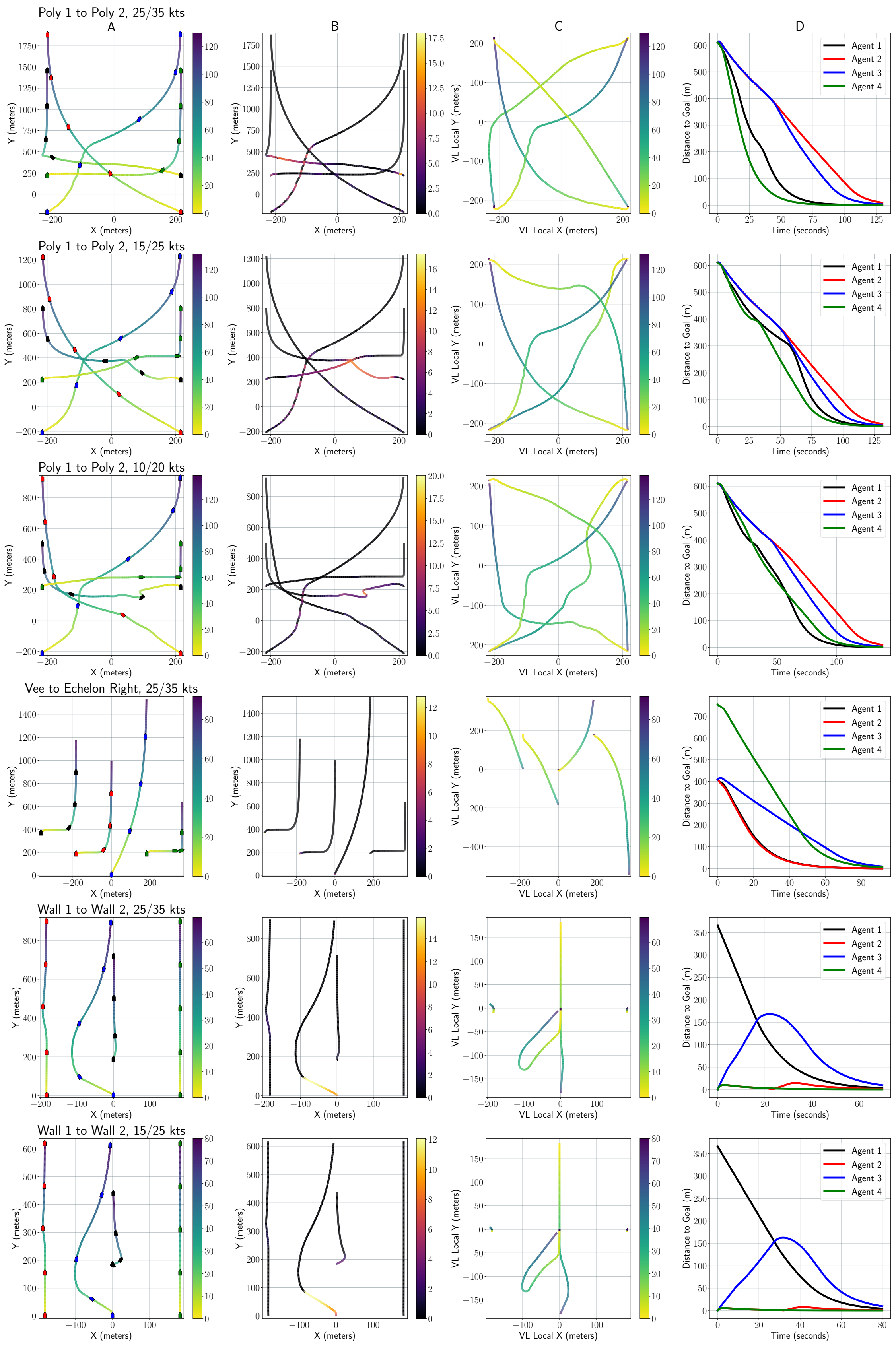
Appendix A. Simulation Results

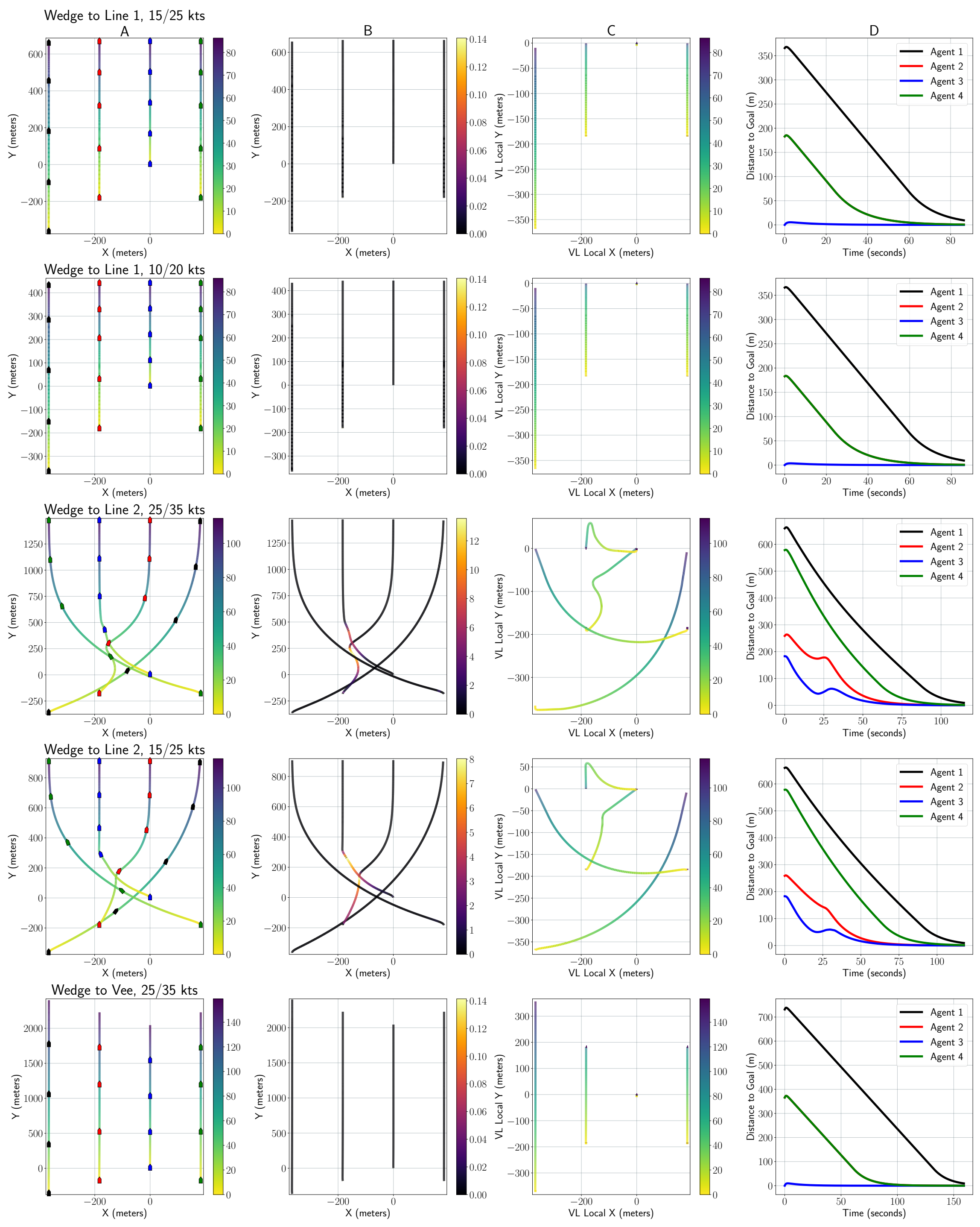
The following pages contain the rest of the simulation results for the formation switches listed in Table 2.


3-way 1 to 3-way 2, 25/35 kts
A
B
C
D
3-way 1 to 3-way 2, 15/25 kts
3-way 1 to 3-way 2, 10/20 kts
Echelon Left* to File*, 25/35 kts
File* to Line*, 25/35 kts
Line* to File*, 25/35 kts
X (meters)
Y (meters)
VL Local X (meters)
VL Local Y (meters)
Time (seconds)
Distance to Goal (m)
Agent 1
Agent 2
Agent 3
Agent 4


Line* to Wedge*, 25/35 kts
Vee* to Echelon Left*, 25/35 kts
Wedge* to Vee*, 25/35 kts
Cup 1 to Cup 2, 25/35 kts
Cup 1 to Cup 2, 15/25 kts
Cup 1 to Cup 2, 10/20 kts
A
B
C
D
X (meters)
Y (meters)
VL Local X (meters)
VL Local Y (meters)
Time (seconds)
Distance to Goal (m)
Agent 1
Agent 2
Agent 3
Agent 4


Echelon Left to File 2, 25/35 kts
Echelon Right to File 1, 25/35 kts
File 1 to Echelon Left, 25/35 kts
Line 1 to Line 2, 15/25 kts
Line 1 to Line 2, 10/20 kts
Line 1 to Wedge, 25/35 kts
A
B
C
D
X (meters)
Y (meters)
VL Local X (meters)
VL Local Y (meters)
Time (seconds)
Distance to Goal (m)
Agent 1
Agent 2
Agent 3
Agent 4


Poly 1 to Poly 2, 25/35 kts
A
B
C
D
Poly 1 to Poly 2, 15/25 kts
Poly 1 to Poly 2, 10/20 kts
Vee to Echelon Right, 25/35 kts
Wall 1 to Wall 2, 25/35 kts
Wall 1 to Wall 2, 15/25 kts
X (meters)
Y (meters)
VL Local X (meters)
VL Local Y (meters)
Time (seconds)
Distance to Goal (m)
Agent 1
Agent 2
Agent 3
Agent 4


Wedge to Line 1, 15/25 kts
Wedge to Line 1, 10/20 kts
Wedge to Line 2, 25/35 kts
Wedge to Line 2, 15/25 kts
Wedge to Vee, 25/35 kts
A
B
C
D
X (meters)
Y (meters)
VL Local X (meters)
VL Local Y (meters)
Time (seconds)
Distance to Goal (m)
Agent 1
Agent 2
Agent 3
Agent 4

Appendix B. Comparison of 2018 Observed to Simulated Times

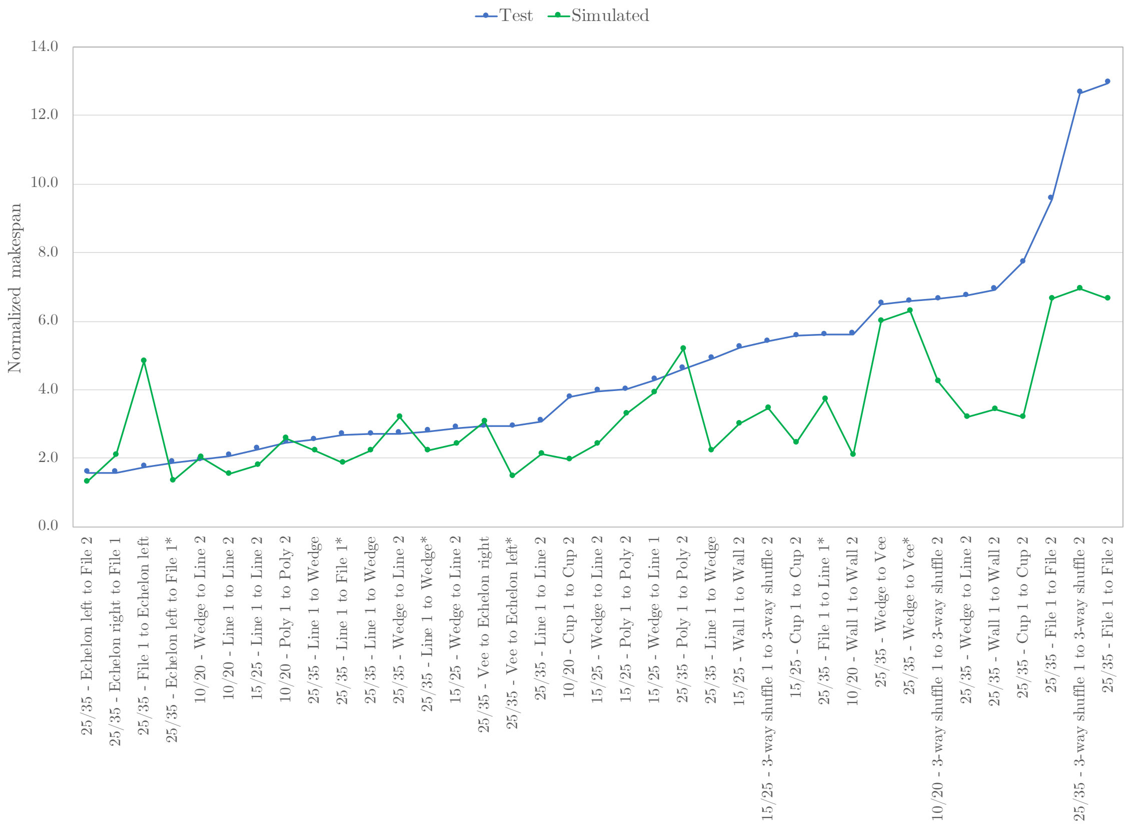


Figure 24. Normalized makespan of 37 formation switches. Normalized makespan of field tests $\hat{M}_{\text{test}}$ (blue, median = 4.0) and normalized makespan of simulations $\hat{M}_{\text{sim}}$ (green, median = 2.6). Ordered by $\hat{M}_{\text{test}}$. Makespans are normalized by lower-bound makespan M_{LB}.

Appendix C. Tables

Table 1. Convergence times (summary).

-	With priorities	ORCA only
Min	63.2	63.2
1st Qu.	117.4	123.4
Med.	138.4	157.0
Mean	138.6	163.4
3rd Qu.	164.1	196.2
Max.	234.0	300.0
Timeouts (300s)	0	3

Table 2. Formation switch makespan: July 2018 test. Purple and blue colors indicate slower and faster formation switches, respectively. Tests are grouped by the starting and ending formation speed ratios (first column) and ordered by increasing normalized test makespan $\hat{M}_{\text{test}}$. *adaptive formation switch. The full runs are completely plotted in Appendix A.

Speeds (knots)	Formation switch From	To	Heading (degrees)	M_{test} (mm:ss)	M_{sim} (mm:ss)	M_{LB} (mm:ss)	$\hat{M}_{\text{test}}$	$\hat{M}_{\text{sim}}$
10/20	Wedge	Line 2	135	01:46	01:50	00:54	2.0	2.0
10/20	Line 1	Line 2	135	02:07	01:34	01:02	2.1	1.5
10/20	Poly 1	Poly 2	180	01:53	01:58	00:46	2.5	2.6
10/20	Cup 1	Cup 2	135	01:35	00:49	00:25	3.8	2.0
10/20	Wall 1	Wall 2	135	02:13	00:50	00:24	5.6	2.1
10/20	3-way shuffle 1	3-way shuffle 2	315	03:17	02:06	00:30	6.6	4.2
15/25	Line 1	Line 2	135	02:01	01:35	00:53	2.3	1.8
15/25	Wedge	Line 2	135	02:08	01:47	00:44	2.9	2.4
15/25	Wedge	Line 2	135	02:56	01:47	00:44	4.0	2.4
15/25	Poly 1	Poly 2	100	02:23	01:57	00:36	4.0	3.3
15/25	Wedge	Line 1	315	01:16	01:10	00:18	4.3	3.9
15/25	Wall 1	Wall 2	135	01:33	00:54	00:18	5.2	3.0
15/25	3-way shuffle 1	3-way shuffle 2	315	02:08	01:22	00:24	5.4	3.4
15/25	Cup 1	Cup 2	270	01:45	00:46	00:19	5.6	2.5
25/35	Echelon left	File 2	310	00:46	00:38	00:29	1.6	1.3
25/35	Echelon right	File 1	90	00:46	01:01	00:29	1.6	2.1
25/35	File 1	Echelon left	315	00:42	01:57	00:24	1.7	4.8
25/35	Echelon left*	File 1*	315	00:39	00:28	00:21	1.9	1.4
25/35	Line 1	Wedge	270	00:30	00:26	00:12	2.5	2.2
25/35	Line 1*	File 1*	270	00:56	00:39	00:21	2.7	1.9
25/35	Line 1	Wedge	115	00:32	00:26	00:12	2.7	2.2
25/35	Wedge	Line 2	135	01:31	01:47	00:33	2.7	3.2
25/35	Line 1*	Wedge*	310	00:33	00:26	00:12	2.8	2.2
25/35	Vee	Echelon right	90	01:13	01:17	00:25	2.9	3.1
25/35	Vee*	Echelon left*	315	01:11	00:36	00:24	2.9	1.5
25/35	Line 1	Line 2	315	02:14	01:32	00:44	3.1	2.1
25/35	Poly 1	Poly 2	100	01:54	02:09	00:25	4.6	5.2
25/35	Line 1	Wedge	135	00:58	00:26	00:12	4.9	2.2
25/35	File 1*	Line 1*	315	01:57	01:18	00:21	5.6	3.7
25/35	Wedge	Vee	270	02:34	02:22	00:24	6.5	6.0
25/35	Wedge*	Vee*	135	02:08	02:03	00:19	6.6	6.3
25/35	Wedge	Line 2	315	03:45	01:47	00:33	6.7	3.2
25/35	Wall 1	Wall 2	135	01:22	00:41	00:12	6.9	3.4
25/35	Cup 1	Cup 2	330	01:37	00:40	00:13	7.7	3.2
25/35	File 1	File 2	135	02:50	01:58	00:18	9.6	6.7
25/35	3-way shuffle 1	3-way shuffle 2	315	03:36	01:59	00:17	12.7	6.9
25/35	File 1	File 2	90	03:50	01:58	00:18	12.9	6.7
							Median	
10/20							3.1	2.1
15/25							4.1	2.7
25/35							3.1	3.2

ORCID

Joshua Vander Hook https://orcid.org/0000-0002-0493-237X
William Seto https://orcid.org/0000-0002-7053-1013
Viet Nguyen https://orcid.org/0000-0002-5722-430X
Zaki Hasnain https://orcid.org/0000-0002-3722-9015
Carlyn-Ann Lee https://orcid.org/0000-0001-6206-071X
Tyler Halpin-Chan https://orcid.org/0000-0001-6313-9761
Varun Varahamurthy https://orcid.org/0000-0002-0432-5043
Moises Angulo https://orcid.org/0000-0002-5145-7321

References

Alonso-Mora, J., Breitenmoser, A., Rufli, M., Beardsley, P., and Siegwart, R. (2013). Optimal reciprocal collision avoidance for multiple nonholonomic robots. In *Distributed Autonomous Robotic Systems*, pages 203–216. Springer.

Alt, H. and Welzl, E. (1988). Visibility graphs and obstacle-avoiding shortest paths. *Zeitschrift für Operations-Research*, 32(3-4):145–164.

Antonelli, G., Arrichiello, F., Casalino, G., Chiaverini, S., Marino, A., Simetti, E., and Torelli, S. (2014). Harbour protection strategies with multiple autonomous marine vehicles. In *International Workshop on Modelling and Simulation for Autonomous Systems*, pages 241–261. Springer.

Arrichiello, F., Chiaverini, S., and Fossen, T. I. (2006). Formation control of underactuated surface vessels using the null-space-based behavioral control. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5942–5947. IEEE.

Boyell, R. L. (1976). Defending a moving target against missile or torpedo attack. *IEEE Transactions on Aerospace and Electronic Systems*, 12(4):522–526.

Breitenmoser, A., Schwager, M., Metzger, J.-C., Siegwart, R., and Rus, D. (2010). Voronoi coverage of non-convex environments with a group of networked robots. In *2010 IEEE International Conference on Robotics and Automation*, pages 4982–4989. IEEE.

Camacho, E. F. and Alba, C. B. (2013). *Model predictive control*. Springer Science & Business Media.

Dai, S.-L., He, S., Lin, H., and Wang, C. (2017). Platoon formation control with prescribed performance guarantees for usvs. *IEEE Transactions on Industrial Electronics*, 65(5):4237–4246.

Duarte, M., Gomes, J., Costa, V., Rodrigues, T., Silva, F., Lobo, V., Marques, M. M., Oliveira, S. M., and Christensen, A. L. (2016). Application of swarm robotics systems to marine environmental monitoring. In *OCEANS 2016-Shanghai*, pages 1–8. IEEE.

Fan, J., Li, Y., Liao, Y., Ma, T., Ge, Y., and Wang, Z. (2020). A formation reconfiguration method for multiple unmanned surface vehicles executing target interception missions. *Applied Ocean Research*, 104:102359.

Ferreau, H., Kirches, C., Potschka, A., Bock, H., and Diehl, M. (2014). qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363.

Fiorini, P. and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research*, 17(7):760–772.

Fu, M. and Wang, D. (2018). The bioinspired model-based hybrid sliding-mode formation control for underactuated unmanned surface vehicles. *Journal of Control Science and Engineering*, 2018.

Ge, S. S. and Cui, Y. J. (2002). Dynamic motion planning for mobile robots using potential field method. *Autonomous robots*, 13(3):207–222.

Halpin-Chan, Mr. Tyler (2019). Autonomous swarms of high speed maneuvering surface vehicles.

Hsu, D., Latombe, J.-C., and Kurniawati, H. (2006). On the probabilistic foundations of probabilistic roadmap planning. *The International Journal of Robotics Research*, 25(7):627–643.

Huntsberger, T., Aghazarian, H., Castano, A., Woodward, G., Padgett, C., Gaines, D., and Buzzell, C. (2008). Intelligent autonomy for unmanned sea surface and underwater vehicles. *AUVSI Unmanned Systems North America*.

Huntsberger, T. and Woodward, G. (2011). Intelligent autonomy for unmanned surface and underwater vehicles. In *OCEANS 2011*, pages 1–10. IEEE.

Ji, J., Khajepour, A., Melek, W. W., and Huang, Y. (2017). Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. *IEEE Transactions on Vehicular Technology*, 66(2):952–964.

Kavraki, L., Svestka, P., and Overmars, M. H. (1994). *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*, volume 1994. Unknown Publisher.

Kunchev, V., Jain, L., Ivancevic, V., and Finn, A. (2006). Path planning and obstacle avoidance for autonomous mobile robots: A review. In *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, pages 537–544. Springer.

Kuwata, Y., Wolf, M. T., Zarzhitsky, D., and Huntsberger, T. L. (2014). Safe maritime autonomous navigation with colregs, using velocity obstacles. *IEEE Journal of Oceanic Engineering*, 39(1):110–119.

Liu, Y. and Bucknall, R. (2015). Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Engineering*, 97:126–144.

Liu, Y. and Bucknall, R. (2016). The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method. *Applied Ocean Research*, 59:327–344.

Liu, Y. and Bucknall, R. (2018a). Efficient multi-task allocation and path planning for unmanned surface vehicle in support of ocean operations. *Neurocomputing*, 275:1550–1566.

Liu, Y. and Bucknall, R. (2018b). A survey of formation control and motion planning of multiple unmanned vehicles. *Robotica*, 36(7):1019–1047.

Mahacek, P., Kitts, C. A., and Mas, I. (2011). Dynamic guarding of marine assets through cluster control of automated surface vessel fleets. *IEEE/ASME Transactions on Mechatronics*, 17(1):65–75.

Marino, A., Antonelli, G., Aguiar, A. P., Pascoal, A., and Chiaverini, S. (2014). A decentralized strategy for multirobot sampling/patrolling: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 23(1):313–322.

Ono, M., Quadrelli, M., and Huntsberger, T. L. (2014). Safe maritime autonomous path planning in a high sea state. In *American Control Conference (ACC), 2014*, pages 4727–4734. IEEE.

Park, J., Kim, D., Yoon, Y., Kim, H., and Yi, K. (2009). Obstacle avoidance of autonomous vehicles based on model predictive control. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 223(12):1499–1516.

Peng, Z., Wang, D., and Wang, J. (2015). Cooperative dynamic positioning of multiple marine offshore vessels: A modular design. *IEEE/ASME Transactions On Mechatronics*, 21(3):1210–1221.

Peng, Z., Wang, J., and Wang, D. (2017). Distributed maneuvering of autonomous surface vehicles based on neurodynamic optimization and fuzzy approximation. *IEEE Transactions on Control Systems Technology*, 26(3):1083–1090.

Phillips, D. T. and Garcia-Diaz, A. (1981). *Fundamentals of network analysis*. Prentice Hall.

Raboin, E., Švec, P., Nau, D., and Gupta, S. K. (2013). Model-predictive target defense by team of unmanned surface vehicles operating in uncertain environments. In *2013 IEEE International Conference on Robotics and Automation*, pages 3517–3522. IEEE.

Reitz, B. C. and Wilkerson, J. L. (2020). Test and evaluation of autonomous surface vehicles: A case study. In *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 839–850. IEEE.

Rossi, F., Bandyopadhyay, S., Wolf, M., and Pavone, M. (2018). Review of multi-agent algorithms for collective behavior: a structural taxonomy. *IFAC-PapersOnLine*, 51(12):112–117.

Rowley, J. (2018). Autonomous unmanned surface vehicles (usv): A paradigm shift for harbor security and underwater bathymetric imaging. In *OCEANS 2018 MTS/IEEE Charleston*, pages 1–6. IEEE.

Snape, J., Van Den Berg, J., Guy, S. J., and Manocha, D. (2011). The hybrid reciprocal velocity obstacle. *IEEE Transactions on Robotics*, 27(4):696–706.

Tange, O. (2011). Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47.

Unknown (1942). *Motor Torpedo Boats, Tactical Orders and Doctrine*. US Government Printing Office, Washington D.C. Available for download at 'https://maritime.org/doc/pt/doctrine/index.htm'.

Valada, A., Velagapudi, P., Kannan, B., Tomaszewski, C., Kantor, G., and Scerri, P. (2014). Development of a low cost multi-robot autonomous marine surface platform. In *Field and service robotics*, pages 643–658. Springer.

Van Den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011a). Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer.

Van Den Berg, J., Guy, S. J., Lin, M., and Manocha, D. (2011b). Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer.

Van den Berg, J., Lin, M., and Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In *2008 IEEE International Conference on Robotics and Automation*, pages 1928–1935. IEEE.

Vander Hook, J., Seto, W., Nguyen, V., Hasnain, Z., Gallagher, L., Halpin-Chan, T., Varahamurthy, V., and Angulo, M. (2019). Autonomous swarms of high speed maneuvering surface vessels for the central test evaluation improvement program. In *Unmanned Systems Technology XXI*, volume 11021, page 110210M. International Society for Optics and Photonics.

Wang, D., Fu, M., Ge, S. S., and Li, D. (2020). Velocity free platoon formation control for unmanned surface vehicles with output constraints and model uncertainties. *Applied Sciences*, 10(3):1118.

Warren, C. W. (1990). Multiple robot path coordination using artificial potential fields. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 500–505. IEEE.

Wilkie, D., Van Den Berg, J., and Manocha, D. (2009). Generalized velocity obstacles. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5573–5578. IEEE.

Wolf, M. T., Rahmani, A., de la Croix, J.-P., Woodward, G., Vander Hook, J., Brown, D., Schaffer, S., Lim, C., Bailey, P., Tepsuporn, S., et al. (2017). Caracas multi-agent maritime autonomy for unmanned surface vehicles in the swarm ii harbor patrol demonstration. In *Unmanned Systems Technology XIX*, volume 10195, page 101950O. International Society for Optics and Photonics.

Woodward, G., Gierach, M., Schaffer, S., Chu, S., Estlin, T., Castano, R., and Li, Z. (2017). Adaptive auv in-situ sensing system. In *OCEANS–Anchorage, 2017*, pages 1–4. IEEE.

Yu, X., Hsieh, M. A., Wei, C., and Tanner, H. G. (2019). Synchronous rendezvous for networks of marine robots in large scale ocean monitoring. *Frontiers in Robotics and AI*, 6:76.

Zereik, E., Bibuli, M., Mišković, N., Ridao, P., and Pascoal, A. (2018). Challenges and future trends in marine robotics. *Annual Reviews in Control*, 46:350–368.

Zhang, F., Marani, G., Smith, R. N., and Choi, H. T. (2015). Future trends in marine robotics [tc spotlight]. *IEEE Robotics & Automation Magazine*, 22(1):14–122.

Zhou, D., Wang, Z., Bandyopadhyay, S., and Schwager, M. (2017). Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells. *IEEE Robotics and Automation Letters*, 2(2):1047–1054.

Zoss, B. M., Mateo, D., Kuan, Y. K., Tokić, G., Chamanbaz, M., Goh, L., Vallegra, F., Bouffanais, R., and Yue, D. K. (2018). Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies. *Autonomous Robots*, 42(8):1669–1689.

How to cite this article: Vander Hook, J. V., Seto, W., Nguyen, V., Hasnain, Z., Lee, C.-A., Gallagher, L., Halpin-Chan, T., Varahamurthy, V., & Angulo, M. (2022). Swarms of Pirates: Red Team Exercises Using Autonomous High-Speed Maneuvering Surface Vessels. *Field Robotics, 2*, 872–909.