High Speed Convoy in Unstructured Indoor Environments

Namya Bagree¹, Charles Noren², Damanpreet Singh¹, Matthew Travers², and Bhaskar Vundurthy²

Abstract—Practical operations of coordinated fleets of mobile robots in different environments reveal benefits of maintaining small distances between robots as they move at higher speeds. This is counter-intuitive in that as speed increases, increased distances would give robots a larger time to respond to sudden motion variations in surrounding robots. However, there is a desire to have lower inter-robot distances in examples like autonomous trucks on highways to optimize energy by vehicle “drafting” or smaller robots in cluttered environments to maintain communication, etc. To achieve this fleet behavior, this work introduces a model based control framework that directly takes non-linear system dynamics into account. Each robot is able to follow closer at high speeds because it makes predictions on the state information from its adjacent robots and biases it’s response by anticipating adjacent robots’ motion. The decentralized nature of our framework leads to the ability of expanding to an arbitrary number of robots without intractable computational burden. In contrast to existing controllers, our non-linear model based predictive decentralized controller is able to achieve lower inter-robot distances at higher speeds. We demonstrate the success of our approach through simulated and hardware results on mobile ground robots.

Index Terms—convoy, non-linear model predictive control, multi-agent, mobile robots, platoon, planning

I. INTRODUCTION

Fleets of mobile robots that are not completely autonomous have shown practical benefits of moving closer at high speeds by optimizing resources and improving efficiencies. An example of this is a group of trucks being able to reduce fuel consumption by reducing aerodynamic drag when they follow at lower inter-vehicle distances [1], [2]. However, reducing inter-vehicle distances reduces the amount of time available to respond to sudden reactive motions of other robots. This can lead to instabilities within the control system. To address these problems, our work investigates non-linear control techniques for autonomous robots that drive closer at high speeds in cluttered environments.

While advantages of a fleet of robots exist in many domains [1], smaller mobile robots are able to exemplify many scenarios of driving fast at close distances in unstructured regions. In search and rescue missions, a practical challenge is maintaining consistent communication, especially in cluttered environments where loss of direct line of sight can hinder inter-robot communications unpredictably [3]. This emphasizes the need to maintain low inter-robot distances in a group of autonomous mobile ground robots operating as a team.

In this work, we present a non-linear model based controller that helps address many issues with the current state of the art methods for fleets of mobile robots operating in cluttered environments. Each agent receives state information from adjacent robots that it uses to predict their future states and in turn, as feedback to optimize its path. This feedforward planning component and feedback control component in our framework help the agents better anticipate and react to unexpected motions in the fleet as compared to purely reactive or proactive solutions.

In addition, we designed the control framework such that it is largely decentralized where each robot computes its own predictive and reactive behaviors based on the information of only those robots that are adjacent to it. This allows us to extend our framework to an arbitrary number of robots with a constant computational requirement on each robot.

The key contributions of our work are:

1) A decentralized non-linear model based predictive control framework to command a fleet of arbitrary number of vehicles with minimal inter-robot distances;
2) An integrated obstacle avoidance trajectory generator with our control framework to operate in unstructured environments;
3) Benchmarking coordinated convoy control from prior works and present a detailed comparison against our control framework in simulation and on hardware.

II. PROBLEM DEFINITION

Consider the schematic illustrated in Fig. 2 where the ‘ego vehicle’ is denoted with the superscript $i$. We assume an increasing index along the trajectory and thus refer to the neighboring vehicles as the lead vehicle and the follow vehicle.
denoted by $i - 1$ and $i + 1$, respectively. Given $L$ agents in the fleet, the state and control trajectories for the agent $i$ over a time horizon of length $N$ are given by $X^i = [x^i_0, x^i_1, ..., x^i_N]$ and $U^i = [u^i_0, u^i_1, ..., u^i_{N-1}]$, respectively, where $i \in \mathcal{I}_L = \{1, 2, \ldots, L\}$.

The objective of the convoy problem as we define it in this work is to design a distributed convoy controller that generates a control output $u^i_0$ at every time instant to transfer a convoy of robots on a predefined trajectory. In particular, the desired objective of the convoy controller for every ego vehicle is to:

1) Maintain a fixed distance $d_{ref}$ from its lead vehicle;
2) Simultaneously operate at desired higher speeds;
3) React to disturbances in lead and follow vehicles, and prioritize the structure of the convoy.

III. LITERATURE REVIEW

To provide context for our specific control approach, we review relevant prior works that address the convoy control problem defined in Section II. Despite several use cases for convoys [1], we focus our discussion on works for unstructured and cluttered environments.

One of the earliest works [4] in convoy control mimic a leader-follower behavior where each vehicle estimates and stores the path of its predecessor as a set of points. The follower then estimates the predecessor’s path curvature around a selected target and follows the trajectory. An extension to this work is shown by authors in [5] where position measurements are stored over time. A spline-approximation technique is then applied to obtain a smooth reference path for the underlying motion controllers. However, these methods force the robot to track the exact positions of its predecessors restricting system flexibility around obstacles.

The authors in [6] provide a framework to switch between exact pose tracking and flexible path search and tracking based on the environment. This allows the trajectory following convoy algorithms to operate in real world conditions. However, there is no interaction between the outputs of the two tracking methods, reducing the controller performance due to conflicting switch decisions when operating in a cluttered environment.

Two implementations have been able to provide a framework with an integrated obstacle avoidance module in unstructured environments. The first implementation [7] defines a robust solution using an adaptive inter-distance control and leader pose ($x^i_{0-1}$) estimation. However, there are no common velocity based control parameters and no prediction on future paths for adjacent agents ($X^i_{-1}, X^i_{+1}$). This leads to higher angular variation as inter-robot distances reduce, resulting in wavy motions and higher error in tracking at high speeds.

The second obstacle avoidance implementation is presented in [8]. The control incorporates a passivity-based MPC method with a traversibility map based planner. However, this framework doesn’t take feedback from following vehicles ($i + 1$), that can lead to high inter-robot distances in cluttered environments. They also rely on continuous communication between the vehicles and a base node to operate.

One other work that doesn’t apply the “Follow the Leader” framework as discussed earlier is shown by the authors in [2]. They include all vehicles in the control framework via a centralized optimization problem to achieve desired straight velocity profiles. The cost terms aim to optimize fuel efficiency while ensuring desired safe following distance. The computational costs increases at least quadratically with increase in agents and must be solved on a single node. This requires high-rate continuous communication of robots with the base node for safe convoying.
All these solutions look at the problem as a “follow-the-leader” problem and do not take following agents \((i + 1, i + 2, \ldots)\) into consideration. While following in cluttered environments, obstacle positions can vary with time. Consider Fig. 3a where agent \(i + 1\) is forced to slow down. The “follow-the-leader” problem reacts to this as seen in Case A where agents \(i\) and \(i - 1\) move at normal speeds. This can lead to communication drops between agents and also a loss of simple following trajectory if the gaps exceed a threshold. Case B is a more desirable scenario where leading agents get feedback from follower agent motions, and are able to maintain a stable convoy.

The high-level controller behaviors listed in Section IV can be expressed in an optimal control framework. The optimal control framework presented in this work expresses the behaviors described in 1) and 2) as terms in the controller objective. Agent reactivity to disturbances described in behavior 3) is encoded in the proposed controller’s cost structure. The proposed controller is detailed in the following section.

A. Convoy Controller

For the \(i\)th robot in the convoy, the discrete-time optimal control problem framework is posed as:

\[
\begin{align*}
\min_{\mathcal{U}_i} & \quad C_{\text{traj}}(X^i, U^i) + C_{\text{convoy}}(X^i, X^{i-1}, X^{i+1}) \\
\text{subject to} & \quad x^i_{k+1} = f(x^i_k, u^i_k, \Delta t), \forall k = 0, \ldots, N - 1, \\
& \quad x^i_0 = x^i(0), u^i_0 = u^i(0), \forall i \in \mathcal{I}_L,
\end{align*}
\]

where:

\[
C_{\text{traj}}(X^i, U^i) = \sum_{k=0}^{N-1} (x^i_k - x^i_{\text{traj},k})^T Q(x^i_k - x^i_{\text{traj},k}) + u^i_k^T R_u u^i_k + (x^i_N - x^i_{\text{traj},N})^T Q_f(x^i_N - x^i_{\text{traj},N}),
\]

\[
C_{\text{convoy}}(X^{i-1}, X^i, X^{i+1}) = \sum_{k=0}^{N-1} (x^i_k - x^{i,\text{ref},i-1,i})^T Q_{\text{lead}}(x^i_k - x^{i,\text{ref},i-1,i}) + (x^i_k - x^{i,\text{ref},i+1,i})^T Q_{\text{follow}}(x^i_k - x^{i,\text{ref},i+1,i}),
\]

and where \(x_k \in \mathbb{R}^m, u_k \in \mathbb{R}^n\). The state-transition model for the \(i\)th robot is defined as \(f(x^i_k, u^i_k, \Delta t)\). Superscript \(\text{ref}\) refers to the reference point defined between the agents with the indices following \(\text{ref}\) (i.e. either \(i - 1, i\) or \(i, i + 1\)). The computation of this point is described in the following section. As the run-time cost is evaluated over the same time interval, the run-time costs in 2) and 3) may be collapsed into a single quadratic cost expression. This new expression is defined as

\[
g(x_k, u_k) = (x^i_k - Q^{-1}_T y_T)^T Q_T (x^i_k - Q^{-1}_T y_T) - y_T^T Q_T y_T + Z_T + u^i_k^T R_u u^i_k,
\]

where:

\[
Q_T = Q + Q_{\text{lead}} + Q_{\text{follow}}
\]

\[
y_T = Q_{\text{traj}}(x_k) + Q_{\text{lead}}(x_k^{\text{ref},i-1,i}) + Q_{\text{follow}}(x_k^{\text{ref},i,i+1})
\]

\[
Z_T = (x_{\text{traj},k})^T Q_{\text{traj}}(x_k) + (x^{\text{ref},i-1,i})^T Q_{\text{lead}}(x_k^{\text{ref},i-1,i}) + (x^{\text{ref},i,i+1})^T Q_{\text{follow}}(x_k^{\text{ref},i,i+1}).
\]

This operation is detailed further in Appendix A. A similar combination of quadratic expressions can be performed on the terminal cost, yielding \((Q_T y_T, Z_T, F_T)\) respectively. These parameters may be used to rephrase the terminal cost:

\[
\phi(X_N) = (x^i_N - Q_T^{-1} y_T)^T Q_T x^i_N - Q_T^{-1} y_T)^T - y_T^T F_T y_T + Z_T.
\]

Thus the cost function may be rephrased as:

\[
J = C_{\text{traj}} + C_{\text{convoy}} = \sum_{k=0}^{N-1} \{g(x_k, u_k)\} + \phi(X_N).
\]

Furthermore, by linearizing the system around \(x_k, u_k\) and defining \(A_k = \frac{\partial}{\partial x_k} f(x_k, u_k)\) and \(B_k = \frac{\partial}{\partial u_k} f(x_k, u_k)\), the optimization problem can be interpreted and solved online as an iterative Linear Quadratic Regulator (LQR). This yields a control law:

\[
u_k = -[R_k + B_k^T P_{k+1} B_k]^{-1} B_k^T P_{k+1} A_k (x^i_k - Q_T^{-1} y_T) \]

with \(P_k\) representing the solution to the Riccati Equation and \(K_k\) being the optimal control gain matrix.

B. Controller Implementation and Design Discussion

The first component of the cost function is a quadratic trajectory tracking cost penalizing deviations from a given convoy trajectory: \(x_{\text{traj}}\). This cost is defined in 2). In this work, the reference path \(x_{\text{traj}}\) is provided to the controller as either a pre-defined path or created for each individual robot from observing the motion of other agents in the convoy (e.g. “follow-the-leader” style approaches [4, 5]).

The convoy cost \(C_{\text{convoy}}\) penalizes deviations from the convoy structure over the future horizon. This cost is defined in Equation 3 where \(x^{\text{ref},i,i-1,i}\) and \(x^{\text{ref},i,i+1,i}\) are the reference positions of the \(i\)th robot given the positions of the \(i - 1\) and \(i + 1\) cars, respectively, and \(Q_{\text{lead}}, Q_{\text{follow}}\) are tunable positive semi-definite constant matrices. The state includes the \(x\) and \(y\) coordinates, vehicle orientation \(\psi\) and velocity \(v\). The control includes acceleration \(a\) and steering angle \(\delta\).

The computation of \(x^{\text{ref}}\) is based on the desired inter-robot distance, \(d^{\text{ref}}\). This desired inter-robot distance is defined as

\[
d^{\text{ref}} = \lambda_1 v_t + \lambda_2 (v_t - v_{t-1}) + K,
\]

where \(v_t\) is the desired target velocity and \(v_t\) corresponds to current velocity for agent \(i\). \(\lambda_1\) and \(\lambda_2\) are tunable parameters where \(\lambda_1\) and \(\lambda_2\) are non-negative values. \(K\) is a constant minimum inter-robot distance for safe operation. For this work, only the current velocities of the ego vehicle and the prior robot in the convoy structure are considered.

As shown in Fig. 2, the reference positions for agents \(i - 1\) and \(i + 1\) over horizon \(1 : N\) are recovered by performing
a open-loop forward rollout using the linearized dynamics at the \(i - 1\) or \(i + 1\) agent’s state. The \(i - 1\) and \(i + 1\) agent’s current velocity and steering are assumed to be constant over the rollout. The reference positions for \(X_{ref}^{i}\) are set by moving backwards \(d_{ref}^{i}\) along the convoy trajectory from the predicted \(i - 1\) agent positions \(X_{i-1}^{t-1}\), and moving the same distance ahead of the \(i + 1\) agent positions \(X_{i+1}^{t+1}\).

The reactivity described in Behavior 3) of II aims to prevent collisions between agents due to sudden variations in speed. To enable this behavior, the controller computes a weighting factor, \(w_{convoy}\), between the costs (2) and (3) during run-time. This factor is based on the desired convoy spacing and the current Euclidean distance \(dist_{i,j}\) between agents \(i\) and \(j\). These weights are multiplied to the \(Q\) matrices in (3) and the weighting factor is computed as:

\[
Q_{lead} = w_{i,i-1} * Q_{lead} \\
Q_{follow} = w_{i,i+1} * Q_{follow}
\]

\[
w_{i,j} = \begin{cases} 
1 + \frac{w_{far} \times (dist_{i,j} - d_{ref}^{i})}{dist_{i,j}} & \text{if } dist_{i,j} \geq d_{ref}^{i} \\
1 + \frac{w_{near} \times (d_{ref}^{i} - dist_{i,j})}{dist_{i,j}} & \text{otherwise.}
\end{cases}
\]

The proposed formulation allows the robot to track its predecessor and follower through the predicted \(d_{ref}^{i}\) terms, while also tracking its desired planned path. The proposed additional convoy cost provided in the optimal control formulation may be interpreted as a modification of the local linearization point used in the LQR. This modification of the coordinate transfer from the reference path \((x_{ref}^{i})\) through both the user-defined weightings \((Q, Q_{lead}, Q_{follow})\) and reference trajectories of the leading \((X_{lead})\) and following \((X_{follow})\) robots in the fleet.

**Algorithm 1** Convoy controller with obstacle avoidance for robot \(i\)

**Input:** Robot states \(x_{0}^{i-1}, x_{0}^{i}, x_{0}^{i+1}\)

**Output:** Control sequence \(u_{i}\)

while Robots are in convoy, \(i \in I_{L}\) do

Run convoy controller, Section IV-A

if Output path \(X_{i}\) is obstacle-free then

return Control sequence \(u_{i}\)

else

Define \(D_{LookAhead}\) based on velocity \(v_{t}\)
Calculate desired velocity \(v_{T}\) and direction \(\theta_{T}\)
Run the local planner \[10\]
Get the modified obstacle-free path \(X_{i}^{*}\)
Send \(X_{i}^{*}\) into a path following iLQR controller
return Control sequence \(u_{i}\)

end if

end while

C. Local Planner and Trajectory Controller

When we switch to this mode, based on Algorithm \[1\], we lose some properties of the optimal convoy following system, to counter this and ensure high speed following around obstacles while maintaining the convoy formation, an additional velocity scaling term is included:

\[
v_{T} = (1 + \alpha)v_{te}
\]

\[
\alpha = \lambda_{3}(d_{1} - d_{ref}) - \lambda_{4}d_{2}
\]

Where \(\lambda\)'s are tunable parameters, \(v_{te}\) is the desired target velocity from the convoy controller, \(v_{T}\) is the modified desired target velocity for the planner. \(v_{t}\) is the robot velocity, \(v_{l}\) is the leader’s velocity, \(d_{1}\) is the distance between the robot and the leader along the trajectory and \(d_{2}\) is the distance between the robot and the follower along the trajectory.

The desired direction is selected based on the output of the convoy controller. This is done to track the desired controller path to the greatest extent before the robot switches back into the convoy controller mode. The direction \(\theta_{T}\) is defined via the look ahead distance \(D_{LookAhead}\) along the optimal trajectory generated by the convoy controller.

The local planner takes these inputs and generates a feasible path that avoids obstacles while tracking outputs from the convoy controller. This work is based off the planner defined in \[10\]. The selected path is then sent to an iLQR trajectory following controller to generate the control sequence.

V. Simulation Results

The performance of the presented controller (“Convoy Controller”) is characterized in a variety of simulated environments shown in Fig. \[4\]. The underlying simulator is Gazebo \[11\]. As discussed in Section III, there are many different approaches to enforcing convoy structure at the control and planning level. In order to develop a comparison between the presented methodology and existing literature, a baseline “Base Controller” is developed by combining the local planning and distance variation behaviors from \[7\] with addition modifications from \[6\] \[12\] \[13\]. This combination creates a decentralized controller, similar to our proposed method, and outperforms the individual performance of each work separately with respect to minimizing inter-agent distances without breaking convoy formation.

An initial comparison is done on straight line and low curvature turn path. Next, an \(\infty\) loop with a 20m radius is simulated at speeds varying between 4 and 8 m/s in order to test controller performance in continuous turns. Finally, runs are performed on tight turns, a tunnel environment and a racetrack. Additional constraints on agent operability and speed are added on these runs to understand fleet performance in edge scenarios. Two error metrics are used to compare the results between both controller performances. The first metric is as follows:

\[
e_{m_{1}} = \begin{cases} 
abs((d_{i-1,i+1}/2) - d_{i-1,i}) & i \in (1, L) \\
abs((d_{L-2,L}/2) - d_{L-1,L}) & i = L
\end{cases}
\]

This metric aims to understand how well a robot is able to maintain a position midway between its adjacent robots. Variation in this metric can help understand accordion-like
behaviors that negatively affects fleet performance with continuous acceleration and braking requirements. However, this error metric can maintain a low value with all robots maintaining an equally large following distance, which is not desirable. To that end, we define a second error metric as follows:

\[ e_{m_2} = \text{abs}(d_{i-1,i} - d_{\text{desired}}) \]

This metric measures distance from the desired gap between robots and along with the first metric, can provide a thorough understanding of the convoy performance.

A. Straight line and low curvature runs

The first set of simulations were performed on a group of robots operating on simple paths while starting from scattered initial locations. The aim of this experiment is to understand how quickly various controllers settle and how the error values vary during that period. As can be seen in Fig. 5, our convoy controller is able to settle faster while maintaining lower values on both error metrics. This displays faster convoy structure formation through our framework as each robot has a cost associated with all of its adjacent agents.

B. High speed \( \infty \) loop

We run robots in an \( \infty \) loop at various speeds to understand system capability limits. Fig. 6 provides error metric graphs run at various target speeds between 4 - 8 m/s. It can be seen through the figures that at initial speeds, error metrics for both controllers settle, while our control achieves this faster. However, as speeds increase, the difference in performance between our convoy controller and the base controller increases, until a point when the base controller breaks and isn’t able to maintain a convoy structure.

C. Tight turns

A series of waypoints are sent to the lead robot, the other robots have no information about these points. On running the base controller, the following agents tend to overshoot due to the presence of sharp turns. This behavior is seen even if we tune down the look-ahead value \([12]\) to zero, depicting an inherent unwillingness to follow through such sharp turns at high speeds. The error metric comparison between controllers can be seen in Fig. 7a. Our controller performs better where usual convoying techniques overshoot and have difficulty in tracking such tight turns.

D. Tunnel environments

During operation, especially in indoor environments at high speeds, an agent might get stuck. To simulate this, we initially stop the first robot in place for 8 seconds and then revert to normal conditions. We simulate this through narrow tunnels. In Fig. 7b the error metrics are capped for our approach whereas they continuously rise with the base controller as the last robot isn’t able to get back into the convoy. Due to the cost term corresponding to the immediate lead and follower in the convoy controller cost definition, as soon as the one of the agents isn’t maintaining desired motion, the inter-robot gaps increase and the corresponding cost term shoots up. This causes the remaining agents to slow down and come to a stop until the slowed down follower agent rejoins, capping the error metric.
E. Race track

We simulate a constrained outdoor environment at 4m/s. We also add an additional constraint that one of the robots isn’t able to achieve speeds higher than 3 m/s, which would mimic a robot with operational issues or a heterogeneous set of robots. We run this setup on a race track, shown in Fig. 4f. The error metric are seen in Fig. 7c. The initial performance is similar, until a point where the leader makes a turn after the straight. On the straight, the base controller notices a continuous drop in performance that it is unable to recover past a turn, and the slow robot can no longer track the lead along the track. Our system on the other hand adapts to this robot and despite losing a little performance, is able to maintain an upper limit on the error metric and continue on the desired path.

Overall, our controller can follow at closer distances at high speeds than seen in prior research, bring down error metrics at faster rates and can adapt to variations in environments and robot operational conditions, through a single multi-objective optimal convoy control cost.

VI. EXPERIMENTS

A. System Overview

Each robot in our setup consists of Traxxas remote controlled trucks fitted with a communication node, LiDAR and IMU sensors, a Jetson AGX Xavier and a motor controller. For our perception framework, we make use of Super Odometry [14], which is an IMU-centric pipeline that provides estimates of each agent’s odometry. To ensure neighbouring robots don’t create drifts in odometry in low LiDAR feature regions such as narrow corridors, the LiDAR data was filtered in directions of the forward and rear robots. This ensures robust perception performance in all environments.

A base station is used to provide input commands to the convoy leader, either via a joystick or through a waypoint sharing interface. All systems are run over ROS and use the DDS protocol for real-time communication.

B. Results

1) Straight Line: As discussed in Section V-A, we test similar conditions on the three ground robots. It can be seen that the error metrics in Fig. 10a and 10b that the metrics vary in a similar manner to those in Fig. 5a and 5b where


| Environment    | Average Error Metric 1 (m) | Average Error Metric 2 (m) |
|----------------|----------------------------|----------------------------|
|                | Base Controller | Our Controller | Base Controller | Our Controller |
| Straight Line  | 1.35           | 0.31           | 1.08           | 0.44           |
| Sine Curve     | 1.12           | 0.30           | 0.80           | 0.47           |
| ∞ loop         | 1.69           | 0.32           | 2.82           | 0.73           |
| Tight Turn     | 0.72           | 0.18           | 0.65           | 0.37           |
| Tunnel         | 10.20          | 1.08           | 10.94          | 2.34           |
| Race Track     | 3.77           | 1.45           | 2.92           | 1.02           |
| Column Room    | 1.91           | 0.97           | 5.61           | 0.95           |
| Long Corridor  | 3.78           | 0.89           | 5.44           | 0.68           |

TABLE I: Overall results comparison

VII. CONCLUSIONS

We have defined an optimal decentralized control system to run on a multi-agent convoy. Our design incorporates future predictions on the predecessor and successor agents and solves a cost minimization incorporating robot states and our convoy controller is able to achieve steady state values faster than the existing base controller. This also re-validates simulation results on a real system.

2) Column room: Such an environment (Fig. 8b) illustrates convoy agility due to continuous obstacle avoidance requirements in low-curvature maneuvers while maintaining a during high-speed convoy maneuvers. The room has dimensions of 35m x 20m. The operator commands a general direction of motion to the lead vehicle and all three robots make use of the convoy controller to follow closely while achieving speeds of 4 m/s. The robots make continuous turns to not hit walls, columns and other obstacles. The robots are run in various directions around this room to cover a total distance of over 500m. The error metrics between our and the base controller are shown in Fig. 10c and 10d. Despite having to continuously accelerate, decelerate and turn in such an environment, the convoy is able to maintain an average gap of just over 5m while operating at 4 m/s compared to 9m following distances in the base controller. We also run the same path with a single robot and notice an increase in average speed by under 5%, displaying the fact that our framework doesn’t reduce overall robot capabilities while increasing agents.

3) Long corridors: In narrow long corridors with doorways (Fig. 8c), we illustrate high-speed convoy performance while adapting to turns and doorways. These environments contain debris as shown in Fig. 9 testing robustness in our control system and mimicking search and rescue operations. The performance difference can be seen in in Fig. 10c and 10d. The average distances are just under 4m due to the increased average speeds, which is significantly lower than lowest gaps of 9-15m seen at comparable speeds in other research papers and our base controller. The deviations in the metrics only take place when the robots make a turn into another corridor, move over rough grounds, or navigate through doorways. The performance drop of a convoy against a single robot is less than 2%, indicating almost similar times to achieve the goal. The robots were able to navigate over 3 km in such environments without running into each other or leaving a robot behind, displaying a robust and safe system.
controls, allowing the agents to operate closer to each other at high speeds. This in turn allows robots to operate well in environments that require continuous sharp turns and variations in speeds. The framework operates in a decentralized manner, resulting in no additional computational constraints while increasing the number of robots being deployed. We have been able to show, through simulation and hardware experimentation, the improvement in performance against current state of the art methods and have been able to cut down following distances against state of the art by half. The system can potentially be improved in the future by incorporating environment dependent control optimizations and integrating hybrid communication control methods with increased data transfer to reduce state prediction computational loads.

REFERENCES

[1] S. Nahavandi, S. Mohamed, I. Hossain, D. Nahavandi, S. M. Salaken, M. Rokonuzzaman, R. Ayoub, and R. Smith, “Autonomous convoying: A survey on current research and development,” IEEE Access, vol. 10, pp. 13 663–13 683, 2022.

[2] V. Turri, B. Besselink, and K. H. Johansson, “Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning,” IEEE Transactions on Control Systems Technology, vol. 25, no. 1, pp. 12–28, 2017.

[3] S. Scherer et al., “Resilient and modular subterranean exploration with a team of roving and flying robots,” Field Robotics Journal, pp. 678–734, May 2022.

[4] J. Yazbeck, A. Scheuer, and F. Charpillet, “Decentralized near-to-near approach for vehicle platooning based on memorization and heuristic search,” in 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 631–638.

[5] G. Nestlinger, J. Rumetshofer, and S. Solmaz, “Leader-based trajectory following in unstructured environments: from concept to real-world implementation,” Electronics, vol. 11, no. 12, 2022. [Online]. Available: https://www.mdpi.com/2079-9292/11/12/1866

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

[7] X. Zhao, W. Yao, N. Li, and Y. Wang, “Design of leader’s path following system for multi-vehicle autonomous convoy,” in 2017 IEEE International Conference on Unmanned Systems (ICUS), 2017, pp. 132–138.

[8] J. Shin, D. Kwak, and J. Kim, “Autonomous platooning of multiple ground vehicles in rough terrain,” Journal of Field Robotics, vol. 38, 09 2020.

[9] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological systems,” in International Conference on Informatics in Control, Automation and Robotics, 2004, pp. 222–229.

[10] J. Zhang, C. Hu, R. Chadha, and S. Singh, “Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation,” Journal of Field Robotics, vol. 37, 04 2020.

[11] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), vol. 3, 2004, pp. 2149–2154 vol.3.

[12] A. Bayuwinda, J. Ploeg, E. Lefeber, and H. Nijmeijer, “Combined longitudinal and lateral control of car-like vehicle platooning with extended look-ahead,” IEEE Transactions on Control Systems Technology, vol. 28, no. 3, pp. 790–803, 2020.

[13] M.-M. Mohamed-Ahmed, A. Naamane, and N. K. M’Sirdi, “Path tracking for the convoy of autonomous vehicles based on a non-linear predictive control,” in The 12TH International Conference on Integrated Modeling and Analysis in Applied Control and Automation, Lisborne, Portugal, Sep. 2019. [Online]. Available: https://hal-amu.archives-ouvertes.fr/hal-02479106

[14] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, “Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments,” in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021, pp. 8729–8736.