Safe Sequential Path Planning Under Disturbances and Imperfect Information

Somil Bansal*, Mo Chen*, Jaime F. Fisac, and Claire J. Tomlin

Abstract—Multi-UAV systems are safety-critical, and guarantees must be made to ensure no unsafe configurations occur. Hamilton-Jacobi (HJ) reachability is ideal for analyzing such safety-critical systems; however, its direct application is limited to small-scale systems of no more than two vehicles due to an exponentially-scaling computational complexity. Previously, the sequential path planning (SPP) method, which assigns strict priorities to vehicles, was proposed; SPP allows multi-vehicle path planning to be done with a linearly-scaling computational complexity. However, the previous formulation assumed that there are no disturbances, and that every vehicle has perfect knowledge of higher-priority vehicles’ positions. In this paper, we make SPP more practical by providing three different methods to account for disturbances in dynamics and imperfect knowledge of higher-priority vehicles’ states. Each method has different assumptions about information sharing. We demonstrate our proposed methods in simulations.

I. INTRODUCTION

Recently, there has been an immense surge of interest in using unmanned aerial systems (UASs) for civil purposes [1]–[4]. Many of these applications will involve unmanned aerial vehicles (UAVs) flying in urban environments. As a result, government agencies such as the Federal Aviation Administration (FAA) and National Aeronautics and Space Administration (NASA) of the United States are trying to develop new scalable ways to organize an air space in which potentially thousands of UAVs can fly together [5], [6].

One essential problem that needs to be addressed is how a group of vehicles in the same vicinity can reach their destinations while avoiding collision with each other. In some previous studies that address this problem, specific control strategies for the vehicles are assumed, and approaches such as induced velocity obstacles have been used [7]–[9]. Other researchers have used ideas involving virtual potential fields to maintain collision avoidance while maintaining a specific formation [10], [11]. Although interesting results emerge from these studies, simultaneous trajectory planning and collision avoidance were not considered.

Trajectory planning and collision avoidance problems in safety-critical systems have been studied using Hamilton-Jacobi (HJ) reachability analysis, which provides guarantees on the success and safety of optimal system trajectories [12]–[16]. In this context, one computes the reachable set, defined as the set of states from which the system can be driven to a target set. HJ reachability has been successfully used in applications involving systems with no more than two vehicles [13], [17]–[19]. However, HJ reachability cannot be directly applied to systems involving multiple vehicles due to its exponentially scaling computational complexity.

To overcome this problem, [20] presents sequential path planning (SPP), in which vehicles are assigned a strict priority ordering. In SPP, higher-priority vehicles ignore the lower-priority vehicles, which must take into account the presence of higher-priority vehicles by treating them as induced time-varying obstacles. Under this structure, computation complexity scales just linearly with the number of vehicles. In addition, a structure like this has the potential to flexibly divide up the airspace for the use of many UAVs; this is an important task in NASA’s concept of operations for UAS traffic management [6].

The formulation in [20], however, ignores disturbances and assumes perfect information about other vehicles’ trajectories. In presence of disturbances, a vehicle’s state trajectory evolution cannot be precisely known a priori; thus, it is impossible to commit to exact trajectories as required in [20]. In such a scenario, a lower-priority vehicle needs to account for all possible states that the higher-priority vehicles could be in. To do this, the lower-priority vehicle needs to have some knowledge about the control policy used by each higher-priority vehicle. The main contribution of this paper is to take advantage of the computation benefits of the SPP scheme while resolving some of its practical challenges. In particular, we achieve the following:

• incorporate disturbances into the vehicle models,
• analyze three different assumptions on information to which lower-priority vehicles may have access,
• for each information pattern, propose a reachability-based method to compute the induced obstacles and the reachable sets that guarantee collision avoidance as well as successful transit to the destination.

II. PROBLEM FORMULATION

Consider \( N \) vehicles, denoted \( Q_i, i = 1, \ldots, n \), whose dynamics are described by the ordinary differential equation

\[
\begin{align*}
\dot{x}_i &= f_i(t, x_i, u_i, d_i), & t \leq t_{\text{STA}}^i \\
u_i &\in U_i, d_i &\in D_i, & i = 1, \ldots, N
\end{align*}
\]

(1)

where \( x_i \in \mathbb{R}^{n_i} \), \( u_i \) denote the state and control of \( i \)th vehicle \( Q_i \), respectively, and \( d_i \) denotes the disturbance experienced by \( Q_i \). In general, the physical meaning of \( x_i \) and the dynamics \( f_i \) depend on the specific dynamic model of \( Q_i \), and need not be the same across the different vehicles. \( t_{\text{STA}}^i \) in (1) denotes the scheduled time of arrival of \( Q_i \).

For convenience, we will use the sets \( \mathbb{U}_i, \mathbb{D}_i \) to denote the set of functions from which the control and disturbance

This work has been supported in part by NSF under CPS:ActionWebs (CNS-931843), by ONR under the HUNT (N00014-08-0696) and SMARTS (N00014-09-1-1051) MURI ands and by grant N00014-12-1-0609, by AFOSR under the CHASE MURI (FA9550-10-1-0567). The research of J. F. Fisac has received funding from the “la Caixa” Foundation.

* Both authors contributed equally to this work. All authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley. {somil, mochen72, jfisac, tomlin}@eecs.berkeley.edu
functions $u_i(\cdot), d_i(\cdot)$ can be drawn. Let $p_i \in \mathbb{R}^p$ denote the position of $Q_i$. Denote the rest of the states $h_i$, so that $x_i = (p_i, h_i)$. The initial state of $Q_i$ is given by $x_{i0}$. Under the worst case disturbance, each vehicle aims to get to some set of target states, denoted $T_i \subset \mathbb{R}^n$, by some scheduled time of arrival $t_{i}^{STA}$. On its way to $T_i$, each vehicle must avoid the danger zones $A_{ij}(t)$ of all other vehicles $j \neq i$ for all time. In general, the danger zone can be defined to capture any undesirable configurations between $Q_i$ and $Q_j$. In this paper, we define $A_{ij}(t)$ as

$$A_{ij}(t) = \{ x_i \in \mathbb{R}^n : \| p_i - p_j(t) \|_2 \leq R_c \}$$

(2)

the interpretation of which is that a vehicle is in another vehicle’s danger zone if the two vehicles are within a Euclidean distance of $R_c$ apart.

The problem of driving each of the vehicles in (1) into their respective target sets $T_i$ would be in general a differential game of dimension $\sum_i n_i$. However, due to the exponential scaling of the complexity with the problem dimension, an optimal solution is computationally intractable even for $N > 2$, with $n_i$ as small as 3.

In this paper, we assume that vehicles have assigned priorities as in the SPP method [20]. Since the analysis in [20] did not take into account the presence of disturbances $d_i$ and limited information available to each vehicle, we extend the work in [20] to answer the following:

1) How can each vehicle guarantee that it will reach its target set without getting into any danger zones, despite the disturbances it and other vehicles experience?

2) How should each vehicle robustly handle situations with limited information about the state, control policy, and intention of other vehicles?

### III. Background

This section provides a brief summary of [20], in which the SPP scheme is proposed under perfect information and absence of disturbances. Here, the dynamics of $Q_i$ becomes

$$\dot{x}_i = f_i(t, x_i, u_i), \quad t \leq t_{i}^{STA}$$

$$u_i \in U_i, \quad i = 1, \ldots, N$$

(3)

where the difference compared to (1) is that the disturbance $d_i$ is no longer a part of the dynamics.

In order to make the $N$-vehicle path planning problem safe and tractable, a reasonable structure is imposed to the problem: the vehicles are assigned a strict priority ordering. When planning its trajectory to its target, a higher-priority vehicle can disregard the presence of a lower-priority vehicle. In contrast, a lower-priority vehicle must take into account the presence of all higher-priority vehicles, and plan its trajectory in a way that avoids the higher-priority vehicles’ danger zones. For convenience and without lost of generality, let $Q_i$ be the vehicle with the $i$th highest priority.

Under the above convention, each vehicle $Q_i$ must take into account time-varying obstacles induced by vehicles $Q_j, j < i$, denoted $O_i^j(t)$ and represent the set of states that could possibly be in the danger zone of $Q_j$. Optimal safe path planning of each lower-priority vehicle $Q_i$ then consists of determining the optimal path that allows $Q_i$ to reach its target $T_i$ while avoiding the time-varying obstacles $G_i(t)$, defined by

$$G_i(t) = \bigcup_{j=1}^{i-1} O_i^j(t)$$

(4)

Such an optimal path planning problem can be solved by computing a backward reachable set (BRS) $V_i(t)$ from a target set $T_i$ using formulations of HJ VI such as [12], [14], [16], [21]. For example, to compute BRSs under the presence of time-varying obstacles, the authors in [21] augment system with the time variable, and then applied reachability theory for time-invariant systems. To avoid increasing the problem dimension and save computation time, for the simulations of this paper we utilize the formulation in [16], which does not require augmentation of the state space with the time variable.

Starting from the highest-priority vehicle $Q_1$, one computes the BRS $V_1(t)$, from which the optimal control and trajectory $x_1(\cdot)$ to $T_1$ can be obtained. Under the absence of disturbances and perfect information, obstacles induced by a higher-priority vehicle $Q_j$, starting with $j = 1$, for a lower-priority vehicle $Q_i$ is simply the danger zone centered around the position $p_j(\cdot)$ of each point on the trajectory:

$$O_i^j(t) = \{ x_i : \| p_i - p_j(t) \| \leq R_c \}$$

(5)

Given $O_i^j(t), j < i$, and continuing with $i = 2$, the optimal safe trajectories for each vehicle $Q_i$ can be computed. All of the trajectories are optimal in the sense that given the requirement that $Q_i$ must arrive at $T_i$ by time $t_{i}^{STA}$, the latest departure time $t_i^{DT}$ and the optimal control $u_i^*(\cdot)$ that guarantees arrival by $t_{i}^{STA}$ can be obtained.

To compute $V_i(t)$ using the method in [16], we solve the following HJ VI for $t \leq t_i^{STA}$:

$$H_i(t, x_i, \lambda) = \min_{u_i \in U_i} \lambda \cdot f_i(t, x_i, u_i)$$

(7)

where $\lambda$ is the gradient of the value function, $D_{x_i}V_i$, and $l_i(x_i), g_i(t, x_i)$, $V_i(t, x_i)$ are implicit surface functions representing the target $T_i$, the time-varying obstacles $G_i(t)$, and the backward reachable set $V_i(t)$, respectively:

$$x_i \in T_i \iff l_i(x_i) \leq 0$$

$$x_i(t) \in G_i(t) \iff g_i(t, x_i) \leq 0$$

$$x_i(t) \in V_i(t) \iff V_i(t, x_i) \leq 0$$

(8)

The optimal control is given by

$$u_i^*(t, x_i) = \arg \min_{u_i \in U_i} \lambda \cdot f_i(t, x_i, u_i)$$

(9)
IV. DISTURBANCES AND INCOMPLETE INFORMATION

Disturbances and incomplete information significantly complicate the SPP scheme. The main difference is that the vehicle dynamics satisfy (1) as opposed to (3). Committing to exact trajectories is therefore no longer possible, since the disturbance \(d_i(\cdot)\) is \textit{a priori} unknown. Thus, the induced obstacles \(O^i_1(t)\) are no longer just the danger zones centered around positions. We present three methods to address the above issues. The methods differ in terms of control policy information that is known to a lower-priority vehicle about a higher-priority vehicle, and have their relative advantages and disadvantages depending on the situation. The three methods are as follows:

- **Centralized control**: A specific control strategy is enforced upon a vehicle; this can be achieved, for example, by some central agent such as an air traffic controller.
- **Least restrictive control**: A vehicle is required to arrive at its targets on time, but has no other restrictions.
- **Robust trajectory tracking**: A vehicle declares a nominal trajectory which can be robustly tracked.

In general, the above methods can be used in combination in a single path planning problem, with each vehicle independently having different control policies. Lower-priority vehicles would then plan their paths while taking into account the control policy information known for each higher-priority vehicle. For clarity, we will present each method as if all vehicles are using the same method of path planning.

For simplicity of explanation, we assume that no static obstacles exist. If static obstacles do exist, the time-varying obstacles \(G_i(t)\) simply become the union of the induced obstacles \(O^i_1(t)\) in (4) and the static obstacles.

A. Method 1: Centralized Control

The highest-priority vehicle \(Q_1\) first plans its path by computing the BRS (with \(i = 1\))

\[
V_i(t) = \{x_i : \exists u_i(\cdot) \in U_i, \forall d_i(\cdot) \in D_i, x_i(\cdot) \text{ satisfies (1), } \forall s \in [t, t^*_{STA}], x_i(s) \notin G_i(s), \exists s \in [t, t^*_{STA}], x_i(s) \in T_i\} \tag{10}
\]

Since we have assumed no static obstacles exist, we have that for \(Q_1, G_i(s) = \emptyset \forall s \leq t^*_{STA}\), and thus the above BRS is well-defined. This BRS can be computed by solving the HJ VI (6) with the following Hamiltonian:

\[
H_i(t, x_i, \lambda) = \min_{u_i \in U_i} \max_{d_i \in D_i} \lambda \cdot f_i(t, x_i, u_i, d_i) \tag{11}
\]

where \(l_i(x_i), g_i(t, x_i), V_i(t, x_i)\) are implicit surface functions representing the target \(T_i\), \(G_i(t), V_i(t)\), respectively. From the BRS, we can obtain the optimal control

\[
u_i^*(t, x_i) = \arg \min_{u_i \in U_i} \max_{d_i \in D_i} \lambda \cdot f_i(t, x_i, u_i, d_i) \tag{12}
\]

Here, as well as in the other two methods, the latest departure time \(t^*_{LDT}\) is then given by \(\arg \sup_{t \geq 0} x_{j0} \in V_i(t)\).

If there is a centralized controller directly controlling each of the \(N\) vehicles, then the control law of each vehicle can be enforced. In this case, lower-priority vehicles can safely assume that higher-priority vehicles are applying the enforced control law. In particular, the optimal controller for getting to the target, \(u_i^*(t, x_i)\) can be enforced. In this case, the dynamics of each vehicle becomes

\[
\begin{align*}
\dot{x}_i &= f_i^*(t, x_i, d_i) = f_i(t, x_i, u_i^*(t, x_i), d_i) \\
d_i &\in D_i, \quad i = 1, \ldots, N, \quad t \in [t^*_{LDT}, t^*_{STA}] 
\end{align*} \tag{13}
\]

where \(u_i\) no longer appears explicitly in the dynamics.

From the perspective of a lower-priority vehicle \(Q_j\), a higher-priority vehicle \(Q_j, j < i\) induces a time-varying obstacle that represents the positions that could possibly be within the capture radius \(R_c\) of \(Q_j\) under the dynamics \(f_j^*(t, x_j, d_j)\). Determining this obstacle involves computing a forward reachable set (FRS) of \(Q_j\) starting from \(x_j(\cdot)_{\text{LDT}} = x_j(0)\). The FRS \(W_j(t)\) is defined as follows:

\[
W_j(t) = \{y \in \mathbb{R}^n : \exists d_j(\cdot) \in D_j, x_j(\cdot) \text{ satisfies (13), } x_j(t) = y\} \tag{14}
\]

The FRS can be computed using the following HJ VI:

\[
\begin{align*}
D_i W_j(t, x_j) + H_j(t, x_j, D_x W_j) &= 0, t \in [t^*_{LDT}, t^*_{STA}] \\
W_j(t^*_{LDT}, x_j) &= \tilde{l}(x_j) \\
H_j(t, x_j, \lambda) &= \max_{d_j \in D_j} \lambda \cdot f_j^*(t, x_j, d_j) 
\end{align*} \tag{15}
\]

where \(\tilde{l}\) is chosen to be such that \(\tilde{l}(y) = 0 \Leftrightarrow y = x_j(t^*_{LDT})\).

The FRS \(W_j(t)\) represents the set of possible states at time \(t\) of a higher-priority vehicle \(Q_j\) given all possible disturbances \(d_j(\cdot)\) and given that \(Q_j\) uses the feedback controller \(u_j^*(t, x_j)\). In order for a lower-priority vehicle \(Q_i\) to guarantee that it does not go within a distance of \(R_c\) to \(Q_j, Q_i\) must stay a distance of at least \(R_c\) away from the set \(W_j(t)\) for all possible values of the non-position states \(h_j\). This gives the obstacle induced by a higher-priority vehicle \(Q_j\) for a lower-priority vehicle \(Q_i\) as follows:

\[
O_i^j(t) = \{x_i : \text{dist}(p_i, P_j(t)) \leq R_c\} \tag{16}
\]

where the \(\text{dist}(\cdot, \cdot)\) function represents the minimum distance from a point to a set, and the set \(P_j(t)\) is the set of states in the FRS \(W_j(t)\) projected onto the states representing position \(p_j\), and disregarding the non-position dimensions \(h_j:\)

\[
P_j(t) = \{p_j : \exists h_j, (p_j, h_j) \in W_j(t)\}. \tag{17}
\]

Finally, taking the union of the induced obstacles \(O_i^j(t)\) as in (4) gives us the time-varying obstacles \(G_i(t)\) needed to define and determine the BRS \(V_i(t)\) in (10). Repeating this process, all vehicles will be able to plan paths that guarantee the vehicles’ timely and safe arrival.

B. Method 2: Least Restrictive Control

Here, we again begin with the highest-priority vehicle \(Q_1\) planning its path by computing the BRS \(V_i(t)\) in (10). However, if there is no centralized controller to enforce the control policy for higher-priority vehicles, weaker assumptions must be made by the lower-priority vehicles to ensure collision avoidance. One reasonable assumption that a lower-priority vehicle can make is that all higher-priority vehicles

1In practice, we define the target set to be a small region around the vehicle’s initial state for computational reasons.
follow the least restrictive control that would take them to their targets. This control would be given by
\[ u_j(t, x_j) \in \begin{cases} \mathcal{U}_j(t, x_j) \text{ given by } (12) & \text{if } x_j(t) \in \partial V_j(t), \\ \mathcal{U}_j & \text{otherwise} \end{cases} \quad (18) \]

Such a controller allows each vehicle to use any controller, except when it is on the boundary of the BRS, \( \partial V_j(t) \), in which case \( u_j^*(t, x_j) \) given by (12) must be used to get to the target safely and on time. This assumption is the weakest one that could be made by lower-priority vehicles given that the higher-priority vehicles will get to their targets on time. Suppose a lower-priority vehicle \( Q_j \) assumes that higher-priority vehicles \( Q_j, j < i \) use the least restrictive control strategy in (18). From the perspective of \( Q_i \), a higher-priority vehicle \( Q_j \) could be in any state that is reachable from \( Q_j \)'s initial state \( x_j(t_j^{\text{LDT}}) = x_{j0} \) and from which the target \( T_j \) can be reached. Mathematically, this is defined by the intersection of a FRS from the initial state \( x_j(t_j^{\text{LDT}}) = x_{j0} \) and the BRS defined in (10) from the target set \( T_j \), \( V_j(t) \cap \mathcal{W}_j(t) \). In this situation, since \( Q_j \) cannot be assumed to be using any particular feedback control, \( \mathcal{W}_j(t) \) is defined as

\[ \mathcal{W}_j(t) = \{ y \in \mathbb{R}^n_j : \exists u_j(\cdot) \in \mathcal{U}_j, \exists d_j(\cdot) \in \mathcal{D}_j, \]
\[ x_j(\cdot) \text{ satisfies } (1), x_j(t_j^{\text{LDT}}) = x_{j0}, x_j(t) = y \} \quad (19) \]

This FRS can be computed by solving (15) without obstacles, and with

\[ H_j(t, x_j, \lambda) = \max_{u_j \in \mathcal{U}^j} \max_{d_j \in \mathcal{D}_j} \lambda \cdot f_j(t, x_j, u_j, d_j) \quad (20) \]

In turn, the obstacle induced by a higher-priority \( Q_j \) for a lower-priority vehicle \( Q_i \) is as follows:

\[ \mathcal{O}^i(t) = \{ x_i : \text{dist}(p_i, \mathcal{P}_j(t)) \leq R_c \}, \quad \mathcal{P}_j(t) = \{ p_j : \exists h_j(p_j, h_j) \in \mathcal{V}_j(t) \cap \mathcal{W}_j(t) \} \quad (21) \]

C. Method 3: Robust Trajectory Tracking

Although it is impossible to commit to and track an exact trajectory in the presence of disturbances, it may still be possible to robustly track a nominal trajectory with a bounded error at all times. If this can be done, then the tracking error bound can be used to determine the induced obstacles. Here, computation is done in two phases: the planning phase and the disturbance rejection phase. In the planning phase, we compute a nominal trajectory \( x_{r,j}(\cdot) \) that is feasible in the absence of disturbances. In the disturbance rejection phase, we compute a bound on the tracking error.

In the planning phase, planning is done for a reduced control set \( \mathcal{U}^p \subset \mathcal{U} \), as some margin is needed to reject unexpected disturbances while tracking the nominal trajectory. In the disturbance rejection phase, we determine the error bound independently of the nominal trajectory. Let \( x_j \) and \( x_{r,j} \) denote the states of the actual vehicle \( Q_j \) and an arbitrary nominal trajectory, respectively, and define the tracking error \( e_j = x_j - x_{r,j} \). When the error dynamics are independent of the absolute state as in (22) and also (7) in [13], we can obtain error dynamics of the form

\[ e_j(t) = f_j(e_j, u_j, u_{r,j}, d_j), \quad u_j \in \mathcal{U}_j, u_{r,j} \in \mathcal{U}^p_j, d_j \in \mathcal{D}_j, \quad t \leq 0 \quad (22) \]

To obtain bounds on the tracking error, we first conservatively estimate the error bound around any reference state \( x_{r,j} \), denoted \( \mathcal{E}^c_j = \{ e_j : \| p_{e,j} \|_2 \leq R_{EB} \} \), where \( p_{e,j} \) denotes the position coordinates of \( e_j \) and \( R_{EB} \) is a design parameter. We next solve a reachability problem with its complement \( \mathcal{E}^c_j \), the set of tracking errors violating the error bound, as the target in the space of the error dynamics. From \( \mathcal{E}^c_j \), we compute the following BRS:

\[ \mathcal{Y}_j(t, 0) = \{ y : \forall u_j(\cdot) \in \mathcal{U}_j, \exists u_{r,j}(\cdot) \in \mathcal{U}^p_j, \exists d_j(\cdot) \in \mathcal{D}_j, e_j(\cdot) \text{ satisfies } (22), e_j(t) = y, \exists s \in [t, 0], e_j(s) \in \mathcal{E}^c_j \} \quad (23) \]

where the Hamiltonian to compute the BRS is given by:

\[ H_j(e_j, \lambda) = \max_{u_j \in \mathcal{U}_j} \min_{u_{r,j} \in \mathcal{U}^p_j} \min_{d_j \in \mathcal{D}_j} \lambda \cdot f_j(e_j, u_j, u_{r,j}, d_j) \quad (24) \]

Letting \( t \to -\infty \), we obtain the infinite-horizon control-invariant set \( \Omega_j := \lim_{t \to -\infty} (\mathcal{Y}_j(t, 0))^c \). If \( \Omega_j \) is nonempty, then the tracking error \( e_j \) at flight time is guaranteed to remain within \( \Omega_j \subseteq \mathcal{E}^c_j \) provided that the vehicle starts inside \( \Omega_j \) and subsequently applies the feedback control law

\[ \kappa_j(e_j) = \arg \max_{u_j \in \mathcal{U}_j} \min_{u_{r,j} \in \mathcal{U}^p_j} \min_{d_j \in \mathcal{D}_j} \lambda \cdot f_j(e_j, u_j, u_{r,j}, d_j) \quad (25) \]

The induced obstacles by each higher-priority vehicle \( Q_j \) can thus be obtained by:

\[ \mathcal{O}_j(t) = \{ x_i : \exists y \in \mathcal{P}_j(t), \| p_{i} - y \|_2 \leq R_c \}, \quad \mathcal{P}_j(t) = \{ p_j : \exists h_j(p_j, h_j) \in \mathcal{V}_j(t) \cap \mathcal{W}_j(t) \} \quad (26) \]

where the “+” in (26) denotes the Minkowski sum.

Since each vehicle \( Q_j, j < i \), can only be guaranteed to stay within \( \Omega_j \), we must make sure during the path planning of \( Q_i \) that at any given time, the error bounds of \( Q_i \) and \( Q_j \), \( \Omega_i \) and \( \Omega_j \), do not intersect. This can be done by augmenting the total obstacle set by \( \Omega_i \):

\[ \hat{G}_i(t) = G_i(t) + \Omega_i \quad (27) \]

Finally, given \( \Omega_i \), we can guarantee that \( Q_i \) will reach its target \( T_i \) if \( \Omega_i \subseteq \hat{T}_i \); thus, in the path planning phase, we modify \( \hat{T}_i \) to be \( \tilde{T}_i := \{ x_i : \hat{G}_i + x_i \subseteq \tilde{T}_i \} \), and compute a BRS, with the control authority \( \mathcal{U}^p_i \), that contains the initial state of the vehicle. Mathematically,

\[ \mathcal{V}_i^{\text{int}}(t, t_i^{\text{STA}}) = \{ y : \exists u_i(\cdot) \in \mathcal{U}^p_i, x_i(\cdot) \text{ satisfies } (3), \]
\[ \forall s \in [t, t_i^{\text{STA}}], x_i(s) \notin \hat{G}_i(s), \]
\[ \exists s \in [t, t_i^{\text{STA}}], x_i(s) \in \tilde{T}_i, x_i(t) = y \} \quad (28) \]

The Hamiltonian to compute \( \mathcal{V}_i^{\text{int}}(t, t_i^{\text{STA}}) \) and the optimal control for reaching \( \tilde{T}_i \) are given by (7) and (9) respectively. The nominal trajectory \( x_{r,i}(\cdot) \) can thus be obtained by using vehicle dynamics (3), with the optimal control \( u_i^{\text{opt}}(\cdot) \). From the resulting nominal trajectory \( x_{r,i}(\cdot) \), the overall control policy to reach \( \tilde{T}_i \) can be obtained via (25).

V. NUMERICAL SIMULATIONS

We demonstrate our proposed methods using a four-vehicle example. Each vehicle has the following model:
\[ \dot{p}_{x,i} = v_i \cos \theta_i + d_{x,i}, \]
\[ \dot{p}_{y,i} = v_i \sin \theta_i + d_{y,i}, \]
\[ \ddot{\theta}_i = \omega_i + d_{\theta,i}, \]
\[ \| (d_{x,i}, d_{y,i}) \| \leq d_x, \| d_{\theta,i} \| \leq d_{\theta} \]

where \( p_i = (p_{x,i}, p_{y,i}), \theta_i, d = (d_{x,i}, d_{y,i}, d_{\theta,i}) \) respectively represent \( Q_i \)'s position, heading, and disturbances in the three states. The control of \( Q_i \) is \( u_i = (v_i, \omega_i) \), where \( v_i \) is the speed of \( Q_i \) and \( \omega_i \) is the turn rate; both controls have a lower and upper bound. For illustration purposes, we choose \( v = 0.5, \bar{v} = 1, \bar{\omega} = 1 \); however, our method can easily handle the case in which these inputs differ across vehicles and cases in which each vehicle has a different dynamic model. The disturbance bounds are chosen as \( d_x = 0.1, d_y = 0.2, d_{\theta} = 0.2 \), which correspond to a 10% uncertainty in the dynamics.

The initial states of the vehicles are given as follows:
\[ x_1^0 = (-0.5, 0, 0), \quad x_2^0 = (0.5, 0, \pi), \]
\[ x_3^0 = (-0.6, 0.6, 7\pi/4), \quad x_4^0 = (0.6, 0.6, 5\pi/4). \] (29)

Each of the vehicles has a target set \( T_i \) that is circular in their position \( p_i \) centered at \( c_i = (x_{c,i}, y_{c,i}) \) with radius \( r \):
\[ T_i = \{ x \in \mathbb{R}^3 : \| p_i - c_i \| \leq r \}. \] (30)

For the example shown, we chose \( c_1 = (0.7, 0.2), c_2 = (-0.7, 0.2), c_3 = (0.7, -0.7), c_4 = (-0.7, -0.7) \) and \( r = 0.1 \). The setup of the example is shown in Fig. 1a.

Using the SPP algorithms presented, we obtain \( t_{LDT}^1 = 1, 2, 3, 4 \) assuming \( t_{STA}^i = 0 \). Note that even though \( t_{STA}^i \) is assumed to be same for all vehicles in this example for simplicity, our method can easily handle the case in which \( t_{STA}^i \) is different for each vehicle.

For each proposed method of computing induced obstacles, we show the vehicles’ entire trajectories (colored dotted lines), and overlay their positions (colored asterisks) and headings (arrows) at a point in time in which they are in relatively dense configuration. In all cases, the vehicles are able to avoid each other’s danger zones (colored dashed circles) while getting to their target sets in minimum time. In addition, we show the evolution of the BRS over time for \( Q_3 \) (green boundaries) as well as the obstacles induced by the higher-priority vehicles (black boundaries).

For the example shown, we chose \( c_1 = (0.7, 0.2), c_2 = (-0.7, 0.2), c_3 = (0.7, -0.7), c_4 = (-0.7, -0.7) \) and \( r = 0.1 \). The setup of the example is shown in Fig. 1a.

Using the SPP algorithms presented, we obtain \( t_{LDT}^1 = 1, 2, 3, 4 \) assuming \( t_{STA}^i = 0 \). Note that even though \( t_{STA}^i \) is assumed to be same for all vehicles in this example for simplicity, our method can easily handle the case in which \( t_{STA}^i \) is different for each vehicle.

For each proposed method of computing induced obstacles, we show the vehicles’ entire trajectories (colored dotted lines), and overlay their positions (colored asterisks) and headings (arrows) at a point in time in which they are in relatively dense configuration. In all cases, the vehicles are able to avoid each other’s danger zones (colored dashed circles) while getting to their target sets in minimum time. In addition, we show the evolution of the BRS over time for \( Q_3 \) (green boundaries) as well as the obstacles induced by the higher-priority vehicles (black boundaries).

For the example shown, we chose \( c_1 = (0.7, 0.2), c_2 = (-0.7, 0.2), c_3 = (0.7, -0.7), c_4 = (-0.7, -0.7) \) and \( r = 0.1 \). The setup of the example is shown in Fig. 1a.

Using the SPP algorithms presented, we obtain \( t_{LDT}^1 = 1, 2, 3, 4 \) assuming \( t_{STA}^i = 0 \). Note that even though \( t_{STA}^i \) is assumed to be same for all vehicles in this example for simplicity, our method can easily handle the case in which \( t_{STA}^i \) is different for each vehicle.

For each proposed method of computing induced obstacles, we show the vehicles’ entire trajectories (colored dotted lines), and overlay their positions (colored asterisks) and headings (arrows) at a point in time in which they are in relatively dense configuration. In all cases, the vehicles are able to avoid each other’s danger zones (colored dashed circles) while getting to their target sets in minimum time. In addition, we show the evolution of the BRS over time for \( Q_3 \) (green boundaries) as well as the obstacles induced by the higher-priority vehicles (black boundaries).

Fig. 2: Evolution of the BRS and the obstacles induced by \( Q_1 \) and \( Q_2 \) for \( Q_3 \) in the centralized control method.

Fig. 1b shows simulated trajectories in the situation where each vehicle uses \( u_i^*(t, x_i) \) in (12). In this case, vehicles appear to deviate slightly from a straight line trajectory towards their targets, just enough to avoid higher-priority vehicles. The deviation is small since the centralized controller is quite restrictive, making the possible positions of higher-priority vehicles cover a small area. In the dense configuration at \( t = -1.0 \), the vehicles are close to each other but still outside each other’s danger zones.

Fig. 2 shows the evolution of the BRS for \( Q_3 \) (green boundary), as well as the obstacles (black boundary) induced by the higher-priority vehicles. The size of the obstacles remains relatively small. \( t_{LDT}^i \) values for the four vehicles (in order) in this case are \(-1.35, -1.37, -1.94 \) and \(-2.04 \). They are relatively close for the vehicles, because the obstacles generated by higher-priority vehicles are small and hence do not affect \( t_{LDT}^i \) of the lower-priority vehicles significantly.

A. Least Restrictive Control

Fig. 1c shows the simulated trajectories in the situation where each vehicle assumes that higher-priority vehicles use the least restrictive control to reach their targets, as described in IV-B. Fig. 3 shows the BRS and induced obstacles for \( Q_3 \).

\( Q_1 \) (red) takes a relatively straight path to reach its target. From the perspective of all other vehicles, large obstacles are induced, since lower-priority vehicles make the weak assumption that higher-priority vehicles are using the least restrictive control. Because the obstacles induced are so large, it is optimal for lower-priority vehicles to wait until higher-priority vehicles pass. As a result, a dense configuration is never formed, and trajectories are relatively straight. The \( t_{LDT}^i \) values for vehicles are \(-1.35, -1.97, -2.66 \) and \(-3.39 \). Compared to the centralized control method, \( t_{LDT}^i \) decrease significantly except for \( Q_1 \), which need not account for any moving obstacles.

From \( Q_3 \)'s (green) perspective, the large obstacles induced by \( Q_1 \) and \( Q_2 \) are shown in Fig. 3 as the black boundaries. As the BRS (green boundary) evolves over time, its growth gets inhibited by the large obstacles for a long time, as evident
Eventually, the boundary of the BRS reaches the initial state of $Q_3$ at $t = t_{LDT}^3 = -2.66$.

B. Robust Trajectory Tracking

In the planning phase, we reduced the maximum turn rate of the vehicles from 1 to 0.6, and the speed range from $[0.5, 1]$ to exactly 0.75 (constant speed). With these reduced control authorities, we determined from the disturbance rejection phase that any nominal trajectory from the planning phase can be robustly tracked within a distance of 0.075.

Fig. 1d shows vehicle trajectories in the situation where each vehicle robustly tracks a nominal trajectory. Fig. 4 shows the BRS evolution and induced obstacles for $Q_3$.

At $t = -0.89$, eventually, the boundary of the BRS reaches the initial state of $Q_3$ at $t = t_{LDT}^3 = -2.66$.

VI. CONCLUSIONS

We have proposed three different methods to account for disturbances and imperfect control policy information in sequential path planning; these three methods can be used independently across the different vehicles in the path planning problem. In each method, different assumptions about the control strategy of higher-priority vehicles are made. In all of the methods, all vehicles are guaranteed to successfully reach their respective destinations without entering each other’s danger zones despite the worst-case disturbance the vehicles could experience.

REFERENCES