Optimal Trajectory for Network Establishment of Remote UAVs

Prachya Panyakeow†, Ran Dai‡, and Mehran Mesbahi†

Abstract—This paper provides two approaches to establish a proximity network among a collection of unmanned aerial vehicles (UAVs) that are initially scattered in space. The goal is to find the shortest trajectories that bring the UAVs to a connected formation where they are in the range of detection of one another and headed in the same direction to maintain the connectivity. Constant-speed unicycles are chosen to represent UAV kinematics flying steady around cruising speed. Pontryagin Minimum Principle (PMP) is used to determine the control law and path synthesis for the UAVs under the turn-rate constraints. We introduce an algorithm to search for the optimal solution when the final network topology is specified; followed by a nonlinear programming method in which the final configuration is emerged from the optimization routine under the constraints that the final topology is connected. Simulation results along with the discussion on the performance of both methods are provided.

I. INTRODUCTION

Research in a multi-vehicle formation has been a topic of interest in the recent years due to broad applications in many areas including cooperative control of unmanned aerial vehicles (UAVs), formation control [4], [5], and flocking [11]. The paradigm of having one UAV, equipped with all necessary instruments, performing all mission tasks raises a number of issues related to the cost of development, integration, and time to complete these tasks. This has motivated the concept of multi-vehicle systems. In such a setup, the communication between the vehicles defined by the topology of the network has a significant effect on the performance of the multi-vehicle system. When the relative positions among the vehicles are considered in the construction of a geometry-based network (via wireless network technology), the information-exchange link is assumed to exist when vehicles are located within a specified range of each other. Since UAVs are assigned to perform tasks such as monitoring, imaging, reconnaissance, data processing, etc, separately at different locations, the relative distances between each vehicle may fall over the maximum communication range. In order to reestablish network connectivity for data exchange and resume mission operations as a group, the problem of controlling vehicles that are initially out of range of detection to the area where they can sense each other becomes important.

The recent works on network connectivity include [12], [13], [15] which focus on maintaining network connectivity during the entire mission. In [14], the authors derived the optimal control scheme to establish the connectivity among a group of spacecraft. In this paper, we consider a group of \( n \) nonholonomic vehicles that are initially out of the communication range, and propose methods to establish connectivity with optimal trajectories, subject to a turn-rate constraint. The unicycles are chosen to capture the dynamics of UAVs that fly at constant cruising speed as opposed to linear models. This model is widely used in many works related to UAVs such as [6], [7], [11]. Since the fuel usage is a prominent concern, in the case of constant speed vehicles, the optimality refers to minimizing maneuver time or path-length. The final configuration is in the form of relative positions and headings between connected vehicles with a graph representation of the connectivity in the form of a tree network. The solution is derived using Pontryagin Minimum Principle (PMP). However, the question of path synthesis encourages us to look into the Dubin’s problem [1], [2] that involves finding the shortest smooth path between two given points by a car, for which the starting and ending directions are specified. With the final orientation of the vehicles constrained to head in the same direction, the optimal solution for each vehicle comprises of three segments: two circular arcs of maximum curvature and a straight line tangent to the arcs. The task of determining how to combine these pieces is derived by solving a set of necessary conditions.

In this paper, we develop two approaches to establish the connectivity among UAVs. First we propose the algorithm to search for the global optimal solution when a target “tree” graph connectivity structure has been specified. However, searching for the best solution across all possible trees requires long computational time when applying to more than a few UAVs. The alternative approach is to transform the nonlinear optimal control into a parameter optimization problem without specifying the target graph, relaxing the on/off connectivity by using an exponential penalty function [3] to approximate the communication strength, and imposing the constraint that the final topology is connected. The control scheme remains the same while the time on each path segment is expressed as an unknown parameter. The connectivity constraint is then satisfied using the Cholesky decomposition. The solution can subsequently be solved using nonlinear programming solvers [8]. Simulation examples and the discussion regarding the performance of both methods are then presented.

II. PRELIMINARY AND PROBLEM FORMULATION

We consider a group of \( n \) homogeneous vehicles in 2D space represented by non-holonomic unicycle model

\[
\dot{x}_i(t) = V_i \cos \theta_i(t), \quad \dot{y}_i(t) = V_i \sin \theta_i(t), \quad \dot{\theta}_i(t) = u_i(t),
\]
where \( i = 1, \ldots, n \) and \((x_i, y_i)\) is the location of vehicle \( i \) measured relative to the earth frame. Every vehicle is assumed to have the same unit constant speed. The heading angle defines the orientation \( \theta_i \in (-\pi, \pi] \) and its rate defines the input \( u_i \). The constrained turn-rates are imposed as

\[
-\dot{\theta}_{\text{max}} \leq u_i(t) \leq \dot{\theta}_{\text{max}}.
\]  

(2)

without loss of generality, we assume that \( \dot{\theta}_{\text{max}} = 1 \). We use the graph \( G = (I, E) \) to represent the connection topology between vehicles. \( I = \{1, \ldots, N\} \) is the set of vertex indices where each vertex represents a vehicle and \( E \subseteq I \times I \) is the set of edges. The connection among the vehicles in the undirected network \( G \) is expressed by the entries of the adjacency matrix \( A(G) = [a_{ij}] \) with \( a_{ij} = 1 \) when \((i, j) \in E\), and \( a_{ij} = 0 \) otherwise.

An edge between vehicle \( i \) and vehicle \( j \) indicates that vehicle \( i \) is in the sensing range \( D \) of vehicle \( j \) and vice versa, i.e., vehicle \( i \) is adjacent to vehicle \( j \) or

\[
(i, j) \in E \iff d_{ij} = \|(x_i, y_i) - (x_j, y_j)\| \leq D.
\]

Vehicles \( i \) and \( j \) are said to be connected if \( G \) contains a path from \( i \) to \( j \). A group of vehicles is connected if every pair of \( i \) and \( j \) in \( G \) are connected.

There are several matrix representations of a graph \( G \). The degree matrix \( \Delta(G) \) is a diagonal matrix given by \( [\Delta(G)]_{ii} = \sum_{j=1}^{n} a_{ij} \) and \( [\Delta(G)]_{ij} = 0 \) (\( i \neq j \)). The incidence matrix \( B(G) = [b_{lj}] \in \mathbb{R}^{N \times |E|} \) is the matrix where each column corresponds to an edge in \( E \). The column \( l \in \{1, \ldots, |E|\} \) corresponds to the edge \((i, j) \in E \) and has entries \( b_{lj} = 1 = -b_{lj} \), where the remaining entries of the column are zero.

The Laplacian matrix \( L(G) \) is a square matrix defined as \( L(G) \equiv \Delta(G) - A(G) \). We denote the eigenvalues of \( L(G) \) by \( \lambda_1(G) \leq \lambda_2(G) \leq \ldots \leq \lambda_N(G) \), where \( \lambda_2(G) > 0 \) is a necessary and sufficient condition for connectivity of \( G \).

A tree is a connected graph without cycles and a connected graph with the least number of edges. A tree of \( n \) vertices has \( n-1 \) edges.

The goal of our paper is to control \( n \) unicycle vehicles that are initially out of sensing range to follow shortest trajectories under the turn-rate constraints to a final state where all the vehicles are connected and heading in the same direction. We refer to the graph representation at time \( t_f \) as \( G = G(q_f) \), where \( q_f \) is a short-hand notation for the collection of vehicle states at the final time. Since all UAVs flies at the same speed, every vehicle will travel the same path-length until it reaches the final location.

The problem can be decomposed into two parts: (1) searching for the optimal controls that bring the vehicles to a specified tree-graph connectivity structure, headed in the same direction, and (2) comparing the solutions among all possible target tree-graphs. The method to solve step (1) is presented in the following section. The “tree” structure is chosen as it is a connected graph with the least number of edges, or in our case, the least number of final constraints.

To simplify the problem, we assume that the closed path connecting initial locations of all vehicles forms a convex polygon as shown in Fig. 1(a). This is to avoid having some vehicles initially lie in the convex hull of other vehicles.

### III. Optimal Trajectory Planning for Target Tree-Graph Connectivity

The time-optimal control problem that brings \( n \) unicycle vehicles (1) to a specified tree-graph connectivity structure and heading in the same direction, is formulated as follow:

\[
\min_{u_1, \ldots, u_n} \int_0^{t_f} 1 \, dt, \quad \text{subject to}
\]

\[
x_i(t) = \cos \theta_i(t), \quad i = 1, \ldots, n \quad \text{system dynamics}
\]

\[
y_i(t) = \sin \theta_i(t),
\]

\[
\dot{\theta}_i(t) = u_i(t),
\]

\[
x_i(0) = x_{i0}, \quad i = 1, \ldots, n \quad \text{initial condition}
\]

\[
y_i(0) = y_{i0}, \quad i = 1, \ldots, n \quad \text{control constraint}
\]

\[
\|u_i(t)\| \leq 1, \quad i = 1, \ldots, n \quad \text{control constraint}
\]

\[
\forall (i,j) \in T, \quad (x_i(t_f) - x_j(t_f))^2 + (y_i(t_f) - y_j(t_f))^2 \leq D^2,
\]

\[
\theta_i(t_f) - \theta_j(t_f) = 0,
\]

where \( T \) is an edge set that forms a tree, \( q = [q_1, \ldots, q_n]^T \), \( q_i = [x_i, y_i, \theta_i]^T \), \( q_f = [q_{f1}, \ldots, q_{fn}]^T \), \( f_i = [\cos \theta_i, \sin \theta_i, u_i]^T \), and \( u = [u_1, \ldots, u_n]^T \).

The final constraints can be written as a smooth end-point manifold

\[
M_f = \{ q_f : \Omega(q_f) = 0 \},
\]

where \( \Omega(q_f) = [\Omega_1(q_f), \ldots, \Omega_{n-1}(q_f), \Omega_1(q_f), \ldots, \Omega_{n-1}(q_f)]^T \), \( \Omega_1(q_f) = (x_i(t_f) - x_j(t_f))^2 + (y_i(t_f) - y_j(t_f))^2 + s_i^2 - D^2, \) \( s_i \) is a slack variable, and \( \Omega_l(q_f) = \theta_i(t_f) - \theta_j(t_f), l = 1, \ldots, n - 1 \).

The PMP approach involves introducing of an adjoint variable vector \( \lambda = [\lambda_1, \ldots, \lambda_n]^T \) where \( \lambda_i = [\lambda_{xi}, \lambda_{yi}, \lambda_{\theta i}]^T \) and construct a Hamiltonian

\[
H(q, \lambda, u) = 1 + \lambda^T f = 1 + \sum_{i=1}^{n} \lambda_{xi} \cos \theta_i + \lambda_{yi} \sin \theta_i + \lambda_{\theta i} u_i.
\]

where \( \|H(q, \lambda, u)\| = 1 + \lambda^T f = 1 + \sum_{i=1}^{n} \lambda_{xi} \cos \theta_i + \lambda_{yi} \sin \theta_i + \lambda_{\theta i} u_i \). The following proposition provides necessary conditions for optimality of the control law.
Proposition 3.1: (PMP-Endpoint Manifolds) Suppose $u^*$ transfers the system from an initial state to a state in the final manifold $M_f$ with the minimum cost. Then $u^*$ satisfies the following necessary conditions:

1) There exists a nonzero costate $\lambda$, such that $\dot{\lambda} = -\frac{\partial H}{\partial \nu}$.
2) $u^* = \arg \min J(q, \lambda, u)$, $\forall t \in [0, t_f]$.
3) $H(q, \lambda) = 0$, $\forall t \in [0, t_f]$, and
4) $\lambda(t_f)$ is orthogonal to the end-point manifold $M_f$.

Proof: See [1]: Proposition 3.10.

A. Characterizing the Optimal Control

The following proposition presents the control law for problem (3) which is similar to that of Dubin’s car [1].

Proposition 3.2: The control law for (3) is of the form:

$$u^*_i = \begin{cases} -1 & \lambda_{\theta i} < 0 \text{ (turn right maximum)}, \\ 0 & \lambda_{\theta i} = 0 \text{ (go straight)}, \\ 1 & \lambda_{\theta i} > 0 \text{ (turn left maximum)}. \end{cases} \tag{5}$$

Proof: From Proposition 3.1, the control law is derived from the pointwise minimizing argument of the Hamiltonian

$$u^* = \arg \min_{\|u\| \leq 1} H(q, \lambda, u) = \arg \min_{\|u\| \leq 1} \sum_{i=1}^{n} \lambda_{\theta i} u_i,$$

which states that the optimal control only depends on the sign of $\lambda_{\theta i}$, i.e., $u^*_i = \begin{cases} -1 & \lambda_{\theta i} < 0 \text{ (turn right)}, \\ 0 & \lambda_{\theta i} = 0 \text{ (go straight)}, \\ 1 & \lambda_{\theta i} > 0 \text{ (turn left)}. \end{cases}$ However, if $\lambda_{\theta i}$ vanishes on a non-zero time-interval $\mathcal{T}$, PMP cannot provide information on the optimal control. For this singular case, we look at the first condition of Proposition 3.1 which requires

$$\dot{\lambda}_{\theta i} = -\frac{\partial H}{\partial \theta_i} = 0, \quad \dot{\lambda}_{\nu i} = -\frac{\partial H}{\partial \nu_i} = 0, \quad \lambda_{\theta i} = \lambda_{\theta i} \sin \theta_i - \lambda_{\nu i} \cos \theta_i. \tag{6}$$

This shows that both $\lambda_{\theta i}$ and $\lambda_{\nu i}$ are constants. Furthermore, the third condition of Proposition 3.1 demands that

$$H|_{\lambda_{\theta i}=0, t=0} = 1 + \sum_{i=1}^{n} \lambda_{\theta i} \sin \theta_i = 0, \forall t \in [0, t_f]. \tag{7}$$

Since $\lambda$ is a non-zero vector, $[\lambda_{\theta i}, \lambda_{\nu i}]$ cannot be zero and (7) can only hold if and only if $\theta_i(t)$ is constant or $u_i(t) = 0$ for all $t \in \mathcal{T}$; this yields the control law (5).

B. Optimal Path Synthesis

Since the control of problem (3) comprises of having each vehicle turn left or right at the maximum rate or go straight, the problem reduces to finding the sequence of commands and the duration spent on each segment. The following proposition presents the control sequence for each vehicle in (3).

Proposition 3.3: The control for each vehicle for the Optimal Target Tree problem (3) is of the form:

$$u^*_i(t) = \begin{cases} \pm 1 & 0 \leq t < t_i, \\ 0 & t_i \leq t \leq t_f - \tilde{t}_i, \\ \pm 1 & t_f - \tilde{t}_i \leq t \leq t_f, \end{cases} \tag{8}$$

where $t_i$ and $\tilde{t}_i$ are switching times. The trajectory during the intervals $0 \leq t < t_i$, $t_i \leq t \leq t_f - \tilde{t}_i$, and $t_f - \tilde{t}_i \leq t \leq t_f$ are called first, second, and third segment for vehicle $i$, respectively.

Proof: The last condition of Proposition 3.1 requires $\lambda(t_f)$ to be orthogonal to $M_f$. In other words, $\lambda(t_f)$ in the space spanned by $\nabla q \Omega(q_f)$. Since $\Omega_i(q_f)$ and $\Omega_f(q_f)$ are prescribed by the different pair of vehicles that form an edge of a tree-graph, they are linearly independent, and $M_f$ possesses a unique tangent plane at every point $q_f$ in $M_f$.

The final condition for the adjoint variable becomes

$$\lambda(t_f) = \sum_{i=1}^{n-1} \nu_i \nabla q \Omega_i(q_f) + \tilde{\nu} \nabla q \tilde{\Omega}(q_f) \tag{9}$$

and

$$= \sum_{i=1}^{n-1} B(T) \nu_i \left[ \begin{array}{c} x_i(t_f) - x_j(t_f) \\ y_i(t_f) - y_j(t_f) \\ 0 \end{array} \right] + \tilde{\nu} \left[ \begin{array}{c} 0 \\ 0 \\ 1 \end{array} \right],$$

where $\nu_i$ and $\tilde{\nu}_i$ are multipliers and $B(T)$ is the $i$th-column of the incidence matrix of the tree $T$ representing the edge $l_i$ connecting vehicles $i$ and $j$.

From (9), we have $\lambda_{\theta i}(t_f) = \sum_{i=1}^{n-1} B(T) \nu_i \neq 0$. This means that every vehicle needs to make the final turn to get to the prescribed relative distance and heading in the same direction, hence $u^*_i(t) = \pm 1$ when $t_i \leq t \leq t_f$.

Meanwhile, (6) indicates that $\lambda_{\theta i}(t_f)$ evolves sinusoidally when $u_i(t) = \pm 1$ and that $\lambda_{\theta i}(t_f - t_i) = 0$. Proposition 3.2 suggests that the control vanishes throughout the time interval $\mathcal{T}$ that $\lambda_{\theta i}(t) = 0$. Therefore, $\exists t_i \leq t_f - \tilde{t}_i$ such that $u_i(t) = 0$ when $t_i \leq t \leq t_f - \tilde{t}_i$ or every vehicle flies on a straight path before reaching the final turn. In order to connect the last two segments of the optimal trajectory with the initial headings and locations, it is transparent that the vehicle has to turn at its maximum rate in the first segment as $\lambda_{\theta i}(t)$ evolves sinusoidally toward zero when $t = t_i$. This thus implies that the control sequence is of the form (8).

Now, solving for the solution of (3) reduces to determining the choices of controls $u_i(t < t_i)$ and $u_i(t_f - \tilde{t}_i \leq t < t_f)$ (either 1 or -1) and the switching times $t_i$ and $\tilde{t}_i$. For $n$ vehicles, we have $2^n$ candidates for the optimal solution: turn left or right on the first and third segments and for $n$ vehicles; the method to select this will be presented in the next subsection. Meanwhile, the switching time $t_i$ and $\tilde{t}_i$ are obtained by solving the following necessary conditions.

Proposition 3.4: (Path Synthesis) The switching times $t_i, \tilde{t}_i, i = 1, \ldots, n$ in (8) and the final time $t_f$ of the Optimal Target Tree problem (3) are determined by solving the following intermediate/final conditions:

1) $(2n - 2)$ final constraints of the states: $\forall (i, j) \in T$, $(x_i(t_f) - x_j(t_f))^2 + (y_i(t_f) - y_j(t_f))^2 + s_{ij}^2 - D^2 = 0$ and $\theta_i(t_f) - \theta_j(t_f) = 0$,
2) $(n - 1)$ constraints of the slack variable: $\forall (i, j) \in T$, $\nu_{\theta j} (x_i(t_f) - x_j(t_f))^2 + (y_i(t_f) - y_j(t_f))^2 - D^2 = 0$,
3) $n$ intermediate constraints of the adjoint variable: $\lambda_{\theta i}(t_f) = \lambda_{\theta i} \sin \theta_i(t_f) - \lambda_{\nu i} \cos \theta_i(t_f) = 0$.
4) $n$ initial constraints of the adjoint variable: $\lambda_{\theta i}(t_f) = \sum_{i=1}^{n-1} B(T) \nu_i \nabla q \Omega_i(q_f) + \tilde{\nu} \nabla q \tilde{\Omega}(q_f) dt$,
5) a final condition of the Hamiltonian $H(q(t_f), \lambda(t_f)) = 0$, 4632
where \( \nu_{ij} \equiv \nu_t \) and \( s_{ij} \equiv s_t \) are multiplier/slack variable for edge \( l \) between vehicle \( i \) and \( j \), \( l = 1, \ldots, n - 1 \).

**Proof:** Let \( u_t \) be a short-hand notation for \( u_t(0 \leq t < t_i) \) and \( \tilde{u}_t \) for \( u_t(t_i \leq t \leq t_f) \). We now have \( 5n - 2 \) unknowns which include \( t_i, t_i, i = 1, \ldots, n \), \( \nu_t, \tilde{u}_t, s_t, (l = 1, \ldots, n - 1) \) and \( t_f \), hence \( 5n - 2 \) nonlinear equations are required to solve for the unknowns.

Starting with system dynamics, and integrating the dynamics through the switching times to the final time yields

\[
\begin{align*}
x_i(t_f) &= x_i(t_0) + \sin(\theta_{f,i} + u_i t_i) - \sin \theta_{i} + \cos(\theta_{f,i} + u_i t_i) (t_f - t_i - \tilde{t}_i) \\
y_i(t_f) &= y_i(t_0) + \cos(\theta_{f,i} + u_i t_i) - \cos \theta_{i} + \sin(\theta_{f,i} + u_i t_i) (t_f - t_i - \tilde{t}_i)
\end{align*}
\]

\[
\theta_i(t_f) = \theta_i(t_0) + u_i t_i + \tilde{u}_i \tilde{t}_i, \quad i = 1, \ldots, n.
\]

(10)

Substituting these into the final constraints \( (x_i(t_f) - x_j(t_f))^2 + (y_i(t_f) - y_j(t_f))^2 - l_{ij}^2 = 0 \) and \( \theta_i(t_f) - \theta_j(t_f) = \pi, (i, j) \in T \), gives us \( 2n - 2 \) nonlinear equations as a function of \( t_1, \ldots, t_n \), and \( t_f \).

The condition for \( s_t \) is derived from \( \nabla_s \Omega_f(q_f) = \nu_t s_t = 0 \). Multiplying the final constraint with \( \nu_t \) yields \( \nu_t x_i(t_f) - x_j(t_f))^2 + (y_i(t_f) - y_j(t_f))^2 - l_{ij}^2 = 0 \), another \( n - 1 \) equation.

Next, the necessary conditions associated with \( \lambda_{\theta_t} \) are derived base on the second segment where \( \lambda_{\theta_t} = 0 \Rightarrow \lambda_{\theta_t} = 0 \). Substituting \( \lambda_{x_t} \) and \( \lambda_{\theta_t} \) with the value in (9) yields

\[
\begin{align*}
\left[\sum_{j \in N_i} \nu_{ij} f_j(x_j(t_f) - x_i(t_f))\cos \theta_i(t_f) \right] = \\
\left[\sum_{j \in N_i} \nu_{ij} f_j(y_j(t_f) - y_i(t_f))\sin \theta_i(t_f) \right]
\end{align*}
\]

where \( N_i \) is a neighboring index set of vertex \( i \) of a tree defined as \( N_i = \{ j : (i, j) \in T \} \); this gives us another \( n \) equations.

Subsequently, the final constraints \( \lambda_{\theta_t}(t_f) \) can be derived by integrating \( \lambda_{\theta_t} \) in (6) toward the final time:

\[
\lambda_{\theta_t}(t_f) = \sum_{i=1}^{n-1} B(T) \dot{s}_t_i = \int_{t_f}^{t_i} \lambda_{x_t} \sin \theta_i(t) - \lambda_{\theta_t} \cos \theta_i(t) \, dt = \lambda_{\theta_t} \left[ \sum_{i=1}^{n} \nu_{ij} f_j(x_j(t_f) - x_i(t_f))\cos \theta_i(t_f) + \left(\sum_{j \in N_i} \nu_{ij} f_j(y_j(t_f) - y_i(t_f))\sin \theta_i(t_f) \right) \right]
\]

Finally, the last equation involves the Hamiltonian that vanishes along the optimal trajectory toward the final time

\[
H(t_f) = 0 = \frac{1}{2} \sum_{i=1}^{n} \lambda_{x_t}(t_f) \cos \theta_i(t_f) + \lambda_{\theta_t}(t_f) \sin \theta_i(t_f) = 1 + \sum_{i=1}^{n} \left[ \frac{1}{2} \left( \sum_{j \in N_i} \nu_{ij} f_j(x_j(t_f) - x_i(t_f))\cos \theta_i(t_f) + \left(\sum_{j \in N_i} \nu_{ij} f_j(y_j(t_f) - y_i(t_f))\sin \theta_i(t_f) \right) \right) \right] \sin \theta_i(t_f)
\]

These give us \( 5n - 2 \) nonlinear equations to solve for \( 5n - 2 \) unknowns which can be done using natural numerical tools.

**C. Proposed Algorithm for Finding the Optimal Path**

It is important to choose the control candidate corresponding to the global optimal solution that yields the shortest path. For \( n \) vehicles, we have \( 2^{2n} \) candidates where each vehicle has a choice to turn right or left in the first and third segments. Taking a wrong turn results in a longer path corresponding to a local optimal solution. Fig. 2(a) shows the optimal trajectory candidate RLL-RRL (a short hand notation for vehicles 1, 2, and 3 turn right, left, and left during the third segments). Fig. 2(b) shows the global optimal solution RRL-RLL with \( t_f = 8.3 \) seconds.

The time-optimal path for each vehicle consists of two line segments of maximum curvature in concatenation with a straight line tangent to the arc. We noticed that if the arc covers more than \( \pi \), the turning other direction may result in a shorter path. All of the arcs in Fig. 2(b) are less than \( \pi \), hence yielding the global optimal solution. We use this fact to determine the control for each vehicle as follow:

**Step 1 (Determine the Candidate)** Initial data are the given initial conditions \( [x_{i0}, y_{i0}, \theta_{i0}] \). We use the centroid of the polygon \( [x_c, y_c]^T \) to estimate as the point that every vehicle needs to travel to. The idea is depicted in Fig. 1(b). We then use the line connecting each initial location to the centroid as an initial guess for the optimal path. Since these lines do not have the same length and the turning portion of the path are not considered, this guess may not result in the right candidate. Let \( r_{ic} \) denote a vector from \( [x_{i0}, y_{i0}]^T \) to \( [x_c, y_c]^T \) and \( v_{i0} = [cos \theta_{i0}, sin \theta_{i0}]^T \) denote vehicle \( i \) initial velocity. The angle between \( v_{i0} \) and \( r_{ic} \) is denoted as \( \alpha_i \). Vehicle \( i \) needs to turn from its initial heading to coincide with \( r_{ic} \). We determine the choice of control by choosing the direction with a smaller turn (less than \( \pi \)). The control for the first segment is chosen from

\[
u_i^*(t < t_i) = \begin{cases} 
1 & k_z(v_{i0} \times r_{ic}) > 0, \\
-1 & k_z(v_{i0} \times r_{ic}) \leq 0,
\end{cases}
\]

(11)

where \( k_z \) is the unit vector perpendicular to the xy-plane.

Furthermore, we use the vector sum of all \( r_{ic} \) to estimate the final heading of the formation \( \theta_i(t_f), \forall i \). Let \( v_f = [cos \theta_i(t_f), sin \theta_i(t_f)]^T = \sum_{i=1}^{n} r_{ic} \) denote the final velocity of the formation. The angle between \( v_f \) and \( r_{ic} \) is denoted as \( \alpha_i \). The control for the third segment is then chosen in the direction corresponding to the smaller turn

\[
u_i^*(t_f - \tilde{t}_i < t < t_f) = \begin{cases} 
1 & k_z(r_{ic} \times v_f) > 0, \\
-1 & k_z(r_{ic} \times v_f) \leq 0,
\end{cases}
\]

(12)

It is also noted that if either \( k_z(v_{i0} \times r_{ic}) \) or \( k_z(r_{ic} \times v_f) \) both have a small value or value close to \( \pi \), we need to find the solution for both candidates and compare the results. This will be discussed further in Step 3.

**Step 2 (Solve for the Switching Times)** The switching times is determined by solving nonlinear equations in Proposition 3.4. This often involves the use of a numerical...
the unknowns are $t_i, \tilde{t}_i, \nu_l, \tilde{\nu}_l, s_l, \text{ and } t_f$, where $i = 1, \ldots, n, l = 1, \ldots, n - 1$. According to Fig. 1(b), the vehicle needs to turn and align with $r_{ic}$, then proceed straight toward the centroid. This estimation is made as if the vehicle does not move forward while turning.

As the maximum turn-rate is set to 1, the time required for vehicle $i$ to turn toward the centroid equals the angle between $v_{i0}$ and $r_{ic}$. Hence, the initial guess for $t_i$ should be $\alpha_i$. Similarly, the initial guess for $\tilde{t}_i$ is made to be $\tilde{\alpha}_i$. Furthermore, with a unit speed, the time to travel to $[x_{ic}, y_{ic}]^T$ equals $\|r_{ic}\|$ and the total time for vehicle $i$ can be estimated as $\|r_{ic}\| + \alpha_i + \tilde{\alpha}_i$. Since every vehicle should have the same $t_f$, the initial guess is picked as the average of this value

$$\frac{1}{n} \sum_{i=1}^{n} (\|r_{ic}\| + \alpha_i + \tilde{\alpha}_i).$$

As for $\nu_l$, Proposition 3.4 (3) shows that the velocity vector during the second segment of vehicle $i$ is formed by the linear combination of that from its neighbors with the factor of $\nu_{ij}, j \in N_i$. Since the final location for each vehicle is likely to stay inside the convex hull of the initial locations, $\nu_l$ shall have the same sign. Therefore we choose the initial guess as $\forall l, \nu_l = 1$ and similarly $\tilde{\nu}_l = 1$. For $s_l$, since the final constraint in (3) is expected to close to equality, we choose $\forall l, s_l = 0$.

**Step 3 (Verify Results)** It is important to verify if the solver gives $0 < t_i, \tilde{t}_i < \pi$ when we pick $0 < t_i, t_i = 1$. If not, it means that our estimation using the centroid of the polygon is not valid. When this happens, we need to compare the result with the candidate where vehicle $i$ turns the other direction. Furthermore, if $t_i$ has the value close to 0 or $\pi$, it is not clear whether the control $u_i$ yields the global solution.

Our proposed algorithm can be summarized in Protocol 1. It is noted that solving the Optimal Target Tree-graph problem across all possible labeled tree provides a method to solve the original problem that requires $\lambda_2(G) > 0$. In fact, solving for $\lambda_2(G) > 0$ induces a connected graph $G$ which necessarily contains at least one tree $T$. However, the number of labeled trees on $n$ nodes is $n^{n-2}$, which makes the search for the best solution across all possible trees impractical when applying this algorithm to more than five UAVs due to the long computational time. In the next section, we introduce a relaxation method to solve the original problem for the larger number of UAVs.

**IV. NONLINEAR PROGRAMMING METHOD**

In this section, we present a NLP algorithm to search for the parameterized optimal solutions. We focus on searching for the optimal controls and switching times that satisfy the terminal graph connectivity constraints. The existence of an edge between two vehicles is encoded as an integer variable constrained by Euclidean distance at time $t_f$. The elements of the Laplacian for the graph at the final time are therefore linear functions of these nonlinear integer variables. The complications of constraining $\lambda_2$ for this integer-valued Laplacian matrix suggest finding a relaxation to the discrete form of the connectivity function. The data communication rate using WiFi devices, for example, drops off as the distance between the vehicles grows beyond a given threshold. We use the bump function [3]

$$w(d_{ij}) = 1/(1 + e^{\alpha(d_{ij} - \rho)}),$$

where $\alpha = \frac{1}{p_2 - p_1} \log \left(\frac{1-c}{c}\right)$ and $\rho = \frac{p_1 + p_2}{2}$, to represent the communication strength between a pair of UAVs. This leads to a weighted adjacency matrix with entries $a_{ij} = w(d_{ij}(t_f))$ to represent the element $a_{ij}$ for graph $G$ at time $t_f$. The plot of this function is shown in Fig. 3. However, constraining $\lambda_2$ for the weighted Laplacian matrix is still non-trivial. We thus use the Cholesky decomposition to convert the eigenvalue constraint to an equivalent semidefinite constraint.

**Corollary 4.1.** For a graph Laplacian $L(G)$, there $\exists \mu$ such that $L(G) + \mu 11^T/n > 0$ is equivalent to $\lambda_2(G) > 0$.

**Proof:** See [14]: Corollary 6.4.

**Lemma 4.2.** [9] Every symmetric positive definite matrix $M$ has a unique factorization of the form $M = LL^T$, where $L$ is a lower triangular matrix with positive diagonal entries.

From Cholesky decomposition, when the network is connected, we can find a lower triangular matrix $L$ with positive diagonal entries for the positive definite matrix $L(G) + \mu 11^T/n$. We use this property to transform the problem into the following parameter optimization problem where the matrix $L$, the switching time $t_i, \tilde{t}_i, i=1, \ldots, n$, and $t_f$ are expressed as unknown variables, without specifying the target graph:

$$\min_{t_1, \ldots, t_n, \tilde{t}_1, \ldots, \tilde{t}_n, t_f} t_f$$

subject to

$$\begin{align*}
\theta_i(t_f) &= \theta_j(t_f) \\
\rho_{ij}(t_f) &= ||(x_i(t_f), y_i(t_f)) - (x_j(t_f), y_j(t_f))|| \\
a_{ij} &= 1/(1 + e^{\alpha(d_{ij}(t_f) - \rho)}) \forall i, j \in 1, \ldots, n \\
L(G) + \mu 11^T/n - \xi I &= LL^T \\
L &= \begin{bmatrix}
\ell_{11} & \ell_{12} & \ell_{13} & \cdots & \ell_{1n} \\
\ell_{21} & \ell_{22} & 0 & \cdots & \cdots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
\ell_{n1} & \cdots & \cdots & \cdots & \ell_{nn}
\end{bmatrix}, \ell_{ii} > 0, \forall i \in 1, \ldots, n,
\end{align*}$$

where $x_i(t_f)$ and $y_i(t_f)$ is derived from (10), the control $u_i$ is derived from (11) and (12), and $\xi \in \mathbb{R}$ is a positive small number to guarantee that the weighted graph is connected. In this method, the final connected configuration corresponds with the optimal trajectories is obtained from the optimization routine. However, due to the nonlinearity of the problem,
this method may return a local optimal solution depending on the initial picks of the unknown variables.

V. SIMULATION RESULTS

Fig. 4 depicts the trajectories of five unicycle-UAVs corresponding to the shortest travel time to establish the tree-graph connectivity using the method developed in §III. The tree is prescribed as a Path-Graph where vehicle 1 is connected to 2, vehicle 2 to 3, vehicle 3 to 4, and vehicle 4 to 5, and \( t_f = 11.35 \) seconds. On the other hand, Fig. 5 shows the trajectories from the NLP method with \( t_f = 11.00 \) seconds as we pick \( \alpha = 4.4, \rho = 1, \) and \( \xi = 0.3. \) We notice that the final time from the NLP method is slightly faster than that from the Optimal Target Tree-Graph method. This is because the NLP algorithm produces a different connected configuration, e.g., vehicle 5 is “more” connected to vehicle 1 and 2, but “less” connected to 4.

VI. CONCLUSION

In this paper, two approaches for establishing a proximity network among a collection of scattered unicycle-vehicles are developed. The control law and path synthesis are derived using Pontryagin minimum principle. The algorithm for finding the optimal solution for the fixed final network topology is first introduced, followed by a nonlinear programming method in which the final connected configuration corresponds with the optimal paths emerges from the optimization routine. Simulation results along with the discussion on the performance of both methods have been provided.

There are several possible extensions of this work. First, the assumption that the initial locations of the UAVs form a convex polygon should be relaxed. This may require a set of commands that allow some vehicles to loiter while other vehicles arrive into the area. The other extension involves allowing the UAV speed to vary between stall and maximum speeds with some vehicle traveling faster than others if necessary. This can open up the possibility of developing an algorithm that not only brings the UAVs to a connected network, but also forms a desired formation shape.

REFERENCES