2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Clothoid-based Moving Formation Control Using Virtual Structures

Brian Merrell, Greg Droge

Abstract— **Formation control is a canonical problem in multi-agent systems as many multi-agent problems require agents to travel in coordination at some point during execution. This paper develops a method for coordinated moving formation control by building upon existing virtual structures approaches to define the relative vehicle positions and orientations and building upon clothoid-based motion planning to create the desired motion of the structure. The result is a coordinated formation control method that respects individual curvature constraints of each agent while allowing agents to track their desired positions within the formation with asymptotic convergence.**

I. Introduction

Multi-agent robotic systems are becoming increasingly prevalent in a host of applications, including security and surveillance [1], factory automation [2], inspection operations [3], transit [4] and convoying applications [5], distributed sensing [6], and target tracking networks [7], to name a few. A nearly universal requirement of multi-agent systems is that, at some point in their mission, they move together in coordinated motion, often in tight formation. Thus, formation control forms a fundamental enabling capability for multi-agent operations. The goal of this work is to create a moving formation control approach that will enable a group of vehicles to maintain a rigid, relative geometric configuration as they travel through a series of waypoints to a goal destination.

The work herein combines three major elements. The first is the use of virtual structures, which allow the desired positions of vehicles in the formation to be defined in terms of relative displacement to a single oriented point, e.g., [8]. Second, continuous-curvature paths [13], [11] are developed to design the virtual structure motion. Third, a zero-error trajectory tracking control law, [11] [14], is used to allow vehicles to converge to and track a sufficiently smooth trajectory.

This work extends continuous-curvature paths to provide sufficiently smooth paths for use with the zero-error tracking controller. It also extends the definition of a virtual structure to create executable trajectories for the followers. Furthermore, a motion model will be used for the virtual leader that directly considers curvature constraints for the virtual structure and is used to create a new clothoid for transition between straight lines and arcs. The use of this clothoid allows for a rapid evaluation of the curvature requirements imposed on each agent by the movement of the virtual structure. By evaluating solely the resulting formation movement over a single clothoid cycle (on the order of seconds of a trajectory), a formation designer could ensure curvature constraints are satisfied for an extended trajectory.

The remainder of this paper will proceed as follows. Preliminaries for formation definition, continuous-curvature paths, and trajectory tracking are presented in Section II. The virtual structure will be extended in Section III and a method for evaluating the curvature requirements for each agent will be given. The motion of the virtual structure will then be defined in Section IV. The paper ends with a brief example in Section V and concluding remarks in Section VI.

II. Preliminaries

This section presents key background information for the development of the zero-error moving formation control approach discussed in the sequel.

A. Formation Definition

Formation control is the coordinated effort of multiple agents to achieve or travel in some desired configuration. While there are a myriad of methods to define and achieve formation control, this work takes a rigid, geometric approach to defining the desired relative placement of the vehicles. Meaning that, at any snapshot in time, the desired relative position of vehicles forms a particular geometric shape.

A virtual structure approach, [8], is used to allow the formation to move about the environment. The virtual structure approach creates an image of the desired formation and uses this image to create references of what the desired formation state should be as depicted in Figure 1. The placement of each agent within the structure can be defined in terms of the "virtual leader" configuration consisting of a reference position, q_l, and an orientation, ψ_l. This configuration is referred to as a "virtual leader" since it may not be an actual physical agent. Following the developments in [9], the desired position of agent i at time t, $q_{d_i}(t)$, can be defined with respect to the leader as:

$$\begin{aligned} q_{d_i}(t) &= R(\psi_l(t))\tau_i + q_l(t) \\ R(\psi_l(t)) &= \begin{bmatrix} \cos\psi_l(t) & -\sin\psi_l(t) \\ \sin\psi_l(t) & \cos\psi_l(t) \end{bmatrix}, \end{aligned} \tag{1}$$

where τ_i is the desired relative offset of agent i from the virtual leader.

The notation $\mathcal{C}^k$ is used to denote the set of functions that are k-times continuously differentiable. In Section IV, a virtual leader trajectory in $\mathcal{C}^4$ will be produced to enable the development of trajectories that are sufficiently smooth for a follower vehicle to track.

Email: brian.merrell@aggiemail.usu.edu, greg.droge@usu.edu.
Department of Electrical and Computer Engineering, Utah State University, Logan, UT 84322, USA.

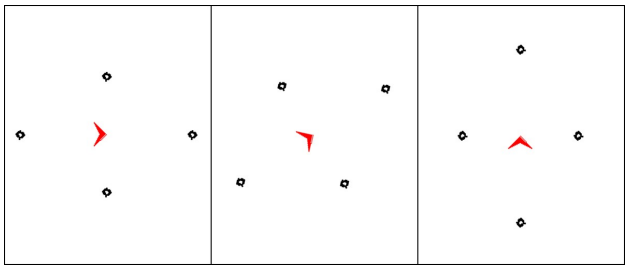

Fig. 1. A virtual structure shown in different configurations. The red icon denotes the virtual leader and the circles represent the desired follower positions.

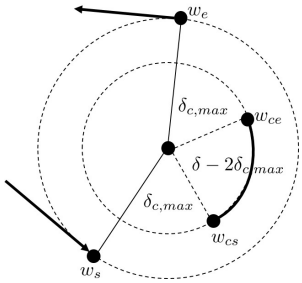

Fig. 2. Example of a continuous curvature turn where w_s represents the start of the transition clothoid, w_{cs} is where the path begins a circular arc, w_{ce} is where the circular arc ends and a clothoid begins, and w_e is the point where the clothoid ends, δ is the toal change in orientation while $\delta_{c,max}$ is the change while converging to maximum curvature.

B. Dynamic Motion Models

The motion for each agent is modeled using a smooth, unicycle kinematic model, e.g., [10], [11]. The states include the two-dimensional position of the vehicle, (q_1, q_2), its orientation ψ, and translational and rotational velocities, (v, ω), respectively. The model used to define the motion of the virtual leader is a slightly altered form of the unicycle model to accommodate the smoothness requirements needed in the sequel. It assumes a constant virtual leader velocity, v_l, the rotational velocity is replaced by a curvature term, $\omega_l = v_l \kappa_l$, and derivatives of the curvature are included as additional states. Thus, the full leader state is given as follows.

$$x_l = \begin{bmatrix} q_{l_1} \\ q_{l_2} \\ \psi_l \\ \kappa_l \\ \sigma_l \\ \gamma_l \end{bmatrix}, \dot{x}_l = \begin{bmatrix} v_l \cos(\psi_l) \\ v_l \sin(\psi_l) \\ v_l \kappa_l \\ \sigma_l \\ \gamma_l \\ u_l \end{bmatrix}, \tag{2}$$

where κ_l is the curvature of the leader trajectory and its first, second, and third derivatives are σ_l, γ_l, and u_l. It is assumed that $\kappa_l \in [-\kappa_{max}, \kappa_{max}]$, $\sigma_l \in [-\sigma_{max}, \sigma_{max}]$, and $u_l \in [-u_{max}, u_{max}]$.

C. Continuous Curvature Paths

The virtual leader motion model allows for direct consideration of the path curvature. To directly consider the maximum curvature in planning, Dubins paths were developed to find the shortest path between any two oriented waypoints [12]. Given a constant forward velocity, minimum paths are found by either executing three maximum curvature circles (CCC) or a circle, line, then a circle (CLC). However, Dubins paths assume that the vehicle curvature can be changed instantaneously. Continuous curvature paths extend the idea of Dubins paths by employing a smooth transition between desired curvatures, known as clothoids, as shown in Figure 2. The smooth transition are achieved by linearly changing the curvature at a maximum curvature rate. In [13], constant curvature turns (CCTurn) replace the Dubins circle, although planning is nearly identical.

It was shown in [11] that a truncated form of (2) could be used to develop CCPaths[1]. Using a motion model for the clothoid provides the added benefit of time-indexing, enabling the paths to be used in trajectory tracking control laws.

While the discussion of planning shortest paths is left to [13], the generation of the CCTurn is important for understanding of the development of sufficiently smooth trajectories in Section IV. As shown in Figure 2, a CCTurn has the vehicle turn as quickly as possible from one straight line to another while considering constraints on curvature and its derivatives. This change in orientation is given by the deflection angle, δ. The turn consists of three phases. The first phase is to move through a clothoid from zero curvature to the maximum curvature. The second phase is to execute a circular arc at maximum curvature. The third phase is to reflect the first clothoid and transition from maximum curvature to zero curvature. If δ is small enough, the vehicle never reaches maximum curvature before transitioning back to zero curvature, thus never executing the second phase consisting of the maximum curvature arc.

D. Trajectory Tracking

The virtual leader trajectory will be used to create a trajectory for each agent to follow. While there are numerous techniques for tracking a trajectory (i.e., have $q(t) \longrightarrow q_d(t)$ as $t \longrightarrow \infty$), a simplified form of the control law developed in [14] and modified in [11] is used. The actual trajectory tracking control law employed is not relevant except for two properties. The first being that [11] requires the reference trajectory to be in $\mathcal{C}^3$. The second being that the tracking controller must have asymptotic convergence.

III. The Virtual Structure

The virtual structure in (1) combined with a trajectory produced using the virtual leader model in (2) allows for the creation of a virtual structure motion model where each agent can use the tracking capabilities of the controller to move in a rigid formation. This section derives the generic trajectory that each agent should nominally follow. It is then shown that the structure of the virtual leader motion model can be exploited to evaluate characteristics of the resulting follower trajectories.

[1]The truncation comes from a lesser requirement for smoothness than $\mathcal{C}^4$.

A. Follower Trajectory Generation

The following lemma states that the trajectory generated from the virtual leader motion model and virtual structure will produce a sufficiently smooth trajectory to be used with the trajectory tracking control. The proof of the lemma gives the required equations for implementation.

Lemma 1: Using a virtual leader motion model defined in (2) and the leader-follower relationship defined in (3), the desired trajectory for the follower vehicle will be three times continuously differentiable (i.e. $q_{d_i}(t) \in \mathcal{C}^3$). *Proof:* To concisely show the smoothness properties of the produced trajectory, notation is introduced for ψ_l and its derivatives. The leader orientation vector is denoted as $h_l = \begin{bmatrix}\cos(\psi_l) & \sin(\psi_l)\end{bmatrix}^T$. The derivatives of ψ_l are denoted as $\dot{\psi}_l = \omega_l$, $\ddot{\psi}_l = \alpha_l$, and $\psi^{(3)} = \zeta_l$ where $\omega_l = v_l\kappa_l$, $\alpha_l = v_l\sigma_l$, and $\zeta_l = v_l\gamma_l$. Given a scalar ρ and the $\frac{\pi}{2}$ rotation matrix, J, we denote $\hat{\rho}$ as the skew symmetric matrix

$$\hat{\rho} = \rho J = \rho \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

This allows for a concise expression of the orientation vector derivative as $\dot{h}_l = \hat{\omega}_l h_l$. Similarly, the derivative for the rotation matrix, $R(\psi_l)$ can be written as $\dot{R}(\psi_l) = R(\psi_l)\hat{\omega}_l$, e.g. [15]. Repeated use of the product rule on (3) allows for the follower position and derivative to be expressed as

$$\begin{aligned} q_{d_i} &= R(\psi_l)\tau_i + q_l \\ \dot{q}_{d_i} &= R(\psi_l)\hat{\omega}_l\tau_i + \dot{q}_l \\ \ddot{q}_{d_i} &= R(\psi_l)(\hat{\omega}_l^2 + \hat{\alpha}_l)\tau_i + \ddot{q}_l \\ q_{d_i}^{(3)} &= R(\psi_l)(\hat{\omega}_l^3 + \hat{\zeta}_l + 3\hat{\omega}_l\hat{\alpha}_l)\tau_i + q_l^{(3)} \end{aligned}. \tag{3}$$

The derivatives of the virtual leader position can be written using (2) to express $\dot{q}_l = v_l h_l$ and then use repeated use of the product rule to write

$$\begin{aligned} \dot{q}_l &= v_l h_l \\ \ddot{q}_l &= v_l^2\kappa_l J h_l \\ q_l^{(3)} &= v_l^2\sigma_l J h_l - v_l^3\kappa_l^2 h_l \end{aligned}. \tag{4}$$

As (3) and (4) are written using purely continuous variables, this completes the proof. ■

B. Curvature Properties for Follower Trajectories

The virtual leader motion model allows for direct limitations on the leader's curvature and its derivatives. As trajectory curvature can directly correlate to executability of the trajectory for physical systems, it is important to understand how the virtual leader's curvature relates to the curvature of the follower vehicles.

Recall that curvature can be defined as $\kappa = \frac{\omega}{v}$ where the velocities can be derived directly from a trajectory in vector form as

$$v_{d_i} = \sqrt{\dot{q}_{d_i}^T\dot{q}_{d_i}},\ \omega_{d_i} = \frac{-\dot{q}_{d_i}^T J\ddot{q}_{d_i}}{\dot{q}_{d_i}^T\dot{q}_{d_i}}. \tag{5}$$

An immediate concern for the follower trajectory is to understand when the virtual structure motion will demand an agent to execute infinite curvature, which is addressed in the following lemma.

Lemma 2: Given a desired offset for agent i, written as $\tau_i = \begin{bmatrix}\tau_{i_1} & \tau_{i_2}\end{bmatrix}^T$, an infinite curvature will be commanded for agent i if and only if *(i)* $\tau_{i_1} = 0$ and *(ii)* $\kappa_l = \frac{1}{\tau_{i_2}}$.

Proof: An infinite curvature command will only come when $v_{d_i} = 0$ (or equivalently if $v_{d_i}^2 = 0$). Equation (5) for v_{d_i} can be used with (3) and $v_{d_i}^2$ can be simplified algebraically to

$$v_{d_i}^2 = v_l^2\kappa_l^2\tau_i^T\tau_i - 2v_l^2\kappa_l\tau_{i_2} + v_l^2. \tag{6}$$

Solving for the value of κ_l where (6) is zero, v_l^2 can be factored out and the quadratic equation used to produce

$$\kappa_l = \frac{\tau_{i_2} \pm \sqrt{-\tau_{i_1}^2}}{\tau_i^T\tau_i}, \tag{7}$$

which only has a real solution when $\tau_{i_1} = 0$ (condition *(i)*). In that case, condition *(ii)* directly falls out. ■

An interesting result from Lemma 2 is that only the vehicles that are to move directly "along side" the virtual leader run the risk of having a zero velocity, or, equivalently, turning in place.

However, it may be desirable to limit the virtual leader motion in such a way that the desired curvature for any follower agent lies below some threshold. This is addressed in the following lemma.

Lemma 3: The curvature for agent i will stay below the maximum threshold κ_{max_i} if, for all time t,

$$\kappa_{max_i} > \frac{v_l\kappa_l^3\tau_i^T\tau_i + [\sigma_l,\ 2v_l\kappa_l^2]\tau_i + v_l\kappa_l}{v_l\left(\kappa_l^2\tau_i^T\tau_i - 2\kappa_l\tau_{i_2} + 1\right)^{\frac{3}{2}}} \tag{8}$$

where time indices on $\kappa_l(t)$ and $\sigma_l(t)$ have been omitted for sake of brevity.

Proof: The proof is trivial in nature as the right-hand side of (8) is the equation for agent i's desired curvature. It can be obtained by combining (3) with (5), using the relation $\kappa_{d_i} = \frac{\omega_{d_i}}{v_{d_i}}$, and simplifying algebraically. ■

Lemma 3 may appear difficult to satisfy at first glance since it requires solving for all possible combinations of κ_l and σ_l through the entirety of the trajectory. This is true, generally. However, it will be shown in Section IV that by using a clothoid approach to virtual leader planning, all possible combinations of κ_l and σ_l occur during the clothoid, so only the clothoid transition need be evaluated.

IV. Continuous Curvature Virtual Leader Trajectory

An example method to create a sufficiently smooth virtual leader trajectory is now shown using clothoids as an inspiration. In this section, two minor contributions are made to the clothoid techniques. First, a sufficiently smooth clothoid is created to satisfy the continuity conditions for a virtual leader. Second, a simplified waypoint planning technique is presented to rapidly plan paths given a sequence of desired position-based waypoints.

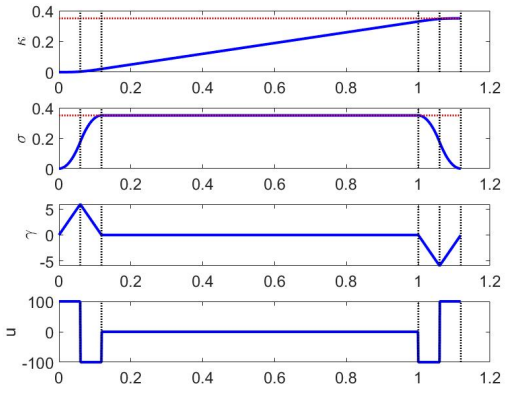

Fig. 3. An illustration of the construction of x_{κ_l} using bang-bang control. The figures from top to bottom are κ_l, σ_l, γ_l, and u_l. The blue line shows the value over time, the red shows bounds, and the black lines show the switching times.

A. A Sufficiently Smooth Clothoid

Recall that a clothoid is used to transfer the vehicle from a straight line along the tangent of an outer circle to a smaller concentric circle where the vehicle is then able to execute its maximum curvature as depicted in Figure 2. The straight line corresponds to κ_l and its derivatives being equal to zero and the inner circle corresponds to $\kappa_l = \pm\kappa_{max}$ with the derivatives again equal to zero.

Thus, once started, the clothoid need only consider κ_l, its derivatives, and the amount of time to stay on the inner circle. The final three elements of x_l in (2) consisting of κ_l and its derivatives is denoted as $x_{\kappa_l} \in \mathbb{R}^3$. The resulting state dynamics form a triple-integrator linear system. For such a constrained system, the fastest way to achieve κ_{max} is to use bang-bang control, e.g. [16]. The control strategy will proceed as follows[2].

1) Use bang-bang control to get from $\begin{bmatrix}\sigma & \gamma\end{bmatrix}^T = \begin{bmatrix}0 & 0\end{bmatrix}^T$ to $\begin{bmatrix}\sigma & \gamma\end{bmatrix}^T = \begin{bmatrix}\sigma_{max} & 0\end{bmatrix}^T$
2) Coast at $\sigma = \sigma_{max}$
3) Use bang-bang control to get from $\begin{bmatrix}\sigma & \gamma\end{bmatrix}^T = \begin{bmatrix}\sigma_{max} & 0\end{bmatrix}^T$ to $\begin{bmatrix}\sigma & \gamma\end{bmatrix}^T = \begin{bmatrix}0 & 0\end{bmatrix}^T$ at the precise time that $\kappa_l = \kappa_{max}$

To define the control, four parameters are first introduced.

- τ_{bb}: The time control is held constant during bang-bang maneuvers
- $\Delta\kappa_{bb}$: The change in curvature during a bang-bang maneuver
- $\Delta\kappa_{\sigma_{const}}$: The change in curvature during coast when $\sigma_l = \sigma_{max}$
- $t_{\sigma_{const}}$: The time at which coast ends

These parameters, and the resulting trajectory for x_{κ_l}, are shown in Figure 3. The control to achieve κ_{max} is stated in the following lemma.

[2]Note that only the convergence to κ_{max} is considered. To move from κ_{max} to zero or from zero to $-\kappa_{max}$ the control inputs need only be reversed.

Lemma 4: Given the dynamics for κ_l in (2) and associated constraints on σ_l and u_l, the following control will move the system from $x_{\kappa_l} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$ to $\begin{bmatrix}\kappa_{max} & 0 & 0\end{bmatrix}^T$ in the minimum time.

$$u_l(t) = \begin{cases} u_{max} & t \in [0,\ \tau_{bb}) \\ -u_{max} & t \in [\tau_{bb},\ 2\tau_{bb}) \\ 0 & t \in [2\tau_{bb},\ t_{\sigma_{const}}) \\ -u_{max} & t \in [t_{\sigma_{const}},\ t_{\sigma_{const}} + \tau_{bb}) \\ u_{max} & t \in [t_{\sigma_{const}} + \tau_{bb},\ t_{\sigma_{const}} + 2\tau_{bb}) \end{cases} \tag{9}$$

where

$$\tau_{bb} = \sqrt{\frac{\sigma_{max}}{u_{max}}},\ t_{\sigma_{const}} = \frac{\Delta\kappa_{\sigma_{const}}}{\sigma_{max}} + 2\sqrt{\frac{\sigma_{max}}{u_{max}}} \tag{10}$$

and

$$\Delta\kappa_{bb} = u_{max}\left(\frac{\sigma_{max}}{u_{max}}\right)^{\frac{3}{2}},\ \Delta\kappa_{\sigma_{const}} = \kappa_{max} - 2\Delta\kappa_{bb} \tag{11}$$

Proof: It is important to recall the constraints, $|u_l| \leq u_{max}$, $|\sigma_l| \leq \sigma_{max}$, and $|\kappa_l| \leq \kappa_{max}$. Bang-bang control has two control intervals. In each interval, the control is held constant at an extreme. Due to the symmetry in the upper and lower constraints, these intervals are equivalent (τ_{bb} represents the length of a single time interval).

To find the switching times and curvature change values, the solution to a linear, time-invariant (LTI) system is employed, e.g., [17]. Namely, given $\dot{x}(t) = Ax(t) + Bu_{const}$, the solution at time t for $x(t)$ is given by the equation

$$x(t) = \underbrace{\exp^{A(t-t_0)}\, x(t_0)}_{\text{zero-input}} + \underbrace{\int_{t_0}^{t} \exp^{A(t-\rho)}\, Bd\rho u_{const}}_{\text{zero-state}}. \tag{12}$$

In the case of $x_{\kappa_l} = \begin{bmatrix}\kappa_l & \sigma_l & \gamma_l\end{bmatrix}^T$, the exponential and integral portions can be written as

$$\begin{aligned} \exp^{A(t-t_0)} &= \begin{bmatrix} 1 & t-t_0 & \frac{1}{2}(t-t_0)^2 \\ 0 & 1 & t-t_0 \\ 0 & 0 & 1 \end{bmatrix} \\ \int_{t_0}^{t} \exp^{A(t-\rho)}\, Bd\rho &= \begin{bmatrix} \frac{1}{6}(t-t_0)^3 \\ \frac{1}{2}(t-t_0)^2 \\ t-t_0 \end{bmatrix} \end{aligned} \tag{13}$$

Assuming that the initial time is 0, τ_{bb} can be found by using (12) and (13) over the first two time intervals. The solution from the first time interval can be fed directly into the zero-input portion of the second time interval as:

$$\sigma(2\tau_{bb}) = \sigma_{max} = \left(\frac{\tau_{bb}^2}{2} + \tau_{bb}^2 - \frac{\tau_{bb}^2}{2}\right)u_{max}, \tag{14}$$

where the first two τ_{bb} terms come from propagating the solution of the first horizon through the zero-input portion on the second horizon. The amount of change in curvature over the bang-bang manuever, $\Delta\kappa_{bb}$, can then be solved for in a similar fashion using the solution for τ_{bb}.

The bang-bang control will be reversed to send σ_l back to zero by the end of the horizon. Over the reversed bang-bang control, the change in curvature will be the same as the as the original bang-bang control, resulting in $2\Delta\kappa_{bb}$ occurring

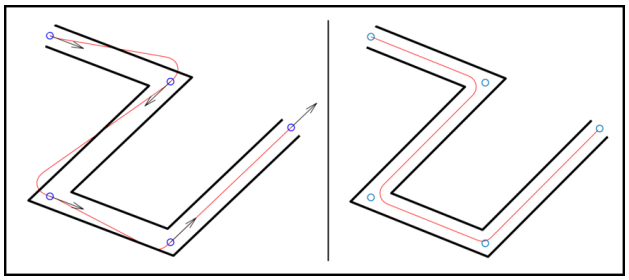

Fig. 4. Example of path points and a CCPath. The left depicts a shortest path between oriented waypoints and the right shows the path generated using position only waypoints.

over the combined bang-bang maneuvers. Thus, to achieve κ_{max} a total curvature change of $\kappa_{max} - 2\Delta\kappa_{bb}$ must occur during the constant σ_{max} interval. Since σ_l is constant over that interval, the curvature in that interval can be written as $\kappa_l(t) = \sigma_{max}(t - 2\tau_{bb}) + \Delta\kappa_{bb}$. The solution of $t_{\sigma_{const}}$ can be solved for from the following equation.

$$\begin{aligned}\kappa_l(t_{\sigma_{const}}) &= \kappa_{max} - \Delta\kappa_{bb} \\ &= \sigma_{max}(t_{\sigma_{const}} - 2\tau_{bb}) + \Delta\kappa_{bb}.\end{aligned} \tag{15}$$

■

One of the major benefits to using a CCTurn approach for the virtual leader path is that all possible combinations κ_l and σ_l are contained in the within the clothoid (both for achieving κ_{max} and κ_{min}). Thus, to see if the trajectory will violate the curvature restraints in Lemma 3, one only need to verify each offset over four clothoid intervals: $-\kappa_{max}$ to 0, 0 to κ_{max}, κ_{max} to 0, and 0 to $-\kappa_{max}$. Figure 6 depicts the verification for the example in Section V.

B. Virtual Leader Trajectory For Structured Environments

While Dubins paths and CCPaths can be used to find the shortest path that moves between a series of waypoints, often passing exactly through each waypoint is not completely necessary. For example, consider a series of waypoints that are taken from a graph representing building corridors. The waypoints may correspond to the center-points of corridor intersections. A planned path through the corridors to get from point A to point B could include multiple intermediary points. It may not be necessary to move directly through those points, but simply to move "near" them as depicted in Figure 4. Thus, this section presents a method to use clothoids to plan a trajectory from point A to B using intermediary points to define required turns, not points that have to be traversed exactly.

The nominal path is defined as an ordered set of n positions to be visited, denoted as Q, where

$$Q = \{q_1, q_2, \ldots, q_n\},$$

and the i^{th} waypoint is in $\mathbb{R}^2$.

As mentioned, the CCTurn forms a fundamental component in the shortest paths in CCPaths [13]. In this work, CCTurns are used to perform a turn between two subsequent line segments made from the ordered set of positions. The process for rapidly assembling the turns at each waypoint can be summarized as:

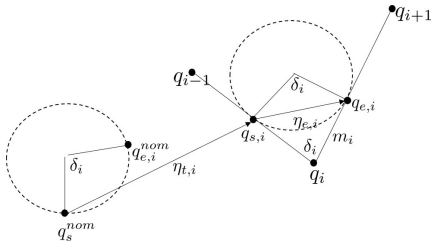

Fig. 5. An illustration of several of the parameters used to rotate and translation the CCTurn into position for maneuvering past the i^{th} waypoint.

1) Determine the deflection angle around the turn i, denoted δ_i. This defines the nominal value for the end waypoint, q_e^{nom}.
2) Rotate the CCTurn so the starting point tangent line is parallel to the angle from q_{i-1}-to-q_i, denoted $\psi_{i-1,i}$.
3) Find the vector that will point from $q_{s,i}$ to $q_{e,i}$.
4) Use the law of sines to solve for the distance between q_i and $q_{e,i}$.
5) Translate the CCTurn into place.

These parameters and their relations are shown in Figure 5. Defining the start and end points of each turn is sufficient for designing the trajectory as straight line segments will be used to connect the turns.

The *deflection angle* at waypoint i can be defined using the unit vectors parallel to line segments q_{i-1}-to-q_i and q_i-to-q_{i+1} as

$$\bar{\eta}_{i-1,i} = \frac{q_i - q_{i-1}}{d_{i-1,i}}, \; \bar{\eta}_{i,i+1} = \frac{q_{i+1} - q_i}{d_{i,i+1}} \tag{16}$$

where $d_{i-1,i} = ||q_i - q_{i-1}||$. The deflection angle about waypoint i can be defined as

$$\delta_i = acos(\bar{\eta}_{i-1,i}^T \bar{\eta}_{i,i+1}). \tag{17}$$

The deflection angle defines the nominal ending waypoint, $w_{e,i}^{nom}$, on the nominal CCTurn. To orient the CCTurn so that $w_{e,i}$ will align with $\bar{\eta}_{i,i+1}$, the nominal *CCTurn is rotated* by the orientation of $\bar{\eta}_{i-1,i}$, denoted as $\psi_{i-1,i}$. This allows for the direct calculation of the *vector that will point* from $q_{s,i}$ to $q_{e,i}$ as

$$\eta_{e,i} = R(\psi_{i-1,i})\big(q_{e,i}^{nom} - q_{s,i}^{nom}\big), \tag{18}$$

where $q_{s,i}^{nom}$ is typically at the origin.

Allowing θ_i to be the angle between $\eta_{e,i}$ and $\bar{\eta}_{i-1,i}$, the law of sines can be used to *find the distance* between $q_{e,i}$ and q_i as

$$m_i = \frac{\sin(\theta_i)}{\sin(\delta_i)} ||\eta_{e,i}||, \tag{19}$$

where $q_{e,i} = q_i + m_i \bar{\eta}_{i,i+1}$ Finally, the *translation vector* for the CCTurn can be calculated as

$$\eta_{t,i} = q_{e,i} - \eta_{e,i}. \tag{20}$$

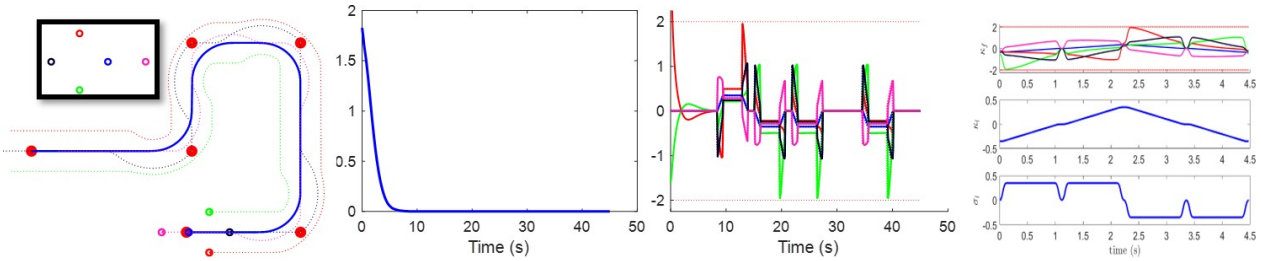

Fig. 6. From left to right, this figure shows the paths, the summed squared error of each vehicle's position to the desired, the curvature for each agent over the entire trajectory, and the curvature simulated over the a single pass through a clothoid. The upper left corner depicts the nominal formation. The lines are color coded between plots for each vehicle.

V. Example

The virtual structure motion was planned for and executed by a group of five simulated agents as depicted in Figure 6. The agents executed a formation defied by the following offset values:

$$\{\tau_i\} = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -1.5 \\ 1.5 \end{bmatrix}, \begin{bmatrix} -1.5 \\ -1.5 \end{bmatrix}, \begin{bmatrix} -3 \\ 0 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\}. \quad (21)$$

The maximum desired curvature for each of the five agents was $\kappa_{max_i} = 2.0$ which was achieved using a value of $v_l = 1$ and maximum values for the leader of $\kappa_{max} = 0.35$ and $\sigma_{max} = 0.35$ with results shown in Figure 6. The virtual leader trajectory was produced as described in Section IV-B using the series of waypoints shown as red circles resulting in the solid blue trajectory. Also shown in Figure 6 is the resulting curvature, which, once vehicles converge to their desired positions, shows that all curvature stays below the threshold as predicted by simulating over solely the clothoid.

VI. Conclusion

In this paper a moving formation control approach has been presented that considers the motion constraints of each agent while simultaneously enabling the rapid planning of the formation through a series of waypoints. The approach centers on the definition of the motion of the virtual leader. A virtual leader motion model was presented that allows for the direct consideration of the curvature constraints for both the movement of the structure as a whole as well as the individual curvature constraints of each vehicle. A new clothoid was created which respected the $\mathcal{C}^4$ conditions. By using the CCPaths generated using this clothoid to define the virtual structure motion, it was sufficient to evaluate a 4.5 second trajectory to determine if a trajectory consisting of an arbitrary number of waypoints would be feasible for execution.

References

[1] X. Zhou, W. Wang, T. Wang, X. Li, and T. Jing, "Continuous patrolling in uncertain environment with the UAV swarm," *PLoS ONE*, vol. 13, no. 8, pp. 1–29, 2018.

[2] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Magazine*, vol. 29, no. 1, pp. 9–19, 2008.

[3] T. Suzuki, T. Sekine, T. Fujii, H. Asama, and I. Endo, "Cooperative formation among multiple mobile robot teleoperation in inspection task," *Proceedings of the IEEE Conference on Decision and Control*, vol. 1, pp. 358–363, 2000.

[4] A. Soni and H. Hu, "Formation control for a fleet of autonomous ground vehicles: A survey," *Robotics*, vol. 7, no. 4, 2018.

[5] S. Wu, Y. Xia, M. Lin, and Y. Luo, "Leader-following Consensus and Trajectory Tracking for Nonholonomic Mobile Robots," in *Proceedings 2018 Chinese Automation Congress, CAC 2018*. Institute of Electrical and Electronics Engineers Inc., jan 2019, pp. 3678–3683.

[6] S. Sarno, M. D'Errico, J. Guo, and E. Gill, "Path Planning and Guidance Algorithms for SAR Formation Reconfiguration: Comparison between Centralized and Decentralized Approaches," *Acta Astronautica*.

[7] H. A. Almurib, P. T. Nathan, and T. N. Kumar, "Control and path planning of quadrotor aerial vehicles for search and rescue," in *Proceedings of the SICE Annual Conference*. Society of Instrument and Control Engineers (SICE), 2011, pp. 700–705.

[8] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Transactions on Control Systems Technology*, vol. 9, no. 6, pp. 777–790, 2001.

[9] G. Droge, "Distributed virtual leader moving formation control using behavior-based MPC," in *Proceedings of the American Control Conference*, vol. 2015-July, 2015.

[10] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006, vol. 9780521862.

[11] C. Ferrin, G. Droge, and R. Christensen, "Zero-error tracking for autonomous vehicles through epsilon-trajectory generation," *Journal of Intelligent and Robotic Systems*, pp. 1–9, 2020 (Under Review).

[12] L. E. Dubins, "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents," *American Journal of Mathematics*, vol. 79, no. 3, p. 497, 1957.

[13] T. Fraichard and A. Scheuer, "From Reeds and Shepp's to Continuous-Curvature Paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.

[14] R. Olfati-Saber, "Near-identity diffeomorphisms and exponential ϵ-tracking and ϵ-stabilization of first-order nonholonomic SE(2) vehicles," in *Proceedings of the American Control Conference*, vol. 6, 2002, pp. 4690–4695.

[15] Z. S. S. Murray, Richard M Li, *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[16] D. Liberzon, *Calculus of variations and optimal control theory: a concise introduction*. Princeton University Press, 2011.

[17] J. P. Hespanha, *Linear systems theory*. Princeton university press, 2018.