Projection-Based Inter-Agent Collision Avoidance in Dual Agent Systems

Vinod P. Gehlot and Mark J. Balas
Mechanical Engineering Department
Texas A&M University, College Station, Texas 77843, USA
email: {vinodgehlot, mbalas}@tamu.edu

Saptarshi Bandyopadhyay, Marco B. Quadrelli, and David S. Bayard
Jet Propulsion Laboratory
California Institute of Technology, Pasadena, California 91125, USA
email: {saptarshi.bandyopadhyay, marco.b.quadrelli, david.s.bayard}@jpl.nasa.gov

***Abstract*— Inter-agent collisions can occur in otherwise dynamically-stable (i.e., Lyapunov stable) dual-agent leader-follower systems. These inter-agent collisions between the leader and the follower happen during the transient phase of the system's evolution, although the steady-state behavior of the system is asymptotically/exponentially stable. Therefore, to avoid such inter-agent collisions, it is essential to control the relative error trajectory between the leader and the follower during the transient phase of the system's evolution. In this paper, we introduce a novel projection operator based model-reference control architecture that can mitigate impending inter-agent collisions by modifying the transient dynamics of relative trajectories. This controller augments the follower's baseline controller and consists of two essential components: a collision-free reference model based on the projection operator and a model reference tracking controller to guide the follower to follow the reference-model. This paper defines the concept of transient-instability in leader-follower systems, introduces collision mitigation controller architecture, and presents an illustrative example demonstrating its effectiveness.**

***Keywords*— Inter-agent collision avoidance; Collision mitigation; Multi-agent systems; Swarms; Interconnected systems; Motion planning; Projection operator.**

I. INTRODUCTION

In a dual agent leader-follower system, the leader is an independent entity; and the follower, as the name suggests, follows the leader at a specified separation distance. Relative position vectors and their associated dynamics are fundamental to the leader-follower formation maintenance. Therefore, the dynamic stability of the relative error dynamics is of paramount importance. We begin our discussion by introducing the mathematical preliminaries of dual agent leader-follower systems.

In the figure below, there are two identical agents — linear time-invariant (LTI) systems — identified by their indices $\mathbf{1}$, and $\mathbf{2}$, and their corresponding state vectors x_1, and x_2, respectively. They are in a leader-follower arrangement, with agent $\mathbf{1}$ as the leader, and agent $\mathbf{2}$ as the follower.

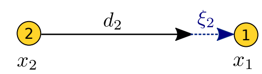

Fig. 1. Two Agent Leader-Follower Formation

The trajectories of the agents evolve according to the dynamics defined in (1). Here, $i \in \mathcal{I} = \{1, 2\}$, is the index of the two agents, x_i is the state vector of the agents that evolves in the $n-$dimensional state space $\mathbf{X} \subseteq \mathbf{R^n}$, $u_i \in \mathbf{U} \subseteq \mathbf{R^m}$, and $y_i \in \mathbf{Y} \subseteq \mathbf{R^p}$ are the $m-$dimensional input and $p-$dimensional output vectors of the agents, respectively. The tuple (A, B, C) are the set of appropriately sized matrices that model the system dynamics.

$$\dot{x}_i = Ax_i + Bu_i \tag{1a}$$
$$y_i = Cx_i \tag{1b}$$

Although not necessary, assume that in (1), the matrix A has at least a one-dimensional null space. This assumption allows for the arbitrary assignment of constraints on a partial set of an agent's state vector; without the use of a constant control effort.

In this formation, agent $\mathbf{1}$ is an independent entity, and agent $\mathbf{2}$, the follower, does not affect its dynamics. Agent $\mathbf{2}$ merely tracks agent $\mathbf{1}$ and maintains a spatial separation of $d_2 \in X$ using the control law

$$u_2 = G\left(y_1 - y_2 - Cd_2\right) = GC\xi_2. \tag{2}$$

Here, $\xi_2 \equiv x_1 - x_2 - d_2 \in \mathbf{R^n}$ is the relative error as measured from agent $\mathbf{2}$, and $G \in \mathbf{R^{m\times p}}$ is a stable closed loop gain matrix that drives the relative error trajectory $\xi_2(t) \to 0$ as $t \to \infty$. Note that d_2 is a vector quantity, and it resides in the null space of the system matrix A ($d \in \ker(A)$), therefore, it can include many more constraints besides distance. Equation (3) is the relative error dynamics of the two agent formation, and a suitable value of the gain matrix G will render the closed loop matrix $A_C \equiv (A - BGC)$ Hurwitz, thereby meeting the formation control objective. We say that agent $\mathbf{2}$ is "looking" at agent $\mathbf{1}$ when it takes control actions based on its sensor observation of agent $\mathbf{1}$ — like in (2).

$$\dot{\xi}_2 = (A - BGC)\xi_2 + Bu_1 \tag{3}$$

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-787-0

The two agent formation discussed thus far is quite common in many applications, and it sets the stage for discussion on transient stability in this paper. Examples of the formation just described include adaptive cruise control in vehicles, autonomous convoy or platooning, autonomous mid-air refueling, and formation flying spacecraft among others. Whatever the application may be, it is important to ensure that the dynamics in (3) are dynamically stable (i.e. Lyapunov stable). In our previous work [1]–[3], we explored the stability and adaptive control of several general formation geometries with large number of agents and arbitrary network topologies. In this paper, we use the two-agent leader-follower formation from Figure 1 to introduce the concept of transient instability in leader-follower systems, and a novel control architecture that mitigates transient stabilities. We define transient instability for the leader-follower formation as follows:

Definition 1.1 (Transient Stability): The dynamics of a two agent leader-follower formation is transient-stable if the relative error trajectory $\xi_2(t) \to 0$ as $t \to \infty$, and

$$\|\xi_2(t)\| \leq \xi_2^{max} \equiv (1-\alpha)\|d_2\|, \tag{4}$$

for all $t \in \mathbf{R}^+$. The scalar $\alpha \in [0, 1)$ describes a safety perimeter around agent $\mathbf{2}$.

In other words, we say that the two agent system is transient-stable if 1) the relative error trajectory is asymptotically stable, and 2) the agent trajectories evolve collision free. The problem of collision avoidance in formation and swarms is a thoroughly studied subject in the control and robotics literature, but many questions, particularly that of transient stability, still remain unanswered.

When it comes to collision avoidance algorithms in autonomous systems, the paper by Ames et. al [4] is noteworthy. They present a control barrier function based Quadratic Programming (QP) algorithm, that the follower continuously executes to generate collision free trajectories, all the while meeting asymptotic stability of the relative trajectories. And, since the control inputs are generated optimally, we know that the trajectories will be unique. The recent survey by Rossi et. al [5] offers comprehensive outlook on the current state-of-the-art multi-agent coordination and control algorithms. Based on this survey, the vast array of coordination and control algorithms can be classified into two broad categories: predictive and reactive algorithms. In predictive algorithms, optimization based path planning algorithms determine collision free trajectories for the agents to follow. Well known predictive algorithms include Optimal Reciprocal Collision Avoidance (ORCA), and Model Predictive Control and Sequential Convex Programming (MPC-SCP). Reactive algorithms, on the other hand, accomplish collision avoidance on an ad-hoc basis; when a safety-perimeter violation occurs, imminent collision is avoided by recomputing the motion planning algorithm. Voronoi-based [6], and Artificial Potential Functions (APF) [7] are two examples of reactive algorithms. With regards to stability, predictive algorithms, in general, can guarantee asymptotic stability but not collision avoidance, and reactive algorithms, can guarantee collision avoidance, but not asymptotic stability. It is worth mentioning that no algorithm mentioned in [5] can guarantee transient-stability.

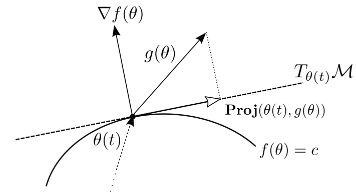

Fig. 2. Projection Operator in Action

Unlike the methods thus described, in this paper, we introduce a smooth, Lipschitz continuous method for collision avoidance that does not require the controller to solve optimization problems continuously in realtime. Moreover, this approach can satisfy dynamic stability and collision avoidance simultaneously. In the proposed method, we introduce a novel reference model for the follower that uses the projection operator to modify the drift vector field $(A - BGC)\xi_2 + Bu_1$ in (3) of the relative error vector $\xi_2(t)$ to generate transient-stable relative trajectories. Since directly differentiating $\xi_2(t)$ can induce unwanted noise into the feedback loop, we instead propose a Luenberger estimator of the form

$$\dot{\hat{\xi}}_2 = (A - BGC)\hat{\xi}_2 + L(\hat{y}_2 - y_2) \tag{5a}$$

$$\hat{y}_2 = C\hat{\xi}_2 \tag{5b}$$

to generate an estimate of the drift vector field in (3). This estimator is embedded within the projection operator to form the reference model, thereby generating relative error trajectories that satisfy the constraint (4). Finally, we augment the baseline control law in (2) with a type-1 tracking control law that tracks the transient stable trajectories generated by the reference model. It is the combination of the reference model, the reference model tracking controller, and the baseline relative error regulator that ensures transient stability in Definition 1.1. Figure 3 shows the proposed control methodology. We present our results in three sections. In Section II, we introduce the fundamentals of the projection operator and a few essential results without proof. In Section III, we introduce the transient instability mitigation architecture and discuss its various components, and present the main theoretical results. Finally, in Section IV, we use an illustrative example to demonstrate the effectiveness of the collision mitigation strategy presented in this paper.

II. The Projection Operator

The projection operator is among several methods in the convex analysis that can solve constrained convex optimization problems. In gradient descent iterations, the projection operator projects the gradient of the cost function onto the constraint manifold, limiting the solution to the convex set defined by the constraints. According to [8], Kreisselmeier and Narendra [9] were the first to use the projection operator to bound time-varying gains in adaptive control systems. And since then, it has been hugely popular in several adaptive

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-787-0

control algorithms [10] [8] [11]. Let $\theta \in \mathbf{R}^n$ be a state vector that evolves according to

$$\dot{\theta}(t) = g(\theta), \tag{6}$$

and suppose that we want the trajectory $\theta(t)$ to stay within a convex set with the boundary $f(\theta) = c$. We can accomplish this by constraining the dynamics in (6) with the projection operator dynamics

$$\begin{aligned}\dot{\theta}(t) &= \mathbf{Proj}(\theta, g(\theta)) \\ &= \begin{cases}\left(I - \frac{\nabla f (\nabla f)^T}{\|\nabla f\|^2} f(\theta)\right) g(\theta) & \text{if } f > 0 \text{ and } \theta^T \nabla f > 0 \\ g(\theta) & \text{otherwise.}\end{cases}\end{aligned} \tag{7}$$

The modification of the dynamics in (6) by (7) guarantees that $\theta(t)$'s trajectory will stay within the convex set

$$\Theta = \{\theta \in \mathbf{R}^n : f(\theta) \leq 0\}. \tag{8}$$

The projection operator achieves this by subtracting from the drift vector field $g(\theta)$, the component of $g(\theta)$ that is parallel to the gradient vector $\nabla f(\theta)$ (see Figure 2). Therefore, drift vector $\mathbf{Proj}(\theta, g(\theta))$ lies on the tangent plane $T_\theta \mathcal{M}$, where $\mathcal{M} = \{\theta : f(\theta) = c\}$ is the constraint manifold. The following is a significant lemma that is useful in stability proofs involving the projection operator.

Lemma 2.1 (Projection Inequality): Let θ^* be point in the interior of the convex set Θ, and let $\Gamma > 0$ be some positive definite and symmetric matrix, then for any other $\theta(t) \in \Theta$,

$$(\theta - \theta_*)^T \left(\Gamma^{-1} \mathbf{Proj}\left(\theta, \Gamma g(\theta)\right) - g(\theta)\right) \leq 0. \tag{9}$$

For the proof of this inequality, please refer to [8]. In this section, we have given a concise summary of the application of the projection operator applied to dynamical systems. For a more thorough treatment on this subject, we ask the reader to refer to [12] [11] [8].

III. The Collision Avoidance and Dynamic Stability Architecture

As discussed earlier, the follower implements the output feedback control law (2) to maintain the separation vector d_2 from the leader, resulting in the closed-loop relative error dynamics (3). Also, as discussed before, we know that the trajectory $\xi_2(t)$ can violate the transient stability criteria in Definition 1.1, even though the closed-loop matrix $A_c \equiv A - BGC$ is Hurwitz, and the input $u_1(t)$ is bounded. Therefore, there is a need to manage and modify the transient dynamics of the vector $\xi_2(t)$ to prevent inter-agent collisions, and hence satisfy the transient stability criterion. One — and possibly the most straightforward — approach, would be to specify the transient stability criteria directly in terms of frequency or time domain specifications. Then, compute the feedback gain G using an appropriate method from classical/modern control theory. This approach can work quite well for multi-agent systems with low cardinalities, like the leader-follower system discussed in this paper. Still, as the number of agents gets more substantial, and communication and sensing topologies can get complicated, it can be challenging to compute gains for the individual agents.

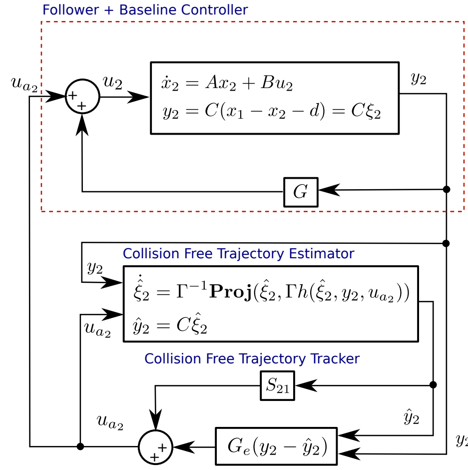

Fig. 3. Follower Collision Mitigation Architecture

Moreover, in [3], we show that eigenvalues of the Laplacian matrix of the network digraph of the formation can inadvertently scale the baseline feedback gains G, thereby causing stability issues in the formation geometry, and all the while degrading the controller performance. Motivated by these issues, and addressing the problem of transient stability in formations with a large number of agents, we introduce the control architecture shown in Figure 3, that can simultaneously satisfy dynamic stability as well as avoid inter-agent collisions. The proposed architecture consists of two essential controller subsystems: the collision-free (CF) estimator, and the collision-free (CF) trajectory tracker. Combined, the two controller subsystems augment the baseline control law (2) to guarantee transient stability. The CF subsystems generate the control vector u_{a_2}, and adds to the baseline control law as

$$u_2 = GC\xi_2 + u_{a_2}. \tag{10}$$

At its core, the CF estimator has a drift vector field

$$h(\hat{\xi}_2, y_2, u_{a_2}) = A_c\hat{\xi}_2 - Bu_{a_2} + L(\hat{y}_2 - y_2), \tag{11}$$

which is the structure of a standard Luenberger observer. Provided the pair (A_c, C) is observable, and the closed-loop estimator matrix $A_c + LC$ is Hurwitz, the estimated state $\hat{\xi}_2(t)$ will converge exponentially to the actual state $\xi_2(t)$. We enclose the estimator drift vector field $h(\hat{\xi}_2, y_2, u_{a_2})$ inside the projection operator. Therefore, we have

$$\begin{aligned}\dot{\hat{\xi}}_2 &= \Gamma^{-1}\mathbf{Proj}(\hat{\xi}_2, h(\hat{\xi}_2, y_2, u_{a_2})) \\ &= \Gamma^{-1} \begin{cases}(I - \frac{\nabla f (\nabla f)^T}{\|\nabla f\|^2} f(\theta))h & \text{, if } f > 0 \text{ and } \hat{\xi}^T \nabla f > 0 \\ h & \text{, otherwise.}\end{cases}\end{aligned} \tag{12}$$

Here,

$$f(\hat{\xi}_2) \equiv \frac{(1+\varepsilon)\|\hat{\xi}_2\|^2 - \|\hat{\xi}_2^{\max}\|^2}{\varepsilon\|\hat{\xi}_2^{\max}\|^2}, \tag{13}$$

which is the constraint vector for the projection operator, and $\xi_2^{\max}$ is the bound on the relative error trajectory from the

 ISBN: 978-1-61208-787-0

transient stability criterion in Definition 1.1. $\varepsilon > 0$ is a scalar that creates a smooth boundary by defining two concentric convex sets

$$\Omega_0 \equiv \{\hat{\xi}_2 \in \mathbf{R}^n : \|\hat{\xi}_2\| \leq \frac{\|\xi_2^{\max}\|}{\sqrt{1+\varepsilon}}\}, \text{ and} \tag{14a}$$

$$\Omega_1 \equiv \{\hat{\xi}_2 \in \mathbf{R}^n : \|\hat{\xi}_2\| \leq \|\hat{\xi}_2^{\max}\|\}. \tag{14b}$$

From Lemma 11.4 in [8], for any trajectory $\hat{\xi}_2(t_i) = \hat{\xi}_{t_i} \in \Omega_0$, the projection operator guarantees that for all $t > t_i$, $\hat{\xi}_2(t) \in \Omega_1$. Hence, the estimated state vector $\hat{\xi}_2(t)$ satisfies the transient characteristics and generates CF trajectories. Moreover, in the following result, we show that even though the estimator dynamics are enclosed within the projection dynamics, the estimated state $\hat{\xi}_2(t)$ exponentially converges to the actual state vector $\xi_2(t)$.

Theorem 3.1 (Collision Free Estimator Stability): The error trajectory $e(t) \equiv \hat{\xi}_2 - \xi_2$, of the estimator dynamics

$$\begin{aligned} \dot{\hat{\xi}}_2 &= \Gamma^{-1}\mathbf{Proj}(\hat{\xi}_2, \Gamma h(\hat{\xi}_2, y_2, u_{a_2})) \\ \hat{y}_2 &= C\hat{\xi}_2, \end{aligned} \tag{15}$$

with $h(\hat{\xi}_2, u_{a_2}, y_2) \equiv A_c\hat{\xi}_2 - Bu_{a_2} + L(\hat{y}_2 - y)$, is exponentially stable.

Proof: For notational convinence, let $z \equiv (\hat{\xi}_2, y_2, u_{a_2})$. Taking the time derivative of $e(t)$, we have

$$\begin{aligned} \dot{e}(t) &= \dot{\hat{\xi}}_2(t) - \dot{\xi}_2(t) \\ &= \Gamma^{-1}\mathbf{Proj}(\hat{\xi}_2, \Gamma h(z)) - h(z) + (A_c + LC)e. \end{aligned}$$

Let $V(e)$ be the positive definite and decresent Lyapunov function associated with the estimator error trajectory $e(t)$, and defined by

$$\lambda_{\min}(\Gamma)\|e\|^2 \leq V(e) \equiv \frac{1}{2}e^T\Gamma e \leq \lambda_{\max}(\Gamma)\|e\|^2.$$

By taking the time derivative of $V(e)$ and using Lemma 2.1, we obtain

$$\begin{aligned} \dot{V}(e) &= e^T\Gamma\dot{e} \\ &= \underbrace{e^T\Gamma\left(\Gamma^{-1}\mathbf{Proj}(\hat{\xi}_2, \Gamma h(z)) - h(z)\right)}_{\leq 0} \\ &\quad + e^T\Gamma(A_c + LC)e \\ &\leq e^T\Gamma(A_c + LC)e. \end{aligned}$$

The closed-loop estimator matrix $A_c + LC$ is Hurwitz, therefore, for a given $Q > 0$, there exists a matrix $\Gamma > 0$ that solves the Lyapunov matrix equation

$$\Gamma(A_c + LC) + (A_c + LC)^T\Gamma = -Q.$$

Therefore,

$$\dot{V}(e) \leq -\frac{1}{2}e^TQe \leq -\frac{1}{2}\lambda_{\min}(Q)\|e\|^2 \leq -\underbrace{\frac{\lambda_{\min}(Q)}{2\lambda_{\max}(\Gamma)}}_{=\mu}V(e)$$

$$\Rightarrow \dot{V}(e) + \mu V(e) \leq 0.$$

Using the integrating factor $e^{\mu t}$, we have

$$\int_0^\tau e^{\mu t}(\dot{V} + \mu V) \leq \int_0^\tau e^{\mu t}\ 0 \ \Rightarrow V(e(\tau)) \leq e^{-\mu\tau}V(0)$$

Further,

$$V(0) \leq \lambda_{\max}(\Gamma)\|e(0)\|^2, \text{ and}$$

$$\sqrt{\lambda_{\min}(Q)}\ \|e(\tau)\| \leq V^{\frac{1}{2}}(e(\tau)) \leq e^{-(\mu/2)\tau}\ V^{\frac{1}{2}}(0).$$

Therefore,

$$\|e(\tau)\| \leq \underbrace{\sqrt{\frac{\lambda_{\max}(\Gamma)}{\lambda_{\min}(\Gamma)}}}_{=K_0}e^{-(\mu/2)\tau}\ \|e(0)\| = K_0e^{-(\mu/2)\tau}\|e(0)\|$$

■

The CF tracker subsystem generates the input vector u_{a_2} so that the follower can track the transient stable relative error trajectories produced by the CF estimator. Fundamentally, the CF tracker is a servomechanism problem, and there are several options for its structure. A PID/LQR controller based type-1 tracker is a perfectly reasonable option. With a view on applying the CF tracker/estimator to more extensive and complex swarms, we opt for an output-feedback model-reference based tracking architecture. A model-reference based approach can readily accept time-varying adaptive gains, which allows for the automation of gain determination in complex formation structures.

Assumption: The output state vector $\hat{\xi}_2(t)$ of the CF estimator can be expressed as a linear combination of basis vectors $\phi_i(t)$ with some coefficient matrix L.

With this assumption, we can write the CF estimator output in the command generator form

$$\begin{aligned} \hat{\xi}_2(t) &= L\phi(t) \\ \hat{y}_2 &= C\hat{\xi}_2(t) \end{aligned} \tag{16}$$

where, $\phi(t) = (\phi_1(t), \ldots, \phi_p(t))^T$, is a column vector of tracking signal basis functions. According to [13], (16) is equivalent to the dynamical system

$$\begin{aligned} \dot{\eta}(t) &= F\eta(t) \\ \hat{y}_2 &= C\eta(t). \end{aligned} \tag{17}$$

The following result is the stability proof of the CF tracker subsystem based on the output-feedback model-reference tracking controller.

Theorem 3.2 (Reference Model Tracking): The follower LTI system

$$\dot{x}_2 = Ax_2 + Bu_2 \tag{18a}$$

$$y_2 = C(x_1 - x_2 - d_2) = C\xi_2, \tag{18b}$$

with the tracking control law

$$u_{a_2} = G_e(y_2 - \hat{y}_2) + S_2\hat{y}_2, \tag{19}$$

for appropriately sized gain matrices G_e, and S_2, will track the command generator reference (CGR) system

$$\hat{\xi}_2(t) = L\phi(t) \quad (20a)$$
$$\hat{y}_2 = C\hat{\xi}_2(t), \quad (20b)$$

such that, both the output relative error vector $e_y(t) \equiv \hat{y}_2(t) - y_2(t)$, and the error between the relative error vectors $e(t) \equiv \hat{\xi}_2(t) - \xi_2(t)$, are driven to the origin exponentially. Provided, the transmission zeros and the poles of the relative error dynamics $\xi_2(t)$ are distinct.

Proof: Let ξ_2^*, u_2^*, and y_2^*, be the ideal trajectory, ideal inputs, and ideal output, respectively. The ideal trajectory, and its associated vectors evolve according to the dynamics

$$\dot{\xi}_2^* = A_c\xi_2^* + Bu_2^* \quad (21a)$$
$$y_2^* = C\xi_2^* = \hat{y}_2. \quad (21b)$$

That is, the ideal trajectory ξ_2^* evolves so that it tracks the CGR system output exactly. Let S be a matrix that relates the ideal trajectories to the CGR system, defined by

$$\begin{pmatrix} \xi_2^* \\ u_i^* \end{pmatrix} = S \begin{pmatrix} \hat{y}_2 \\ 0 \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \begin{pmatrix} \hat{y}_2 \\ 0 \end{pmatrix}. \quad (22)$$

The CGR system can also be expressed in the form of an equivalent LTI system

$$\dot{\eta}(t) = F\eta(t) \quad (23a)$$
$$\hat{y}_2 = C\eta(t) \quad (23b)$$

Using (21), and (23), and taking the time derivative of (22), we have matrix equations

$$(A_cS_{11} - BS_{21})C = S_{11}CF \quad (24a)$$
$$CS_{11} = I, \quad (24b)$$

which are the matching conditions for the tracking problem. The implementation of the tracking control law requires the solution to the matrices S_{11} and S_{21}. According to [14], the solution to the matrices S_{11} and S_{21} exists, provided the transmission zeros of the reference model, and the poles of the plant are distinct, which by assumption is true. We now define the tracking error $\Delta\xi_2 \equiv \xi_2^* - \xi_2$. Taking its time derivative, we have

$$\Delta\dot{\xi}_2 = A_c\Delta\xi_2 - B\Delta u_{a_2} \quad (25a)$$
$$\Delta y_2 = C(\xi_2^* - \xi_2) = y_2^* - y_2 = \hat{y}_2 - y_2, \quad (25b)$$

where, $\Delta u_{a_2} = u_{a_2}^* - u_{a_2}$. Let $\Delta u_{a_2} = G_e\Delta y$, so that, the closed-loop tracking error system

$$\Delta\dot{\xi}_2 = (A_c - BG_eC)\Delta\xi_2 = \tilde{A}_c\Delta\xi_2 \quad (26a)$$
$$\Delta y_2 = C\Delta\xi_2 \quad (26b)$$

is exponentially stable. Therefore, $\Delta\xi_2 \to 0$ as $t \to \infty$, which implies $\xi_2(t) \to \hat{\xi}_2(t)$ as $t \to \infty$. Also, since

$$\begin{aligned} \Delta u_{a_2} &= G_e\Delta y \\ \Rightarrow u_{a_2}^* - u_{a_2} &= G_e(\hat{y}_2 - y_2) \\ \Rightarrow u_{a_2} &= G_e(y_2 - \hat{y}_2) + S_{21}\hat{y}_2 \end{aligned}$$

■

The proposed CF tracking controller tracks the collision-free trajectory generated by the CF estimator. But, only if the follower dynamics are deterministic. In practice, an adaptive control law would determine the gain matrices G_e and S_{21}, which would result in robust tracking performance similar or better to that of integral action in type-1 servomechanisms.

IV. Simulation Result

We use the two simulation runs: Run 1 with CF tracker disabled, and Run 2 with CF tracker enabled, to demonstrate the functioning, and also highlight a few limitations of the CF estimator and tracker subsystem. For the two simulation runs, the leader and follower are double integrator agents with the model $\ddot{x} = u$. The leader and the follower initially rest at their specified separation of 5 meters. After a specified time of about 20 seconds, a position and velocity disturbance is applied to the leader using the input vector u_d, as shown in Figure 5. $\dot{x}_1 = Ax_1 + Bu_1 + u_d$ is the structure of the leader's dynamics with the disturbance vector input u_d. In Figures 6 and 7, the solid horizontal lines named Relative Error Upper limit, and Relative Error Lower limit, reflect the relative error bounds in (3). The dashed lines represent the beginning of the soft constraint boundary for the projection operator; the estimates from the projection operator are allowed to exceed the soft boundary temporarily. Figure 4 shows timed sequence of the leader and follower positions for the two simulation runs. Stiffness in the Ordinary Differential Equations (ODEs) can arise from the projection-based dynamics in the follower reference model, and it can lead to slow convergence when using popular explicit ODE solvers such as the Dormand-Prince RK5(4) [15]; usually, an implicit solver is a better solution. For the simulation runs in this paper, we used the LSODA solver (which is roughly equivalent to ode15s in MATLAB) from SciPy [16], which is a scientific library for Python, and it can automatically switch between implicit and explicit methods to handle stiff ODEs.

Figure 6 shows the results for the simulation run with the CF tracker turned off. There are two important outcomes. First, as expected, the relative error trajectory violates the upper and lower limits for collision avoidance. Second, the CF estimator generated trajectory saturates and never exceeds the transient stability bounds. However, when the actual relative error is within the bounds, the estimator perfectly tracks the actual trajectory. The outcomes of this simulation demonstrate the predictions of Theorem 3.1.

Figure 7 shows the results for the simulation run with the CF tracker turned on. Right away, we see that both the true and the estimated relative trajectories never exceed the transient instability bound; therefore, no collisions occur. Since the CF estimator on the follower does not have access to the leader's input information, whenever the vector u_d is non-zero, a near constant bias/DC component exists between the actual and estimated relative trajectories. The magnitude of the disturbance vector components is deliberately limited to 2 [m] or [m/s]. If the magnitude of the disturbance is any higher, the bias between the actual and estimated trajectories increases and degrades the tracking performance, which can result in inter-agent collisions. It can be shown that the sharing of control information (i.e., $u_1(t)$) from the leader to the follower will resolve the aforementioned issues.

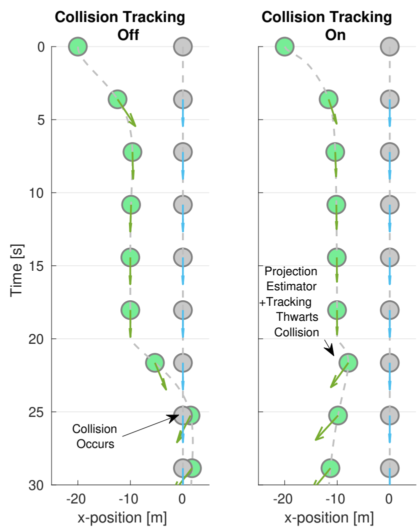

Fig. 4. Sequences of Leader-Follower Motion from Simulation Runs

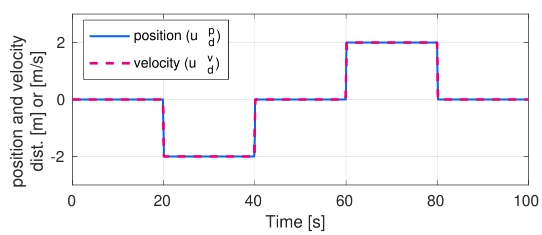

Fig. 5. Leader Disturbance Vector

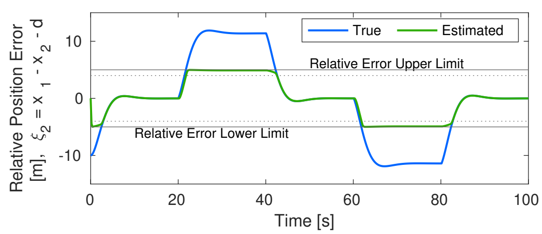

Fig. 6. Follower Trajectory with Tracking Off

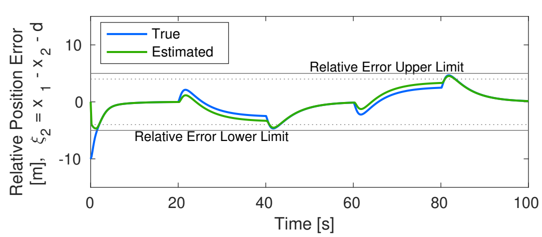

Fig. 7. Follower Trajectory with Tracking On

V. Conclusion

We have developed a novel controller architecture that addresses the problem of transient instabilities in dual-agent formations. We also presented the preliminary theoretical results on the stability of the collision-free estimator and the tracker subsystems. The proposed controller is particularly attractive due to its simplicity and its ability to guarantee both dynamic and transient stability. Future work will focus on 1) developing a comprehensive analytical framework that will investigate robustness to external noise and disturbances. And 2) on generalizing the architecture to larger and more complex formation structures.

Acknowledgment

Part of this research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. ©2020 California Institute of Technology. Government sponsorship acknowledged.

References

[1] V. P. Gehlot and M. J. Balas, "An evolving systems approach to the stable operation of dynamic formations and swarms of autonomous vehicles in a disruptive environment," in *2018 AIAA SPACE and Astronautics Forum and Exposition, AIAA SPACE Forum*. American Institute of Aeronautics and Astronautics, 2018.

[2] ——, "A theoretical framework for the stable operation of autonomous spacecraft formations in the hill-clohessy-wiltshire frame," in *IEEE Southeast Conference*. IEEE, 2019.

[3] V. P. Gehlot, M. J. Balas, and S. Bandyopadhyay, "Dynamic stability and adaptive control of networked evolving formations with weak nonlinearities," in *AIAA Scitech Forum*. AIAA, 2020.

[4] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *53rd IEEE Conference on Decision and Control*, 12 2014, pp. 6271–6278.

[5] F. Rossi, S. Bandyopadhyay, M. Wolf, and M. Pavone, "Review of multi-agent algorithms for collective behavior: a structural taxonomy," *IFAC-PapersOnLine*, vol. 51, no. 12, pp. 112 – 117, 2018, iFAC Workshop on Networked & Autonomous Air & Space Systems NAASS 2018.

[6] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, online collision avoidance for dynamic vehicles using buffered voronoi cells," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1047–1054, 2017.

[7] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, vol. 2, 1985, pp. 500–505.

[8] E. Lavretsky and K. Wise, *Robust and Adaptive Control: With Aerospace Applications*, ser. Advanced Textbooks in Control and Signal Processing. Springer London, 2012.

[9] G. Kreisselmeier and K. Narendra, "Stable model reference adaptive control in the presence of bounded disturbances," *IEEE Transactions on Automatic Control*, vol. 27, no. 6, pp. 1169–1175, 1982.

[10] N. Hovakimyan and C. Cao, *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*, ser. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2010.

[11] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Mineola, New York: Dover Publication, Inc., 2012, p. 586.

[12] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*, ser. Advances in Design and Control. Society for Industrial and Applied Mathematics, 2006.

[13] C. D. Johnson, "Disturbance-accommodating control; an overview," in *1986 American Control Conference*, 1986, pp. 526–536.

[14] M. J. Balas and S. A. Frost, "Adaptive Tracking Control for Linear Infinite Dimensional Systems," ser. Smart Materials, Adaptive Structures and Intelligent Systems, vol. Volume 2: Modeling, Simulation and Control; Bio-Inspired Smart Materials and Systems; Energy Harvesting, 09 2016. [Online]. Available: https://doi.org/10.1115/SMASIS2016-9098

[15] J. Dormand and P. Prince, "A family of embedded runge-kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19 – 26, 1980.

[16] P. Virtanen et al., "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.