

# Leaderless Formation Control using Dynamic Extension and Sliding Control <sup>\*</sup>

Zhibo Zheng<sup>\*</sup> Stephen C. Spry<sup>\*\*</sup> Anouck R. Girard<sup>\*\*\*</sup>

<sup>\*</sup> *Mechanical Engineering at Columbia University, New York, NY  
10027, USA (e-mail: zz2115@columbia.edu)*

<sup>\*\*</sup> *LAM Research Corporation, Fremont, CA 94538*

<sup>\*\*\*</sup> *Aerospace Engineering at the University of Michigan, Ann Arbor,  
48109, USA (phone: 734-647-4692; e-mail: anouck@umich.edu)*

---

**Abstract:** We present a design of a leaderless formation controller for networked vehicle systems, which uses concepts from sliding mode control and dynamic extension. A single gain varies the importance of the relative and absolute position terms, allowing for tight or loose formations. This approach is proven mesh stable. Applications include formation flying of Unmanned Air Vehicles as well as possible extensions to satellites or Autonomous Underwater Vehicles.

Keywords: Autonomous mobile robots, Co-operative control, Decentralized control, Nonlinear control, Sliding-mode control.

---

## 1. INTRODUCTION

A large body of research has been published in recent years about motion control and collaborative control of networked unmanned vehicles. The state of the science and practice in developing networked unmanned vehicle systems requires new developments (for example, control systems with feedback loops through communications channels, dynamic systems formed of several parts that are not physically coupled, etc. . . ), and breakthroughs are needed to effect highly dependable systems, with focus on real-world demonstrations targeting the avionics, maritime and automotive industries.

There are multiple applications for formation maneuvering of multiple vehicles, such as satellite interferometry, gradient/environmental estimation using vehicle formations, etc. . . In addition, moving vehicles as a group simplifies path planning, as a single path can be specified for the group with an associated formation shape (Spry and Hedrick [2004]).

We propose a controller that allows a group of vehicles to maintain a given formation shape using concepts from sliding mode control and dynamic extension. The controller is “leaderless”, that is, no specific vehicle is designated as the leader, which provides redundancy and fault tolerance. In addition, a single gain allows to vary the weight between the absolute and relative position terms of the controller, allowing to place either more emphasis on going to a certain location or more emphasis on maintaining a particular formation shape, and less on reaching an exact location.

The remainder of this paper is organized as follows. Section 2 presents a literature review in the field of control of networked vehicle systems. Section 3 details the leaderless

controller design and proves our approach is mesh stable. Section 4 presents simulation results. Finally, Section 5 draws some conclusions and discusses future work.

## 2. LITERATURE REVIEW

The control of networked vehicle systems has presented a new challenge to control theory. The challenge comes from the distributed nature of the problem (Murray et al. [2003]). For example, in networked multi-vehicle systems, information and commands are exchanged among multiple vehicles, and the roles, relative positions and dependencies of those vehicles change during operations. The control and computer science communities address this challenge in the context of distributed hybrid systems, and contribute complementary views and techniques (see Varaiya et al. [2001]) for an overview of research on distributed hybrid systems).

Many different control strategies have been considered for the coordinated control of multiple vehicle systems. There is no standard taxonomy for the classification of these control and coordination strategies. Generally speaking the problem of the control of formations of vehicles is a single-stage problem as far as there are no modifications to the topology of the formation. Possible approaches for the control of formations of vehicles include leader-follower approaches (Hedrick et al. [1994], Smith and Hadaegh [2002]) or string- and mesh-stable approaches (Swaroop and Hedrick [1996], Pant et al. [2002]).

Researchers working on platooning of passenger vehicles as a means to increase highway capacity without building new highways (Hedrick and Swaroop [August 1993]) first noticed that strings of automatically controlled vehicles exhibited “string instabilities”, i.e., disturbances in the front of the platoon were amplified as they were propagated upstream. Linear transfer function analysis (Swa-

---

<sup>\*</sup> This work was supported in part by United States Air Force, under grant number FA8650-07-2-3744.

roop and Hedrick [1996], Hedrick and Swaroop [August 1993], Sheikholeslam and Desoer [1992]) showed that these instabilities could be eliminated by the introduction of a common reference trajectory for all of the vehicles. If all of the vehicles in the platoon have knowledge of the lead vehicle's absolute velocity (Swaroop and Hedrick [1996]), then "weak string stability" can be achieved, i.e., no disturbance would ever be amplified as it traveled upstream in the platoon. Also, if all of the vehicles in the platoon had knowledge of the relative position error between themselves and the lead vehicle, then "strong string stability" could be achieved, i.e., all downstream disturbances could be geometrically attenuated as they traveled upstream in the platoon (Swaroop and Hedrick [1996]). The lead vehicle information needs to be communicated to all of the vehicles via a wireless communication link.

The concept of string stability is extended to 3-D configurations in (Pant et al. [2002]) and the term "mesh stability" is used to denote the property of disturbance attenuation in multi-dimensions. The paper concentrates on minimizing the communication requirements to achieve mesh stability and analyses systems with "look-ahead" sensor information that could be communicated or sensed directly.

Other possible approaches include virtual structures (Beard et al. [2001], Tillerson et al. [2003], Leonard and Fiorelli [2001]), rigid graphs (Olfati-Saber and Murray [2002]) and potential methods (Leonard and Fiorelli [2001], Zohdi [2003]). Nearest neighbor rules, Lyapunov theory, graph techniques and non-smooth control results are used to study how multiple agents eventually move in the same direction despite the absence of centralized coordination and despite the fact that each agent's set of nearest neighbors changes with time as the system evolves (Tanner et al. [2003a,b]).

### 3. LEADERLESS FORMATION CONTROLLER

We consider the design of mesh-stable controllers for vehicle formations. For the sake of example, we will consider a simple model for small fixed-wing UAVs operating in a 2-D plane. A related, mesh-stable approach for helicopters can be found in (Pant et al. [2001]). A design for fixed wing aircraft can be found in (Spry and Hedrick [2004]).

In our development, we have used a kinematic model to represent the aircraft. The reference frames are shown in Fig. 1.

$$\begin{cases} \dot{x} = u_1 \cos \psi + V_{wx} \\ \dot{y} = u_1 \sin \psi + V_{wy} \\ \dot{\psi} = u_2 \end{cases} \quad (1)$$

Here,  $V_{wx}$  and  $V_{wy}$  represent the velocity due to wind in both  $x$  and  $y$  directions. The directions  $x$  and  $y$  are in a frame that is fixed with respect to the ground. The control variables are  $u_1$  and  $u_2$ , the airspeed and turn rate. In the case of model aircraft it is common to keep the airspeed  $u_1$  within a fairly small range and use the turn rate  $u_2$  as the primary control variable. We are restricting ourselves to aircraft moving in the horizontal plane for now. This is a good assumption, as, at this time, the antennas available to be mounted on small UAVs have strong directional patterns, and wireless communications

between aircraft are more reliable if all aircraft are in the same geometric plane. With a 3D model, the controller design can be adapted to the 3D case. For the purpose of controller design, we set the wind velocities to zero.

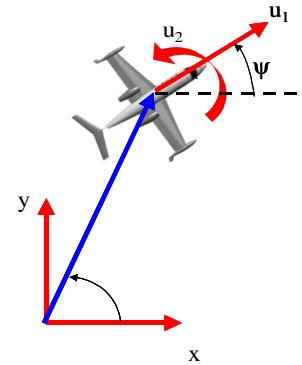


Fig. 1. Variable definition for kinematic, 2D UAV, bird's-eye view with z-axis pointing outwards from the paper.

#### 3.1 System Analysis

Define our state vector to be

$$\mathbf{x} = [x \ y \ \psi]^T$$

and rewrite the system dynamics in the form of

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g_1 u_1 + g_2 u_2.$$

We have

$$f(\mathbf{x}) = 0, \quad g_1 = \begin{pmatrix} \cos \psi \\ \sin \psi \\ 0 \end{pmatrix} \quad \text{and} \quad g_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Thus the controllability matrix

$$C = \begin{pmatrix} \cos \psi & 0 & \sin \psi \\ \sin \psi & 0 & -\cos \psi \\ 0 & 1 & 0 \end{pmatrix}$$

has full rank everywhere, which implies that the system is locally accessible everywhere. Furthermore, since  $f(\mathbf{x}) = 0$  (free dynamics system), the system is controllable.

Choose  $z = h(\mathbf{x}) = \begin{pmatrix} x \\ y \end{pmatrix}$ , we have the observability matrix as

$$O = dG = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -u_1 \sin \psi \\ 0 & 0 & u_1 \cos \psi \\ 0 & 0 & -u_1 (\cos \psi) u_2 \\ 0 & 0 & -u_1 (\sin \psi) u_2 \end{pmatrix}.$$

$O$  has rank 3 if  $u_1 \neq 0$ , which indeed holds for any kind of realistic flight, thus the system is observable.

#### 3.2 Sliding Mode Controller with Dynamic Extension – Single Vehicle Case

We start by defining our position vector  $\eta$ , and our desired position vector  $\eta_d$ .

$$\eta = \begin{pmatrix} x \\ y \end{pmatrix} \quad (2)$$

$$\eta_d = \begin{pmatrix} x_d \\ y_d \end{pmatrix} \quad (3)$$

We define  $e$  to be the position error vector,

$$e = \eta - \eta_d \quad (4)$$

We select our sliding surface to be:

$$S = \dot{e} + Ke \quad (5)$$

If  $S$  goes to zero, then we have exponentially decreasing error dynamics, at a rate set by  $K$ , which is a positive definite matrix that we select as a control parameter.

$$S \rightarrow 0 \Rightarrow \dot{e} \rightarrow -Ke \Rightarrow e \rightarrow 0 \quad (6)$$

Rewriting the equation for  $S$ , we have:

$$S = \dot{\eta} - \dot{\eta}_d + Ke \quad (7)$$

Defining some notation, we set:  $V = \dot{\eta}$  and  $V_R = \dot{\eta}_d - Ke$ . We then have:

$$S = V - V_R \quad (8)$$

We now need to focus on finding a strategy to make the velocity  $V$  go to  $V_R$ . If  $V=V_R$ , then  $S$  equals zero, which drives the position error to zero, and is our goal. We will use a strategy called dynamic extension to that aim. We have:

$$S = \begin{pmatrix} u_1 \cos \psi \\ u_1 \sin \psi \end{pmatrix} - \dot{\eta}_d + Ke \quad (9)$$

We take a derivative of our sliding surface  $S$ .

$$\dot{S} = \begin{pmatrix} \dot{u}_1 \cos \psi - u_1 (\sin \psi) \dot{\psi} \\ \dot{u}_1 \sin \psi + u_1 (\cos \psi) \dot{\psi} \end{pmatrix} - \ddot{\eta}_d + K\dot{e} \quad (10)$$

Rewriting the above equation in matrix form, and remembering that  $\dot{\psi} = u_2$ :

$$\dot{S} = \begin{bmatrix} \cos \psi & -u_1 \sin \psi \\ \sin \psi & u_1 \cos \psi \end{bmatrix} \begin{bmatrix} \dot{u}_1 \\ u_2 \end{bmatrix} - \ddot{\eta}_d + K\dot{e} \quad (11)$$

Setting some notation:

$$A = \begin{bmatrix} \cos \psi & -u_1 \sin \psi \\ \sin \psi & u_1 \cos \psi \end{bmatrix} \quad (12)$$

$$\nu = \begin{bmatrix} \dot{u}_1 \\ u_2 \end{bmatrix} \quad (13)$$

Notice that the  $A$  matrix is invertible for  $u_1 \neq 0$ , and this is not a constraint for controller design as aircraft cannot fly below a certain speed, so that  $u_1 = 0$  is not a possibility for any kind of realistic flight. We then have:

$$\dot{S} = A\nu - \ddot{\eta}_d + K\dot{e} \quad (14)$$

We would like to have exponential dynamics for our sliding surface:  $\dot{S} = -\Lambda S$ , where  $\Lambda$  is a positive definite matrix. Our desired behavior is then:

$$\dot{S} = A\nu - \ddot{\eta}_d + K\dot{e} = -\Lambda S \quad (15)$$

We can then get the vector  $\nu$  from the equation:

$$\nu = A^{-1} [\ddot{\eta}_d - K\dot{e} - \Lambda S] \quad (16)$$

The vector  $\nu$  gives us values for  $u_2$  and  $\dot{u}_1$ . What we really need are values for  $u_1$  and  $u_2$ . We end up passing the  $\nu$  vector through a system that integrates the first field, to obtain the required values.

### 3.3 Formation Control of Multiple Vehicles

*Mesh Stability Analysis* In this section we will perform a stability analysis to show that our formation control law for a V-shaped vehicle formation is mesh stable.

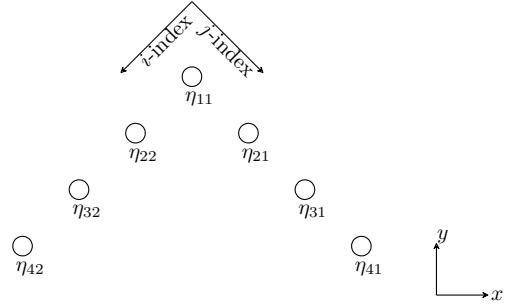


Fig. 2. V-Shape Formation

Each vehicle has an error associated with respect to its desired inertial position, as well as a relative error defined by its position with respect to neighboring vehicles (the vehicle directly in front of it, the vehicle directly behind it, and the vehicle on the same horizontal row). For example,  $\eta_{32}$  has neighbors  $\eta_{22}$ ,  $\eta_{42}$  and  $\eta_{31}$ . To generalize,  $\eta_{i,j}$  has neighbors  $\eta_{i-1,j}$ ,  $\eta_{i+1,j}$  and  $\eta_{i,j-1}$  (Note that  $\eta_{i,j}$  here stands for the position vector of the vehicle indexed  $(i, j)$ ). For a V-shaped formation, the  $j$  index may only be 1 or 2; when  $j$  equals 1, we look at  $\eta_{i,j+1}$  instead of  $\eta_{i,j-1}$ . We can define error as below, where the  $K_r$  gain allows us to adjust the importance of the inertial term and the formation shape terms.

$$e_{i,j} = \eta_{i,j} - \eta_{d_{i,j}} + K_r(\eta_{i,j} - \eta_{i-1,j} - \eta_{d_{(i,j)(i-1,j)}}) + K_r(\eta_{i,j} - \eta_{i+1,j} - \eta_{d_{(i,j)(i+1,j)}}) + K_r(\eta_{i,j} - \eta_{i,j-1} - \eta_{d_{(i,j)(i,j-1)}}). \quad (17)$$

Depending on the formation shape and the vehicle's position in the formation, some robots might not have all the terms (some relative error terms are set to zero if the vehicle is on the edge of the formation and therefore have less than 3 neighbors, for instance).

A sliding control law is applied to drive the error to zero. The sliding surface is defined as follows:

$$S_{i,j} = \dot{e}_{i,j} + q_1 e_{i,j} \quad (18)$$

We must force the control law to converge to zero in order to achieve our control objective. The following control input will serve this purpose.

$$\bar{u}_{i,j} = \frac{1}{1 + 3K_r} [-KS_{i,j} - q_1 \dot{e}_{i,j} + K_r \ddot{e}_{i,j}] \quad (19)$$

where  $K > 0$ ,  $q_1 > 0$  for classic stability, and  $e_{i,j} := \eta_{i-1,j} + \eta_{i+1,j} + \eta_{i,j-1}$ . The over-bar is used to indicate that this is the desired acceleration. We will assume that the actual vehicle acceleration is delayed due to processing and actuator dynamics modeled by the first order system (Pant et al. [2001]):

$$\bar{u}_{i,j} = \tau \dot{u}_{i,j} + u_{i,j} \quad (20)$$

The input acceleration is  $u_{i,j}$ , thus we will use this notation in place of  $\ddot{\eta}_{i,j}$ . Differentiating (17) twice, and substituting (19) and (20) into, we obtain a differential equation which

relates the error dynamics of a particular vehicle with respect to its neighbors.

$$\left( \tau \frac{d^3}{dt^3} + (1 + 3K_r) \left( \frac{d^2}{dt^2} + (K + q_1) \frac{d}{dt} + Kq_1 \right) \right) e_{i,j} = K_r \left( \frac{d^2}{dt^2} + (K + q_1) \frac{d}{dt} + Kq_1 \right) \delta_{i,j} \quad (21)$$

where  $\delta_{i,j} = e_{i-1,j} + e_{i+1,j} + e_{i,j-1}$

Taking the Laplace transform we get

$$\hat{e}_{i,j}(s) = \begin{bmatrix} H(s) & 0 \\ 0 & H(s) \end{bmatrix} \frac{\hat{\delta}_{i,j}(s)}{3}$$

with

$$H(s) = \frac{3K_r(s^2 + (K + q_1)s + Kq_1)}{\tau s^3 + (1 + 3K_r)(s^2 + (K + q_1)s + Kq_1)} \quad (22)$$

Each subsystem within the mesh must be mesh stable in order for the overall structure to be mesh stable (Pant et al. [2002]). Thus, we can now specify acceptable ranges and conditions on our gains  $K$ ,  $K_r$  and  $q_1$ .

Notice that  $(h(t) = L^{-1}\{H(s)\})$  does not change sign for small values of  $\tau$ , thus

$$\|h(t)\|_1 = \sup_w |H(jw)| = H(0) = \frac{3K_r}{1 + 3K_r} \quad (23)$$

$H(0) < 1$  if  $K_r > 0$ .

*Examples* For the sake of example, let's consider a formation of five vehicles organized as shown below.

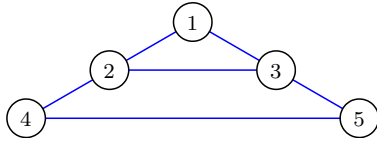


Fig. 3. 5-vehicle V-shaped example formation.

The approach presented in Section 3.2 for a single vehicle is still valid, given that we change the definition of the position error accordingly. As proved above, this approach is mesh stable. For the five vehicle case, the position errors are defined as,

$$\begin{aligned} e_1 &= \eta_1 - \eta_{d_1} + \sum_{j=2,3} \Lambda_R(\eta_1 - \eta_j - \eta_{d_{1j}}) \\ e_2 &= \eta_2 - \eta_{d_2} + \sum_{j=1,3,4} \Lambda_R(\eta_2 - \eta_j - \eta_{d_{2j}}) \\ e_3 &= \eta_3 - \eta_{d_3} + \sum_{j=1,2,5} \Lambda_R(\eta_3 - \eta_j - \eta_{d_{3j}}) \\ e_4 &= \eta_4 - \eta_{d_4} + \sum_{j=2,5} \Lambda_R(\eta_4 - \eta_j - \eta_{d_{4j}}) \\ e_5 &= \eta_5 - \eta_{d_5} + \sum_{j=3,4} \Lambda_R(\eta_5 - \eta_j - \eta_{d_{5j}}) \end{aligned}$$

where  $\Lambda_R$  is a positive definite matrix. If  $\Lambda_R$  is zero we revert to the single vehicle case. If  $\Lambda_R$  is large, more emphasis is placed on relative terms than on absolute position.

We can then define  $S_i = \dot{e}_i + Ke_i$  where “ $i$ ” is the vehicle index. For example, for the first vehicle, after some reorganizing of the terms:

$$\begin{aligned} S_1 &= (I + 2\Lambda_R)\dot{\eta}_1 + \left( -\dot{\eta}_{d_1} \right. \\ &\quad + \Lambda_R(-\dot{\eta}_2 - \dot{\eta}_{d_{12}} - \dot{\eta}_3 - \dot{\eta}_{d_{13}}) \\ &\quad + K(\eta_1 - \eta_{d_1} + \Lambda_R(2\eta_1 - \eta_2 \\ &\quad \left. - \eta_{d_{12}} - \eta_3 - \eta_{d_{13}})) \right) \end{aligned} \quad (24)$$

We can then set  $V = \dot{\eta}_1$  and

$$\begin{aligned} V_R &= (I + 2\Lambda_R)^{-1} \left( -\dot{\eta}_{d_1} \right. \\ &\quad + \Lambda_R(-\dot{\eta}_2 - \dot{\eta}_{d_{12}} - \dot{\eta}_3 - \dot{\eta}_{d_{13}}) \\ &\quad + K(\eta_1 - \eta_{d_1} \\ &\quad \left. + \Lambda_R(2\eta_1 - \eta_2 - \eta_{d_{12}} - \eta_3 - \eta_{d_{13}})) \right) \end{aligned} \quad (25)$$

and apply the approach presented in Section 3.2. Alternatively, we can write the above equations in matrix form:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix} = (I + Q)(\underline{\eta} - \underline{\eta}_d) \quad (26)$$

where  $\underline{\eta} = [\eta_1; \eta_2; \eta_3; \eta_4; \eta_5]$ ,  $\underline{\eta}_d = [\eta_{d_1}; \eta_{d_2}; \eta_{d_3}; \eta_{d_4}; \eta_{d_5}]$ , and

$$Q = \begin{bmatrix} 2\Lambda_R & -\Lambda_R & -\Lambda_R & 0 & 0 \\ -\Lambda_R & 3\Lambda_R & -\Lambda_R & -\Lambda_R & 0 \\ -\Lambda_R & -\Lambda_R & 3\Lambda_R & 0 & -\Lambda_R \\ 0 & -\Lambda_R & 0 & 2\Lambda_R & -\Lambda_R \\ 0 & 0 & -\Lambda_R & -\Lambda_R & 2\Lambda_R \end{bmatrix} \quad (27)$$

It is possible to show that the rows of  $(I + Q)$  are independent for  $\Lambda_R \neq 0$ . This indicates that  $(I + Q)$  is invertible, so that

$$e \rightarrow 0 \Rightarrow \underline{\eta} \rightarrow \underline{\eta}_d \Leftrightarrow \eta \rightarrow \eta_d \quad (28)$$

Finally, one can apply some notation to the multi-vehicle case to make it slightly more general and to mirror the single vehicle case. We define:

$$e_i = \eta_i - \eta_{ir} \quad (29)$$

where  $\eta_{ir} = \eta_{d_i} - \sum_{j \neq i} \Lambda_R(\eta_i - \eta_j - \eta_{d_{ij}})$  is a reference position. We can use this notation for the single vehicle case by setting  $i = 1$ , with  $\eta_{1r} = \eta_{d_1}$  and  $e_1 = \eta_1 - \eta_{d_1}$ .

We let

$$\begin{aligned} S_i &= \dot{e}_i + Ke_i = \dot{\eta}_i - (\dot{\eta}_{ir} - K e_i) \\ &= V_i - V_{ir}, \end{aligned} \quad (30)$$

We now have:

$$\dot{S}_i = \dot{\eta}_i - \dot{V}_{ir} = A_i \nu_i - \dot{V}_{ir}. \quad (31)$$

Letting  $\nu_i = A_i^{-1}[\dot{V}_{ir} - \Lambda S_i]$ , we now have

$$\dot{S}_i = -\Lambda S_i. \quad (32)$$

#### 4. SIMULATION RESULTS

We present simulation results for the controllers described above. We have adapted the simulation to reflect ground

robots, with which we plan to test the algorithms before applying them to Unmanned Air Vehicles. Because of our planned long-term application, we are still requiring that  $u_1 \neq 0$ , even though this is an unusual condition for ground vehicles.

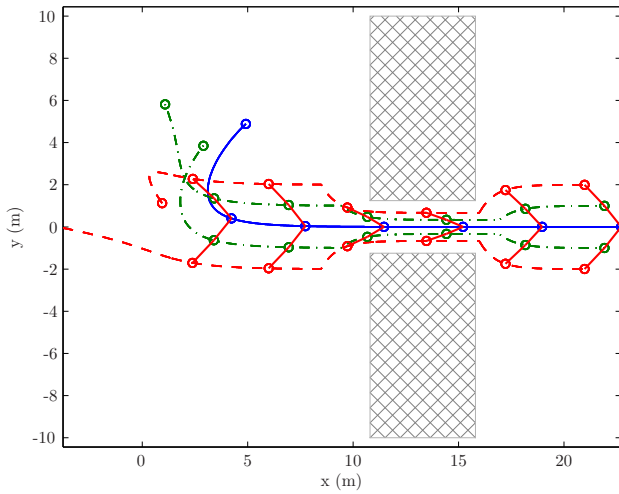


Fig. 4. Robot formation, unconstrained control inputs.

Fig. 4 shows five robots starting at different initial conditions, assembling into a formation, and moving as a formation towards the right of the plot. When detecting the obstacles (at  $t = 20$  sec in this example, assuming perfect sensors), they tighten the formation to “squeeze” through the obstacles, and then expand back to their original shape (at  $t = 35$  sec) beyond the obstacles. We assume obstacle detection is decoupled from obstacle avoidance, and for these simulations we are assuming perfect detection (which can easily be expanded).

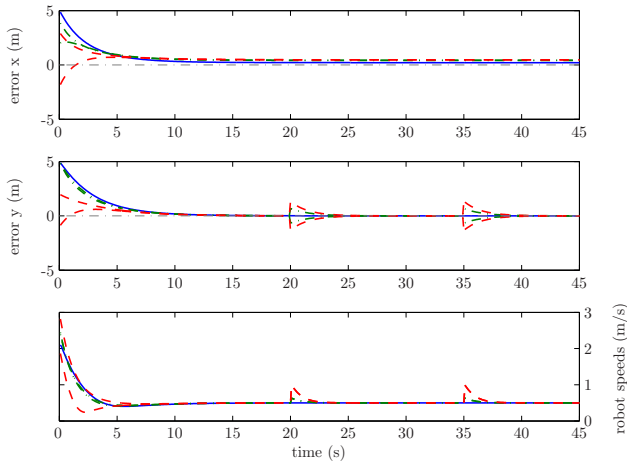


Fig. 5. Errors  $e_x$ ,  $e_y$  (showing exponential convergence to zero) and robot speeds, unconstrained control inputs.

Fig. 5 shows the error signals and the control inputs corresponding to the run in Fig. 4. The errors converge exponentially to zero. Fig. 6 shows that the sliding surface converges to zero as time goes.

In the interest of making the simulations more realistic, we apply the following constraints to the robot dynamics: the minimum speed is 0.4m/s, the maximum speed is 0.6 m/s,

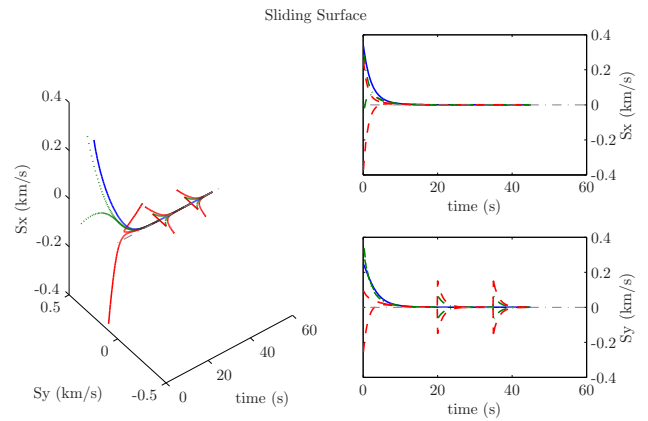


Fig. 6. Sliding surfaces converging to zero exponentially, unconstrained control inputs.

the min/max accelerations are  $+0.5\text{m/s}^2$  and  $-0.5\text{m/s}^2$  respectively, and the maximum turn rate is  $\pm 45$  deg/sec.

Fig. 7 shows the same scenario as Fig. 4, but with constrained control inputs. The error signals, shown in Fig. 8, still converge to zero, although the convergence is not exponential. Fig. 8 also depicts the robot speeds, which remain constrained between the minimum and maximum bounds. The sliding surface converges to zero as time goes at a slower rate compared to the aforementioned unconstrained case, as shown in Fig. 9.

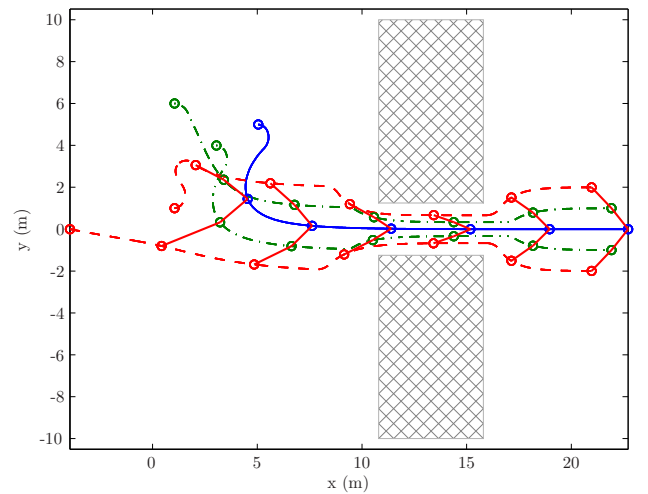


Fig. 7. Robot formation, constrained control inputs.

## 5. CONCLUSION

We have presented a leaderless formation control algorithm for unmanned vehicles, using concepts from sliding mode control and dynamic extension. The control algorithm is “leaderless”, which provides redundancy and fault tolerance, and it is proven mesh-stable. A single gain adjusts the relative importance of formation-maintaining terms versus individual position terms, allowing for both tight and loose formations. Simulation results are presented.

The strategy presented in this research is used for path planning purposes. It generates position commands that are then fed into a vehicle specific autopilot. As such, it

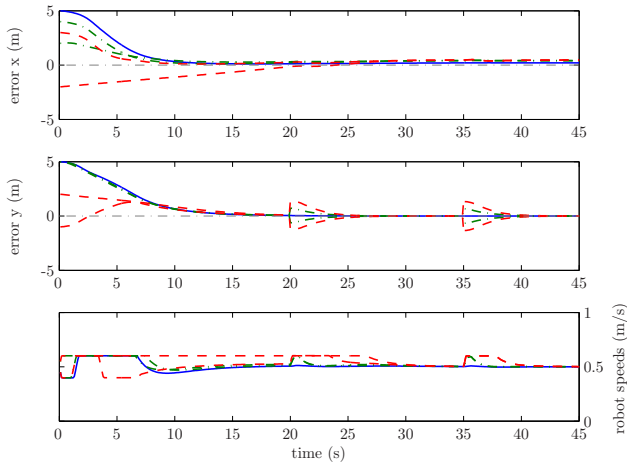


Fig. 8. Errors  $e_x$ ,  $e_y$  and robot speeds for constrained control inputs. A minimum speed was set as well as a maximum speed, as this algorithm will eventually be applied to Unmanned Air Vehicles.

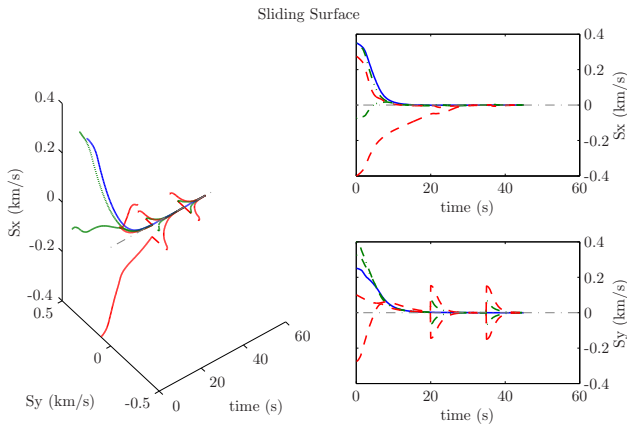


Fig. 9. Sliding surfaces converging to zero at a slower rate compared to the unconstrained control inputs case, constrained control inputs.

can be used with many different types of vehicles. The performance of the vehicle formation, though, will still be somewhat limited by whether the steerable unicycle approximation is reasonable for the vehicle under consideration.

Future work includes testing the algorithm on ground robots, Unmanned Air Vehicles and Autonomous Underwater Vehicles.

#### ACKNOWLEDGEMENTS

This work was supported in part by United States Air Force, under grant number FA8650-07-2-3744.

#### REFERENCES

R.W. Beard, J. Lawton, and F.Y. Hadaegh. A coordination architecture for spacecraft formation control. *IEEE Transactions on Control Systems Technology*, 9(6):777–790, 2001. ISSN 1063-6536. doi: 10.1109/87.960341.

J. K. Hedrick and D. Swaroop. Dynamic coupling in vehicles under automatic control. In *Proc. of the 13th IAVSD Symposium*, August 1993.

J.K. Hedrick, M. Tomizuka, and P. Varaiya. Control issues in automated highway systems. *IEEE Control Systems Magazine*, 14(6):21–32, 1994. ISSN 0272-1708. doi: 10.1109/37.334412.

N.E. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *Proc. of the 40th IEEE Conference on Decision and Control*, volume 3, pages 2968–2973, Orlando, FL, 2001. doi: 10.1109/.2001.980728.

R.M. Murray, K.J. Astrom, S.P. Boyd, R.W. Brockett, and G. Stein. Future directions in control in an information-rich world. *IEEE Control Systems Magazine*, 23(2):20–33, 2003. ISSN 0272-1708. doi: 10.1109/MCS.2003.1188769.

R. Olfati-Saber and R.M. Murray. Graph rigidity and distributed formation stabilization of multi-vehicle systems. In *Proc. of the 41st IEEE Conference on Decision and Control*, volume 3, pages 2965–2971, 2002.

A. Pant, P. Seiler, T.J. Koo, and K. Hedrick. Mesh stability of unmanned aerial vehicle clusters. In *Proc. of the 2001 American Control Conference*, volume 1, pages 62–68, Arlington, VA, 2001. doi: 10.1109/ACC.2001.945514.

A. Pant, P. Seiler, and K. Hedrick. Mesh stability of look-ahead interconnected systems. *IEEE Transactions on Automatic Control*, 47(2):403–407, 2002. ISSN 0018-9286. doi: 10.1109/9.983389.

S. Sheikholeslam and C.A. Desoer. Control of interconnected nonlinear dynamical systems: the platoon problem. *IEEE Transactions on Automatic Control*, 37(6):806–810, 1992. ISSN 0018-9286. doi: 10.1109/9.256337.

R.S. Smith and F.Y. Hadaegh. Control topologies for deep space formation flying spacecraft. In *Proc. of the 2002 American Control Conference*, volume 4, pages 2836–2841, 2002. doi: 10.1109/ACC.2002.1025219.

S. Spry and J.K. Hedrick. Formation control using generalized coordinates. In *Proc. of the 43rd IEEE Conference on Decision and Control*, volume 3, pages 2441–2446, 2004.

D. Swaroop and J. K. Hedrick. String stability of interconnected systems. *IEEE Transactions on Automatic Control*, 41(3):349–57, 1996.

H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents, part I: fixed topology. In *Proc. of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 2010–2015, 2003a. doi: 10.1109/CDC.2003.1272910.

H.G. Tanner, A. Jadbabaie, and G.J. Pappas. Stable flocking of mobile agents, part II: dynamic topology. In *Proc. of the 42nd IEEE Conference on Decision and Control*, volume 2, pages 2016–2021, 2003b. doi: 10.1109/CDC.2003.1272911.

M. Tillerson, L. Breger, and J.P. How. Distributed coordination and control of formation flying spacecraft. In *Proc. of the 2003 American Control Conference*, volume 2, pages 1740–1745, 2003.

P. Varaiya, T. Simsek, and J. Borges de Sousa. Communication and control of distributed hybrid systems - tutorial session. In *Proc. of the 2001 American Control Conference*, pages 4968–83. IEEE, 2001.

T.I. Zohdi. Computational design of swarms. *International Journal for Numerical Methods in Engineering*, 57:2205–2219, 2003.