Euro UAV 2007 Conference and Exhibition, 13-15 June 2007, Paris, France

Cooperative Control of UAV Platoons – A Prototype

Karl Mathia
Renaissance Sciences Corp.
1351 N Alma School Road
Chandler, Arizona 85224, USA
kmathia@rscusa.com

Gerardo Lafferriere
Department of Mathematics
Portland State University
724 SW Harrison Street
Portland, Oregon 97201, USA
gerardoL@pdx.edu

Todd Titensor
Procerus Technologies
478 South Geneva Rd.
Vineyard, Utah 84058, USA
toddt@procerusuav.com

Abstract

A platoon formation of autonomous vehicles refers to a set of spatially distributed vehicles whose dynamic states are coupled through a common, cooperative control law. We present a technical approach and prototype of a cooperative controller for unmanned aerial vehicles (UAVs) and other autonomous robot vehicles. We presented the underlying theory for the decentralized and cooperative control in [8][10]. The present paper focuses on the next development step, i.e. a prototype system. We first review the cooperative controls concept and then discuss disturbance rejection, and path planning and tracking in a cooperative controls context. Finally the prototype design and its hierarchical controls architecture are presented, as well as the high-level software implementation.

1 Introduction

The cooperative control for platoons of multiple UAVs is desired where it offers capabilities not possible with individual UAVs. Examples are maximizing the area coverage for search and rescue operations, cooperative tracking of a moving target, and coopertative engagement. A basic requirement for successful cooperative platoon control is the stable estimation and control of the relative vehicle positions and orientations within a platoon [1]-[5][8][12][23]. Disturbance rejection and path tracking in a team context are also capabilities needed to make platoon control suitable for a variety of applications. The decentralized and cooperative control method presented here combines recent results from dynamical system theory, control theory, and algebraic graph theory [22][6][7]. The concept offers flexibility and ease of use and is applicable to loose and tight formations of various sizes, and to formations with and without leader following. In theory, vehicle fleets ('formations of formations') are also possible [24][25][26][28].

Several publications addressed the underlying theory for decentralized and cooperative control. See [6][7][8][10] and the references therein. A platoon converges to the commanded formation if all selected vehicles achieve their respective relative position within the platoon formation from arbitrary initial positions and then maintain formation by aligning heading and speed with their neighbors. It was shown that this process is asymptotically stable.

The next steps in the development process towards a commercial cooperative control system is a prototype system, including the rejection of steady disturbances, for example steady wind, and the path following problem for platoons. This is presented here. The prototype system includes one cooperative controller per vehicle and will emphasize that a central 'supervisor' controller is only required when a task or mission change is ordered by the human operator. The actual heading and speed control among platoon members is fully decentralized and cooperative. The hierarchical architecture of the cooperative control system is organized at four layers, i.e. human operator, platoon manager, a cooperative controller per vehicle, and a conventional autopilot per vehicle. The hierarchy and its command and data flow are presented, as are some implementation challenges. A prototype design for a hardware-in-the-loop simulation study is outlined and demonstrates the feasibility of cooperative control.

2 Vehicle Dynamics and Formation Control

Several research efforts have addressed cooperative control for autonomous vehicles, see, for example, [5][6][7][22][23][25][26]. While most previously published feedback control laws utilize error

---

*This research effort was supported in part by the U.S. Army under contract W15QKN-07-C-0051.

measurements between the vehicles' desired and actual locations relative to their neighbors [4][10], the method used here defines the control problem as a *stabilizability problem in the class of hybrid control systems*. We formulate it as a hybrid control problem, as it contains the

- continuous-time state of the vehicles, and
- discrete-time inter-vehicle communication.

The UAVs are modeled as linear time-invariant (LTI) systems, while the intervehicle feedback laws rely on the potentially dicrete-time communication caused by intermittent communication. However, this is not addressed here. Instead we assume perfect communication links among vehicles and focus on the control problems discussed below.

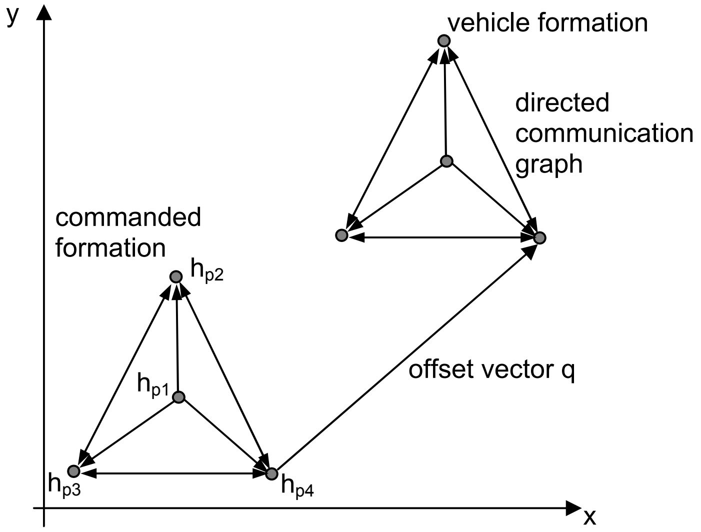
2.1 Formation Definition

A platoon formation of N vehicles is given by its formation configuration vector h,

$$h = h_p \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \in \Re^{2nN}, \tag{1}$$

where '$\otimes$' denotes the Kronecker product, and h_p includes the vehicle absolute positions in the platoons global reference frame. The vector h offers an easy-to-use method to specify formations and is key to the control method. The number of states per vehicle is denoted by n. The N-vehicle platoon is in formation if all vehicles are 'sufficiently close' to the commanded position within the platoon and if all vehicles move at the same speed [7]. For the purpose of this discussion we assume that the vehicles move in a 2-dimensional plane with coordinate axes x and y. See Figure 1.

Figure 1. Defining a platoon formation with four vehicles.



The communication between vehicles is described by a directed communication graph, G=(V,E) , defined by vertices V and edges E. The i-th vertex represents the i-th vehicle, and the edge (i,j) indicates that vehicle j receives information from vehicle i, which we call a neighbor of vehicle j. Let $\Im_i$ denote the set of neighbors to agent i, and let $\left|\Im_i\right|$ denote the number of visible neighbors. All vehicles must agree on a global reference frame, which in practice can be provided by the Global Positioning System (GPS).

2.2 Dynamical Model

The *i*-th vehicle within a platoon of *N* autonomous agents (or vehicles) is modeled by the second-order, linear time-invariant (LTI) system

$$\dot{\boldsymbol{x}}_i = \boldsymbol{A}_{veh}\boldsymbol{x}_i + \boldsymbol{B}_{veh}\boldsymbol{u}_i, \qquad i = 1 \ldots N, \qquad \boldsymbol{x}_i \in \Re^{2n} \tag{2}$$

with

$$\boldsymbol{A}_{veh} = diag\left[\begin{bmatrix} 0 & 1 \\ a_{21}^1 & a_{22}^1 \end{bmatrix}, \ldots, \begin{bmatrix} 0 & 1 \\ a_{21}^n & a_{22}^n \end{bmatrix}\right]$$

$$\boldsymbol{B}_{veh} = \boldsymbol{I}_n \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The signal $\boldsymbol{u}_i$ is the control input for vehicle *i*. Let $n = 2$, with (relative) position and velocity being the two states. The vehicle position and velocity vectors are $\boldsymbol{x}_p = \begin{bmatrix} x & y \end{bmatrix}^T$ and $\boldsymbol{x}_v = \begin{bmatrix} \dot{x} & \dot{y} \end{bmatrix}^T$, respectively. The state vector for vehicle *i* is defined as

$$\boldsymbol{x}_i = \boldsymbol{x}_p \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \boldsymbol{x}_v \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The platoon dynamics combine all vehicle dynamics,

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u}, \tag{3}$$

where $\boldsymbol{x}$ and $\boldsymbol{u}$ denote the augmented vector of all vehicle states and controls, respsectively. $\boldsymbol{A}$ and $\boldsymbol{B}$ are the appropriate size block diagonal matrices, and $\boldsymbol{u}$ is the entire system input.

2.3 Formation Control

The cooperative control problem is to find a control $\boldsymbol{u}$ such that all vehicles converge to the commanded vehicle formation $\boldsymbol{h}$. The distributed aspect of our control method is based on the dependence of $\boldsymbol{u}$ on local and relative state information only, which is received from the neighbors J_i of each vehicle. A central controller is not needed. Furthermore, $\boldsymbol{u}$ is the only dynamic link among the vehicles in a platoon. The convergence and stability properties of the cooperative control method were studied using the platoon dynamics shown in Equation (3), and the required communication links and controller parameters for the individual vehicles are derived from this 'top-down' representation. Of course, the implementation of the cooperative control system will build upon the local controls of each vehicle. As such, a 'bottom-up' representation is established, emphasizing the individual vehicles and their local controls. The objective here is to transition these theoretical results into a practical form that will allow the controls implementation on physical systems. The first step is to convert the 'top-down' approach and representation at the systems (platoon) level to a 'bottom-up' approach and representation of interconnected, but individual vehicles.

'Top-Down' Approach. The 'top-down' approach to formation control is based on the overall platoon representation in Equation 3, which is used in most mathematical analyses of platoon dynamics and stability. Its control $\boldsymbol{u}$ is given by

$$\boldsymbol{u} = \boldsymbol{F} \cdot \boldsymbol{z} = \boldsymbol{F} \cdot \boldsymbol{L}\left(\boldsymbol{h} - \boldsymbol{x}\right), \tag{4}$$

thus the controlled formation is given by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{F}\boldsymbol{L}\left(\boldsymbol{h} - \boldsymbol{x}\right) \tag{5}$$

The commanded formation is denoted $\boldsymbol{h}$. The Laplacian matrix $\boldsymbol{L}$ for the communication graph is defined as

$$\boldsymbol{L} = \boldsymbol{D}_d - \boldsymbol{A}_d,$$

where $\boldsymbol{D}_d$ is the indegree matrix and $\boldsymbol{A}_d$ is the adjacency matrix of the graph. The top-down representation is shown in Figure 2. The formation error vector $\boldsymbol{e}$ is the difference of commanded and actual formation positions and velocities.

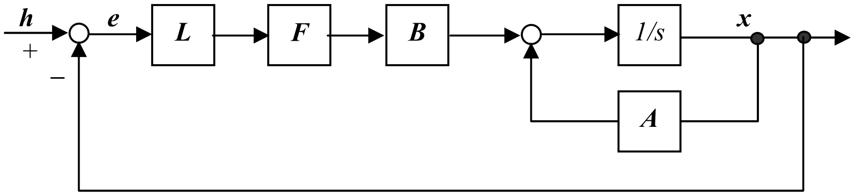


Figure 2. Top-down approach to formation control: system representation.

Note that the basic formation control in Equation 5 provides stability and convergence-to-formation [6][7], but it does not compensate for disturbances, e.g. sensor offset and wind. This is addressed below.

'Bottom-Up' Approach. The conversion from the 'top-down' to a 'bottom-up' platoon representation is needed for the implementation of the local controls within the distributed architecture. The autonomous vehicles achieve their common objective using local control laws only, based on the information exchanged among neighbors. The common objective here is the convergence to the commanded platoon formation. So the question is how to express the platoon-level Equation (5) at the local vehicle level? In particular, what is the feedback $\boldsymbol{z}_i$ for each individual vehicle? The feedback also depends on the communication graph, defined by the Laplacian $\boldsymbol{L}$, and comprises the average relative displacements and velocities of the *i*-th vehicle and its $\left|\Im_i\right|$ 'visible' neighbors. The vehicle's commanded position is $\boldsymbol{h}_i$. As such, every vehicle in the platoon receives the following feedback:

$$\boldsymbol{z}_i = \begin{cases} \dfrac{1}{\left|\Im_i\right|} \displaystyle\sum_{j \in \Im_i} \left(\left(\boldsymbol{x}_i - \boldsymbol{h}_i\right) - \left(\boldsymbol{x}_j - \boldsymbol{h}_j\right) \right) & \text{if } \left|\Im_i\right| > 0 \\ 0 & \text{otherwise} \end{cases}, \quad \text{with } i = 1 \ldots N, \quad \boldsymbol{z}_i \in \Re^{2n} \tag{6}$$

With this feedback law and the vehicle dynamics in Equation (2) then become the controlled vehicle dynamics

$$\dot{\boldsymbol{x}}_i = \boldsymbol{A}_{veh}\boldsymbol{x}_i + \boldsymbol{B}_{veh}\boldsymbol{u}_i = \boldsymbol{A}_{veh}\boldsymbol{x}_i + \boldsymbol{B}_{veh}\boldsymbol{F}_{veh}\boldsymbol{z}_i \tag{7}$$

where $\boldsymbol{F}_{veh}$ is a 'formation gain matrix'. Note that the Laplacian matrix does not appear in the bottom-up representation, because it represents the communication graph at the pltoon-level only. For example, for a vehicle with n=2 we have

$$\begin{bmatrix} \dot{x}_i \\ \ddot{x}_i \\ \dot{y}_i \\ \ddot{y}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_{22}^1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_{22}^n \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \\ y_i \\ \dot{y}_i \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_{i1} \\ u_{i2} \end{bmatrix}$$

The controlled dynamics are

$$\begin{bmatrix} \dot{x}_i \\ \ddot{x}_i \\ \dot{y}_i \\ \ddot{y}_i \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & a_{22}^1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & a_{22}^n \end{bmatrix} \begin{bmatrix} x_i \\ \dot{x}_i \\ y_i \\ \dot{y}_i \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_{i1} & f_{i2} & 0 & 0 \\ 0 & 0 & f_{i1} & f_{i2} \end{bmatrix} \begin{bmatrix} z_{i1} \\ z_{i2} \\ z_{i3} \\ z_{i4} \end{bmatrix}.$$

3 Disturbance Rejection

We studied the distributed formation control under persistent disturbances. We showed under what conditions convergence to formation can be achieved in the presence of disturbances and perturbations of the control input in a platoon context [8]. In fact, a control law that compensates for steady and smooth disturbances (e.g. steady wind) is reviewed here. The formation control method was analyzed and recast as a classical output stabilization problem. Thus far the two main results are:

- The model is output disturbance decoupled if and only if the disturbances are constant. (This has few practical implications.)
- The model converges to formation in the presence of persistent (constant) disturbances in the feedback control loop if we include a dynamic compensator in the loop with an integral term. Such compensator can be designed using standard pole placement techniques.

Persistent disturbances in the feedback control loop are controlled with an additioal integral controller term, i.e. with a dynamic compensator. The controller equations are given below, where $\boldsymbol{x}_i$ denotes the coordinates of vehicle i. The controller variable is an accumulator to account for persistent errors. Note the integral feedback term, with $\boldsymbol{z}$ as the integrator variable.

$$\begin{aligned} \dot{z}_i &= x_i - h_i - \frac{1}{|J_i|}\sum_{j\in J_i}(x_j - h_j) \\ \ddot{x}_i &= -f\left(x_i - h_i - \frac{1}{|J_i|}\sum_{j\in J_i}(x_j - h_j)\right) - g\left(\dot{x}_i - \frac{1}{|J_i|}\sum_{j\in J_i}\dot{x}_j\right) - \mu z_i \end{aligned} \tag{8}$$

The constants *f*, *g*, and μ are the position, velocity, and integral feedback gain. The compensator dynamics also apply to more general second-order dynamics than the vehicle model used here. In particular, the accelerations $\ddot{x}_i$ may also depend directly on $\dot{x}_i$ (not just through the feedback law). The disturbance rejection method was validated using software simulation. An example of resulting platoon trajectories are shown in Figure 3. The commanded formation is a regular pentagon, and the compensator is turned on about a half way through the circular motion to illustrate its effect. Initially the vehicles settle in an irregular pentagon formation. After the compensator is turned on there is some intermediate transient behavior and then the regular pentagon formation is achieved. Note that the compensator cancels the effect of the disturbance on the formation but not on the absolute positions of the agents. Such absolute motions reside in the unobservable space of the model.

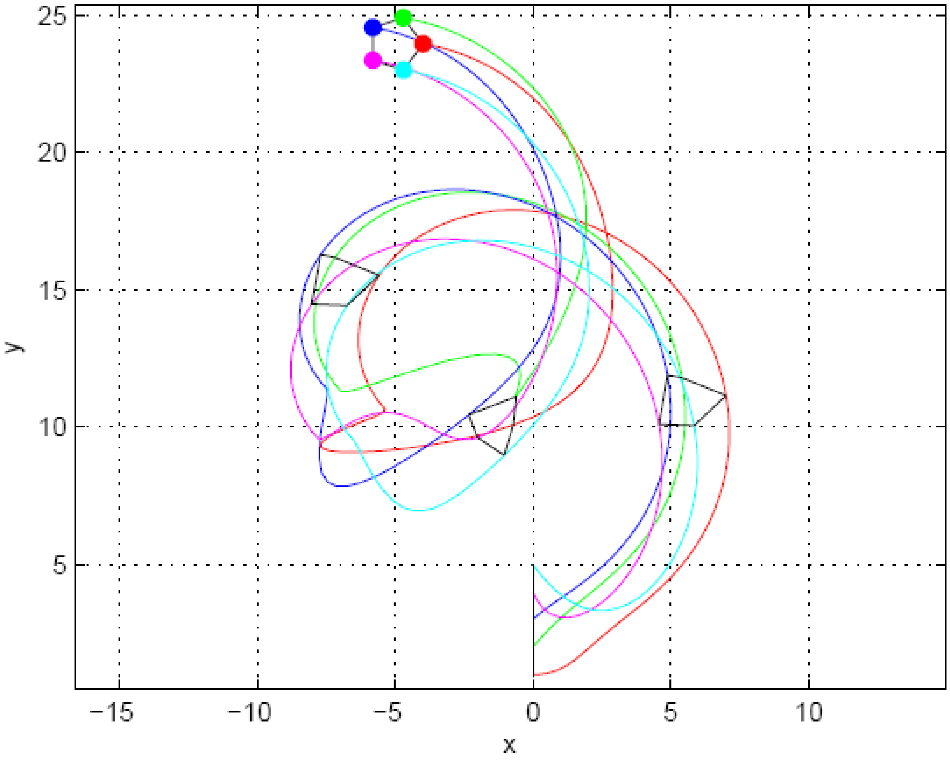


Figure 3. Formation convergence while in circular motion.

4 Path Following

A path following (or tracking) method for platoons of autonomous vehicles is presented here. It enables platoons to follow a specified path. The algorithm extends existing path planning and path following methods for individual vehicles to a team context: the algorithm combines leader-following and cooperative formation control, where one vehicle is designated as the leader. It tracks the specified path while the other vehicles maintain the platoon formation and simultaneously follow the leader. The system dynamics of the controlled vehicle, including both the cooperative and tracking controllers, are described by

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\left(\boldsymbol{F}\left(\boldsymbol{x} - \boldsymbol{h}\right) + \boldsymbol{d}\right) + \boldsymbol{K}\boldsymbol{e}_{track} \tag{10}$$

$$\boldsymbol{e}_{track} = \boldsymbol{r} - \boldsymbol{x}$$

where $\boldsymbol{d}$ is a vector constant disturbance (different for each vehicle), $\boldsymbol{e}_{track}$ is the path following error, and $\boldsymbol{r}$ is the reference/desired path. We call the matrix $\boldsymbol{K}$ the tracking gains. A simulation study of a platoon's path following performance is shown in Figure 4. The vehicles' initial positions in the x/y-plane are aligned on the vertical axis at x=0: (0,1), (0,2), (0,3), (0,4), (0,5). The black circle with radius 10 is the desired path. The red line indicates the path of the leader vehicle, starting at point (0,1). After the initial transitition the leader closely tracks the desired path. The remaining vehicles converge to the commanded formation and simultaneously follow the leader. Note that the final formation is not a completely regular pentagon. The figure illustrates a compromize between tracking and formation control. This control problem will be addressed.

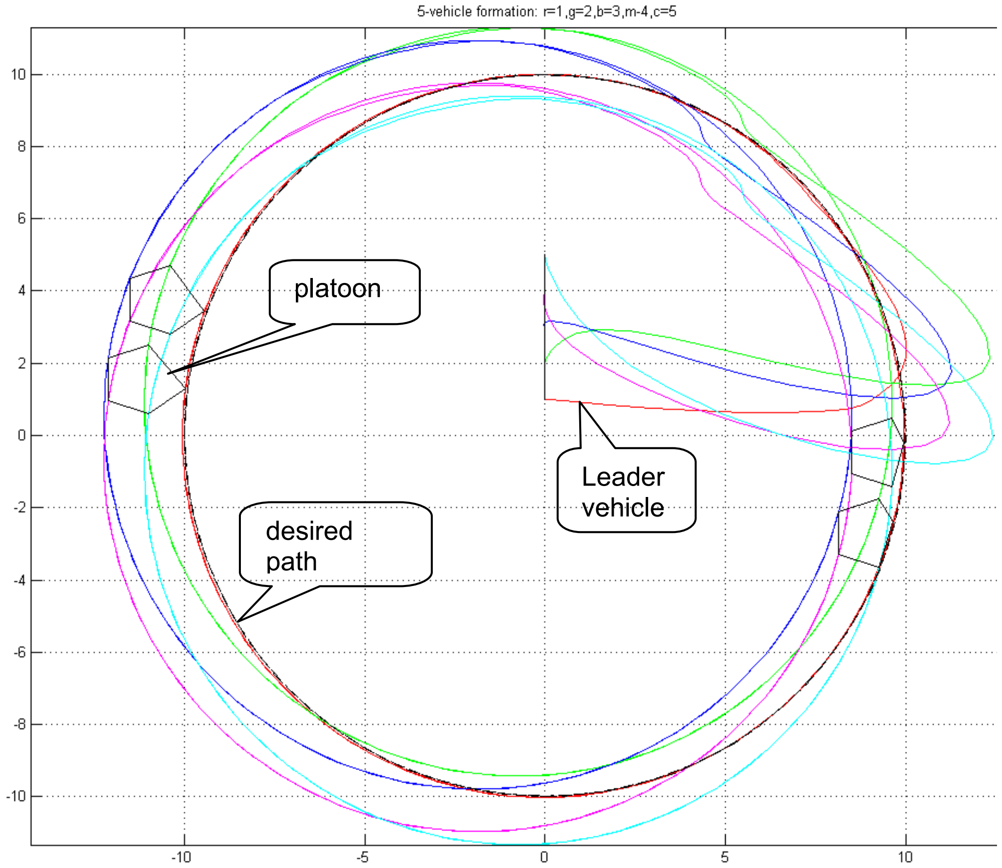


Figure 4. Formation converges while following a circular path using leader-following.

5 Implementation and Hardware-in-the-Loop Simulation

5.1 Hierarchical Architecture

A hierarchical architecture was chosen for the cooperative controls system, motivated by the flexibility and scalability of a 'building block' system can offer. This feature is needed, for example, for adjusting the a platoon size, i.e. adding or removing vehicles. The hierarchy includes four levels of control and allows the implementation of suitable communication interfaces for the desired interoperability between the system components. See Table 1, which also lists examples of typical tasks perfromed at each level.

Level	**Controller**	**Realization**	**Typical Tasks**
1	Supervisor	Human Operator	Strategic planning, exceptions, waypoints
2	Platoon Manager	Software on OCU	Control of the platoon as a single system
3	Cooperative Controller	Hardware/Software	Control of vehicle for cooperative behavior
4	Autopilot	Hardware/Software	Local control

Table 1. Hierarchical levels of control.

Figure 5 shows the Control Flow Diagram with subsystems and interfaces for *N* vehicles. The control hierarchy and four levels of control are emphasized, as well as as the signals sent across the interfaces between the subsystems. Note that only one vehicle is shown in the figure, although a platoon comprises two or more vehicles. The platoon manager (PM) represents the highest level of *automatic* control. The PM is a software systems and resides on the Operator Control Unit (OCU). A copy of the collaborative controller (CC) software resides on each vehicle, together with a conventional autopilot (AP).

Platoon Manager (PM). One PM is needed per platoon. It performs high-level tasks that affect the platoon as a single system. Such actions are, for example, changing the formation pattern, specifying way points for a new path, adding vehicles to a platoon, or removing vehicles from a platoon. Of course, the human operator has override authority.

Cooperative Controller (CC) and Communication Manager (CM). One CC is used per vehicle. The CC performs only the vehicle control functions required for the platoon's cooperative behavior, i.e. to achieve common objectives at the platoon level. The common object here is to achieve and maintain a commanded platoon formation. However, other common platoon objectives are also feasible. The CM provides the communication for the control loop between the CC and the autopilot, but may also be needed for communication between the CC and the PM.

Autopilot. The AP performs the vehicle-specific, local control functions, including vehicle stability, turns, rolls, and velocity adjustments. The AP does is not 'aware' of the vehicle's common objective as part of a platoon.

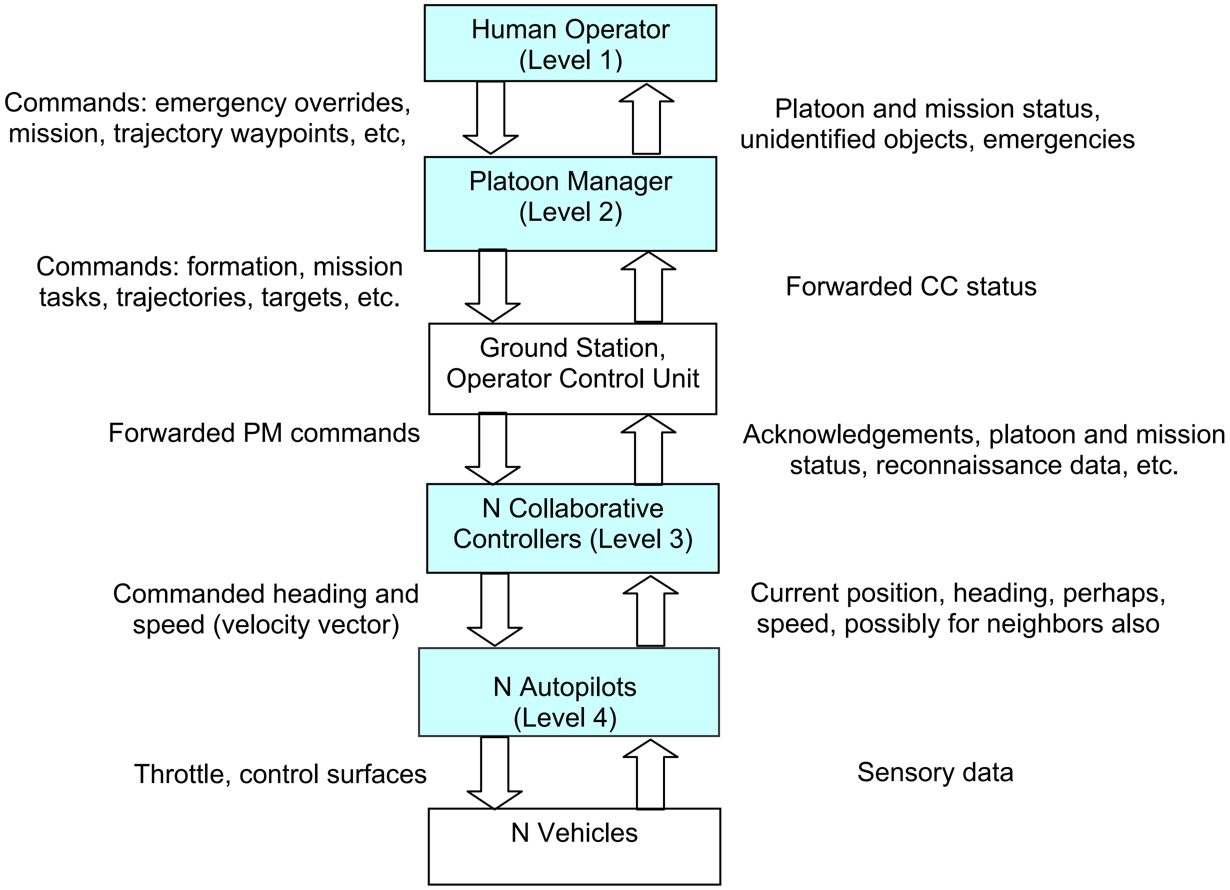


Figure 5. Control Flow Diagram for the proposed systems architecture.

5.2 Interoperability through Flexible and Scalable Building Blocks

A communication architecture and protocol is needed that supports both the flexibility and scalability of above control system hierarchy. All subsystems of the cooperative control system will be ‘building blocks’ that offer the interoperability needed to, for example, add and remove vehicles to/from a platoon, including vehicles from other platoons or networks. As such, both inter-agent communication and inter-network communication is desired. We selected well-known, standard interface protocol candidates that will allow the exchange of system components as long as the functionality is identical. The NATO STANAG 4586 messaging standards is one example. Slight variations in functionality, e.g. command syntax, can be compensated with software changes. The designed architecture includes

- a network capable operator control unit (OCU),
- a platform interface that conforms to the selected messaging standards, and
- OCU support for a minimal messaging set for target and hands-off transfer, geo-location transfer, and tasking.

The next section outlines the autopilot and system platform used for the prototype and hardware-in-the-loop simulation.

5.3 Hardware-in-the-Loop Simulation

A hardware-in-the-loop simulation was implemented with the hardware and software platform from Procerus Technologies, in particular Procerus’ ‘Virtual Cockpit’, i.e. XP-based ground control software for the Kestrel™ Autopilot System, and the Kestrel Autopilot itself. The Aviones flight simulator and the associated DLL libraries are also part of the prototype system.

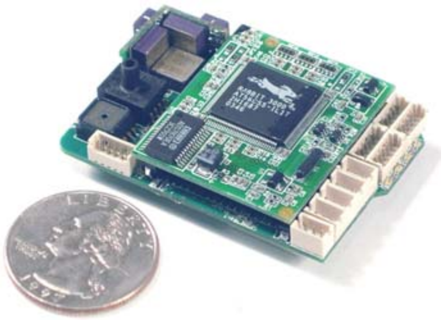
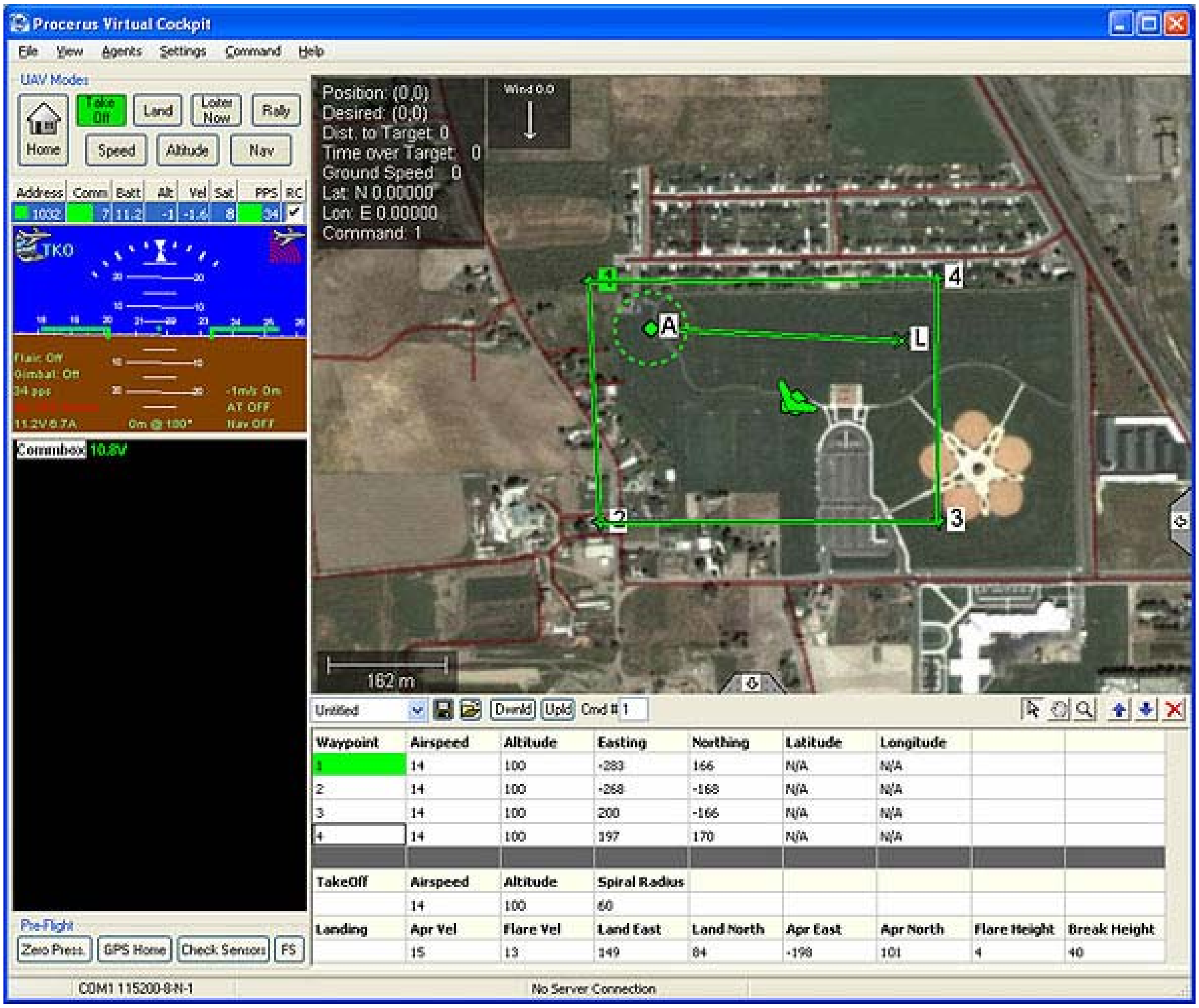


Figure 6. Kestrel Autopilot version 2.2.



Address	Comm	Batt	Alt	Vel	Sat	PPS	RC
1032	7	11.2	-1	-1.6	8	34	✓

Flaic: Off
Gimbal: Off
34 pps
11.2V 6.7A
0m @ 100°
-1m/s 0m
AT OFF
Nav OFF

Commbox 10.8V

Position: (0,0)
Desired: (0,0)
Dist. to Target: 0
Time over Target: 0
Ground Speed: 0
Lat: N 0.00000
Lon: E 0.00000
Command: 1

Cmd # 1

Waypoint	Airspeed	Altitude	Easting	Northing	Latitude	Longitude		
1	14	100	-283	166	N/A	N/A		
2	14	100	-268	-168	N/A	N/A		
3	14	100	200	-166	N/A	N/A		
4	14	100	197	170	N/A	N/A		
TakeOff	Airspeed	Altitude	Spiral Radius					
	14	100	60					
Landing	Apr Vel	Flare Vel	Land East	Land North	Apr East	Apr North	Flare Height	Break Height
	15	13	149	84	-198	101	4	40

COM1 115200-8-N-1

No Server Connection

Figure 7. Virtual Cockpit 2.2 interface.

The Kestrel Autopilot is a small and light miniature autopilot targeted for mini and micro UAVs. It weighs 16.65 grams at (5 cm x 3.5 cm x 1.2 cm), including all sensors (Figure 6). It provides autonomous flight control for single or multiple UAVs. Features include GPS waypoint navigation, as well as autonomous takeoff, flight and auto-landing. Combined with the Virtual Cockpit™ ground control software, the Kestrel provides the features and capabilities required for the prototype. The Virtual Cockpit user interface is shown Figure 7. It provides the operator access to the vehicles from the ground station (operator control unit). It also offers online mission planning, monitoring, in-flight adjustment, and more on a notebook computer and allows operators to configure and monitor the autopilot, upload flight plans and change waypoints.

Our modular approach provides the collaborative control system as a separate software module, also for later insertion into multiple software architectures. Written in standard C-code it will also be of use in multiple operating systems. Here each module communicates with the Kestrel via a standard serial interface. Each Kestrel communicates with the Aviones simulator via a dynamic link library.

For the prototype demonstration, i.e. the hardware-in-the-loop simulation, the Aviones flight simulator allows inter-vehicle communication by simulating an RF modem data link required for the collaborative control. A message sent from a vehicle (Kestrel autopilot) gets transferred over the serial cable to the Aviones simulator, and is then broadcasted out to all other autopilots in the simulated platoon. Each Kestrel autopilot receives the exact same message and filters messages based on their respective vehicle address.

6 Conclusion

The development of a cooperative control system for unmanned aerial vehicles (UAVs) was presented, including the technical approach and prototype implementation. The cooperative controller is targeted for UAV and other autonomous robot vehicles. We previously presented the underlying theory for the decentralized and cooperative control [8][10]. The present paper in turn focused on the next development step, i.e. a prototype system. We first reviewed the cooperative controls concept and then discussed disturbance rejection, and path planning and tracking in a team context. Finally the prototype design and its hierarchical controls architecture was presented, as well as high-level aspects of the software implementation.

7 References

[1] S. Bayraktar, G. Fainekos, and G.J. Pappas, “Experimental Cooperative Control of Unmanned Aerial Vehicles,” *Proc. 43rd IEEE Conf. on Decision and Control*, Bahamas, December 2004.

[2] J.A. Fax and R.M. Murray," Information Flow and Cooperative Control of Vehicle Formations," *Proc. 15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 2002.

[3] J.A. Fax and R.M. Murray, "Graph Laplacians and Stabilization of Vehicle Formations," *Proc. 15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 2002.

[4] A. Jadbabaie, J.Lin, A.S. Morse, “Coordination of Groups of Mobile Autonomous Agents Using Nearest Neighbor Rules,” *IEEE Trans. Automatic Control*, Vol. 48, No. 6, pp. 988-1001, June 2003.

[5] S. Glavaski, M. Chavez, R. Day, A. Williams, P. Nag, and W. Zhang. “*Vehicle Networks: Achieving Regular Formation,*” Proc. American Control Conference, Denver, Colorado, June 4-6, 2003.

[6] G. Lafferriere, J. Caughman, and A. Williams, “Graph Theoretic Methods in the Stability of Vehicle Formations”, *Proc. American Control Conference*, Boston, July 2004.

[7] G. Lafferriere, A. Williams, J. Caughman, and J.J.P. Veerman, “Decentralized Control of Vehicle Formations”, *Systems and Control Letters*, accepted February 2005.

[8] G. Lafferriere and K. Mathia, 'Control of Formations under Persistent Disturbances,' 46th IEEE Conference on Decision and Control, New Orleans, Louisiana, USA, 12-14 Dec. 2007 (to appear).

[9] Jesse Leitner, 'Formation Flying - The Future of Remote Sensing From Space,' 2nd International Symposium on Formation Flying - Missions & Technologies, Washington D.C., Sep. 14-16, 2004.

[10] K. Mathia, G. Lafferriere, and A. Williams, ‘Cooperative Control of Unmanned Vehicle Formations,’ Euro UAV 2006 Conference and Exhibition, Paris, France, 6-8 June 2006.

[11] R. Olfati-Saber and R.M. Murray, "Consensus Problems in Networks of Agents with Switching Topology and Time-Delays," *IEEE Trans. Automatic Control*, vol. 49, pp. 1520-1533, Sep. 2004.

[12] G. Purcell, D. Kuang, S. Lichten, S.-C. Wu, and L. Young, 'Autonomous Formation Flyer (AFF) Sensor Technology Development,' 1998.

[13] C.W. Reynolds, "Steering Behaviors For Autonomous Characters," *Proc. Game Developers Conference*, 1999.

[14] A. Richards and J. How, “Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming,” *Proc. American Control Conference*, May 2002.

[15] K. Sigurd and J. P. How, "UAV Trajectory Design Using Total Field Collision Avoidance," *Proc. AIAA Guidance, Navigation, and Control Conference*, Austin, Texas, August 2003.

[16] D.P. Spanos and R.M. Murray, "Robust Connectivity of Networked Vehicles", *Proc. 43rd IEEE Conf. on Decision and Control*, Bahamas, December 2004.

[17] D. Stipanovic, G. Inalhan, R. Teo, and C. J. Tomlin, "Decentralized Overlapping Control of a Formation of Unmanned Aerial Vehicles," *Automatica*, Vol. 40, No. 8, pp. 1285-1296, August 2004.

[18] D. Swaroop and J.K. Hedrick, “*String Stability of Interconnected Systems*,” IEEE Trans. on Automatic Control, Vol. 41, No. 3, pp. 349-357, March 1996.

[19] H.G. Tanner, G.J. Pappas, and V. Kumar, “Leader to Formation Stability,” *IEEE Transactions on Robotics and Automation*, 20(3):443-455, June 2004.

[20] H.G. Tanner, S. Loizou, and K.J. Kyriakopoulos, "Nonholonomic Navigation and Control of Cooperating Mobile Manipulators," *IEEE Transactions on Robotics and Automation*, vol 19, no 1, pp 53-64, February 2003.

[21] H.G. Tanner, "Flocking with Obstacle Avoidance in Switching Networks of Interconnected Vehicles," *Proc. IEEE Int’l Conference on Robotics and Automation*, New Orleans, LA, April 2004.

[22] J.J.P. Veerman, G. Lafferriere, J. Caughman, and A. Williams, “Flocks and Formations,” *Special issue of the Journal of Statistical Physics*, to appear.

[23] D. Weidow and J. Bristow, 'NASA/DoD University Nano-Satellites for Distributed Spacecraft Control - Military Uses of Small Satellites in Distributed Systems,' 13th AIAA/USU Conference on Small Satellites, Logan, Utah, 1999.

[24] A. Williams, S. Glavaski, and T. Samad, “Formations of Formations: Hierarchy and Stability”, *Proc. American Control Conference*, Boston, July 2004.

[25] A. Williams and S. Glavaski, “Connectivity and Convergence of Formations”, *Proc. American Control Conference*, Portland, June 2005.

[26] A. Williams, G. Lafferriere, and J.J.P. Veerman, “Stable Motions of Vehicle Formations”, Proc Conf. Decision and Control, December 2005.

[27] W.M. Wonham, *Linear Multivariable Control: A Geometric Approach*, 3rd Edition, Springer-Verlag, New York, 1985.

[28] A. Williams, G. Lafferriere, *Stability of Formations with Varying communication Graphs,* in preparation.