On the Cooperative Control of Multiple Unmanned Aerial Vehicles

by C.A. Rabbath[1,2], E. Gagnon[1,3] and M. Lauzon[1]
[1]Defence R & D Canada - Valcartier, QC,
[2]Dept. of Mechanical Engineering, McGill University-Montreal, QC,
[3]Dépt. de génie électrique et de génie informatique, Université Laval, Québec, QC.

Abstract

Unmanned aerial vehicles (UAVs) are rapidly becoming a strategic asset of today's military forces and an enabler of transformation for the civilian airspace community. Cooperative teaming will revolutionize the employment of UAVs by replacing a single vehicle, currently controlled by multiple human operators, with teams of cooperating UAVs monitored (and/or controlled) by a single operator. Therefore, a cooperative control scheme for a team of UAVs will have to ensure the success of team missions by autonomously and optimally assigning tasks to each vehicle and planning the routes and actions of each vehicle in a cohesive manner despite disturbances, such as strong wind gusts, and uncertainties, such as the presence of unpredictable dynamic and static obstacles. Cooperative control software and hardware will have to provide real-time performance with a relatively fast response. To achieve such objectives, there are a variety of challenges that must be overcome by the designers. This article describes some of the issues and challenges currently faced by researchers and developers of tomorrow's teams of aerial robotic systems, and presents possible solutions to the problems at hand. Of particular interest to the electrical and computer engineering community are the issues of communications and computing demands that must be somehow constrained in the design of a multi-UAV cooperative control scheme. Finally, it is important to stress that cooperative control requires a synergy among a variety of disciplines for its effective solution, namely mathematics, computer science, engineering (control, aerospace, electrical, mechanical, communications, robotics), and operations research.

Sommaire

Les drones, ou avions sans pilote, sont maintenant devenus des atouts stratégiques des forces militaires et sont en voie de transformer l'espace aérien civil. La coopération entre drones va révolutionner leur utilisation. Présentement, un véhicule est contrôlé par plusieurs opérateurs. Dans le futur, un opérateur pourra surveiller (et/ou contrôler) plusieurs équipes de drones. Pour ce faire, une stratégie de contrôle d'une équipe de drones devra ordonnancer les tâches et planifier le vol (trajectoire) de chacun des drones de façon autonome et optimale malgré les perturbations, tel que de très forts vents, et les incertitudes, tel que la présence d'obstacles statiques et dynamiques imprévisibles a priori. Le logiciel et le matériel pour le contrôle coopératif devront rencontrer des exigences de temps réel et offrir des réponses relativement rapides. Pour accomplir de tels objectifs, il y a une multitude de défis qui doivent être relevés par les concepteurs. Cet article présente quelques-uns des enjeux et des défis auxquels doivent faire face les chercheurs et les développeurs des systèmes robotiques aériens de demain, et présente des solutions possibles aux problèmes courants. Les défis concernant les communications entre véhicules et les demandes en temps de calculs seront particulièrement d'intérêt pour la communauté d'ingénieurs en électrique et informatique. Finalement, il est important de souligner que solutionner le problème de contrôle coopératif de drones demande une synergie entre plusieurs disciplines, comme les mathématiques, l'informatique, le génie (commande, aérospatiale, électrique, mécanique, communications, robotiques), et la recherche opérationnelle.

1.0 Introduction

Building upon rapid advances in robotics, control, communications and computer technology, UAVs will undoubtedly be called upon to play an ever-increasing role in the civilian and military communities. Military strategic planning has already started to incorporate wide-ranging roles for UAVs, such as tactical surveillance, communications relay, target designation, battle damage assessment and covert payload delivery [1]. In the future, teams of autonomous intelligent vehicles with common mission objectives will be integrated into military force structures. The problem of cooperative control of UAVs concerns the coherent and efficient maneuvering of each member of a group of aerial vehicles (the team) to successfully complete a mission with limited human intervention in a highly unstructured environment [2]. This can be achieved by devising control algorithms, implemented on digital hardware, that allocate tasks to each UAV member of the team, select flight paths and generate the trajectory for each member, and set attitude configurations to aerial vehicles at timely instants, at precise positions or during specific maneuvers such as evasion, combat, reconnaissance, take-off, landing, rendezvous and so on. The control objectives can be characterized as the optimization of a set of designer-specified global functions. A central motivation for the development of cooperative control schemes is that enabling UAV teaming should result in a more effective operational capability than that available through independent control of the UAVs. This idea, along with the need to leave it to a group of machines to effectively perform the dull, dangerous and dirty missions, are in fact the main drivers for the research in cooperative control today [3].

This article discusses some of the challenges currently faced by designers of cooperative control schemes for teams of UAVs, and presents possible solutions to the problems involved. An effective cooperative control strategy should provide close-to optimal, robust, real-time performance with a relatively fast response from the team. Specifically, decentralization of the cooperative control problem, to improve the team's robustness to failures and to reduce computing costs while satisfying global mission objectives, must be addressed. Team autonomy must be achieved via algorithms for the scheduling of tasks, the planning of vehicle paths and the generation of trajectories for each vehicle. Formulating a tractable optimization problem and effectively implementing cooperative control software with constrained inter-vehicle communications and computations, warranting real-time performance and a fast response, is critical. The latter is especially true for small-scale, expendable UAVs where lightweight, compactness and limited computing power and communications bandwidth are the norm. Finally, the article presents an experimental COTS testbed for the validation of the concepts and extensive testing in (quasi-) realistic scenarios.

2.0 Cooperative Control Of Multiple UAVs

Consider a set of $N \geq 2$ compact, light weight UAVs to be deployed either from a single location (ground or air) or from multiple locations. Each vehicle has its own intrinsic dynamic characteristics (time constants, aerodynamic coefficients, etc.), and computing, sensors, transmitters and actuators hardware. Hardware found on the vehicles is current and/or legacy technology. Prior to the UAVs being deployed, a certain level of information is assumed known and coded within each vehicle, such as information on the team members, the environment, the airspace and ground. Given a pre-specified list of global objectives, to be performed in a certain order and with certain constraints (on timing, location, fuel consumption, etc.), and limited knowledge, embedded in the on-board electronics of each UAV, of the environment (ground and air) and the team members, then the cooperative control problem can be described as follows:

To ensure the success of the team mission (in meeting global objectives) by appropriately (i.e. optimally) assigning tasks to each vehicle and planning the routes and actions of each vehicle (path planning, trajectory generation, low-level commands) in a cohesive manner (e.g. avoiding static and dynamic obstacles) despite disturbances (e.g. wind gusts) and uncertainties (e.g. limited information available to each vehicle).

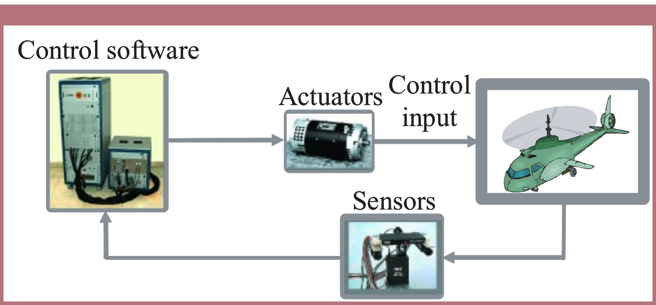


Figure 1: UAV Control System

The cooperative control problem is therefore a form of constrained optimization. The low-level UAV control system, such as the flight control system or autopilot/guidance (Figure 1). Figure 2 presents the schematics of a cooperative control scheme. The allocation of tasks, the planning of the path, the generation of the vehicle trajectories and the low-level control loop are feedback loops that are integral parts of the cooperative control strategy.

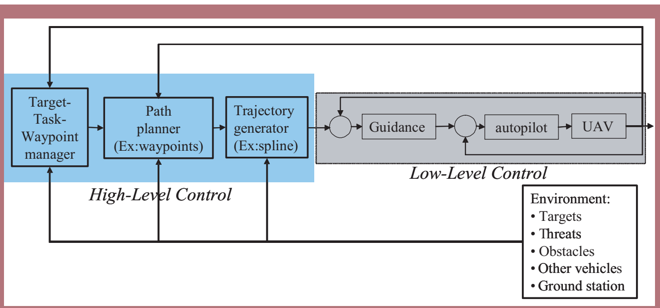


Figure 2: Block Diagram of a Cooperative Control Scheme

3.0 Team Autonomy In Task Allocation, Path Planning And Trajectory Generation

Team autonomy requires algorithms allocating tasks to team members, planning the path of each member and generating trajectories to ensure avoidance of static and dynamic obstacles (Figure 2) and such that prescribed target points are reached at some time instants. It must be emphasized that this procedure should be dynamic to react to changing conditions.

3.1 Task Allocation

To demonstrate a high level of autonomy, the UAV team receives a set of high-level orders from the ground control station (or has a set of pre-programmed orders) and deduces and manages a set of ordered tasks [4]. Task allocation can be exemplified as computing and assigning a set of waypoints to a sub-team of vehicles based on the information (vehicles states, waypoints locations, obstacles, etc.) known at mission pre-planning. However, as the mission is executed, the information about the environment, and the environment itself, will change. Thus, an optimal task allocation decided *a priori* to the execution of the mission will most likely become obsolete as the mission progresses, so the task allocation needs to be updated frequently during the mission. In addition, an approach obviating the need to re-consider the whole problem at once is necessary to avoid excessive computing demands on each vehicle. As suggested in [4], a sub-team problem can be solved, thus diminishing the computing demands, by involving only the vehicles having the largest influence on each other, such as those nearby the area of interest (e.g. close to a waypoint). Then, the task allocation can be based on a sub-team of UAVs, considering the local optimization problem within a prescribed 3D region.

The task allocation (or scheduling) problem can be stated as one of assigning a set of items or ordered tasks to each UAV such that an overall cost function is optimized while a maximum number of tasks is successfully accomplished, based on some knowledge of the environment. Due to the nature of this optimization problem (performing a search over all possibilities and obtaining the "best" solution is impractical), it is sometimes solved via heuristic techniques, such as greedy algorithms [5]. In general, heuristic techniques yield sub-optimal solutions to the problem at hand, although they provide finite-time computations, which increase in complexity as the number of vehicles and tasks increases (computing time increases as well). Graph theory can be used to model the problem, and integer programming can provide an algorithmic solution (Figure 3). In Figure 3, the problem is as follows. Given a set of waypoints/tasks, a 2-UAV team and the knowledge of the environment, perform the ordered set of tasks 1, 2, 3 and 4 as soon as possible.The numbers beside the arrows indicate path duration, the circles correspond to waypoints/tasks, the number of the task is in the top portion of the circle whereas the time required to perform a task is indicated in the bottom portion of the circles. The dotted lines correspond to the solution obtained with a heuristic method. It is clear that the time to execute the 4 tasks, in order, with the 2-UAV team is shorter than the time required with a single UAV.

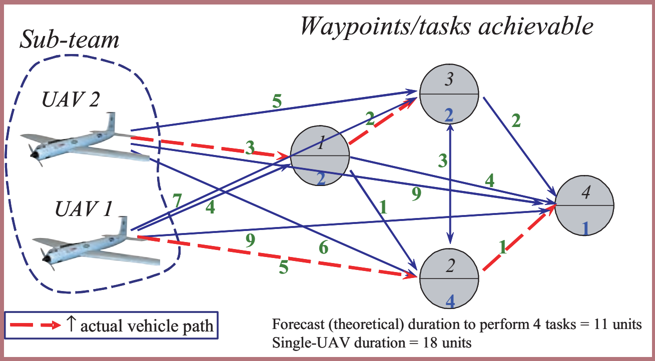

Sub-team
Waypoints/tasks achievable
UAV 2
UAV 1
actual vehicle path
Forecast (theoretical) duration to perform 4 tasks = 11 units
Single-UAV duration = 18 units

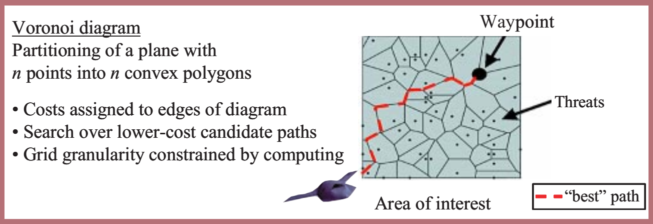
Figure 3: Example of Task Allocation

3.2 Path Planning & Trajectory Generation

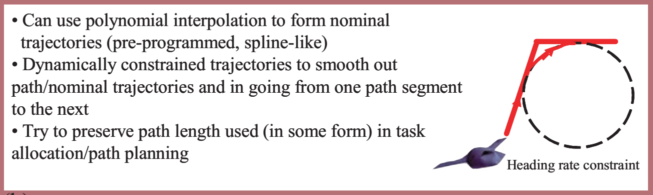
Path planning and trajectory generation can be tackled in combination. UAV trajectories are typically described by an ordered sequence of vehicle states, such as position and speed, over a certain time period. For example, the vehicle path can be based on the goal of reducing the exposure of each vehicle to ground threats, or to allow for the reconnaissance of several objects. With the knowledge of the waypoints and of the states of the nearby UAVs at these points, a trajectory generation algorithm then ensures that each UAV follows the prescribed trajectory. The algorithms must calculate trajectories that respect the vehicles' dynamics (constraints on vehicle states and allowed inputs) as well as the objectives of avoiding static and dynamic obstacles, and reaching a certain set of destinations at prescribed time instants. Specific trajectory profiles can be pre-programmed into each UAV, such as straight level flight or loitering-type flight. Trajectory generation should be updated at a rate fast enough to allow the UAVs to react, in real-time, to the dynamic environment. An approach allowing the UAVs to effectively react to the changing environment is the so-called receding horizon control strategy, where, at every sampling instant, an optimization problem is solved over a finite time horizon while considering updates of the optimization variables and the constraints. Figure 4 illustrates one approach to the combined path planning and trajectory generation problem for a single UAV. First, construct a grid of the region and obtain a set of best paths to reach the waypoints (i.e. intermediate waypoints) by using Voronoi diagrams [6], as shown in Figure 4(a). Second, generate a time trajectory, as shown in Figure 4(b). The challenge is to attribute flight paths to a team of UAVs in real-time and simultaneously.

4.0 Decentralization

A centralized approach in solving the cooperative control problem requires either a single vehicle with large computing capacity, to carry out the bulk of the computations, or a ground station equipped with heavy computing equipment, with which every vehicle in the team must be able to communicate. Either way, the communication costs are enor-


Voronoi diagram
Partitioning of a plane with n points into n convex polygons
• Costs assigned to edges of diagram
• Search over lower-cost candidate paths
• Grid granularity constrained by computing
Waypoint
Threats
Area of interest
"best" path

(a)


• Can use polynomial interpolation to form nominal trajectories (pre-programmed, spline-like)
• Dynamically constrained trajectories to smooth out path/nominal trajectories and in going from one path segment to the next
• Try to preserve path length used (in some form) in task allocation/path planning
Heading rate constraint

(b)

Figure 4: Path Planning & Trajectory Generation

mous, and the lack of robustness of such an approach, that is the fact that the team is highly sensitive to failure of the main computing unit, is a major drawback. Figure 5 illustrates the centralized control of a 3-UAV team, where most of the computations are carried out on-board UAV 1. It is clear that failure of UAV 1 would jeopardize the success of the mission.

The alternative is decentralized control, where the solution to the optimization problem is decomposed into a set of N sub-problems, for instance one per UAV [7]. With decentralized control, the control and the data are distributed among the vehicles. Each team member has its own controller. While no one controller has enough local memory or computational power to solve the entire cooperative control, the team as a whole can. The controllers are somehow coupled, that is they share some information during flight. A major advantage of the decentralized strategy is the gain in the system's fault tolerance, as the robustness of the team to the loss of one or more UAVs during the mission is increased. However, the difficulty lies in determining 1) the decomposition of the original global optimization, 2) the information that should be exchanged among the UAVs, and 3) a way to warrant satisfactory team performance. Decentralization for a 3-UAV team is shown in Figure 6. Each UAV computes a portion of the entire cooperative control problem. The amount of data exchanged among the UAVs depends on the level of decentralization. A highly decentralized cooperative control approach requires minimal data exchanges whereas a lowly decentralized approach requires a maximum exchange of data (i.e. all of the available information). The trade-off between computing/communications costs and the level of decentralization is thus clear.

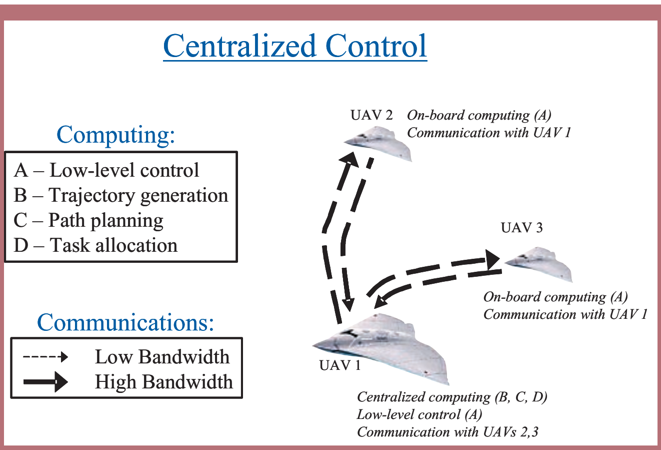


Figure 5: Centralized Control for a 3-UAV Team

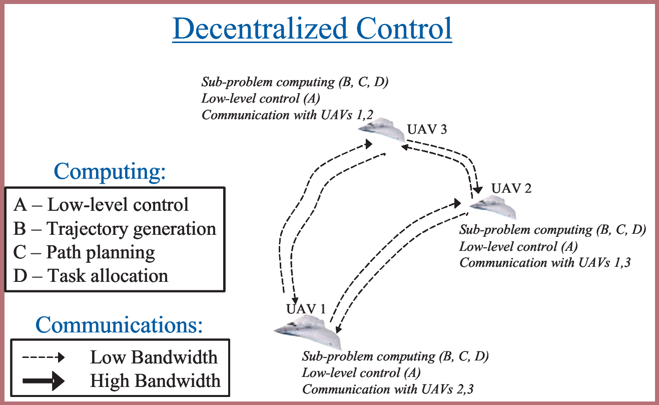


Figure 6: Decentralized Control of a 3-UAV Team

5.0 Computing & Communications

A major challenge with cooperative control, especially for teams with a relatively large number of UAVs, is the effective handling of the computations and the communications such that a fast response is obtained from the team. The control strategy should ideally result in real-time performance with control update and sampling rates that are relatively fast. By real-time execution, it is understood that the computations of the cooperative control algorithms and the communications are performed in a sequenced order at every iteration cycle over a predictable, bounded time period in an irreversible manner.

5.1 Computing

The sustained growth in computational processing power fosters a continuous evolution of control algorithms towards more complex, more capable, more robust solutions. However, UAVs' on-board systems will always have some limitation in regard to the implementation of complex real-time control strategies. Furthermore, in practice, designers strive for control strategies that provide real-time performance at the lowest costs, usually available with low-processing speeds hardware. In real-time control, overruns indicate that the actual execution of the real-time tasks, for each iteration step, has a longer duration than the sampling period. The designer must make sure that computing, communications and conversion times, for the given operating system (having its own figures of latency, jitter and task switching timings), digital hardware and input-output cards, can be performed within the sampling intervals selected to run the digital control algorithms. If overruns occur, then the designer may be forced to increase the sampling period and/or to redesign the control law. An alternative is to select a more powerful digital platform and/or communication medium, whenever it is possible. The issue of high computing demand associated with sophisticated UAV cooperative control strategies is well known [3]. The challenge thus lies in the design of an effective control system that works with the available hardware; that is, a control system warranting fast response as well as real-time performance. In small-scale UAVs, where light weight, compactness and low energy consumption are typical requirements, the control laws should generally be as simple as possible. There is therefore an expected trade-off between the performance level attained with a cooperative control scheme and the threshold on complexity, in both computing and communications. Ways of relaxing this trade-off include the combination of the cooperative control algorithms with real-time distributed and parallel processing within and among UAVs, and the use of dynamic task scheduling. Furthermore, the use of constrained UAV spatial horizons may enable a reduction in computing demands. Algorithm partitioning and distribution support heterogeneous processor architectures, thereby enabling the teaming of legacy UAVs. Figure 7 illustrates the concept of distributed computations for a team of 7 UAVs. At the given time step, 3 sub-teams are defined. The communications, needed to perform the distribution of the cooperative control problem among the UAVs, is shown with dotted lines. The intermittent communications that may arise between any sub-team and/or UAV and with a ground operator is not shown. Figure 7 shows the schematics of the parallel computing carried out within a 2-UAV team (that of UAV 3 and UAV 7), where the cooperative control algorithms are partitioned among the available hardware in order to provide reduced time step requirements and to use more efficiently the

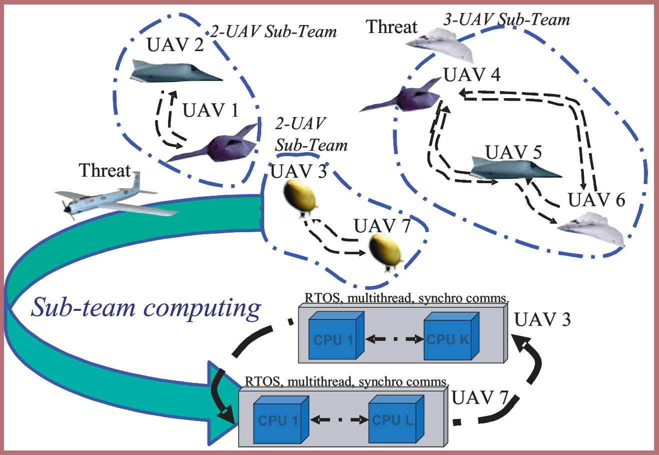


Figure 7: Distributed Computing within the 2-UAV Sub-team

computing resources. It should be noted that parallelism of the computations is achieved in two ways: first, within a UAV, provided it is equipped with a cluster of compact, low-power processors and DSPs connected in a parallel architecture, and, second, among the UAVs of a sub-team, when the inter-UAV communications required for such parallel processing are allowed. The partitioning of the algorithms can be done via heuristic methods, such as greedy algorithms, and more evolved Meta heuristics [5], and via feedback control concepts.

5.2 Communications

The various nodes (CPUs, DSPs, FPGAs, etc.) in a distributed real-time control system transmit messages to and receive messages from each other. A node may be a single UAV, a single CPU within a UAV, or a sub-team of UAVs exchanging with another sub-team of UAVs. There are inherent communication delays between nodes, which may vary due to network load, message priorities, and so on. Depending on the communications type, wireless Ethernet, RS-232, FireWire (IEEE 1394) or Giganet's cLAN for instance, there is a different probability distribution of the communication delays. The real-time operating system should be such that the interrupts are guaranteed to be handled within a certain specified maximum time. Figure 8 illustrates the communications that might be required with a cooperative control strategy not taking into account the requirement of constrained communications bandwidth. In the figure, the dotted lines constitute 2-way real-time communications between two UAVs. It should be noted that these communications are required to solve the real-time control problem without the notion of a sub-team. It is clear from Figure 8 that each UAV communicates, at every time step, with all of its teammates. Such exchanges will most probably result in communication delays, excessive use of communication channels and slow response to threats.

However, by adhering to a finite horizon strategy (i.e. a sub-team of vehicles which exchange data in real-time to solve a sub-optimization problem), communications among UAVs can be effectively constrained. In Figure 9, the spatial horizon of UAV i is given by a circle of radius ri, i =1, 2, 3. At time step k, UAV 1 forms a sub-team with UAV 2; that is, UAV 1 is within the spatial horizon of UAV 2, and UAV 2 lies inside the horizon of UAV 1. Therefore, UAVs 1 and 2 share data in real-time to solve a common optimization and to calculate their respective paths, via parallel computing. It should be noted that the optimization problem for UAVs 1 and 2, at time step k, considers the union of the regions covered by the horizons of both UAVs 1 and 2, which includes UAV 3, as seen in Figure 9. Another sub-team is that of the single UAV 3, which does not receive information from the other two vehicles to compute its path since they lie outside of its horizon. Still, UAV 3 could share information with the rest of the team in an ad-hoc, intermittent fashion.

6.0 Experimental Validation

Commercial software such as Matlab/Simulink^{TM} provides an effective modeling and simulation framework in which to test the cooperative control approaches. With such software, the vehicle dynamics, the interactions among the vehicles and with the environment, and the control laws can be modeled. Furthermore, a variety of test scenarios and fault/uncertain conditions can be implemented and simulated. For example, operationally relevant mission scenarios (e.g. littoral surveillance and interdiction) and performance metrics (e.g. persistence of sensor-coverage over targets, and percentage of targets found) can be verified. In the simulation studies, dynamic aspects should include random time-critical targets and behavior that necessitates team adaptation. Performance metrics should include measurements of the UAV team cooperation and interaction levels. Analysis can be performed to determine how to trade-off performance with team size, on-board processing power and available inter-UAV bandwidth.

Despite the importance of modeling and simulation in the cooperative control design process, control laws can only be truly verified with an experimental testbed involving actual on-board UAV electronics and (quasi-) realistic operating conditions. Figure 10 presents an experimental set-up where an actual small-scale UAV mock-up (remote controlled aerial vehicle) sits on a platform that comprises a set of load cells. These instruments measure the forces acting on the airframe for various commands of aileron, elevator, rudder and motor speed. The cells are connected to a digital computer via acquisition cards and converters. The feedback loop then involves measurement of forces (and computations of moments) for the generation of appropriate commands to the

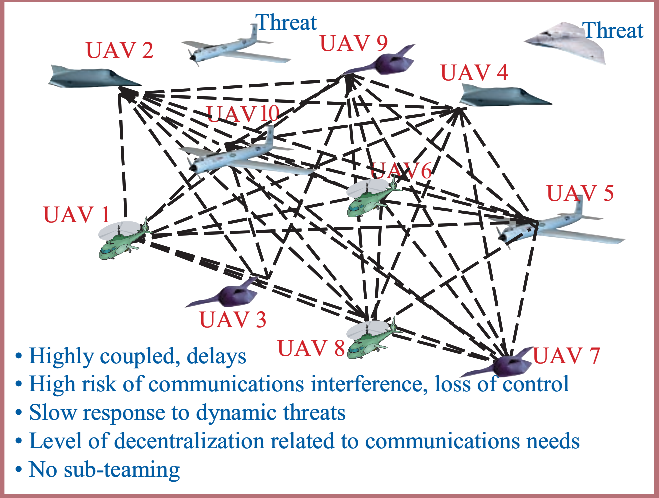

UAV 2
Threat
UAV 9
Threat
UAV 4
UAV10
UAV6
UAV 5
UAV 1
UAV 3
UAV 8
UAV 7
• Highly coupled, delays
• High risk of communications interference, loss of control
• Slow response to dynamic threats
• Level of decentralization related to communications needs
• No sub-teaming

Figure 8: Full Bandwidth Communications

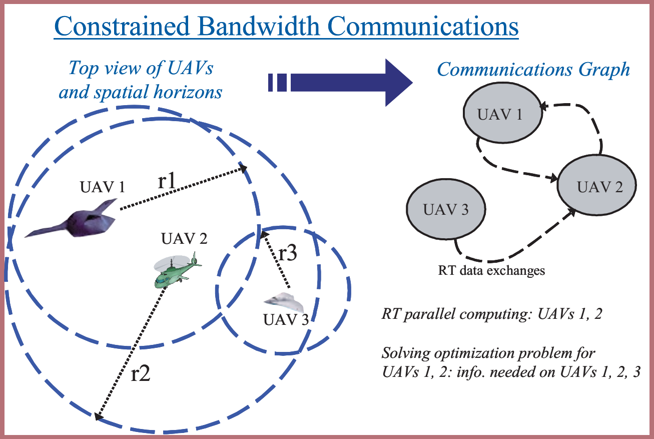

Constrained Bandwidth Communications
Top view of UAVs and spatial horizons
Communications Graph
UAV 1
r1
UAV 2
r3
UAV 3
r2
UAV 1
UAV 2
UAV 3
RT data exchanges
RT parallel computing: UAVs 1, 2
Solving optimization problem for UAVs 1, 2: info. needed on UAVs 1, 2, 3

Figure 9: Constrained Bandwidth Communications