George Vachtsevanos, Ben Ludington, Johan Reimann, Panos Antsaklis, Kimon Valavanis, “Modeling and Control of Unmanned Aerial Vehicles– Current Status and Future Directions,” Workshop on Modeling and Control of Complex Systems (MCCS), Ayia Napa, Cyprus, June 30-July 1, 2005. Also Chapter 9, in Modeling and Control of Complex Systems, CRC Press 2007.

Modeling and Control of Unmanned Aerial Vehicles – Current Status and Future Directions

George Vachtsevanos, Ben Ludington, Johan Reimann, Georgia Institute of Technology
Panos Antsaklis, Notre Dame University
Kimon Valavanis, University of South Florida

Abstract

Recent military and civil actions worldwide have highlighted the potential utility for Unmanned Aerial Vehicles (UAVs). Both fixed wing and rotary aircraft have contributed significantly to the success of several military and surveillance/rescue operations. Future combat operations will continue to place unmanned aircraft in challenging conditions such as the urban warfare environment. However, the poor reliability, reduced autonomy and operator workload requirements of current unmanned vehicles present a roadblock to their success. It is anticipated that future operations will require multiple UAVs performing in a cooperative mode, sharing resources and complementing other air or ground assets. Surveillance and reconnaissance tasks that rely on UAVs require sophisticated modeling, planning and control technologies. This paper reviews the current status of UAV technologies with emphasis on recent developments aimed at UAV improved autonomy and reliability and discusses future directions and technological challenges that must be addressed in the immediate future. We view the assembly of multiple and heterogeneous vehicles as a “system of systems” where individual UAVs are functioning as sensors or agents. Thus, networking, computing and communications issues must be considered as the UAVs are tasked to perform surveillance and reconnaissance missions in an urban environment. The same scenario arises in similar civil applications such as forest fire detection, rescue operations, pipeline monitoring, etc. A software (middleware) platform enables real time reconfiguration, plug-and-play and other quality of service functions. Multiple UAVs, flying in a swarm, constitute a network of distributed (in the spatio-temporal sense) sensors that must be coordinated to complete a complex mission. Current R&D activities are discussed that concern issues of modeling, planning and control. Here, optimum terrain coverage, target tracking and adversarial reasoning strategies require new technologies to deal with issues of system complexity, uncertainty management and computational efficiency [Vachtsevanos, et al, 2004]. We will pose the major technical challenges arising in the “system of systems” approach and state the need for new modeling, networking, communications and computing technologies that must be developed and validated if such complex unmanned systems as UAVs are to perform effectively and efficiently, in conjunction with manned systems, in a variety of application domains. We will conclude by proposing possible solutions to these challenges.

I. INTRODUCTION

The future urban warfare , as well search and rescue, border patrol, Homeland security and other applications, will utilize an unprecedented level of automation in which human-operated, autonomous, and semi-autonomous air and ground platforms will be linked through a coordinated control system. Networked UAVs bring a new dimension to future combat systems that must include adaptable operational procedures, planning and deconfliction of assets coupled with the technology to realize such concepts. The technical challenges the control designer is facing for autonomous collaborative operations stem from real-time sensing, computing and communications requirements, environmental and operational uncertainty, hostile threats and the emerging need for improved UAV and UAV team autonomy and reliability. Figure 1 shows the autonomous control level trend according to the DoD UAV Roadmap [Office of Secretary of Defense, 2002]. The same roadmap details the need for new technologies that will address single vehicle and multi-vehicle autonomy issue. The challenges increase significantly as we move up the hierarchy of the chart shown in Figures 2 (a) and (b) from single vehicle to multi-vehicle coordinated control. Moderate success has been reported thus far in meeting the lower echelon challenges. Achieving the ultimate goal of full autonomy for a swarm of vehicles executing a complex surveillance and reconnaissance mission still remains a major challenge. To meet these challenges, innovative coordinated planning and control technologies such as distributed artificial intelligence (DAI), computational intelligence and soft computing, as well as game theory and dynamic optimization, have been investigated intensively in recent years. However, in this area, more work has been focused on solving particular problems, such as formation control and autonomous search, while less attention has been paid to the system architecture, especially from an implementation and integration point of view. Other significant concerns relate to inter-UAV communications, links to command and control, contingency management, etc.

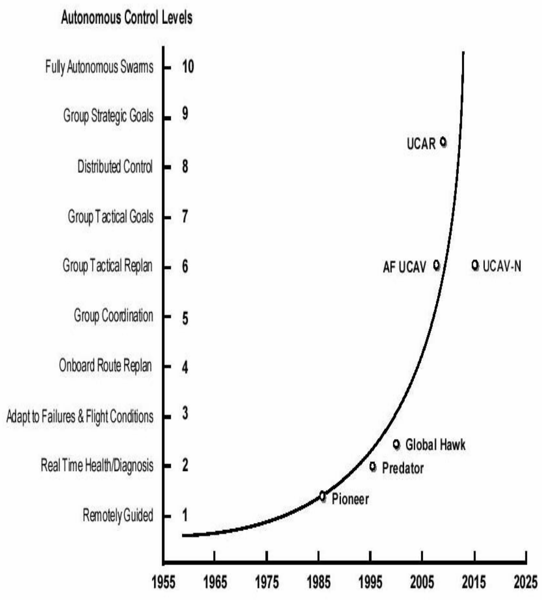

Figure 1: Autonomous Control Level Trend

Autonomous Control Level (ACL) Chart

Note: As ACL increases, capability includes, or replaces, items from lower levels

Level	Level Descriptor	Observe Perception / Situational Awareness	Orient Analysis / Coordination	Decide Decision Making	Act Capability
10	Fully Autonomous	Cognizant of all within Battlespace	Coordinates as necessary	Capable of total independence	Requires little guidance to do job
9	Battlespace Swarm Cognizance	Battlespace inference – Intent of self and others (allies and foes). Complex Intense environment – on-board tracking	Strategic group goals assigned. Enemy Strategy inferred.	Distributed tactical group planning. Individual determination of tactical goal. Individual task planning/execution. Choose tactical targets	Group accomplishment of strategic goal with no supervisory assistance
8	Battlespace Cognizance	Proximity inference – Intent of self and others (allies and foes). Reduced dependence upon off-board data.	Strategic group goals assigned Enemy tactics inferred ATR	Coordinated tactical group planning Individual task planning / execution Choose targets of opportunity	Group accomplishment of strategic goal with minimal supervisory assistance
7	Battlespace Knowledge	Short track awareness – History and predictive battlespace data in limited range, timeframe and numbers.	Tactical group goals assigned. Enemy Trajectory estimated.	Individual task planning / execution to meet goals	Group accomplishment of tactical goal with minimal supervisory assistance
6	Real-Time Multi-Vehicle Cooperation	Ranged awareness – Onboard sensing for long rang, supplemented by off-board data	Tactical group goals assigned Enemy location sensed / estimated	Coordinated trajectory planning and execution to meet goals – Group optimization	Group accomplishment of tactical goal with minimal supervisory assistance
5	Real-Time Multi-Vehicle Coordination	Sensed awareness – Local sensor to detect others, fused with off-board data	Tactical group plan assigned RT Health Diagnosis; Ability to compensate for most failures and flight conditions; Ability to predict onset of failures (e.g. Prognostic Health Mgmt) Group diagnosis and resource management	On-board trajectory replanning – Optimize for current and predictive conditions Collision avoidance	Group accomplishment of tactical plan as externally assigned Air collision avoidance Possible close air space separation for AAR, formation in non-thread conditions

Figure 2(a): The Autonomous Control Level Chart

Autonomous Control Level (ACL) Chart

Note: As ACL increases, capability includes, or replaces, items from lower levels

Level	Level Descriptor	Observe Perception / Situational Awareness	Orient Analysis / Coordination	Decide Decision Making	Act Capability
4	Fault / Event Adaptive Vehicle	Deliberate Awareness – Allies communicate data	Tactical plan assigned Assigned Rules of Engagement RT Health Diagnosis. - Ability to compensate for most failures and flight conditions – inner loop changes reflected in outer loop performance	On-board trajectory replanning – Event driven, Self resource management, Deconfliction	Self accomplishment of tactical plan as externally assigned Medium vehicle airspace separation
3	Robust Response to Real-Time Faults / Events	Health / Status history and models	Tactical plan assigned RT Health Diagnostics Ability to compensate for most control failures and flight conditions	Evaluate status vs. required mission capabilities Abort / RTB if insufficient	Self accomplishment of tactical plan as externally assigned
2	Changeable Mission	Health / Status sensors	RT Health diagnosis Off-board replan	Execute preprogrammed or uploaded plans in response to mission and health conditions	Self accomplishment of tactical plan as externally assigned
1	Execute Preplanned Mission	Preloaded mission data Flight Control and Navigation Sensing	Pre / Post Flight BIT Report status	Preprogrammed mission and abort plans	Wide airspace separation requirements
0	Remotely Piloted Vehicle	Flight Control sensing Nose camera	Telemetered data Remote pilot commands	N/A	Control by remote pilot

Figure 2(b): The Autonomous Control Level Chart

We will review briefly in this paper a few of the challenges referred to above and suggest possible approaches to these problems. The intent is to motivate through application examples the modeling, control and communication concerns and highlight those new directions that are needed to assist in arriving at satisfactory solutions. We will emphasize the synergy of tools and methodologies stemming from various domains as well as the resurfacing of classical mathematical notions that may be called upon now to solve difficult spatio-temporal dynamic situations. Recent advances in computing and communications promise to accommodate the on-line real time implementation of such mathematical algorithms that were considered intractable some years back.

II. System Architecture

While networked and autonomous UAVs can be centrally controlled, this requires that each UAV communicates all the data from its sensors to a central location and receives all the control signals back. Network failures and communication delays are one of the main concerns in the design of cooperative control systems. On the other hand, distributed intelligent agent systems provide an environment in which agents autonomously coordinate, cooperate, negotiate, make decisions and take actions to meet the objectives of a particular application or mission. The autonomous nature of agents allows for efficient communication and processing among distributed resources.

For the purpose of coordinated control of multiple UAVs, each individual UAV in the team is considered as an agent or sensor with particular capabilities engaged in executing a portion of the mission. The primary task of a typical team of UAVs is to execute faithfully and reliably a critical mission while satisfying local survivability conditions. In order to define the application domain, we adopt an assumed mission scenario of a group of UAVs executing reconnaissance and surveillance (RS) missions in an urban warfare environment, as depicted in Figure 3.

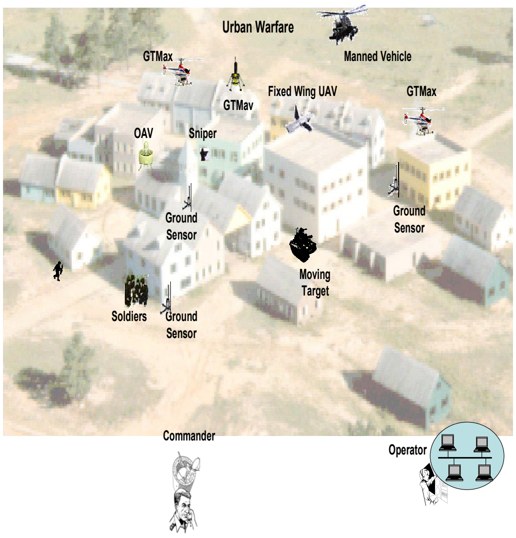

Figure 3: A Team of 5 UAVs Executing RS Missions in an Urban Warfare Environment

A "system of systems" approach suggests a hierarchical architecture for the coordinated control of multiple UAVs. The hierarchical architecture, shown in Figure 4, features an upper level with global situation awareness and team mission planning, a middle level with local knowledge, formation control and obstacle avoidance, and a low level that interfaces with onboard baseline controllers, sensors, communication and weapon systems. Each level consists of several interacting agents with dedicated functions. The formation control problem is viewed as a Pursuit Game of n pursuers and n evaders. Stability of the formation of vehicles is guaranteed if the vehicles can reach their destinations within a specified time, assuming that the destination points are avoiding the vehicles in an optimal fashion. Vehicle model is simplified to point mass with acceleration limit. Collision avoidance is achieved by designing the value function so that it ensures that the vehicles move away from one another when they come too close to each one. Simulation results are provided to verify the performance of the proposed algorithms.

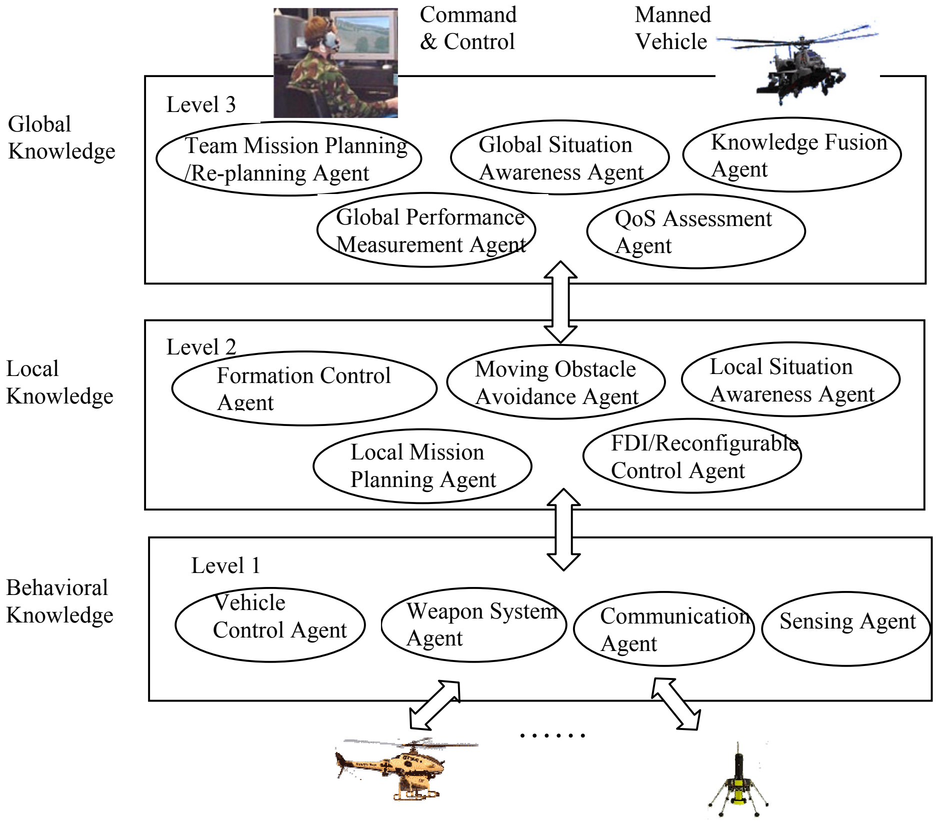

Figure 4: A Generic Hierarchical Multi-agent System Architecture

The highest level of the control hierarchy features functions of global situation awareness and teamwork. The mission planning agent is able to generate and refine mission plans for the team, generate or select flight routes, and create operational orders. It is also responsible for keeping track of the team's plan, goals, and team members' status. The overall mission is usually planned by the command and control center based on the capabilities of each individual UAV agent, and is further decomposed into tasks/subtasks which are finally allocated to the UAV assets (individually or in coordination with other vehicles). This can usually be cast as a constrained optimization problem and tackled with various approaches, such as integer programming, graph theory, etc. Market based methods [Dunbar and Murray, 2002] [Voos, 1999] and especially auction theory [Clearwater, 1996], [Walsh and Wellman, 1998], [Engelbrecht, et. Al 1983] can be applied as a solution to autonomous mission re-planning.

Planning the UAVs' flight route is also an integral part of mission planning. A modified A* search algorithm, which attempts to minimize a suitable cost function consisting of the weighted sum of distance, hazard and maneuverability measures [Bertsekas, 1992], [Vachtsevanos et. al, 1997], can be utilized to facilitate the design of the route planner. In the case of a leader-follower scenario, an optimal route is generated for the leader, while the followers fly in close formation in the proximity of the leader. The global situation awareness agent, interacting with the knowledge fusion agent, evaluates the world conditions based on data gathered from each UAV (and ground sensors if available) and reasons about the enemy's likely actions. Adversarial reasoning and deception reasoning are two important tasks executed here. The global performance measurement agent measures the performance of the team and suggests team re-configuration or mission re-planning, whenever necessary. Quality of service (QoS) is assessed to make the best effort to accomplish the mission and meet the predefined quality criteria. Real world implementation of this level is not limited to the agents depicted in the figure. For example, in heterogeneous agent societies, knowledge of coordination protocols and languages may also reside [Sousa and Pereira, 2003].

III. Formation Control

The problem of finding a control algorithm, which will ensure that multiple autonomous vehicles can maintain a formation while traversing a desired path and avoid inter-vehicle collisions, will be referred to as the formation control problem. The formation control problem has recently received considerable attention due in part to its wide range of applications in aerospace and robotics. A classic example involving the implementation of the virtual potential problem is presented in [Howard et. al, 2000]. The authors performed simulations on a two-dimensional system, which proved to be well behaved. However, as they mention in their conclusion, the drawback of the virtual potential function approach is the possibility of being “trapped” in local minima. Hence, if local minima exist, one cannot guarantee that the system is stable. In [Baras, et. al, 2003], the individual trajectories of autonomous vehicles moving in formation were generated by solving the optimal control problem at each time step. This is computationally demanding and hence not possible to perform in real-time with current hardware.

This paper views the formation control problem from a two player differential game perspective, which provides a framework to determine acceptable initial vehicle deployment conditions but also, provides insight into acceptable formation maneuvers that can be performed while maintaining the formation.

The formation control problem can be regarded as a Pursuit Game, except that, it is in general, much more complex in terms of the combined dynamical equations, since the system consists of n pursuers and n evaders instead of only one of each. However, if the group of vehicles is viewed as the pursuer and the group of desired points in the formation as the evader, the problem is essentially reduced to the standard but much more complex pursuit game.

Differential Game Theory was initially used to determine optimal military strategies in continuous time conflicts governed by some given dynamics and constraints [Isaacs, 1965]. One such application is the so-called Pursuit Game in which a pursuer has to collide with an evading target. Naturally, in order to solve such a problem it is advantageous to know the dynamics and the positional information of both the evader and the pursuer, that is, the Pursuit Game will be viewed as a Perfect Information Game.

Stability of the formation of vehicles is guaranteed if the vehicles can reach their destination within some specified time, assuming that the destination points are avoiding the vehicles in an optimal fashion. It seems counterintuitive that the destination points should be avoiding the vehicles optimally, however if the vehicles can reach the points under such conditions then they will always be able to reach their destination.

As a consequence of our stability criterion, it is necessary not only to determine the control strategies of the vehicles but also the optimal avoidance strategies of the desired points. Let us label the final control vector of the vehicles by $\bar{\phi}$ and the control final vector of the desired points by $\bar{\psi}$. Then, the main equation which has to be satisfied is:

$$\min_{\phi}\max_{\psi}\left[\sum_j V_j \cdot f_j(\vec{x},\phi,\psi) + G(\vec{x},\phi,\psi)\right] = 0 \qquad (1)$$

which has to be true for both $\bar{\phi}$ and $\bar{\psi}$.

The $f_j(\vec{x},\phi,\psi)$ term is the jth dynamic equation governing the system, and the V_j is the corresponding Value of the game. $G(\vec{x},\phi,\psi)$ is a predetermined function which, when integrated, provides the payoff of the game. Notice, that the only quantity that is not specified in the equation is the V_j term.

George Vachtsevanos, Ben Ludington, Johan Reimann, Panos Antsaklis, Kimon Valavanis, “Modeling and Control of Unmanned Aerial Vehicles– Current Status and Future Directions,” Workshop on Modeling and Control of Complex Systems (MCCS), Ayia Napa, Cyprus, June 30-July 1, 2005. Also Chapter 9, in Modeling and Control of Complex Systems, CRC Press 2007.

From the main equation it is possible to determine the retrograde path equations (RPEs), which will have to be solved to determine the actual paths traversed by the vehicles in the formation. However, initial conditions of the retrograde path equations will have to be considered in order to integrate the RPEs. These initial condition requirements provide us with the ability to introduce tolerance boundaries, within which we say that the formation has settled. Such boundaries naturally add complexity to the problem, however they also provide a framework for positional measurement errors.

The above formulation suggests a way for approaching the solution to differential game. However, how does one ensure that inter-vehicle collisions are avoided? To ensure this, it is necessary to consider the payoff function determined by the integral of $\boldsymbol{G}(\vec{x},\boldsymbol{\phi},\boldsymbol{\psi})$. As an example, if we simply seek that the vehicles must reach their goal within a certain time τ, then $\boldsymbol{G}(\vec{x},\boldsymbol{\phi},\boldsymbol{\psi})=1$. This can be verified by evaluating $\int_0^{\tau} \boldsymbol{G}(\vec{x},\boldsymbol{\phi},\boldsymbol{\psi})dt=\tau$. Hence, we have restricted our solutions to the initial vehicle deployment, which will ensure that the vehicles will reach the desired points in τ time. However, if $\boldsymbol{G}(\vec{x},\boldsymbol{\phi},\boldsymbol{\psi})$ is changed to penalize proximity of vehicles to one-another, only initial conditions that ensure collision free trajectories will be valid.

However, $\boldsymbol{G}(\vec{x},\boldsymbol{\phi},\boldsymbol{\psi})$ does not provide the means to perform the actual collision avoidance, but merely limits the solution space. So, in order to incorporate collision avoidance into the controller, one can either change the value function or add terms to the system of dynamic equations.

IV. Two-Vehicle Example

In order to illustrate some of the advantages and disadvantages with the differential game approach to formation control, consider the following system of simple point “Helicopters”, that is, points that can move in three dimensions governed by the following dynamic equations:

$$
\begin{aligned}
\dot{x}_i &= v_{xi} \\
\dot{v}_{xi} &= F_i \cos(\phi_{2i-1})\sin(\phi_{2i}) - k_i \cdot v_{xi} \\
\dot{y}_i &= v_{yi} \\
\dot{v}_{yi} &= F_i \sin(\phi_{2i-1})\sin(\phi_{2i}) - k_i \cdot v_{yi} \\
\dot{z}_i &= v_{zi} \\
\dot{v}_{zi} &= F_i \cos(\phi_{2i}) - k_i \cdot v_{zi}
\end{aligned}
$$

Where $\boldsymbol{i}=1,2$.

The two desired “points” are described by one set of dynamic equations. This simply implies that there is a constant distance separating the two desired points, and that the formation can only perform translations and not rotations in the three dimensional space. Hence the dynamic equations become:

$$
\begin{aligned}
\dot{x}_d &= v_{xd} \\
\dot{v}_{xd} &= F_d \cos(\psi_1)\sin(\psi_2) - k_d \cdot v_{xd} \\
\dot{y}_d &= v_{yd} \\
\dot{v}_{yd} &= F_d \sin(\psi_1)\sin(\psi_2) - k_d \cdot v_{yd} \\
\dot{z}_d &= v_{zd} \\
\dot{v}_{zd} &= F_d \cos(\psi_2) - k_d \cdot v_{zd}
\end{aligned}
$$

In the above dynamical systems, the $\boldsymbol{k_i}$ and $\boldsymbol{k_d}$ factors are simply linear drag terms to ensure that the velocities are bounded, and the $\boldsymbol{F_d}$ and $\boldsymbol{F_i}$ terms are the magnitudes of the applied forces. Figure 5 shows the coordinate system and the associated angles.

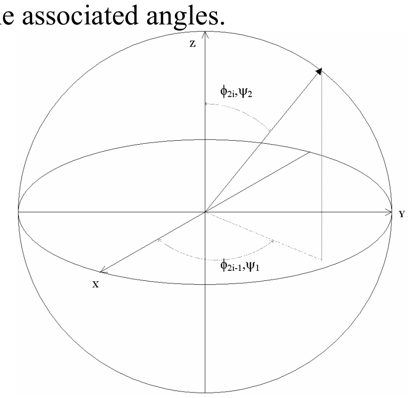

Figure 5: Definition of Angles

Substituting the dynamical equations into the main equation (1), we obtain the following expressions:

$$\min_{\phi}[F_1 \cdot (V_{vx1} \cdot \cos(\phi_1) \cdot \sin(\phi_2) + V_{vy1} \cdot \sin(\phi_1) \cdot \sin(\phi_2) + V_{vz1} \cdot \cos(\phi_2)) + F_2 \cdot (V_{vx2} \cdot \cos(\phi_3) \cdot \sin(\phi_4) + V_{vy2} \cdot \sin(\phi_3) \cdot \sin(\phi_4) + V_{vz2} \cdot \cos(\phi_4))]$$

$$And$$

$$\max_{\psi}[F_d \cdot (V_{vxd} \cdot \cos(\psi_1) \cdot \sin(\psi_2) + V_{vyd} \cdot \sin(\psi_1) \cdot \sin(\psi_2) + V_{vzd} \cdot \cos(\psi_2))] \qquad (2)$$

To obtain the control law that results from the max-min solution of equation (2), the following lemma is used:

Lemma 1:

Let $\boldsymbol{a}, \boldsymbol{b} \in \Re$:

Then $\boldsymbol{\rho} = \sqrt{\boldsymbol{a}^2 + \boldsymbol{b}^2}$

is obtained where

$$\max_{\boldsymbol{\theta}}(\boldsymbol{a} \cdot \cos(\boldsymbol{\theta}) + \boldsymbol{b} \cdot \sin(\boldsymbol{\theta}))$$

$$\cos(\boldsymbol{\theta}) = \frac{\boldsymbol{a}}{\boldsymbol{\rho}}, \text{and } \sin(\boldsymbol{\theta}) = \frac{\boldsymbol{b}}{\boldsymbol{\rho}}$$

and the max is $\boldsymbol{\rho}$

By combining Lemma 1 with Equation 2, the following control strategy for vehicle 1 is found:

$$\cos(\overline{\phi}_1) = -\frac{V_{vx1}}{\rho_1}, \sin(\overline{\phi}_1) = -\frac{V_{vy1}}{\rho_1}$$

$$\cos(\overline{\phi}_2) = -\frac{V_{vz1}}{\rho_2}, \sin(\overline{\phi}_2) = -\frac{\rho_1}{\rho_2}$$

Where

$$\rho_1 = \sqrt{V_{vx1}^2 + V_{vy1}^2}$$

$$and$$

$$\rho_2 = \sqrt{V_{vx1}^2 + V_{vy1}^2 + V_{vz1}^2}$$

Similar results are obtained for vehicle 2. For the optimal avoidance strategy of the desired points, we obtain the following:

$$\cos(\overline{\psi}_1) = +\frac{V_{vxd}}{\rho_{d1}}, \sin(\overline{\psi}_1) = +\frac{V_{vyd}}{\rho_{d1}}$$

$$\cos(\overline{\psi}_2) = +\frac{V_{vzd}}{\rho_{d2}}, \sin(\overline{\psi}_2) = +\frac{\rho_{d1}}{\rho_{d2}}$$

From this, we see that the retrograde equations have the following form:

$$\overset{o}{v}_{x1} = -F_1 \cdot \frac{V_{vx1}}{\rho_2} + k_1 \cdot v_{x1}$$

$$\overset{o}{x}_1 = -v_{x1}$$

$$\overset{o}{V}_{x1} = 0$$

$$\overset{o}{V}_{vx1} = V_{x1} - k_1 \cdot V_{vx1}$$

For this example, the final value will be zero, and occurs when the difference between the desired position and the actual position is zero. Naturally, to obtain a more general solution, a solution manifold should be used; however, in order to display the utility of this approach, the previously mentioned final conditions will suffice. The closed form expression of the value function is then of the form:

$$V_{vx1} = (x_1 - x_d) \cdot \frac{1 - e^{-k_1 t}}{k_1}$$

It should be noted that the above analysis could be performed on a reduced set of differential equations, where each equation would express the differences in distance and velocity, and hence reduce the number of differential equations by a factor of 2. However, for the sake of clarity, the analysis is performed on the actual position and velocity differential equations.

Furthermore, it should also be noted that this solution closely resembles the isotropic rocket pursuit game described in [Isaacs, 1965]. This is due to the fact that the dynamic equations are decoupled, and hence working within a three-dimensional framework will not change the problem considerably.

V. Simulation Results

From the closed form expression of the control presented in the previous section, it is obvious that the optimal strategies are in fact bang-bang controllers. Since the forces in the system are not dependent on the proximity of the vehicles to the desired points, there will always exist some positional error. It is however possible to resolve this problem simply by switching controllers at some error threshold, or introducing terms that minimize the force terms F1 and F2 as the vehicles approach the desired points.

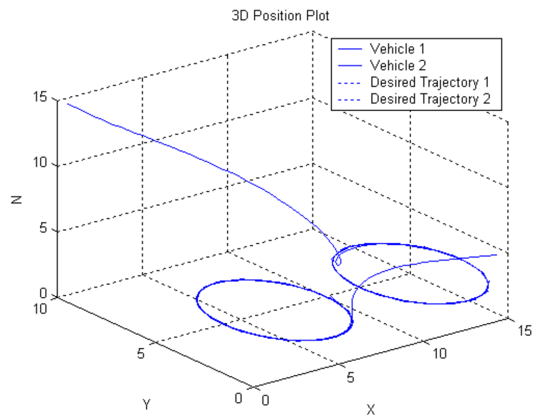

Figure 6: Two-Vehicle Simulation with Sufficient Vehicle Velocities

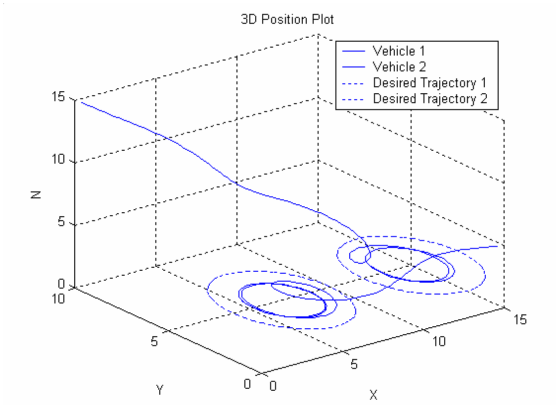

Figure 7: Two-Vehicle Simulation with Insufficient Vehicle Velocities

The above plot shows the tracking capabilities of the derived controller. The two vehicles are attempting to follow two parameterized circular trajectories with a radius of three. In Figure 6 the vehicles can move quickly enough to actually reach the desired trajectories, while in Figure 7 the velocities of the vehicles are not sufficient to reach the desired trajectories. In the latter case, the vehicles simply move in a smaller circle, which ensures that the error remains constant.

The term target tracking is often used to refer to the task of finding/estimating the motion parameters (mainly the location and direction) of a moving target in a time sequence of measurements. This task is achievable as long as the target is within the sensor's field of view (FOV). If it happens that the target keeps moving away to the point it runs o® the FOV, the target tracking task will fail to track the moving target until the target re-enters the sensors FOV. To address such problem, the sensor is mounted on a moving platform such as a UAV. We call the new setup (the sensor plus the UAV) an agent. Thus, we can start a second task, other than the target tracking task, to (reactively or proactively) move the sensor to guarantee that the target stays in view. That second task is what we call the agent placement task. The work presented in this paper is of the active sensing-based target tracking variety, in which both tasks discussed above are integrated.

VI. Target Tracking

There exists a number of efforts to formally describe the dynamic agent placement problem for target tracking. The choice is made to use a formulation of the variety of Weighted Cooperative Multi-robot Observation of Multiple Moving Targets (W-CMOMMT) (Werger and Mataric, 2000), (Werger and Mataric, 2001) since it captures the multiple-observer-multiple-target scenario with target prioritization. W-CMOMMT can be shown to be an NP-hard problem [Hegazy and Vachtsevanos, 2004].

The agent (sensor) placement problem is formulated by defining a global utility function to be optimized given a graph representing the region of interest, a team of agents and a set of targets. A coarse motion model is developed first where target transitions follow a stochastic model described by an M^{th} order Markov chain.

Agents use the model to predict the target locations at future time instants as probability distributions. The algorithm attempts to maximize the coverage by searching for a set of observation points at each time step. A real time dynamic programming tool is called upon to solve the maximization problem. Details of the approach can be found in [Hegzy and Vachtsevanos, 2004].

Particle filters have recently been successful in tracking mobile targets in video (P. Perez and Blake, 2004) (P. Perez and Vermaak, 2002). The video tracking problem consists of determining the position of a target within a particular video frame based on information from all past frames. Information such as size, color, and motion characteristics of the target is known a priori. In the particle filter framework, this information is used to initialize the filter in the first few frames of video. Thereafter, using a model similar to (P. Perez and Blake, 2004) the state of each particle is updated as the video progresses from one frame to the next. At each step, color and motion data is collected for each particle to determine which particles have a high probability of correctly tracking the target. On the next iteration, particles are drawn according to this probability. Thus, successful particles "survive" and are used in subsequent frames, while the other particles "die".

Particle Filtering in a Bayesian Framework

The objective of Bayesian state estimation is to estimate the posterior pdf of a state, x_k, based on all previous measurements, $z_{1:k}$. This pdf, $p(x_k | z_{1:k})$ can be determined in two steps, prediction and update. In the prediction step, the state update model is used to determine the prior pdf $p(x_k | x_{k-1})$. If a first-order Markov model is assumed, then the prior is given as

$$p(x_k | z_{1:k-1}) = \int p(x_k | x_{k-1})\, p\,(x_{k-1} | z_{1:k-1})\, dx_{k-1}$$

After the measurement, z_k is made, the prior is updated using Bayes' rule:

$$p(x_k | z_{1:k}) = \frac{p(x_k | z_k)\, p\,(x_k | z_{1:k-1})}{p\,(z_k | z_{1:k-1})}$$

In most cases, the above equations cannot be determined analytically. The Kalman filter is a well-known exception. However, when a Kalman filter is used, the system must be linear with
Gaussian distributions. The particle filter is one way to estimate the above equations.
A particle filter iteratively approximates the posterior pdf as a set

$$S_k = \left\{ \left\langle x_k^{(i)}, w_k^{(i)} \right\rangle \middle| i = 1, \dots, n \right\}$$

where $x_k^{(i)}$ represents a point in the state space, and $w_k^{(i)}$ is the importance weight associated with this point. The $w_k^{(i)}$ are non-negative, and sum to unity. At each iteration, the particles are updated using the system dynamics and sampling from

$$p\left(x_k^{(i)} \middle| x_{k-1}^{(i)} \right)$$

Measurements are then taken at each particle and the weights are updated using

$$w_k^{(i)} \propto w_{k-1}^{(i)}\; p\left(z_k | x_k^{(i)}\right)$$

If the particles are resampled at each iteration, then the previous weights may be neglected and Eq. 12 becomes

$$w_k^{(i)} \propto p\left(z_k | x_k^{(i)}\right)$$

After the weights are determined to at least a scale factor, they are normalized such that their sum is equal to unity. It has been shown (Arulampalam and Maskell, 2002) that the posterior pdf estimated using particle filtering converges to the actual pdf as the number of particles increases.

A particle filter was used to track a soldier as he maneuvered in an urban environment 2. Frames were grabbed from a movie at a rate of 30 Hz. The movie camera was held by a human operator.

Therefore, there are a number of vibrations in the video, and the zoom is adjusted during the video.

A few frames of the output are shown in Figure 8. The box represents a weighted average of the ten best particles. The set of "lights" in the upper left corner of each frame are used to indicate the output of the neural network. If the lowest "light" is "illuminated," the neural network has output the lowest confidence level. If the second lowest is "illuminated," the neural network has output the second lowest confidence level. If the middle two are "illuminated," the neural network has output the second highest confidence level. If the top three are "illuminated," the neural network has output the highest confidence level.

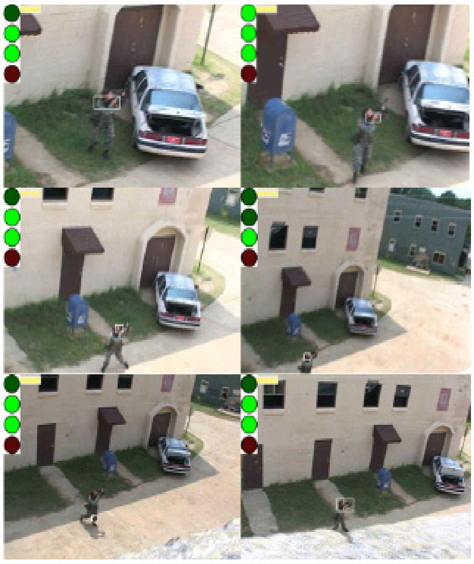

Figure 8: Typical output frames. Each frame is approximately 1.7 seconds apart from each other

VII. New Directions/Technological Challenges

Technological Challenges:

*From single system to "system of systems"

- Modeling– Spatio-temporal modeling paradigms are needed for real-time planning and control of networked systems.
- Control– Hierarchical/Intelligent control of multiple networked systems (agents, sensors); new reasoning paradigms for tracking, pursuit-evasion, surveillance/reconnaissance, coordinated control, planning and scheduling, obstacle avoidance, etc.
- Networking and Communications – Inter-and intra-systems reliable and secure communication protocols; need for command and control and supervisory functions; bandwidth and other Quality of Service requirements.
- Computing– On-platform computational requirements; hardware and software architectures; open systems architectures.
- Sensors and Sensing Strategies – Hardware/Software requirements; performance and effectiveness metrics; networked sensors.
- Performance Metrics/Verification and Validation – Defining metrics for design and performance assessment; formal methods for verification and validation.

The Enabling Technologies:

- New *modeling* techniques are required to capture the coupling between individual system/sensor dynamics, communications, etc. with system of systems behaviors. Hybrid system approaches will play a key role. Means to represent and manage uncertainty. Software models for improved QoS. Spatio-temporal models of distributed agents (sensors) are required to integrate system and motion dependencies, contingency planning, etc.
- Control – Intelligent and hierarchical/distributed control concepts must be developed and expanded to address "system of systems" configurations. Game – theoretic notions

and optimization algorithms running in almost real time to assist in cooperative control and adversarial reasoning. Control of networks of dynamic agents.

- Networking and Communications – Communication protocols and standards.

- Computing – Embedded processing requirements; new and reliable, fault-tolerant computing platforms; software reliability issue.

- Sensors and Sensing Strategies - Innovative concept and technologies in wireless communications; improved and reliable/cost-effective sensor suites; "smart" sensors and sensing strategies; data processing, data mining, sensor fusion, etc.

- Performance Metrics/V&V- Need new system of systems performance and effectiveness metrics to assist in the design, verification/validation and assessment of networked systems.

VIII. CONCLUDING REMARKS

Federated systems consisting of multiple Unmanned Aerial Vehicles performing complex missions present new challenges to the control community. UAVs must possess attributes of autonomy in order to function effectively in a "system of systems" configuration. Coordinated/collaborative control of UAV swarms demands new and novel technologies that integrate modeling, control and communications/computing concerns into a single architecture. Typical application domains include reconnaissance and surveillance missions in an urban environment, target tracking and evasive maneuvers, search and rescue operations, Homeland security, etc. Major technological challenges remain to be addressed for such UAV swarms, or similar federated system of systems configurations to perform efficiently and reliably. Excessive operator load, autonomy issues and reliability concerns have limited thus far their widespread utility. The systems and controls community is called upon to play a major role in the introduction of breakthrough technologies in this exciting area.

IX. REFERENCES

1. Vachtsevanos, G., Tang, L. and Reimann, J. "An Intelligent Approach to Coordinated Control of Multiple Unmanned Aerial Vehicles," Proceedings of the American Helicopter Society 60th Annual Forum, Baltimore, MD, June 7-10, 2004.
2. Office of the Secretary of Defense (Acquisition, Technology, & Logistics), Air Warfare. "OSD UAV Roadmap 2002-2027." December 2002.
3. Sousa, J. B., and Pereira, F., "A framework for networked motion control," Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 1526-1531, Hawaii, USA, December 2003.
4. Howard, M., Hoff, B., Lee, C., "Hierarchical Command and Control for Multi-agent Teamwork", Proceedings of 5th Intl. Conf. on Practical Application of Intelligent Agents and Mult-Agent Technology (PAAM2000), pp. 1-13, Manchester, UK. Apr. 10, 2000.
5. Baras, J. S., Tan, X., and Hovareshti, P., "Decentralized Control of Autonomous Vehicles," Proceedings of the 42nd IEEE Conference on Decision and Control, pp. 1532-1537, Hawaii, USA, December 2003.
6. Dunbar, W. B., Murray, R. M., "Model Predictive Control of Coordinated Multi-Vehicle Formation," Proceedings of the 41st IEEE Conference on Decision and Control, pp. 4631-4636, Las Vegas, USA, December 2002.
7. Voos, H., "Market-based Algorithms for Optimal Decentralized Control of Complex Dynamic Systems," Proc. of the 38th IEEE Conference on Decision and Control, Vol.40, pp.3295-3296, Phoenix, AZ. 1999.
8. Clearwater, S. H. E., "Market-Based Control: A Paradigm for Distributed Resource Allocation," Singapore: World Scientic, 1996.
9. Walsh, W., and Wellman, M., "A Market Protocol for Decentralized Task Allocation," Proc. of the 3rd International Conference on Multiagent Systems, 1998.

10. Engelbrecht, W. R., Shubik, M. , and Stark, R.M., "Auctions, Bidding, and Contracting: Uses and Theory," New York, NY: New York University Press, 1983.
11. Bertsekas, D., “Auction Algorithms for Network Flow Problems: A Tutorial Introduction,” Computational Optimization and Applications, Vol.1, pp.7-66, 1992.
12. Vachtsevanos, G., Kim, W., Al-Hasan, S., Rufus, F., Simon, M., Schrage, D., and Prasad, J. V. R., "Mission Planning and Flight Control: Meeting the Challenge with Intelligent Techniques," Journal of Advanced Computational Intelligence, Vol. 1, (1), pp. 62-70, Oct., 1997.
13. Al-Hasan, S. and Vachtsevanos, G., "Intelligent Route Planning for Fast Autonomous Vehicles Operating in a Large Natural Terrain," Journal of Robotics and Autonomous Systems, Vol. 40, pp. 1-24, 2002.
14. Al-Hasan, S. and Vachtsevanos, G., "A Neural Fuzzy Controller for Moving Obstacle Avoidance," Third International NAISO Symposium on Engineering Of Intelligent Systems, Malaga, Spain, September 24 - 27, 2002.
15. Isaacs, R. "Differential Games: a Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization." New York: John Wiley and Sons, Inc., 1965.
16. Arulampalam, A. and Maskell, S. (2002). A tutorial on particle fiters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing* 50, 174-188.
17. Berchtold, A. and Raftery, A. (2002). The mixture transition distribution model for high-order Markov chains and non-gaussian time series. *Statistical Science* 17(3), 328-356.
18. Kwok, C., Fox, D. and Meila, M. (2003). Adaptive real-time particle fitlers for robot locatlization. In: Proc. of the *IEEE International Conference on Roboticès & Automation*. pp. 2836-2841.
19. Fox, D. (2003). Adapting the sample size in particle filters through kld-sampling. *International Journal of Robotics Research* pp. 985-1002.
20. Loy, G., Fletcher, L. and Apostoloff, N. (2002). An adaptive fusion architecture for target tracking. *Proceedings of the Fifth International Conference on Automatic Face and Gesture Recognition*. pp. 248-253.
21. Hegazy, T. and Vachtsevanos, G. (2004). Dynamic agent deployment for tracking moving targets. *Proceedings of the Twelfth Mediterranean Conference on Control and Automation*. Kusadasi, Aydin, Turkey.
22. Hochbaum, D. (1995). *Approximate Algorithms for NP-Hard Problems*. PWS Publishing Company.
23. Perez, P., Hue, C. and Vermaak, J. (2002). Color-based probabilistic tracking. *Proceedings of the European Conference on Computer Vision*. pp. 134-149.
24. Perez, P., Vermaak, J. and Blake, A. (2004). Data fusion for visual tracking with particles. *Proceedings of the IEEE*. Vol. 92. pp. 495-513.
25. Han, Q.,Ye, Y., Zhang, H. and Zhang, J. (2002). On approximation of max-vertex-cover. *European Journal of Operational Research* (2), 207-220.
26. Werger, B. and Mataric, M. (2000). Broadcast of local eligibility: Behavior-based control for strongly cooperative robot teams. *Proceedings of the Fourth International Conference on Autonomous Agents*. pp. 21-22.
27. Werger, B. and Mataric, M. J. (2001). From insect to internet: Situated control for networked robot teams. *Annals of Mathematics and Artificial Intelligence* 31(1-4), 173-197.