14th ICCRTS
“C2 and Agility”

Title of Paper : Design of a Decision Support Architecture for Human Operators in UAV Fleet C2 Applications

Topic (s) : C2 Architectures and Technologies (Primary)
Experimentation and Analysis

Author (s) : Oktay Arslan [STUDENT], Research Assistant
Controls and Avionics Laboratory, Istanbul Technical University, Maslak, Istanbul, Turkey

Gokhan Inalhan, Assistant Professor
Faculty of Aeronautics and Astronautics, Istanbul Technical University, Maslak, Istanbul, Turkey

Point of contact : Oktay Arslan (oktay.arslan@itu.edu.tr)

Name of the Organization : Controls and Avionics Laboratory

Complete Address : Controls and Avionics Laboratory, Faculty of Aeronautics and Astronautics, Istanbul Technical University, Maslak 34469
Istanbul, Turkey

Telephone : 0090 212 285 3148

Email address : {oktay.arslan, inalhan}@itu.edu.tr

Oktay Arslan is currently a master student, in his second year of study, of interdisciplinary Defense Technologies Program with specialization in Information Technologies subprogram. The copy of the university course schedule for the current and past semesters can be found as attachment in the submission email.

Design of a Decision Support Architecture for Human Operators in UAV Fleet C2 Applications[*]

Oktay Arslan[†] Gokhan Inalhan[‡]

Abstract

UAVs are becoming indispensable assets for military command and control applications such as surveillance, reconnaissance, search and rescue operations because of their superiorities over manned vehicles. Nevertheless, humans are still needed for high-level guidance and commanding of UAVs in those operations for their intelligence and higher level of flexibility in decision making. Within this new position, human operators are responsible for commanding of multiple UAVs under hard timing constraints in a dynamically changing environment. The supervisory control of the UAVs becomes more challenging as the number of the UAVs increases and it is sometimes intractable or infeasible even by a set of operators. In this work, we focus on development of decision support tools in order to improve of the agility of a C2 system for UAV fleets and present the framework of a real-time decision support system for operators who are responsible for high-level decision making in scenarios involving a large number of tasks across multiple UAVs. The decision support system consists of three stages including planning, scheduling and low-level mission specific task planning. The overall system is integrated to a labscale multi-vehicle mission simulator, demonstrating the ability of human operators for exploiting a fuller set of UAV fleet capabilities.

Keywords: decision support system, human supervisory control, multiUAV control

---

[*] This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) Graduate Fellowship Funds

[†] O. Arslan - Research Assistant, Controls and Avionics Laboratory, Istanbul Technical University, Maslak, Istanbul, Turkey `oktay.arslan@itu.edu.tr`

[‡] G. Inalhan - Corresponding Author, Assistant Professor, Faculty of Aeronautics and Astronautics, Istanbul Technical University, Maslak, Istanbul, Turkey `inalhan@itu.edu.tr`

Introduction

The recent developments in the autonomous vehicle technology result in a remarkable increase in usage of Unmanned Air Vehicles (UAVs) in military command and control applications such as surveillance, intelligence or reconnaissances (SIR). Nowadays, UAVs have the capability of performing many complex missions with strict time constraint as well as manned vehicles, even with reduced risks to human pilot. Especially, due to improved human safety, UAVs are, also, preferred in not only military but also civil application such as urban search and rescue operations. However, growing involvement of UAVs in complex applications, several challenging problems arise intrinsically in aerial systems. First of all, cost of low-level remote control of UAVs might be very high and usually requires employment of several human operators for single UAV of limited decisional autonomy. Also, following the taxonomy as presented in [10], the number and types of missions generally require several UAVs operating collaboratively in order to accomplish the operational goals. Therefore, new tools, algorithms and architectures are required to maintain coordinated control of UAV fleets.

Despite the current trend reducing the role of human in military systems and substantial interest focused on the development of higher level of autonomy for UAVs, human operators are still needed for supervisory control of UAV fleets. As proposed in [5], one way to exploit UAVs' capabilities is employing the human operators for higher levels of planning and decision making tasks rather than direct manual control of UAVs. Since human operators have limited capacity of cognitive resources, there has been a great deal of effort in development of systems that allow one operator to manage of a large number autonomous UAVs in a high mission workload environment. While there are some successful works that encourage the controlling of multiple UAVs by a single operator [6], the supervisory control of the UAVs becomes more challenging as the number of the UAVs increases and it is sometimes intractable or infeasible even by a set of operators due to high degree of mental workload. Therefore, new C2 systems (human operators with their supporting information systems and decision aid tools) are needed to be developed in order to increase the agility of the overall system including human operators in means of responsiveness [1]. There are various factors that effect the cognition of the operators, but one of the most important and time consuming process is deliberation of the which tasks to perform and scheduling of these tasks regarding the temporal and resource constraints. Since optimization of human interaction with an automated system through supervision is very difficult, one way to mitigate saturation of operators is providing real-time decision aid for planning and scheduling of these tasks. Such a decision support system contributes too much to the human operators to command and control of the UAV fleets effectively in a timely manner. Because in many of the proposed systems [9, 11], the duration of the planning and scheduling of the tasks is ignored and there are assumptions that planning and scheduling are instantaneous. However, these processes take time in the case in real applications and cause to saturate the cognition of human operators and therefore they have to be handled as fast as possible with the help of realtime generated decision aids.

In this work, we are interested in development of more agile C2 system for UAV fleets by developing decision support tools and present the framework of a realtime decision support system for operators who are responsible for high-level decision making in scenarios involving a large number of tasks across multiple UAVs. Then, the overall system is integrated to a labscale multi-vehicle mission simulator, demonstrating the ability of human operators for exploiting a fuller set of UAV fleet capabilities. The organization of this work is as follows: In Section 1, the proposed decision support architecture and its main segments are described in detail. Then, the general design and the architecture of the multi vehicle mission simulator is briefly presented in the next section. Finally, the integration of the decision support architecture to the simulator and its implementation are given and some successful results of the scheduling algorithm is also given.

1 Design of a Decision Support Architecture

The problem of planning and scheduling of tasks can be solved by human operator easily in the recent supervisory control architectures that based on simultaneous control of four or five UAVs by a single operator. However, as the number of the tasks and UAVs increases like order of hundreds, this problem becomes very challenging and time-consuming and it is required to design a higher level layer which is solely responsible for deciding which actions to perform and temporal allocations of them.

The decision support architecture that is envisioned to provide real-time decision aid to the manned vehicle or ground operators is embedded over the UAV architectures given in [7, 13] and it consists of four main segments,namely, planner, scheduler, resource and task manager as it is shown in Figure 1. With including of this new highest layer, there is an operator which is mainly responsible for deliberation of tasks and temporal and resource allocation for them. Also, the set of human operators who are responsible for supervisory control in the underlying UAV control architecture defines a new type of resources in this extended architecture. After defining resource and time consistent tasks with the aid of the decision support system, this operator delegate or assign them to the available supervisory operator and allocated UAVs via human machine interfaces.

1.1 Planner

The planning is the problem of deciding which action sets to perform during the scenario. Planner is triggered by an external monitored event such as bombing, threat detection or a task request from mission control center. Following the taxonomy as presented in [10], the planning step proceeds by selecting suitable action from a predefined action sets by an expert system. The planner continues its execution collaboratively with resource and task manager until there exists consistent resource allocation and set of tasks.

Events/Requests The set of phenomenon in the mission environment corresponds to the events and the set of commands from mission control center or other peers also correspond to the requests. Basically, each event and request have a different level of priority and importance and require different type of actions to be selected by the planner.

Action sets The actions are the basic components of missions like loitering at a specific point or returning to the base. Each missions can be performed by different types of UAVs in the inventory. Each mission has different goals and functional/information requirements and requires different number of UAVs to be performed.

1.2 Integration of Planner and Scheduler

Resource manager Since each task requires different type and number of UAVs, it is required to allocate enough number of UAVs for each required type. Therefore, resource manager is responsible for resource allocation satisfying time and environment constraints in the asset inventory. However, these tasks are not necessarily mutually exclusive and one type of UAV may have the capability of performing multiple tasks and there may be several different types of allocations for the same problem. Therefore, resource allocation must be done wisely and the solution must be selected from the set of allocation by considering further requests.

Task manager This manager is responsible for defining specification of each task by examining the action sets found by planner.

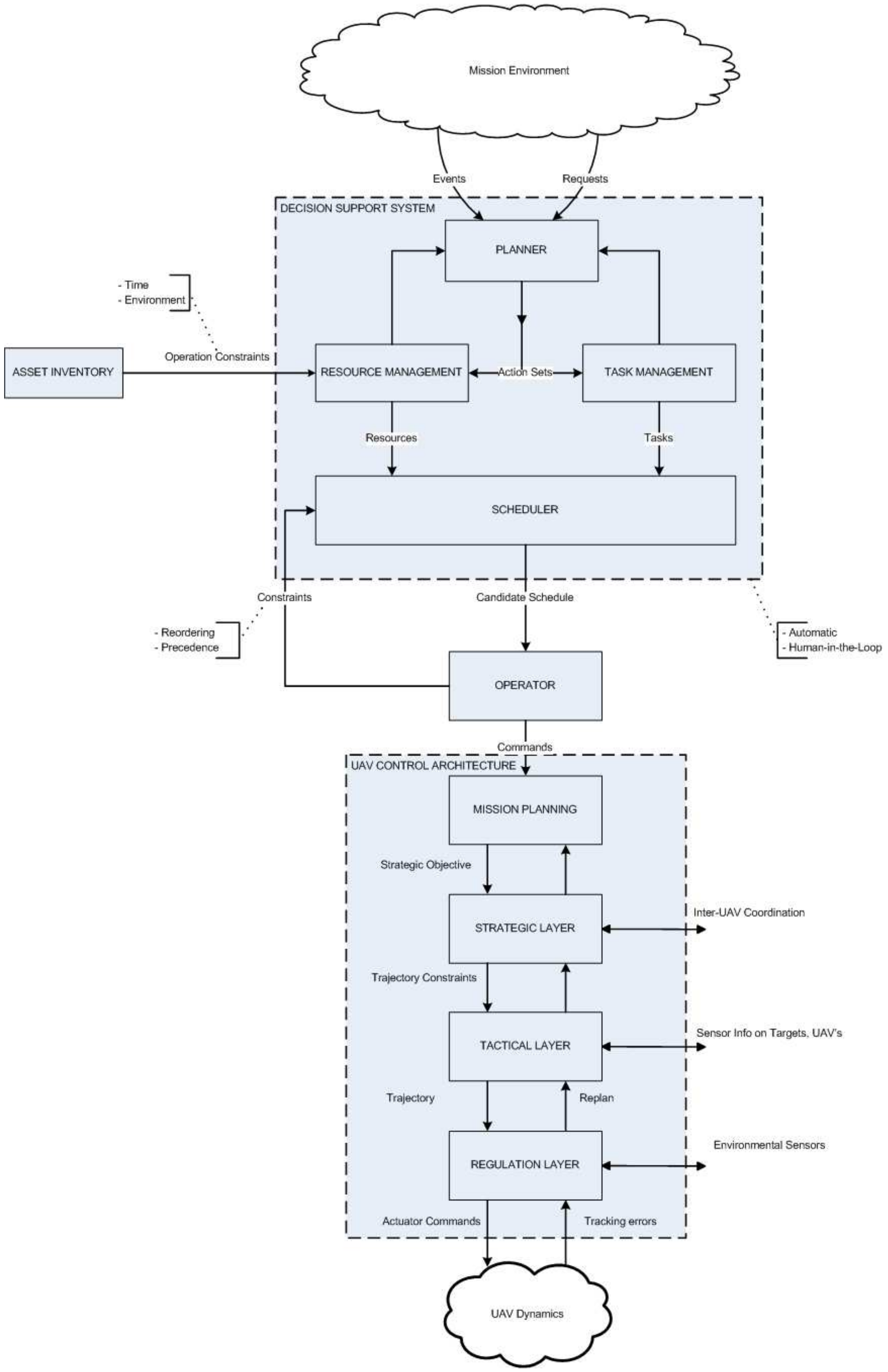

Figure 1: Event-Driven Decision Support Architecture

1.3 Scheduler

The scheduling is the problem of temporal allocation of resources to the tasks. Before the scheduling process, a set of resources must be allocated regarding operation constraints (time, environment) and a group of tasks must be defined regarding to the selected actions and resources. Then, the scheduler finds a candidate schedule for these tasks under resource constraints by satisfying specified time windows constraints. The human operator has the right of the modifying the candidate solution or rescheduling the problem with new precedence and temporal constraints. This scheduling process is executed until the operator confirms and submits the candidate solution as a mission commands to the allocated UAVs and corresponding supervisor operator.

Operation Constraints Due to complex nature of the missions and resources, a scenario intrinsically contains different types of constraints such as time, precedence. Structural dependencies between missions, physical and logistic constraints define a set of temporal constraints. For example, a target must be destroyed (attack) within some periods of time immediately after its designation (reconnaissance), the total completion time of set of missions assigned to a UAV cannot exceed its endurance and it must return to base for maintenance before a certain amount of time from endurance.

As classified in [10], each activities (UAV missions) has different characteristics based on its goals, functional/information requirements. For instance, the target acquisition mission requires path planning (areas to search and waypoints to the area of interest), threat area and forbidden zone information or cargo mission requires path planning (route from origin to destination), forbidden zone information and scheduling mechanism. This kind of planning problems are solved in the underlying mission planning layer of UAV architecture.

2 General Design and Architecture of the Simulator

The general design of the mission simulator is structured around two layers: the visualization and mission layer. These two layers represent two different data bus structures and data flows. As seen in Figure 3, simulation elements such as piloted vehicle simulator, unmanned vehicles, real unmanned vehicles (ground vehicles and micro-UAVs), ground stations and the simulation control computers carry distinct wired and wireless connections to these two data layers.

Figure 2: Multiple monitor visualization as seen by the pilot during a joint mission with unmanned helicopters

Visualization Layer entails the passage of the visualization and simulation related data packets (i.e. packets which result in a coherent visual picture of the whole scenario to the operators and the simulator pilot) across the wired ethernet network using UDP packets. The visualization layer uses open-source FlightGear flight simulator packet structure to allow direct integration to the flight simulator visualization elements. These visualization elements include the three panel environment display for the pilot of the manned simulator (as shown in Figure 2) and the pilot/operator panel for tactical/simulation displays.

The Mission Layer is accomplished via wireless communications (serial and Ethernet) across each unique entity existing within the simulation using predefined data packet numbers and structures. Mission critical data such as target assignments, payload readings, commands and requests are delivered through this wireless mission layer link.

The hardware structure within the network simulator is tailored to mimic the distributed nature of each of the vehicle's processors and communication modules. Open-source flight simulation software, FlightGear, is modified for networked operations and it is used as the 3D visualization element for the pilot and the mission controls. The UAV dynamics and low-level control algorithms are embedded within the xPC target computers. Equipped with 3D fight simulation displays and touch-screen C2 interface at the desktop pilot level, the platform also allows us to rapidly prototype and test pilot-unmanned fleet supervisory control and pursuit-evasion game designs. In addition, the unique design enables seamless integration of real unmanned air vehicles within a simulated scenario. Hardware-in-the-loop testing of network bus compatible mission computers and avionics systems provides us with validation of the C2 architectures and the hardware designs on a realistic lab-scale platform before the actual flight experiments. A more detailed explanation of the components of the multi-vehicle mission simulator is given in [2].

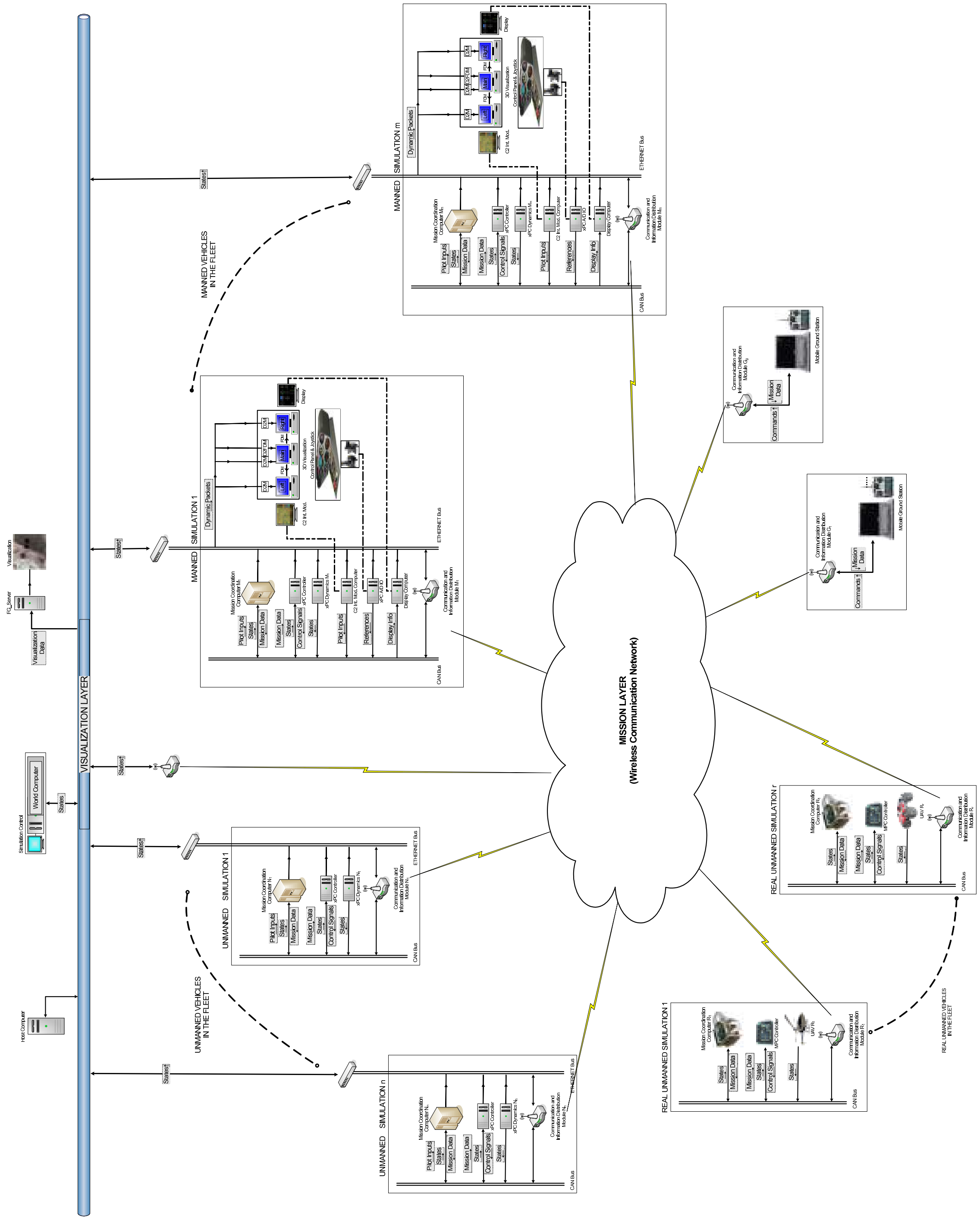

Figure 3: General Architecture of the Multi-vehicle Mission Simulator

3 Integration of the Decision Support System to the Mission Simulator

3.1 Development of the Algorithms

3.1.1 Planner

The planner of the decision support system is written by using C++ language and it is currently under development. A GUI based frontend will be designed for the human operators after the integration of overall subparts of the system.

3.1.2 Scheduler

In the scheduling problem, we assume that the set of tasks requiring resources are specified in advance and focus on temporal allocation of a set of activities satisfying resource and strict time constraints as fast as possible. We use the Resource-Constrained Project Scheduling Problem with Minimum and Maximum time lags (RCPSP/max) as a reference [4]. Specifically, to handle executional uncertainty in the dynamic mission environment, Solve & Robustify approach is used as a base algorithm. The algorithm used in the Solve step, Earliest Start Time Algorithm (ESTA) [12], is modified with temporal space partitioning to provide real-time solutions to the operator. Benchmark problem comparisons with the classical ESTA formulation for two hundred tasks indicates that the proposed temporal space partitioning approach improves the computation time forty-fold while only incurring five percent increase in the total completion of the tasks. After successful temporal allocation of the actions, the low level task planning problem is solved by the algorithm given in [8] and the complete algorithmic structure of the $ESTA_P$ can be found in [3].

Both algorithms, $ESTA_P$ and $ESTA$ are implemented in C++ using Eclipse IDE and the CPU times given in the table are measured on a PC with Intel Pentium(R) D CPU 3.40 Ghz processor under Fedora Core 9. The maximal horizon h_{max} is set to 5000 in order to find a solution quickly by searching within a sufficiently large horizon. The following performance measures are calculated for comparative analysis of both algorithms on different problem sets:

$N_{feas}\%$ the percentage of problems feasibly solved for each benchmark set

t_{mks} average makespan of the solutions

t_{cpu} average CPU-time in seconds spent to solve instances of the problem set

N_{pc} the number of leveling precedence constraints posted to solve a problem

$\Delta_{LB}\%$ the average of percentage relative deviation from known lower bound

Table 1: Performance of the algorithms (UBO50)

UBO50	t_{mks}	$\Delta_{LB}\%$	t_{cpu}	$N_{feas}\%$	N_{pc}
ESTA$_{P11}$	217.671	28.757	0.360	77.778	54.471
ESTA$_{P15}$	217.099	28.247	0.383	78.889	55.141
ESTA$_{P16}$	217.306	27.531	0.400	80.000	56.694
ESTA$_{P17}$	**218.973**	**27.459**	**0.373**	**81.111**	**56.904**
ESTA$_{P20}$	218.466	27.462	0.452	81.111	60.480
ESTA$_{P21}$	218.699	27.595	0.462	81.111	59.384
ESTA	**213.603**	**24.455**	**4.004**	**81.111**	**74.890**

Table 2: Performance of the algorithms (UBO100)

UBO100	t_{mks}	$\Delta_{LB}\%$	t_{cpu}	$N_{feas}\%$	N_{pc}
ESTA$_{P12}$	**423.167**	**30.970**	**3.075**	**86.667**	**120.077**
ESTA$_{P15}$	419.705	29.833	3.121	86.667	123.538
ESTA$_{P20}$	418.436	29.697	3.166	86.667	128.910
ESTA$_{P25}$	419.141	30.100	3.345	86.667	132.974
ESTA	**407.286**	**25.645**	**79.214**	**86.667**	**195.753**

Table 3: Performance of the algorithms (UBO200)

UBO200	t_{mks}	$\Delta_{LB}\%$	t_{cpu}	$N_{feas}\%$	N_{pc}
ESTA$_{P18}$	770.974	29.970	26.761	85.556	254.961
ESTA$_{P20}$	765.481	28.987	33.866	85.556	261.091
ESTA$_{P25}$	**763.474**	**28.603**	**31.782**	**86.667**	**275.897**
ESTA$_{P30}$	759.766	27.987	33.159	85.556	281.169
ESTA	**751.962**	**26.946**	**1462.83**	**86.667**	**461.436**

3.2 C2 Implementation: Hardware-in-the-loop Testing of a Large-scale Autonomous Target-Task Assignment Problem for a UAV Network

As a first step coordination algorithm implementation within the mission simulator, we considered one of the basic problems. The problem consists of n targets to be visited by m UAVs and the vehicles should autonomously find the waypoint selections that results in the minimum total path traveled by the UAV fleet. In addition, all the mission coordination and the communication between the units had to be done autonomously without any human intervention. To address the first step of this problem, we have developed a large-scale distributed task/target assignment method that allows autonomous and coordinated task-assignment in real-time for a UAV fleet. By using delayed column generation approach on the most primitive non-convex supply-demand formulation, a computationally tractable distributed coordination structure (i.e. a market created by the UAV fleet for tasks/targets) is exploited. This particular structure is solved via a fleet-optimal dual simplex ascent in which each UAV updates its respective flight plan costs with a linear update of way-point task values as evaluated by the market. The complete theoretical treatment and algorithmic structure of this problem can be found in [8].

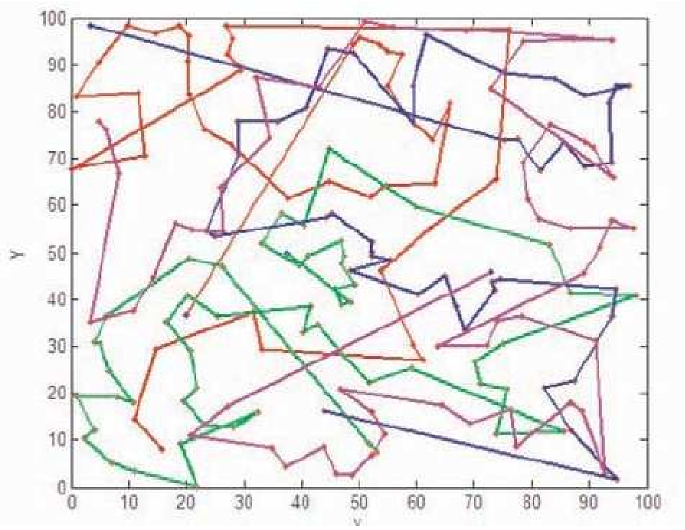

Figure 4: The waypoint routes for a random pop-up task-target assignment for four vehicles. The algorithm is implemented further in the receeding horizon mode for five hundred waypoints.

3.3 C2 Application: Expansion of the Human-Machine-Interface (HMI) to decision-support system for manned and unmanned fleets

The development of agile C2 system for UAV fleets is very hard task, since the agility characteristics of a such system itself includes several key dimensions which should be examined in very different domains and context: robustness, resilience, responsiveness, flexibility, innovation, adaptation [1] . In this application, we focus on development of decision-support tools in order to improve the agility of C2 system in the sense of responsiveness, since the problem of high level UAV fleet coordination and the underlying low-level missions are performed in a very high tempo.

A key part of such an event-driven process interaction hinges on the UAVs to autonomously coordinate the target/task assignments and distributions. The experimental illustration of this within the mission simulator is illustrated in [2]. However, it is important to note that supervisory control and interruption is desired in all mission critical phases. Towards this goal, we have developed a Human-Machine-Interface display systems which allow this supervisory control capability over unmanned vehicle networks over the manned simulation platform. Basically, this HMI display systems enhanced with decision support capability generate candidate sched-

Figure 5: Command ana Control Interface GUI

ule which has temporal flexibility as much as possible and is robust to executional uncertainty in mission environment in a very brief time intervals. Then, the human operator examines the proposed schedule and makes modifications or posts additional precedence constraints between activities if required. Generation of robust schedules are very crucial, since the mission environment is inherently very challenging and includes too many misfortune like asset breakdowns or delays in activity duration. The developed algorithm has the capability of adapting the current solution into the new situation very fast, hence it contributes the human operator to handle with the perturbation in the mission environment. In addition to showing manned vehicle flight information data, this HMI display system also tracks and monitors the action of unmanned vehicle fleet within the joint mission.

4 Conclusions

In this work, we have focussed on the problem of the planning and scheduling of tasks in UAV fleet C2 applications and decision support architecture is presented for real-time decision generation for operators who are responsible for high-level decision making in scenarios involving a large number of tasks across multiple UAVs. The main segments of the proposed decision support architecture and its integration to a labscale mission simulator is explained in detail. Finally, the developed Human

Machine Interfaces and the performance of the implemented algorithms for scheduling and C2 applications is given.

References

[1] D.S. Alberts and R.E. Hayes. *Power to the edge: command, control in the information age*. Command and Control Research Program (CCRP) Publication Series, 2003.

[2] O. Arslan, B. Armagan, and G. Inalhan. *Development of a Mission Simulator for design and testing of C2 Algorithms and HMI Concepts across Real and Virtual Manned-Unmanned Fleets*. Lecture Notes in Control and Information Sciences. Springer, 2008.

[3] Oktay Arslan and Gokhan Inalhan. An event driven decision support algorithm for command and control of UAV fleets. St. Louis, Missouri, USA, June 2009.

[4] M. Bartusch, R. H. Möhring, and F. J. Radermacher. Scheduling project networks with resource constraints and time windows. *Annals of Operations Research*, 16(1):199–240, 1988.

[5] S. Bruni, S. Mercier, and P. J. Mitchell. Automation architecture for single operator, multiple UAV command and control. *The International C2 Journal*, 1(2):1–24, October 2007.

[6] M. Cummings and S. Guerlain. Human performance issues in supervisory control of autonomous airborne vehicles. In *AUVSI Unmanned Systems North America Conference*, 2004.

[7] DN Godbole. Control and coordination in uninhabited combat air vehicles. In *American Control Conference, 1999. Proceedings of the 1999*, volume 2, 1999.

[8] S. Karaman and G. Inalhan. Large-scale Task/Target Assignment for UAV Fleets Using a Distributed Branch and Price Optimization Scheme. In *Int. Federation of Automatic Control World Congress (IFAC WC2008), Seoul, June*, 2008.

[9] T.W. McLain and R.W. Beard. Coordination Variables, Coordination Functions, and Cooperative Timing Missions. *Journal of Guidance, Control & Dynamics*, 28(1):150–161, 2005.

[10] C. Nehme, M. Cummings, and J. Crandall. UAV Mission Hierarchy, 2006.

[11] H.V.D. ParunaK, S. Brueckner, and J. Sauter. Digital pheromone mechanisms for coordination of unmanned vehicles. In *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*, pages 449–450. ACM Press New York, NY, USA, 2002.

[12] N. Policella. *Scheduling with uncertainty: A proactive approach using Partial Order Schedules*. PhD thesis, University of Rome “La Sapienza”, Rome, March 2005.

[13] S. Shankar Sastry. Hybrid control synthesis Real-Time control problems for UAV, 1999.