Hardware on-Board Implementation of a Possible Solution to Introduce UAVs into Non Segregated Areas

Sergio Taraglio[1], Vincenzo Nanni[1] and Damiano Taurino[2,3]
[1]*Robotics Lab, ENEA, Via Anguillarese 301, Rome, Italy*
[2]*Deep Blue, Piazza Buenos Aires 20, Rome, Italy*
[3]*Facoltà di Ingegneria, University of Florence, Via di Santa Marta 3, Florence, Italy*

Keywords: Game Theory, Software Agents for Intelligent Control Systems, Distributed Control Systems, Separation Assurance, Unmanned Aerial Vehicles, Engineering Applications.

Abstract: The implementation of a distributed control algorithm for the safe flight of Unmanned Aerial Vehicles (UAVs) in traffic conditions in an embedded processor unit is presented. Details on the implementation are given. The control algorithm is designed in order to avoid separation infringements with aircraft in the neighbourhood. The algorithm is grounded on Satisficing Game Theory and supplies manoeuvres aimed at providing separation distance among aircraft. Some results of simulated flight trials of UAVs in segregated and non segregated areas are reviewed. Some experiments on the capability of the embedded algorithm to pilot a simulated UAV in different traffic condition are presented.

1 INTRODUCTION

Extensive use of UAVs is a reality in military operations in the Middle East, even if in tele-controlled mode (Schneider, 2011). On the contrary their use in civil environments is still limited to brief missions and experimental flights in restricted airspaces where they cannot mix with civilian aircraft. Nevertheless the market for these applications is forecasted to be quickly surging, pushing for the introduction of unmanned aircraft in non segregated areas (Lukovic, 2011). In a near future UAVs and commercial aircraft will share the same airspace; several projects are currently working on the definition of a realistic integration scenario (MIDCAS project website, 2012). Recently many international and national organizations have oriented themselves towards a more restrictive interpretation of unmanned aircraft, steering to remotely piloted air systems (RPAS) instead of autonomous ones (ICAO, 2011).

The UAV flight can be controlled in different ways, from remote to autonomous through waypoints and autopilot modules, see e.g. (Carbone et al., 2006)(Carbone et al., 2007). Autonomy can be generally defined as the ability to take decisions without human intervention. Autonomy has been, and may continue to be, one of the bottlenecks for future UAV development, as it has been pointed out in (Roch, 2008).

The key issue for the introduction of UAVs in non segregated airspaces is represented by the ability to "keep at a distance" from all the other flying vehicles in the neighbourhood (separation distance assurance and collision avoidance).

In previous work (Taurino et al., 2010a) (Taurino et al., 2010b) a game theory approach to the problem of separation assurance has been proposed: the ARCA algorithm (Adaptive Routing and Conflict management for UAVs). This algorithm is a high level control algorithm and must not be thought as a reactive sense and avoid one. It is conceived as a mid-term approach to foresee possible too close distances with other traffic, intervening very early.

In this paper the integration of the ARCA module is presented. The algorithm is implemented in a flight ready hardware processor, and it is made cooperate with the vehicle auto pilot. The full hardware-in-the-loop simulation of operations is presented and discussed.

The ARCA algorithm takes routing decisions through the knowledge of the flight data of all the nearby aircraft. These data are collected via the ADS-B messages, continuously broadcasted by every aircraft (Stamper, 2005). Up to now these data are transmitted, but their potential benefit is not yet

Taraglio S., Nanni V. and Taurino D..
Hardware on-Board Implementation of a Possible Solution to Introduce UAVs into Non Segregated Areas.
DOI: 10.5220/0004029501770182
In *Proceedings of the 9th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2012), pages 177-182
ISBN: 978-989-8565-21-1

exploited. The ADS-B exploitation is currently object of intense work by several international institutions (see e.g. SESAR Work Package 5 (Sesar, 2012)). The ARCA algorithm thus does not need additional data or interaction, but it exploits already available signals for a new and valuable purpose.

In the second section of the paper the algorithm is briefly reviewed. In the third section the available experimental result are reviewed together with the used experimental setup. In the fourth section the implementation on board the Embedded Processing Unit (EPU) is presented together with the CANOpen implementation of the on board communications among the various devices composing the system; hardware-in-the-loop tests are also reported. In the fifth section of the paper the conclusions are drawn.

2 THE ARCA ALGORITHM

The ARCA algorithm is inspired by Game Theory and in particular it is based on the Satisficing Game Theory approach (SGT) (Stirling, 2003). Basically, SGT computes the best choice among various alternatives by means of two utility functions, rejectability and selectability, which respectively represent costs and benefits for each agent while making a choice. There may exist dependence between utility functions of different agents that implements an "altruistic" behaviour.

This approach has already proven to be effective in dealing with several real-world problems (e.g. packet routing (Boyan et al., 1994) and (Choi et al., 1996), transport logistics (Wolpert et al., 1999), automated car driving (Stirling, 2003), airborne self-separation (Bellomi et al., 2008)) by providing robust and dependable solutions that can be achieved with limited computational resources.

In the UAV context each UAV can be considered as an agent working under the SGT framework. Selectability and rejectability of each agent are respectively the benefits and costs of the UAV's available manoeuvring choices. Benefits are essentially proportional to the optimality of the possible route to reach the final destination (sortest possible delay). Costs are proportional to the risk of infringing the separation with other vehicles. Each aircraft computes the rejectability and selectability of each considered manoeuvre and then selects the option maximising the difference between the selectability and rejectability utilities. Plainly speaking, each aircraft selects the best path with respect to the minimum risk.

2.1 2-D ARCA Algorithm

Let us consider a single flight level. At each time step, each UAV collects information from all other aircraft within its communication range (viewable aircraft). This information includes position, speed, destination, actual heading, flight time (basically an ADS-B frame). Each UAV may choose one of five directions: flying straight, moderate or sharp turn to the left, moderate or sharp turn to the right (-10°,-5°,0°,5°,10°). For each UAV a priority set is defined as that of all viewable aircraft with higher ranking than itself that could be conflicting for some heading choices (parents). The rank may be assigned using different criteria, e.g. it can be based on the aircraft accumulated delay, using the flight time information in the ADS-B data.

The rejectability of agents is unconditioned: each UAV matches the linear extension of each of its directional options with the linear projections of current headings of all aircraft in its priority set. Each projected conflict adds a weight to the rejectability function related to that directional option, depending on distance in time and severity of the conflict. Then a normalization is performed. This increases the rejectability of flight options that lead to conflicts (small separations), with more weight for incidents closer in time. On the other hand the selectability function reflects goal achievement. The selectability is influenced by the preferences of other agents, it is composed of two terms: the base selectability (current UAV heading preferences) and the parent selectability (higher priority agents preferences). In non restricted airspace the commercial aircraft have always higher priorities than the UAVs for obvious safety purposes.

After computing the rejectability and the selectability, each agent chooses its heading change, maximizing the difference between selectability and rejectability. Each satisficing agent looks for the highest gain, with the lowest risk, taking also into account preferences of other agents, thus obtaining a solution that could be effective for the whole system.

2.2 3-D ARCA Algorithm

The three dimensional version of the algorithm takes into account two additional choices for the flight path: it is possible to climb or descend one flight level (1000 ft). The altitude change choice is taken through the continuous monitoring of the minimum value for rejectability with respect to direction. If this minimum is larger than a given threshold, it implies that the current flight level is too crammed

and that a change of flight level is advisable. Whether to climb or to descend is chosen still evaluating the minimum rejectability on the two different flight levels. Once that the altitude change has been decided, the agent, while vertically moving, considers itself as still belonging to the starting flight level for half of the path, then as belonging to the arrival flight level, and computes the ARCA algorithm consequently. Naturally while changing level any further altitude change is inhibited. The altitude change is not indefinitely allowed, e.g. the descent is inhibited if the agent is already flying too low. This 3D ARCA algorithm automatically considers the altitude change choice as possessing a lower priority with respect to manoeuvres in the same flight level. This is driven by economic considerations (level changing is more expensive in terms of fuel consumption) but it can be easily changed to adapt to specific operative scenarios.

3 EXPERIMENTAL RESULTS

The ARCA algorithm is an add-on for the vehicle autopilot. The autopilot computes the route towards the next waypoint and notifies the ARCA module which, in turn, checks cyclically the danger associated with the present UAV route and possible deviations, taking into account the air traffic in the neighbourhood. In order to avoid possible separation miss it can "steer" the UAV, through the adding of an extra waypoint in the waypoint list of the autopilot.

The airspace where the experiments have been performed has been graphically displayed through the use of an interface based on the Google Earth Plug-in. The geographical area used is that of central Italy, above Rome.

The ARCA algorithm accesses the various UAVs simulators, one for each UAV, with bi-directional socket links and receives data from the ADS-B Parser in order to take into account the commercial aircraft routes.

Each UAV has been physically simulated via the JSBSim aircraft simulator, which is an Open Source flight dynamics modeller (JSBSim.sourceforge.net, 2012). A parser for the ADS-B messages has been implemented on the SBS-1 system of the Kinetic Avionic (Kinetic Avionic website, 2012); it outputs the aircraft relevant data to the ARCA algorithm.

The chosen ARCA target processor is already tested and cleared for avionic purposes, but does not offer a too brilliant computing performance (approx. 125 Dhrystone MIPS, as a Pentium 90 PC). The algorithm has thus been lightened of the computation of the parent selectability. This means that each agent does not take into account the flight preferences of the parent aircraft. Nevertheless the performance of the simplified version versus the full algorithm is very similar if not better, see (Taurino et al., 2011). The flight trial simulations reviewed in the following are relative to the use of the simplified algorithm.

Table 1: Choke point scenario.

UAV number	Missed Separations		System Efficiency	
	Mean	Std.Dev.	Mean	Std.Dev.
3	0,000	0,000	0,995	0,001
5	0,200	0,551	0,968	0,029
10	0,233	0,430	0,907	0,036
15	0.933	0,868	0,852	0,030

Table 1 summarizes the results relative to one experiment where cinematic aircraft have been used (vehicles at constant speed and turning on the spot without dynamics). The results shown are relative to a choke point scenario (Taurino et al., 2010b). All aircraft start on a circle of radius 60 nautical miles and head directly through the circle centre towards the opposite side all with the same speed, statistics is on 30 trials. This is a well known hard challenge to any separation assurance method. The *system efficiency* is here computed in terms of flight time, $SE = 1/N \sum_i t_i/(t_i + t_d)$, as the average over the number of UAVs of the ratio between ideal flight time (no detour) and ideal flight time plus delay time introduced by the aircraft manoeuvres. Thus maximum efficiency is 1.

3.1 Simulated 2D Flight Trials

In the first batch of experiments a set of two or four UAVs are dynamically simulated in a segregated space, i.e. with no ADS-B flight data from commercial aircraft. Two experiments are presented: in the first, two UAVs are flown one against the other specifying the start of each one as the destination of the other. In the second, four UAVs are placed at the four corners of a square and flown towards the facing corner, passing all by the square centre. In the first case one of the two has a right of way over the other, i.e. a higher priority. In the second one a ranking in priority is set up.

In the second batch of experiments the UAVs perform the same flight paths but in a non segregated airspace (commercial traffic ADS-B data considered), this to show that the algorithm is able to

avoid separation infringements also with commercial flights.

The results for the segregated space case are presented in Table 2 (statistics on 30 trials). The results of the full traffic case are relative to six trials only, due to the impossibility to perform accelerated time simulations. In these simulations all the traffic is artificially considered at the same altitude in order to stress the algorithm and in order to guarantee the existence of several conflicts among the UAVs and the civilian traffic. There is an average number of separation infringements equal to 1.33 and a large degradation in the UAV system efficiency (average value is 0.775).

Table 2: Segregated space physical UAVs only results.

UAV number	Missed Separations		System Efficiency	
	Mean	Std.Dev.	Mean	Std.Dev.
2	0,000	0,000	0,942	0,039
4	0,000	0,000	0,945	0,049

A second set of simulation has been performed only considering aircraft flying at very high altitude, i.e. those flying at cruise speed and altitude (runway crossing scenario). The number of tests performed is three employing four simulated UAVs. In this case the average number of conflicts reduce to zero with an average UAV system efficiency of 0.912.

3.2 Simulated 3D Flight Trials

Here the civil aircraft picked up by the ADS-B antenna have been distributed on three adjacent flight levels (9000, 10000 and 11000 feet), using a simple altitude discretization. The four UAVs move in the middle flight level and are allowed to climb and descend two levels. Ten full simulations have been performed, with an average number of 0.4 missed separations (all between an UAV and a civilian aircraft) and an average traffic of 16.1 civilian aircraft. The average number of flight level changes was of 12.2, the average duration of a single simulation was of 75.5 minutes, see figure 1. Here it can be seen the user graphical display based on the Google Earth plug-in. The light blue aircraft are the commercial ones, while the red ones are the simulated UAVs, the circle surrounding each vehicle has a 2.5 nautical miles radius: if two circles don't intersect, the separation is assured.

The analysis of the missed separations in the simulations shows that these are caused by the unrealistic features of the simulation setup. Since the aircraft altitudes are discretized, an aircraft departing from or arriving to the Fiumicino airport will

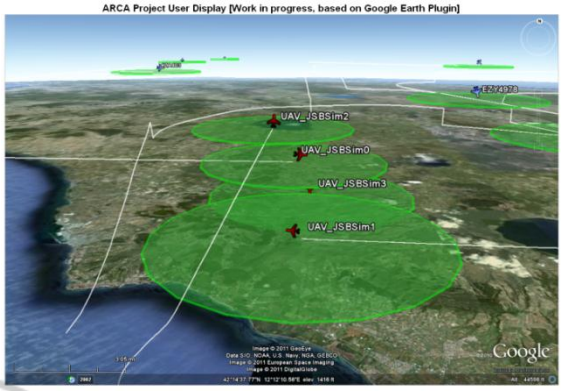

Figure 1: 3D Algorithm: westbound and eastbound UAV are on the same flight level while northbound is on top and southbound on the bottom.

quickly change flight level possibly suddenly appearing at too close quarters with an UAV. In any case such a discretization is a way to highly stress the algorithm and the results in terms of missed separation are quite interesting and show that the 3D algorithm greatly enhances the results of the 2D one.

4 HW IMPLEMENTATION

The focal point of the present paper is the implementation and the testing of the algorithm onto the EPU that will be mounted on board the actual UAV, see figure 2. The communication among the various subsystems of the UAV is based on a CAN (Controller Area Network) bus, a communication tool among devices and microcontrollers without the need for a host computer, typically in the automotive milieu.

Since the aim is to test the algorithm running on the EPU, all the other components have been simulated. The overall organization of the developed system is depicted in figure 3. The graph can be logically separated into three different parts: in the centre the UAV subsystems (i.e. the EPU running the algorithm and all the other avionic components: GPS, altimeter, etc.); in the left part the ADS-B subsystem still on board the UAV; in the right one the UAV simulator. Actually the "other avionics" box is here not considered since its data can be obtained directly from the simulator telemetry via its CAN interface.

Hardware-in-the-loop simulations have been performed, i.e. experiments similar to those described in the previous section with the main difference of having the EPU, and the ARCA algorithm on it, controlling the autopilot and thus the

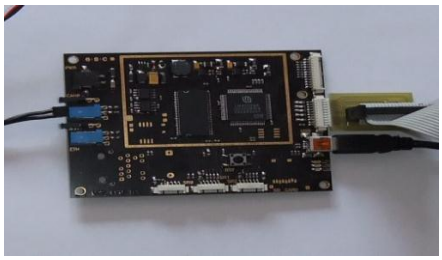

Figure 2: The EPU board with the CAN bus on is left.

simulated UAV. Once that the EPU overcomes the tests, it may be plugged-in aboard the actual physical UAV.

The integration work has been distributed in the three different areas shown in figure 3: ADS-B interface; embedded algorithm implementation; simulation interface. The two interfaces are needed in order to furnish data to the EPU through the CAN bus, while the algorithm has been embedded through the EPU development kit.

The management of everything concerning the CAN bus has been approached via the CANOpen standard (CANopen Wiki, 2012) and specifically through the use of the CanFestival open source software project (CANFestival.org, 2012). The use of such an instrument allows for a relatively easy realization of all the CAN bus standard details, such as device heartbeat, data dictionaries, callback functions etc.. The algorithm needs to fetch all the relevant ADS-B data made available by the interface, it also needs the UAV telemetry data from the simulator and should write onto the simulator the proper steering commands. In order to be certain of the complete arrival of ADS-B data, a callback function strategy has been implemented over the CAN bus communication. The first datum written by the ADS-B interface is the number of civilian aircraft in the area. As soon as data from an aircraft is written to the EPU, the callback function updates a data counter; when this data counter becomes equal to the number of aircraft the algorithm can be started.

Some early experiments have been conducted showing the separate functionality of every part of the system. Finally the EPU has been made actually control a UAV simulation. Two kind of experiments have been repeatedly carried out: in the first two UAVs (one controlled by the EPU and one freely flying in a straight line) are made fly one against the other with no external civilian traffic considered; in the second the same situation is considered but with the actual civilian traffic taken into account.

In both cases the EPU is capable to steer the UAV off the other's route making it safely fly to its

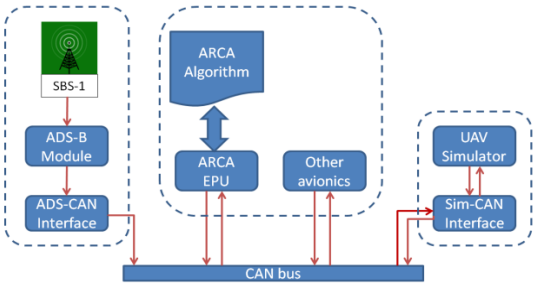

Figure 3: The hardware-in-the-loop system architecture.

final destination. In traffic conditions, ARCA is able to make the UAV avoid all the aircraft, both UAV and civilian ones (with the limitations exposed in section 3.2). The behaviour of the EPU is practically indistinguishable from the one of the software implementation of the algorithm.

5 CONCLUSIONS

In this work it has been briefly reviewed the ARCA algorithm. Several simulated flight trial experiments have been presented showing that this approach can provide acceptable performance in ensuring separation and trajectory efficiency.

The implementation of this algorithm onto a flight ready embedded processing unit (EPU) has been described, presenting some of the hardware and software details. The final outcome of this effort is a hardware physical component that can be directly plugged in the avionic system of a UAV.

The presented operative tests of the hardware component show that the ARCA EPU is capable of piloting a simulated UAV in a simulated flight test scenario, i.e. it safely maintains the UAV separated from all the other air traffic present in the area, both UAV and civilian one. The algorithm hardware implementation has the same performance as the software simulations.

From the control strategy point of view, the implemented algorithm is a hierarchical multi agent control system based on Satisficing Game Theory. It is well known that the symmetrical cooperative algorithms for separation assurances give near optimal solutions in term of global efficiency (Porretta et al., 2010). This algorithm uses instead a ranking in priority which may better reflect the real world situation. The ARCA algorithm (in its full version) features a dynamical computation of the priority ranking based on different flight considerations. Another key aspect is that the ARCA algorithm does not need information exchange

among the various agents in the scene, but only the passive collection of already available ADS-B flight data. This distributed multi agent control system can be exploited in different fields. Presently it is being considered for the collaborative control of truck traffic in restricted areas such as freight logistic centres and for the cooperative control of robotic underwater swarms.

ACKNOWLEDGEMENTS

This work has been developed within the Adaptive Routing and Conflict Management for UAVs (ARCA) project, funded by the EUROSTARS Programme of the European Commission.

The authors wish to thank Dr. Luigi La Porta for the helpful discussions and enlightening collaboration on the CANOpen and CANFestival related issues.

REFERENCES

Bellomi F., Bonato R., Nanni V., Tedeschi A., 2008. Satisficing Game Theory for Conflict Resolution and Traffic Optimisation. *Air Traffic Control Quarterly*, Vol. 16, no. 3, 211-233.

Boyan J., Littman M., 1994. Packet routing in dynamically changing networks: A reinforcement learning approach. in *Advances in Neural Information Processing, Systems - 6*, 671–678. Morgan Kaufmann.

CANFestival.org, computer software, 2012. Available from: http://www.canfestival.org/, [24 February 2012]

CANopen Wikipedia, 2012. Available from: http://en. wikipedia.org/wiki/CANopen, [24 February 2012]

Carbone C., Ciniglio U., Corraro F. Luongo S., 2006. Decision-Making Algorithms for Aircraft Autonomous Collision Avoidance. In *Proc of 5th EUROCONTROL Innovative Research Workshop & Exhibition*, Bretigny-sur-Orge, France.

Carbone C., Ciniglio U., Corraro F., Luongo S., 2007. Autonomous Aircraft Separation Assurance And Collision Avoidance Algorithm. In *Proc of the 7th ATM R&D Seminar*, Barcelona, Spain.

Choi S. P. M., Yeung D. Y., 1996. Predictive Q-routing: A memory based reinforcement learning approach to adaptive traffic control. In *Advances in Neural Information Processing Systems - 8*, 945–951. MIT Press.

ICAO International Civil Aviation Organization, 2011. Circular 328.

JSBSim.sourceforge.net, computer software 2012. Available from: http://jsbsim.sourceforge.net, [24 February 2012].

Kinetic Avionic website, 2012. Available from: http://ww w.kinetic-avionics.com, [24 February 2012].

Lukovic M., 2011. The Future of the Civil and Military UAV Market, in *Frost & Sullivan Market Insight*, 28 June 2011.

MIDCAS project website, 2012. Available from: http:// www.midcas.org, [24 February 2012].

Porretta M., Schuster W., Majumdarand A., Ochieng W., 2010. Strategic Conflict Detection and Resolution Using Aircraft Intent Information, *The Journal Of Navigation*, 63, 61–88.

Roch J. L., 2008. AIR4ALL: Feedback from Questionnaire and Workshop 1, EUROCONTROL, Brussels, 6 May 2008.

Schneider D., 2011. Top 11 Technologies of the Decade – Drone Aircraft. *IEEE Spectrum*, vol 48, no. 1, p 43, January 2011.

SESAR project website, 2012. Available from: http:// www.sesarju.eu/programme/workpackages

Stamper W., 2005. Understanding mode S technology. *Defense Electronics*.

Stirling W. C., 2003. *Satisficing Games and Decision Making: With Applications to Engineering and Computer Science*. Cambridge University Press, 2003.

Taurino D., Frau G., Beato V., Tedeschi A., 2011. ADS-B Based Separation Support for General Aviation. In *Proceedings of ATACCS*, Barcelona, Spain, 26-27 May.

Taurino D., Taraglio S., Tedeschi A., Pasquini A., Nanni V., 2010. Increasing the Autonomy of Unmanned Aircraft Vehicles with a Game Theory Approach. In *Proc of Intl. Conf. on Intelligent Unmanned Systems*, Bali, Indonesia, Nov 3-5.

Taurino D., Tedeschi A., Sanchez A., Flores A., Sysala R., Suchanek P., Nanni V., Taraglio S., 2010. Adaptive Routing and Conflict Management for Unmanned Aircraft Vehicles. In *Proc of Robotics and Applications (RA2010)*, Boston, Mass. USA, Nov 1-3.

Wolpert D., Wheeler K., Tumer K., 1999. Automated design of multi-agent systems. In *Proc. of the 3rd Int. Conf. of Autonomous Agents*.