

A MULTIPLE ROBOT SOLUTION FOR URBAN RECONNAISSANCE

B. Janko, C.P. Turner, K.S. Cave-Ayland, M.K. Pittuck, S.O.H. Madgwick, and W.S. Harwin

Published in International Journal of Intelligent Defence Support Systems (Inderscience.com) 4 no.1 (2011): 70-86 doi: 10.1504/IJIDSS.2011.037808

ABSTRACT

Coordinating aerial and ground based robotic platforms together with sensor payloads is described and some results are presented on the deployment of these robots in an urban environment for reconnaissance, in particular to find military threats. This work was done with the aim of competing in the UK MOD Grand Challenge competition in August 2008. This paper reviews the structure, control, coordination and integration of the robotic platforms for this event.

1. INTRODUCTION

In 2007 the UK Ministry of Defence announced a competition with the aim of exploring problems in urban warfare, in particular reconnaissance [1]. The competition was held in August 2008 at Copehill Down, a village purpose built for training on Salisbury Plain in Wiltshire and received entries from 12 teams. The competition was to detect four threat types: snipers, improvised explosive devices, improvised armed vehicles and gunmen. The T3 team successfully competed in the qualifying rounds and the final, and won the prize for the 'Best Use of National Talent'. In the preliminary proving rounds both ground and air vehicles were deployed but in the final event only the aerial vehicles were deployed. The T3 team was coordinated by Thales UK, and members included Thales UK, Bristol University Vision and Processing group, Ryland Research, a small business that provided a tracked chassis for the ground vehicles, Carvec Remote Vehicle Control, a small business that provided a stabilised model helicopter for the aerial vehicle and a group of undergraduate engineers from the University of Reading, School of Systems Engineering that was responsible for the robot control and coordination.

The challenge was to create a system with a high degree of autonomy that in a given hour would detect, identify, monitor and report a range of military threats in an urban environment measuring approximately 150m × 150m. The challenge was divided into two separate tasks, autonomous robot navigation and automatic threat detection.

This work contrasts with more traditional work on UAV coordination where a number of (usually fixed wing) air vehicles are used to search a larger (non urban) area for specific threats such as fires[2]. Fixed wing UAVs require some care in reconciling the limits of their Dubin paths, with the need to automatically track specific targets, and path optimisation based on Bayesian methods or mutual information has proved to be successful in these instances [3,4]. Although giving greater autonomy to the aerial vehicles there is little expectation that the air vehicles will be able to identify disparate and obscured targets in the clutter of an urban environment. The importance of the human interface in the coordination of multiple UAV's is discussed in [5] showing how autonomous search can be complemented by intervention from a human operator in a small flock of UAVs. The concept of sensor networks is relevant to this work and was explored in [6]. This work opens up the possibility of distributed sensors being used both for acquiring information and providing state

information for controlling individual vehicles. Coordination of ground and air vehicles is discussed by Hsieh et al. in [7] where benefits similar to those found in this work are identified, namely the ability of air vehicles to cover large areas at low resolution relatively rapidly, complements the ability of ground based robots to carry larger sensors to gain more accurate information about specific targets.

This paper outlines the structure of the ground and aerial vehicles used to compete in the MOD Grand Challenge. The primary mission of the robots was to carry a range of sensors into a set of usable positions within the hour allotted for the search and assist in the movement of data through a mesh network in order to get information back to the forward operating vehicle. The paper outlines the interfaces needed to coordinate the mission before going into details on the control of the individual ground and aerial vehicles. A common approach was attempted for the major components of the work including the control, the communication methods (intra and inter robot) and safety. The results of deployment of the robots in the urban environment are presented and finally a discussion is given on the extension of the concept for possible deployment in military implementations.

2. STRUCTURE OF THE ROBOTS AND SYSTEMS DEPLOYED

The principal components in the system were a forward operating vehicle (FOV), two unmanned aerial vehicles (UAV) based on a standard model helicopter, and two autonomous ground based mobile robots (AMR) based on small tracked platforms standing about 600 mm high.

The Forward Operating Vehicle was a modified Ford transit van and contained the mission control computers, the communications equipment and power. The vehicle was driven to the forward operating point and, for a specific reconnaissance mission within the competition, two operators could then deploy the four robots and then specify the robot paths and analyse the acquired sensor information. Data from the four robots was stored and processed at the FOV thus, for example, the image processing software for threat detection could be applied to the data stream coming from any of the deployed cameras. Two primary methods of external communications were used for increased reliability and redundancy. These were high power Wi-Fi (802.11a) and VHF data modems. The use of 802.11a wireless links is a well researched method in use with robotics and UAVs [8], and a high gain directional dish antenna on the FOV increased the effective range of the signal. When possible this aerial tracked a mesh node carried by one of the UAVs. The 802.11a wireless Ethernet was assumed to give a high bandwidth, but relatively low reliability link in an urban area. To compliment this data channel an additional VHF channel was used as backup that provided a low bandwidth data channel, with a high level of reliability. The combination of these two allowed the critical data and the location of the robots to be relayed back to the FOV as well as allowing manual remote control as a backup. Finally a standard 35 MHz channel for model aircraft was used as the uplink from the FOV to the UAVs.

2.1. Sensor payload

Sensor payloads were carried by both the aerial vehicles and ground robots. Indeed the primary purpose of the AGRs was to carry sensors into the area of interest. Where these sensors were directional these could be mounted on a pan tilt unit mounted on the robots and thus steered towards targets of interest, either automatically or on instruction from the FOV.

The sensor payload on the UAV (Unmanned Aerial Vehicle) consisted of a daylight camera, a thermal camera and the UAV location from the GPS. The sensor payload on the AMR (Autonomous Mobile Robot) was more general and included GPS, daylight and thermal cameras, as well as specialised sensors to detect specific threats. This data was transmitted back to the FOV for analysis via the 802.11a mesh network.

2.2. Human Machine Interfaces (HMI)

The HMI design for the base station in the FOV required a fast and intuitive interface. The HMI had three component windows. The first tracked the positions of the UAVs and AMRs, and displayed video feeds from all platforms, and ran the threat detection algorithms. This component also displayed data from the autonomous mobile robots sensors and sent individual destination waypoints to the AMRs. The second window provided path planning facilities for the AMR and allowed the operator to take over manual control of the ground vehicles when necessary. The third window provided information on the location and status of the UAV. It also provided tools for planning the UAV flight paths and for raster based area searches. These components were able to coordinate the information from multiple ground and aerial platforms although were only tested on the two UAVs and two AMRs that were deployed in the tests in the competition.

A key component across all components of the HMI was the GPS coordinate translation that enabled the user to click on a map and convert that point into latitude and longitude for any of the GPS equipment. It also allowed a GPS data sent from the ground and aerial platforms to be plotted on the HMI so that the user could see where each of the platforms was located.

A screen shot of the mission control software relating to the UAV is shown in figure 1. A similar interface was used to direct the operations of the AMRs. The interface was also used to predict future positions of the robot, and in the case of the UAV, to evaluate the accuracy of the controller model during periods of unreliable communications. The user interface also provides additional mission control functionalities such as: play/pause mission, add/remove/edit waypoints. The FOV was also used to analyse data returning from the deployed sensors, either by passing the data stream into threat detection software, or by providing images to the operators.

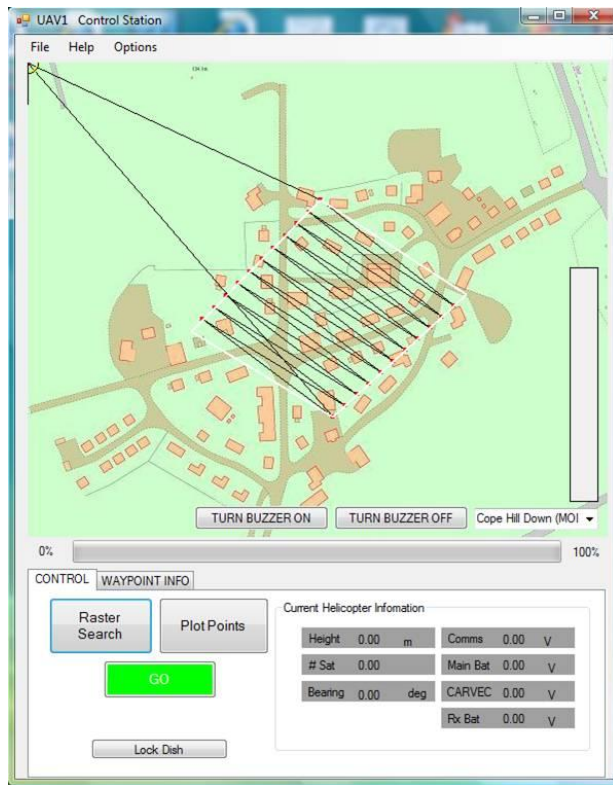


Figure 1. Mission control application, black lines indicate the flight path of the UAV over a village.

2.3. Autonomous mobile robots (AMR)

The mobile robots were designed to be robust and modular so different sensor payloads could be easily integrated. The tracked robots were received late in the project so much of the initial work was done on a bespoke robot with two drivable wheels and rear castors. This allowed the basic hardware and software structure to be evaluated prior to transferring the components to the tracked vehicle provided by Ryland Research. The initial motor controllers were bespoke CAN based controllers replaced in the final design by controllers based on an RS232 protocol. In both cases a forward velocity could be demanded and the changing position of the wheels or tracks could be monitored. In addition to odometers, other sensors used for navigation were a digital compass, a GPS receiver, and forward and down looking LIDAR. Ultrasonic range finders, and bump sensors formed part of the safety system and a camera was initially used for navigation prior to the installation of the higher resolution thermal and daylight cameras for threat detection.

The internal architecture was designed to be modular so that different sensor payloads could be added and local processing was available for compression and other computations. The AMR contained two computers running Microsoft Windows, one for sensor fusion, localisation, obstacle avoidance and local control, the second for the sensor processing in particular data from the LIDAR sensors.

An Ethernet router was used to pass messages between the computers in the AMRs and out to the mesh network. A VHF modem was available as a backup data channel. A CAN bus was used for the communication between the microcontrollers. This was chosen because it is commonly integrated with PIC family of microchips and it allowed a simple bus requiring only one pair of wires simplifying

the cabling. The CAN protocol is error correcting allowing it to work in noisy environments, and has a smaller computing overhead in comparison with Ethernet.

In the final design an additional PIC microcontroller converted CAN packets to Ethernet packets and the VHF radio was connected to one of the computers thus eliminating all RS232 connections along the principal data routes. All sensor data was relayed back to the FOV allowing the operator to monitor the status of the two AMRs. Full manual control by the operators was available at any time should either of the vehicles get into difficulty.

Although a SLAM (Simultaneous Location and Mapping) based navigation was initially planned for the AMRs the accuracy of the U-BLOX GPS receiver achieved positional accuracies of less than 2 metres which was sufficient to allow AMR navigation to be programmed as GPS way-points in a similar way to the software used for the UAV. The ultrasound sensors then provided a safety tier for the autonomous navigation in the relatively wide streets of Copehill Down. A small amount of compensation was necessary since the tracks on the AMR caused a straight-line programmed path to result in a path with a slight curve.

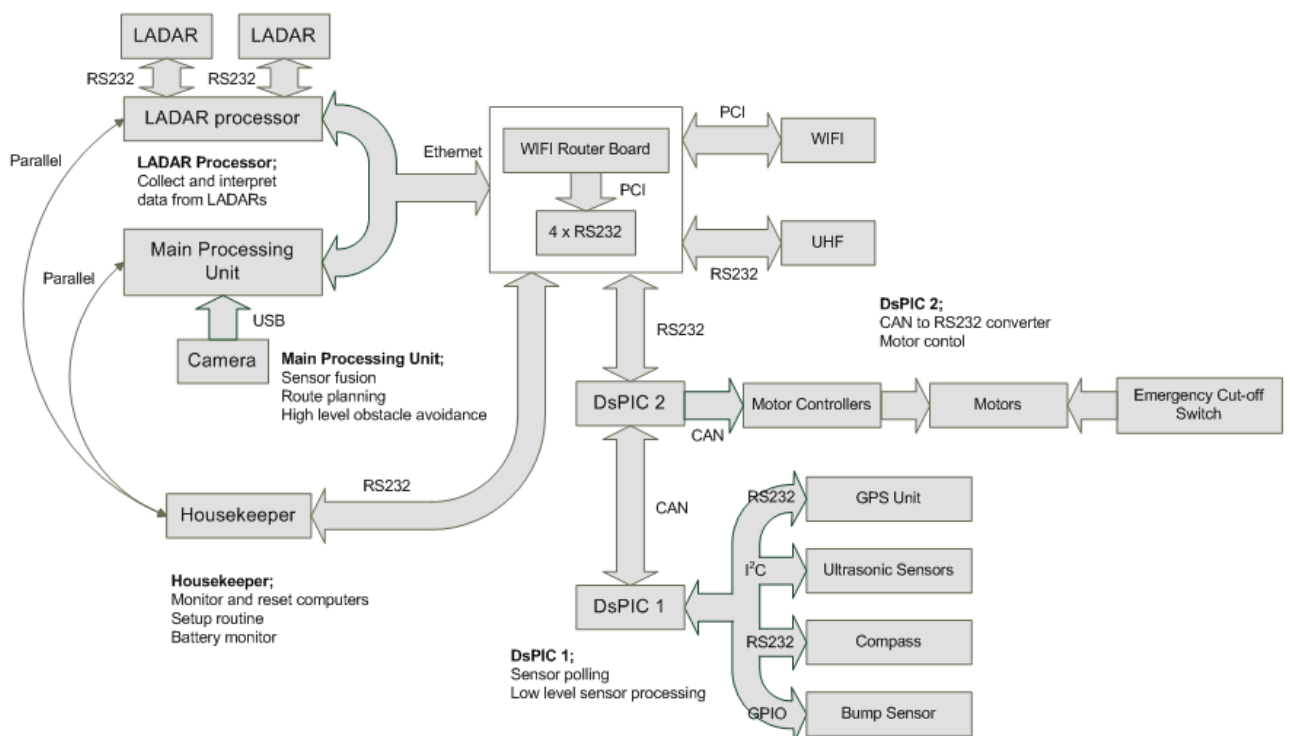


Figure 2. Internal hardware architecture of the AMRs

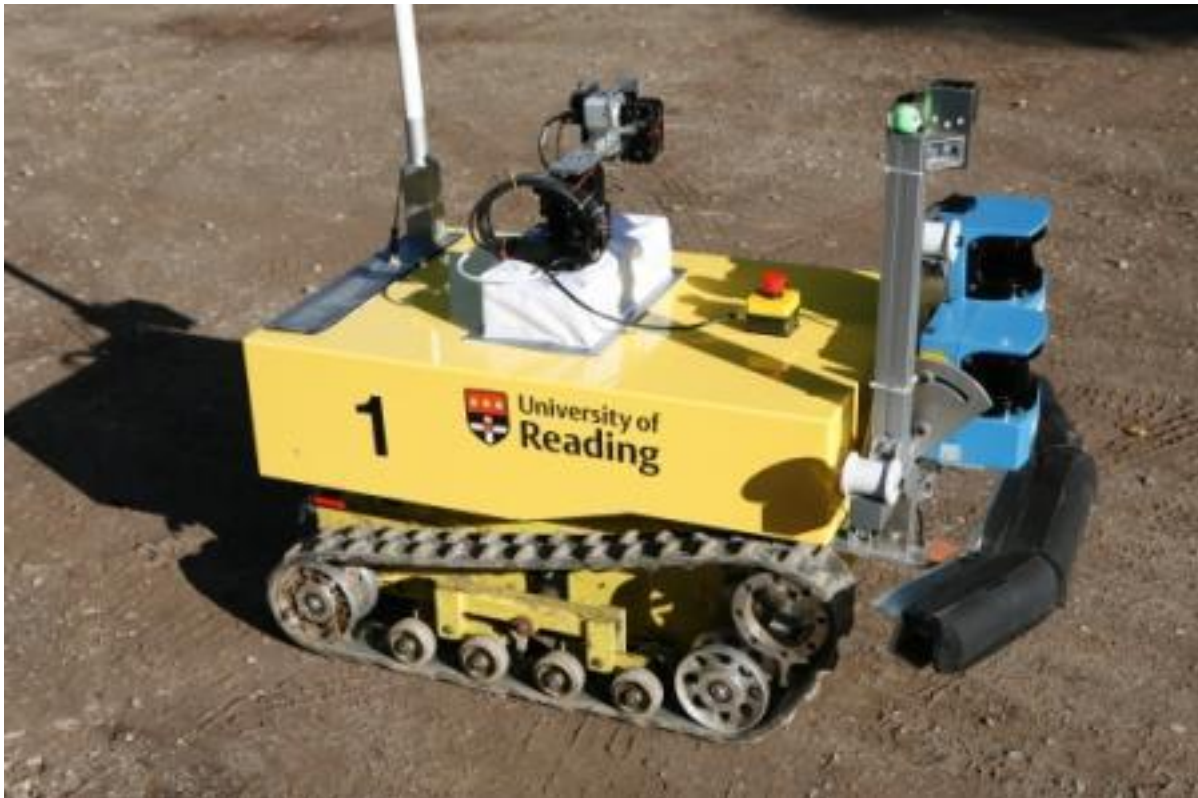


Figure 3. An AMR showing sensor payload, including two LIDARs, cameras, magnetic compass, and aerials for GPS, VHF and 802.11a. Other sensors were also carried as needed.

2.4. Safety considerations for the autonomous ground robot

The maximum speed of the AMR was similar to a fast walk so operators were permitted to approach the vehicle. An emergency stop switch was located in front of the steerable sensor pack, cutting all power to the motor controllers. Ultrasound sensors were used to monitor the area within 10 metres of the AMRs and to trigger a 'software stop'. A further tier of safety was provided by the front bump sensors that likewise triggered a 'software stop' if operated. A software based emergency stop button was also available via the HMI.

A PIC based housekeeper monitored communications from the internal hardware. Specifically the two Microsoft Windows based computers were required to generate a heart beat. If this was not detected by the housekeeper it triggered a hardware reboot. All PIC microcontrollers had internal watchdog timers that automatically reset the individual microcontrollers should the firmware stop functioning correctly.

Action in the event of an identified failure was to stop the AMR and await commands from the FOV. For local testing, as well as bringing the robots out of the FOV, a games controller could be plugged in for local control. One button on the games controller was assigned to be a 'Deadman's handle'.

2.5. Unmanned Aerial Vehicles (UAV)

Figure 4 shows the design of the UAV hardware system architecture. Software ran on a mission control computer operated in the FOV and interfaced to a standard remote control transmitter. This provided commands to the aerial platform. A computer on the air frame (a dsPIC30F4018 microcontroller) intercepted information packets from the principal flight controller and transmitted

information to the FOV via a VHF modem. This computer also measured information from a battery level sensor and from three ultrasound range sensors [9]. At the FOV this information was decoded in the high level mission control computer to close the control loop. Not shown in figure 4 was a second computer mounted on the UAV that was used to coordinate airborne sensors, in particular the two cameras, and act as a node in the mesh network.

2.5.1. The Aerial platform

The Aerial platform was provided by Carvec Remote Vehicle Control and was based on a model helicopter air frame, the Kestrel 2000 with a bespoke flight control system. The Carvec control system stabilised the Kestrel in all six degrees of freedom and offered a mode of operation where the R/C transmitter channels functioned as independent velocity demands on the x, y, z and yaw axis of the platform. In this mode, any velocity demand given to the system would be met with a preset controlled acceleration or deceleration rate; if all velocity demands are zero, the system held its current position; subject to the drift of the IMU (Inertial Measurement Unit) [10].

As well as being a stabilised flying platform, the Carvec platform also featured a stabilised camera mount with controllable tilt and roll; a 2.4 GHz video downlink with telemetry data video overlay; and an on-board flight recorder connected via a serial feed which contains all data collected by on-board sensors. The sensors making up the system were: a 16-channel GPS receiver, three axis magnetometer, three axis accelerometers, three axis gyroscopes, barometer, rotor RPM sensor, main engine, and a separate battery level sensor for each on-board battery (main battery level, auxiliary battery level, RC battery level). This data was encoded by the data parsing computer and transmitted to the high level control computer. All sensory data were available in real-time via an onboard serial link.

2.5.2. Justification for path control via the ground station

A decision was made early to minimise the modification to the aerial platform, thus leaving intact all onboard control systems and their associated safety features. This predicated that the primary mission control systems would be done via a relatively slow control loop that would include the computers in the FOV. An important consequence of this approach is that it was relatively easy to include a human test pilot who could take over manual control of the UAV if there was any suspicion of unintended behaviour. For the most part the UAV operated without test pilot help and was able to operate at significant distance (up to 800 metres) from the FOV.

2.5.3. The R/C interface

The interface was made via the standard trainer port available on most high end Remote Control transmitters. This port expected a pulse position modulated (PPM) waveform with fixed period of 20 ms made up of a number of pulses equal to the number of R/C channels + 1. The relative position of each pulse was proportional to each channel value. The PPM information was then replicated via a microcontroller to successfully gain control of the R/C channels.

The 20ms period of the waveform had implications for the control loop as it meant that any one control channel could take up to 20ms to update. This was a bottle-neck in the system; an artefact of the historical development of R/C equipment. The 115.2 kbps baud connection and efficient 3 byte packet communication to the control computer ensured that any other delays due to communications were kept to a minimum.

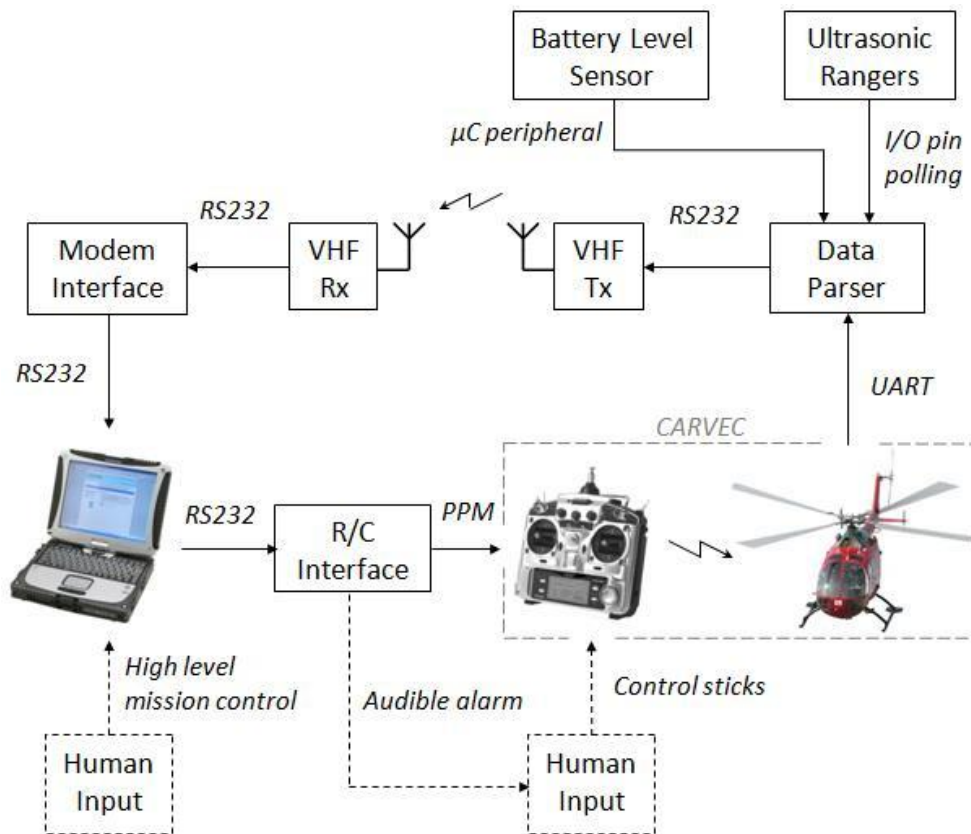


Figure 4. UAV navigational hardware architecture

2.5.4. High level mission control software for the UAV

Figure 5 shows the software structure for control and command of the UAVs. The software design was intended to be flexible rather than efficient so all messages were passed using UDP (User Datagram Protocol). Elements could therefore be easily separated across the mesh network. There were two programmes that ran on the mission controller associated with the hardware for the uplink (RC/Control) and downlink respectively. If the uplink module received a UDP message it sent it out to the helicopter. If the downlink module received a message from the helicopter then it sent it out to the network in a UDP message. All inter-process messages were logged for post flight analysis by an independent module shown in Figure 5.

Associated with the UAV control software was a module for steering the high gain dish antenna. This module received the GPS data giving the current location of the helicopter as well as the location of the FOV (the location of the antenna itself). It then generated the correct control signals to orientate the high gain dish antenna towards the helicopter. This ensured the antenna for high bandwidth wireless communication (802.11a) was pointed towards the UAV.

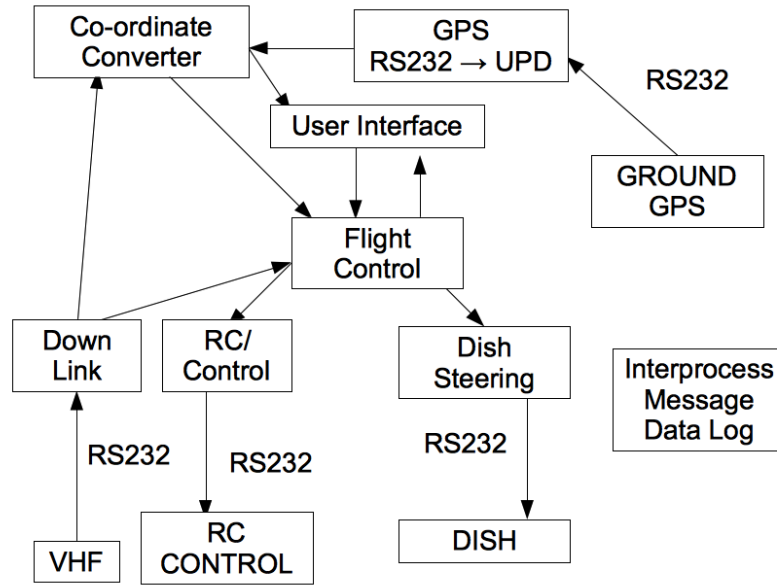


Figure 5. UAV Software structure on the high level mission controller

2.5.5. UAV flight Simulator

A UAV simulator application was created as a development tool which enabled the complete system to be tested without the use of the system hardware. Mathematical models for the simulator were based on the state space model representation of the aerial platform. Using this, output telemetry data was calculated whilst sensor and communication performance data was entered to the simulator user interface manually. The UDP communication system allowed the simulator to mimic actual in-flight data packets and test control software as if in a real flight.

2.5.6. UAV control system design

As the control for the UAV flight path was done through computers in the FOV, both the forward path (UAV control demands) and feedback path (UAV sensory data) were subject to delays. Delays anywhere in the control loop are prone to introduce instabilities. Though sources of the delays were variable (e.g. GPS 4 Hz up-dates, PPM channel update latency), analysis found the expected delay in the system to be a nominal 40 ms. Another consideration required by the controller design is the non-linearity inherent in the Carvec low-level control. These take the form of the acceleration and velocity limits affecting each channel.

2.5.7. System identification

A system model was needed for the controller design and for the simulator, and since the existing stabilised platform was too complex to model based on first principles a systems identification approach was taken. Through the analysis of the data provided by the onboard black box flight recorder and ground based software data-logger application, a mathematical model of the system could be estimated. The system was modelled as a collection of separate single input single output systems; assumed to be decoupled by the Carvec inner-loop controller. To ensure that the input upper and lower limits did not create non-linearity in the system, all demanded velocity and accelerations profiles were kept within the working linear range. An off-line Least Squares algorithm was used to estimate the separate system models shown here as equations 1 to 3. These represent the degrees of freedom relating to a forward, sideways, and rotational motion in the yaw.

$$\text{-----} \quad \text{-----} \quad (1)$$

$$\text{-----} \quad \text{-----} \quad (2)$$

$$\text{-----} \quad \text{-----} \quad (3)$$

Equations 1 to 3 allowed the assumption that the system could be considered as decoupled degrees of freedom, and that the change in position in each, was a result of the integrated demanded input velocity for each. This provided a simple model of the system for use in both the controller, and the UAV simulator software, and as expected the predominant behaviour is a gain and an integrator. Equation 2 is unstable in open loop but was easily controlled since the time constant was significantly larger than the delay in the control loop.

2.5.8. *Closed-loop control*

In order to correct for the disturbances it was necessary to use the feedback of the sensory data onboard the UAV to create a closed loop controller. To account of the delays in the controller, the use of a Smith predictor [11] architecture was employed. This control architecture can be seen in figure 6. The working principle was to use a model of the UAV to provide the constant feedback of the system for a given instant of time. The model states are corrected using the actual UAV sensory data, taking into account the constant latency of the data.

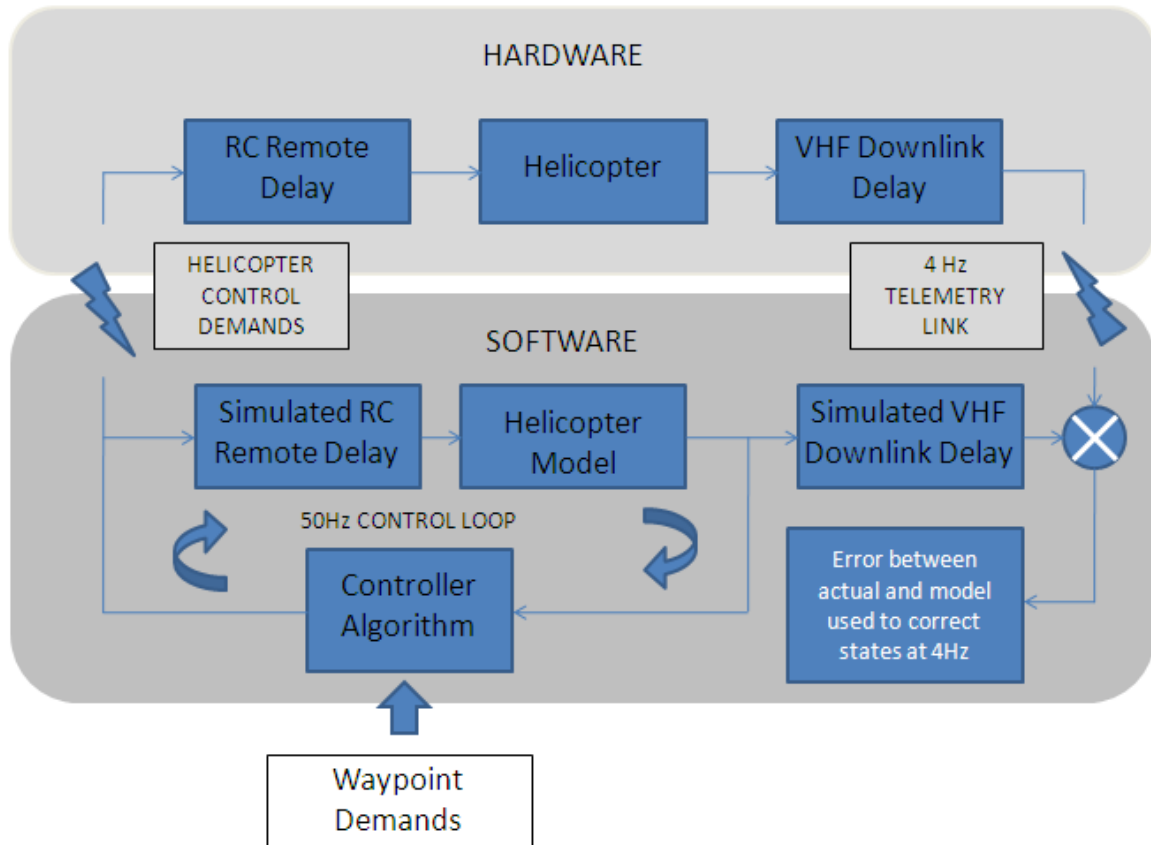


Figure 6. Smith predictor used for a single degree of freedom of the UAV

2.6. Safety considerations for the aerial vehicle.

Central to all safety systems of the UAV was the test pilot. Once airborne the test pilot held the Kestrel 2000 R/C transmitter unit in the training mode for the remainder of the flight, and could thus readily take over for manual overrides. A video downlink screen displayed the helicopter's view and vital sensor data (e.g. batteries, GPS), and the UAV mission control alarm was available to the test pilot. The test pilot was also forewarned of faults detected in the UAV control software via an audible mission alarm. This sounded upon request of the mission processor, or if any piece of software did not respond to a heartbeat or if communication link was determined to be unreliable.



Figure 7. UAV platform with sensory payload consisting of 3 cameras; high-resolution, low-resolution and thermal

3. RESULTS

The rules of the MOD Grand Challenge competition allowed an hour to gather data, and an additional 20 minutes to give a report of the threats identified in the search area. This resulted in a deployment plan with two phases. In the first phase an initial reconnaissance was done using one of the UAVs, using the daylight camera to identify areas of possible interest. This was a raster-scan of the area of interest. Concerns over wind disturbance, reliable communication links and the lack of additional proximity sensing on the UAV required the UAVs to fly above the level of the tallest building so it was not possible, either in the initial phase or the second phase, to deploy the UAVs under autonomous control to get more detailed information. In the second phase the two AMRs were deployed with the planned path based on the information from the UAV raster-scan. While the AMRs were in line of sight communication with the FOV the directional antenna could be used to ensure that the wireless Ethernet communication was kept open. During this phase a second UAV was deployed to act as a mesh node above the area of interest so that line of sight communications would continue with the UAV acting as a mesh node between the FOV and the AMR.

During the proving event this two phase approach was shown to be a reasonable approach and all the threats were identified allowing the team to qualify for the final event. During these proving events a problem with the mesh re-networking came to light. The mesh network nodes remained static for the duration of the event, thus sometimes allowing Ethernet data to travel via the UAV but only if the nodes happened to configure to use the UAV node. This problem was not resolved in time for the final event so only the first phase of the search was possible.

The performance of the UAVs proved to be robust in a range of weather conditions including moderate winds and rain. However in very wet conditions the downlink data links tended to be

unreliable so where sensors produced a high volume of data this was archived on board and analysed once the UAV landed. The VHF downlink and 35MHz RC uplink proved to be sufficiently reliable during wet conditions to allow Smith Predictor based control to operate unhindered.

4. DISCUSSION

4.1.1. Autonomous mobile robots

The use of GPS based path control was satisfactory for the UK MOD competition since the wide roads at Copehill Down were well within the margin of error for the GPS. There were also several alternative strategies available should the GPS accuracy fall below acceptable limits including the opportunity to take over direct control of the AMR. Greater flexibility in a more confined environment would undoubtedly be needed in more realistic environments and a SLAM navigation system should probably be considered.

The modularity and redundancy architecture implemented proved to be useful to adapt to different situations and sensors as well as giving a level of robustness should a component fail. Low level sensor acquisition communicated to the main processing unit via Ethernet means that sensors can be changed/upgraded with minimal firmware alterations. Also the embedded microcontrollers are inherently more reliable than computers, and since they could put data directly into the mesh network they increased the reliability of the system as a whole.

The architecture does produce its own problems. The modular nature has increased the complexity of the overall system and has required message protocols to be developed to enable communication between the processors. Although care has been taken in their design, this adds to the potential for unanticipated situations with unknown results. Likewise distributing computation between the computers and microcontrollers resulted in an increase in the number of cables and therefore a problem with connectors and wires coming loose in a relatively harsh environment.

Lastly, there remains a single potential point of failure onboard the AMR, that is if the router board fails then all communications within and outside of the platform are lost. This would disable the platform. However, the commercial router board chosen included its own watchdog timer which would at least reset the board on failure.

The primary difficulty with robot deployment was that the original nodes for the mesh network did not function as a dynamic mesh. Once a network had been established the network did not re-mesh if a node was added or lost. A last minute purchase of equipment that did implement correct re-meshing was only partially successful when replaced with higher specification equipment, because the supplier could not provide a full set of replacement nodes in time for the final event.

4.1.2. Unmanned aerial vehicles

The implementation of closed loop control meant that the UAV was capable of following mission flight trajectories and rejecting disturbances caused by wind and gusts. The combination of the CARVEC onboard low-level stabilisation system and the developed high-level closed loop control provided a robust and reliable system; capable of maintain positions and trajectories in 40 knots gusts. With the navigation and mission flight paths entrusted to the control software, a user was able to focus on the role of surveillance and target identification.

The use of a Smith Predictor architecture was successful. It not only provided a feedback path of the control system compensating for delay, it also allowed the control system to function when the downlink link became intermittent, or even lost altogether. Although the simplified decoupled model of the UAV was a good representation of the system in normal conditions, an initial test flight had revealed a fault. When heading at a high velocity in to the wind, the UAV lost height. This was due to the total lift provided by the rotor being insufficient to counter both the headwind and the UAV's weight. This behaviour meant that the assumption of decoupled degrees of freedom was incorrect. The immediate solution was to limit velocities although multi-input multi-output models would be a more satisfactory solution. The UAV was built with the possibility of adding 3 ultrasonic range sensors for obstacle avoidance. Two strategies were considered: (1) a safeguard on a pre-planned mission flight paths that would normally avoid known obstacles but may encounter problems whereupon the sensors would alert the safety pilot, (2) flying the UAV at low speeds and using the sensors to automatically navigate the UAV around or over obstacles. Neither strategy was tested and as a result all flights were chosen to rise vertically to a height greater than any buildings or other obstacles in the mission area and maintain that height or greater until just before landing.

Coordinating multiple UAVs on a single computer should be relatively easy since two applications can be run with different UDP ports. This was only a limitation of the hardware and CPU speed.

Key to the success of the research was the use of the software and safety architectures. The use of modularised distributed control software provided a flexible method of development of small UAVs for urban environments. Safety systems embedded through all stages of design enabled the rapid development and progress of the control system.

4.1.3. Coordination

The combination of aerial and ground robots was successful, and relied on the UAV to be light and fast to gain a first view of the situation, and deploy the slower AMRs with a sensor rich payload to gain more detailed information. Despite the problems with remeshing, the networked communication system is an idea worth pursuing since the low latency would allow interesting situations, such as using cameras on the UAV's to act as additional sensors for automatic control or for locating and mapping the ground vehicles in more complex or confined environments. The concept also would allow for intelligent patching of the network, thus multiple UAVs each acting as a node could move to better positions over the search area so as to increase the reliability of data communication as per the demands of the situations encountered by the ground vehicles.

On this occasion much of the coordination between the robots was managed by the operators, but moving this burden to an intelligent agent would be relatively simple.

5. CONCLUSIONS

The system structure outlined proved to be highly effective in prototyping robotic vehicles for urban reconnaissance. The techniques described allowed rapid prototyping of all the elements, and successful integration into a working system. The method was also cost effective, using off the shelf technologies as far as possible together with well documented software and control techniques. The technique also allowed for a good degree of safety for both the ground and the aerial vehicles although to be useful a production version would need to eliminate the need for the test pilot, and there will be some problems in doing so without compromising safety. It is difficult to predict the

final cost of a multiplatform coordinated urban surveillance system for military use although costs were kept low in this instance by using and adapting commercial off the shelf components. Ultimately the robots and the techniques described can only be considered as a proof of concept, robust software and hardware for military deployment would require significantly more time and expense to produce.

Although this research has explored the benefits of the system architecture for a single UAV, the system is easily scalable and would be a suitable development platform for the control of multiple UAVs.

Acknowledgements: We are grateful for the input to this work from the T3 Thales team during the two weeks at Copehill Down, in particular Darren Wright, Bruce Nicholson, Ian Pawson, Kieran Sloyan, Julian Whiffen, Nigel Fraiser Ker, Angus Johnson and John Cunningham. We are also grateful to Nigel Fraiser Ker for the photographs used in figures 3 and 7.

6. REFERENCES

1. "Welcome to the Grand Challenge", URL: <http://www.challenge.mod.uk> (2008)
2. L. Merino, F. Caballero, JR Martinez-de Dios, J. Ferruz and A. Ollero "A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires" *Journal of Field Robotics Wiley Periodicals* **23**(3/4) pp. 165 - 184 (2006)
3. D.T. Cole, A.H. Goktogan, P. Thompson and S. Sukkarieh "Mapping and Tracking" *IEEE Robotics & Automation Magazine* **16**(2) pp. 22-34 (2009)
4. J.P. How, C.S. Fraser, K.C. Kulling, L.F. Bertucelli, O. Toupet, L. Brunet, A.G. Bachrach and N.. Roy "Increasing autonomy of UAVs" *Robotics & Automation Magazine IEEE Institute of Electrical and Electronics Engineers* **16**(2009) pp. 43-51 (2009)
5. X. Ding, M. Powers, M. Egerstedt, S.Y.I.H.R. Young and T. Balch "Executive Decision Support" *IEEE robotics & automation magazine Institute of Electrical and Electronics Engineers* **16**(2) pp. 73-81 (2009)
6. P. Corke, R. Peterson and D. Rus "Networked Robots: Flying Robot Navigation Using a Sensor Net" *11th International Symposium Robotics Research Springer Tracts on Advanced Robotics (STAR) Springer-Verlag* (2003)
7. M.A. Hsieh, A. Cowley, J.F. Keller, L. Chaimowicz, B. Grocholsky, V. Kumar, C.J. Taylor, Y. Endo, R.C. Arkin, B. Jung, D.F. Wolfe, G.S. Sukhatme and D.C. MacKenzie "Adaptive teams of autonomous aerial and ground robots for situational awareness" *Journal of Field Robotics* **24** (11) pp. 991 - 1014 (2007)
8. Chen-Mou Cheng et al. 'Performance Measurement of 802.11a Wireless Links from UAV to Ground Nodes with Various Antenna Orientations', International Conference on Computer Communications and Networks, 303-308 9th Oct. (2006).
9. S. Madgwick, C. Turner and W.S. Harwin "Adaptation of an Commercially Available Stabilised R/C Helicopter to a Fully Autonomous Surveillance UAV" *Bristol International Unmanned Air Vehicle Systems (UAVS) Conference* isbn 978-0-9552644-5-0 (March 2009)
10. 'Core System Installation & User Manual', CARVEC Technical report, www.carvec.co.uk, September 2005.
11. Smith O.J.M. (1959) 'A controller to overcome deadtime'. ISA journal vol 6 pp 28-33 referenced in Becerra V. "Time delay compensation" Lecture notes for the course "case studies in control", U. Reading last delivered 2005