

Adaptation of an Commercially Available Stabilised R/C Helicopter to a Fully Autonomous Surveillance UAV

S.O.H. Madgwick, C.P. Turner, W.S Harwin

University of Reading

siu05som@reading.ac.uk

ABSTRACT

This paper presents the development of an autonomous surveillance UAV that competed in the Ministry of Defence Grand Challenge 2008. In order to focus on higher-level mission control, the UAV is built upon an existing commercially available stabilised R/C helicopter platform. The hardware architecture is developed to allow for non-invasion integration with the existing stabilised platform, and to enable the distributed processing of closed loop control and mission goals. The resulting control system proved highly successful and was capable of flying within 40knot gusts. The software and safety architectures were key to the success of the research and also hold the potential for use in the development of more complex system comprising of multiple UAVs.

BIOGRAPHY

Sebastian Madgwick is an MEng Cybernetics undergraduate student currently in his final year. His research interests include: robotics, embedded systems, control systems, and Human Machine Interfaces. For his third year project he developed an autonomous UAV. For his final year research project he is developing accelerometer sensory systems; for human motion capture. After graduating he plans to stay in research, possibly by doing a PhD.

Chris Turner is an MEng Cybernetics undergraduate student currently in his final year. For his third year project he researched Autonomous UAV control, and for his final year project he is developing hands-free UAV control techniques using head and eye tracking technologies. Post graduation he intends to go into the control engineering industry.

Professor William Harwin is head of the interactive systems research group at the University of Reading with a keen interest in novel human-system information exchanges. He is a leading researcher in the field of robotic mediated neuro-rehabilitation having previously lead research in rehabilitation and assistive robotics. He also maintains a strong research interest in haptic interfaces as these form the basis for much of his research and provide interesting opportunities for innovative solutions to difficult problems in human-machine interaction.

INTRODUCTION

This paper describes an Unmanned Aerial Vehicle (UAV) system developed as a part of a larger project working towards an entry in the Ministry of Defence grand challenge (1). This was an event inspired by the DARPA grand challenge (2) and required teams to develop autonomous robots that given an hour, are able to detect and locate specific military threats in a $150\text{m} \times 150\text{m}$ urban environment. The challenge was divided into two separate tasks; (1) autonomous robot navigation and (2) automatic threat detection. This paper is only concerned with tackling the first of these tasks. In itself, this task encompasses a number of potential challenges including: limited mission time, navigation of uneven terrain and obstacles, and complicated search areas.

The UAV was designed to take on a number of roles during the grand challenge event; with the use of interchangeable payloads. Using optical or thermo-imaging cameras the UAV acts as a scout; searching for possible areas of interest at high altitude. Using the cameras at low altitudes allows the UAV to conduct more detailed searches aiming to detect specific threats. Whilst sensing payloads are not installed, a communications payload allowed the UAV to act as radio relay for allied ground based robots.

For either role the UAV adopts during a mission, the task can be broken down into a series of separate waypoints to be executed. In order to acquire these waypoints a constant communication link with a base station computer was necessary.

The UAV system described in this paper built upon an existing, commercially available, stabilised platform developed by CARVEC (3). This avoided the complexities of low level control and stabilisation allowing development to focus on higher levels of control and intelligence.

THE CARVEC PLATFORM

The CARVEC platform consists of a combination of the CARVEC flight control system and a conventional R/C (Radio Controlled) model helicopter air frame; the kestrel 2000. The platform is controlled by a pilot using a conventional R/C FM transmitter. The system is stabilised in all six degrees of freedom and offers a mode of operation where the R/C transmitter channels function 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 preset controlled acceleration or deceleration rate; if all velocity demands are zero, the system will hold its current position; subject to the drift of the IMU (Inertial Measurement Unit).

As well as being a stabilised flying platform, the CARVEC platform also features a stabilised camera

mount with controllable tilt and roll; a 2.4GHz 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 are: a 16-channel channel GPS receiver, triple axis magnetometer, triple axis accelerometers, triple axis gyroscopes, barometer, rotor RPM sensor, main engine, and a separate battery level sensor for each on-board battery. All sensory data were available real-time via an onboard serial link.



Figure 1. UAV platform with sensory payload consisting of 3 cameras; high-resolution, low-resolution and thermal

As a large exposed-blade aircraft, the platform must take account of the associated safety issues. The CARVEC reference manual (4) describes the automated start-up and shutdown sequences used by the platform, for which, safety measures are hardcoded at every stage. Other safety features include an automated return to base upon the loss of the R/C transmitter signal, and an emergency throttle cut switch on the pilot's controls. The CARVEC platform offers a proven flight controller and aerial platform package. The final UAV system created from this is simply the result of the addition of an integrated navigation and mission processor.

REASON FOR OFF-BOARD CONTROL

The use of an on-board mission processor would have required intercepting CARVEC control lines on the platform. Not only would this have led to the modification of a proven system, it would have also disabled all existing CARVEC safety features. The implementation of an on-board mission processor would require it to be in permanent control of the platform and would lead to the R/C transmitter being unable to function as a reliable manual override. On these grounds it was decided that all integration should be non-invasive; i.e. only points CARVEC intended for integration would be used and that the inner controller

would not be modified in any way; only added to. An important consequence of this approach is that a human pilot would be able to take manual control of the UAV at any point.

SYSTEM ARCHITECTURE

1.1. Hardware

Integration of the mission processor to the CARVEC platform meant mimicking R/C control signals that represent velocity demands from the R/C transmitter; and intercepting the flight recorder serial feed to gain access to all sensor data. In addition to the existing sensors, the UAV system was designed to include three ultrasonic rangers and an additional battery level sensor for the extra on-board electronics.

Figure 2 shows the design of the UAV hardware system architecture. In this, the image of the laptop represents the base station control computer (mission processor) and the images of the R/C transmitter and helicopter represent the existing CARVEC platform components. Linking the UAV platform and base station computer together are the components that make the uplink, and the downlink. Together the uplink and downlink are the ‘air to ground’ control loop.

Components making up the uplink are the R/C interface and R/C transmitter. Making up the downlink is the data parser, VHF modems (transmitter and receiver) and modem interface. In implementing this hardware architecture, this project saw the development and construction of the R/C interface, modem interface and data parser. The VHF modems and ultrasonic rangers were sourced as off-the-shelf components.

The three pieces of hardware were developed around the dsPIC30F4018 microcontroller featuring a 16-bit architecture running at 20MIPS (5), which provides a range of peripherals that meet the requirements of all three of the intended applications. The use of such a powerful microcontroller allowed flexibility and reduced software development times. The system structure is shown in figure 2.

1.1.1. R/C interface

The main purpose of the R/C interface is to allow the mission processor control of the R/C transmitter. It is a safety critical part of the control loop and therefore must be completely reliable; any delays arising from this module had to be kept to an absolute minimum. The module interfaces to the R/C transmitter through the trainer port (6); a connector on R/C transmitters which allows a second handset to provide control signals.

The control signal required by the port was found through the reverse engineering of an old R/C transmitter. The trainer port was found to emit a waveform with fixed period of 20ms 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 and position of the control stick on the R/C transmitter. Replicating the waveform on a microcontroller output pin connected to the trainer port successfully gained control of the R/C channels. The waveform was later found to be known as a PPM wave.

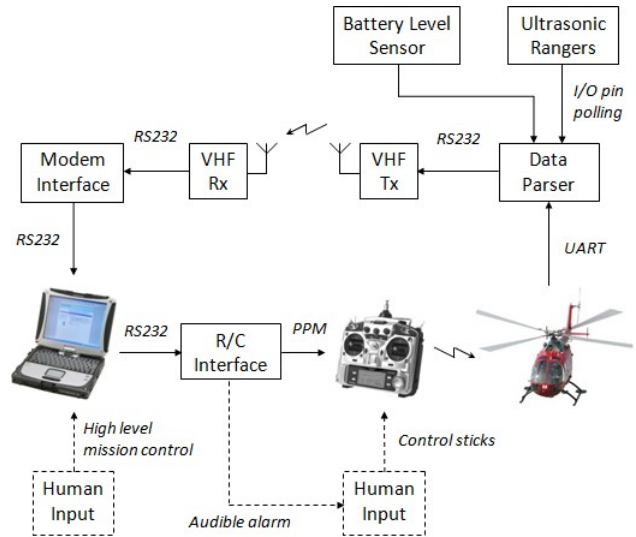


Figure 2. UAV navigational hardware architecture

The 20ms period of the waveform had implications to the control loop as it meant that any one control channel could take up to 20ms to update. This would be an inevitable bottle-neck in the system; an artefact of the historical development of R/C equipment. The 115.2kbps 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.

An important feature of the R/C interface is an audible alarm that alerts the pilot to manually override the UAV. This sounded upon request of the mission processor, or will automatically sound if any piece of software crashes or if communication link becomes unreliable. The R/C interface also has an additional mode where rather than generating the PPM wave, it samples it. This serves three purposes: (1) Calibration of PPM values, (2) A source of control for the UAV simulator, (3) Data logging of UAV system inputs for system identification.

1.1.2. Modem interface

The modem interface serves a minor role in the UAV system. Its purpose is to interface the control computer with the VHF receiver modem. In doing this, data is buffered between differing baud speeds, and any

overhead required by the VHF modem is dealt with by the interface. Essentially this ensures that the control computer connects to a fix baud uni-directional serial feed regardless of changing VHF modems or settings used.

The nature of the downlink design means that the baud rate of the modem connection can precisely match the baud rate of the incoming serial bit-stream. This not only reduces the likelihood of framing errors but also permits the use of non-standard baud rates over the VHF modems. This is made possible as the data parser and modem interface are based on the same microcontroller with matching clock frequencies and hardware UARTs.

1.1.3. Data parser

The purpose of the data parser was to continuously collect all desired data on-board the UAV and to organise it in into efficient packet protocol to be sent through the VHF modems, ultimately reaching the control computer. The main source of this data is through the interception of the CARVEC serial feed. The CARVEV serial feed continuously streams out all sensor data in six separate 32 byte packets. However, as only a small portion of this data is required by the mission processor, the data parser is required to filter each packet and decode only the bytes containing necessary information.

The data parser also operates the three on-board ultrasonic rangefinders. Each ranger is connected to two I/O pins; one being the pin of an input capture module. Using this hardware allows accurate measurements to be made in the background of main code execution. The data parser is deliberately powered from the same source as the on-board VHF modem. This allows the on-board A/D converter to monitor the battery level of all the on-board electronics. Aside from collecting all the required UAV data, a crucial function of the data parser is to organise the data into prioritised efficient packets. This is explained in section 6.

1.2. Communications

The use of 802.11a wireless links are a well researched method in use with robotics and UAVs (7). Unfortunately, the urban environment that the system is designed to operate in did not suit this method and so low band width VHF modems have been used instead. These provide a reasonably reliable data link over a long distance but require a very efficient packet protocol to be used so as to make best use of the low band width. In order to do so, the data parser organises collected data into packet types for which each packet type has an associated priority and timer. The data parser ensures that the VHF transmitter is constantly transmitting at a maximum rate by passing newly obtained data in order of priority. If at any point a packet type timer expires, it immediately becomes the highest priority. The data

packets in descending priority order are: yaw, GPS & altitude, ultrasonic sensors, all other sensors, and delay packet.

1.3. Software

The UAV software consists of numerous separate modularised applications, with most applications capable of operating indecently. The separate components that make up the system can be separated into 3 different groups: hardware interface applications, control processing applications, and the mission control interfaces. Inter-application communication was achieved through the use of the uDP (User Datagram Protocol) internet protocol. This method would also allow applications running on separate computers to communicate provide they had a network connection.

1.3.1. Hardware interfaces

In order to interface the mission processor and control software with the UAV system hardware, three pieces of software accompanied the developed hardware; the modem interface GUI (Graphical User Interface), R/C interface GUI, data logger.

The modem interface GUI held the RS232 connection with the modem interface hardware and decoded the packets transmitted from the UAV. As a stand-alone application, the GUI would display the UAV navigational data as well as a real-time evaluation of the downlink quality. Whilst running as part of the full system, the application also broadcasted the decoded data via the uDP protocol.

The R/C interface GUI held the RS232 connection with the R/C interface hardware. As a stand-alone application the GUI would enable the manual control of the UAV control channels as well as decoding and displaying the demanded controls sent from the R/C handset (if plugged present). Operating as part of the full system, the R/C interface receives UAV control demands from the control software to display and transmit to the UAV.

As a safety and debugging tool, a data logging there was also background application. This would passively pick up on all uDP broadcasts and record the data to file with a time stamp. The data provided allowed system identification techniques to evaluate the system performance with the delays introduced by all communication channels.

1.3.2. Control processing applications

There were two applications that made up the control components of the system. A simple background application was used to convert the GPS latitude and longitude data into Cartesian coordinates in the mission map frame. The calibration for this conversion would

require prior ground measurements made corresponding to recorded GPS positions, and a Least Squares algorithm used to create a conversion matrix. The main control application would process waypoint velocity profile information received from the mission control software the inputs system inputs to the closed loop controller. The controller feedback is provided by the downlink interface uDP packets, and provides outputs as the packets for the R/C interface.

1.3.3. 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. The simulator processes were based on the state space model representation of the modelled UAV system. Using this, output telemetry data is calculated whilst sensor and communication performance data is 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.

1.3.4. Mission control software

Although the UAV control software was comprised of many applications, a user would only be concerned with one, the mission control software. A screen shot of the mission control software can be seen in figure 3. The purpose of the interface is to provide the user with all required data in a simple, easy-to-process form. The UAV could be controlled via individual point-and-click waypoints or though automatically plotted raster search flight paths over target areas.



Figure 3. Mission control application, black lines indicate the flight path of the UAV over a village.

The on-screen displays provides the user with a current UAV position on the onscreen map, as well as the UAV estimated position according to controller. These separate UAV positions provide the user with a simple evaluation of the controller model accuracy during periods of unreliable communications. The user interface also provides useful mission control functionalities such as: play/pause mission, add/remove/edit waypoints.

AUTONOMOUS CONTROL

The autonomous control processes of the system translates the user define mission waypoints and search areas into the trajectories of the UAV. The trajectories can only be maintained with constant feedback of the UAV's sensor data.

1.4. Control system design

As the control processing is done off board the UAV, both the forward path (UAV control demands) and feedback path (UAV sensory data) will be subject to delays. Delays anywhere in the control loop are prone to introduce instabilities and oscillations to the UAV's motion. Though some sources of the delays are variable (e.g. GPS 4Hz up-dates, PPM channel update latency), analysis found the expected delay in the system to be a nominal 40ms. 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.

The first step in the design process was to employ system identification techniques to confirm an understanding of the system to be controlled.

1.5. System identification

The existing stabilised platform was essentially a black box system; too complex to model based on first principles. Through the analysis of the data provided by 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.

$$\frac{\text{Distance Traveled Forward}}{\text{Demanded Forward Velocity}} = \frac{1.108}{s + 0.000274} \quad (1)$$

$$\frac{\text{Distance Traveled Sideways}}{\text{Demanded Sideways Velocity}} = \frac{1.006}{s - 0.001856} \quad (2)$$

$$\frac{\text{Total Angle of Rotation}}{\text{Demanded Angular Velocity}} = \frac{1.13}{s + 0.00093} \quad (3)$$

Equations 1 to 3 confirmed that the system functions 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; a vital development tool. Figure 4 shows the results for a single degree of freedom, where the model was verified against the actual UAV for given inputs applied to both.

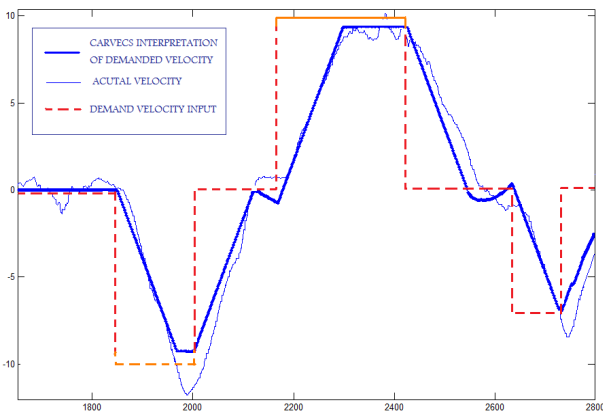


Figure 4. Comparison of UAV model and actual outputs to step inputs

1.6. Open-loop control

The first stage in the controller design was to develop the forward path of the system and achieve open loop control of the UAV. trajectory can be compared to the mission trajectory.

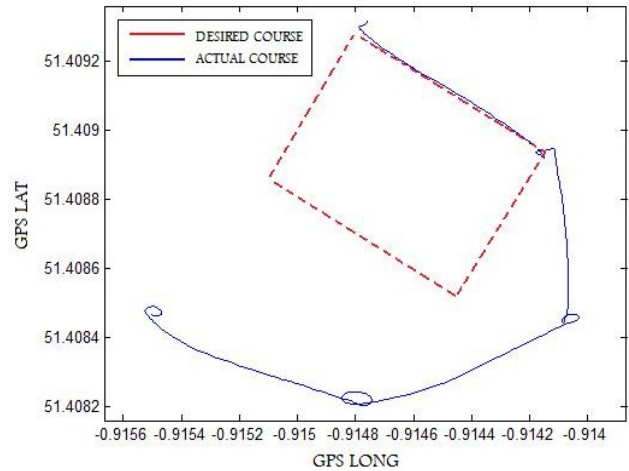


Figure 5. Desired and actual trajectory of the UAV in open loop control

In open loop control, demanded waypoints from the mission processor would be processed into velocity and acceleration profiles based on the UAV model. The results of this are shown in figure 5, where the actual UAV

In figure 5, the first ‘straight’ of the UAV’s path can be seen to match the desired trajectory; however, after the first corner the path drifts. This is due to disturbances in the actual system not represented in the model introducing errors at each corner. The result was each change in yaw would undershoot by 30 to 40 degrees. This error would accumulate over each flight and result in unacceptable errors. Also observed in figure 5 are the small circles around each corner. These can be explained by the coupling still present between the yaw and the roll. However, as the on-board controller accounts for and ultimately corrects for these motions, the assumption of decoupled degrees of freedom was considered to still hold.

1.7. Closed-loop control

In order to correct for the disturbances seen in figure 5, it is 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 (8) architecture was employed. This control architecture can be seen in figure 6. The working principle is 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.

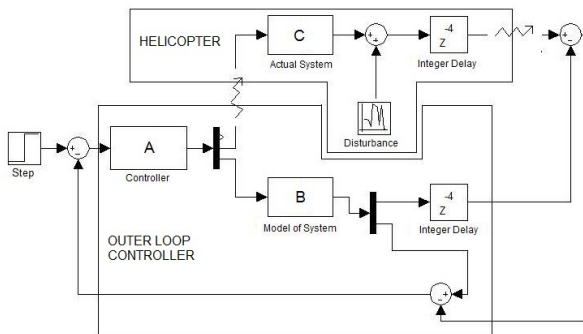


Figure 6. Smith predictor used for a single degree of freedom of the UAV

DISCUSSION

The implementation of closed loop control meant that UAV was capable of following mission flight trajectories and rejecting disturbances caused wind and gusts. The combinations 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 40knotts gusts. With the navigation and mission flight paths entrusted to the control software, a user is able to focus on the role of surveillance and treat identification. For a UAV with a sensory payload this would be done using 2 on-board cameras, one daylight, and one thermal.

The use of a Smith Predictor architecture proved to be very successful. IT not only provided a feedback path f the control system compensating for delay. It also allowed the control system to function when the downlink link became intermittent, or even lost all together. 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 solution was to limit velocities. The UAV was built with the functionality to use 3 ultrasonic rangers for obstacle avoidance with the option of taking on two strategies: (1) mission flight paths should avoid obstacles and the sensors alert the safety pilot if otherwise, (2) flying at low speeds the sensors can be used to automatically navigate the UAV around or over obstacles. Only the former method was used throughout this research.

1.7.1. Safety

Central to all safety systems of the UAV was the safety pilot. Their equipment included: the R/C transmitter for manual overrides, a video downlink screen displaying the helicopter's view and vital sensor data (e.g. batteries, GPS), and the UAV mission control alarm. During any flight, it was the experienced safety pilot whom would decide if they were happy with the UAV behaviour, and take over control accordingly. For faults in the UAV system or performance that the safety pilot may not detect, there was the mission alarm. This was a buzzer attached to the pilot's R/C transmitter. This alarm would sound of specific predefined safety critical events such as any unacceptable battery levels, GPS faults or failure, and object proximity detection. The alarm would also sound if any of the UAV control software becomes unresponsive. This was achieved through the use of heartbeat signals between applications and the R/C interface hardware. IF any heartbeat failed or the R/C interface lost physical connection with the UAV control PC, the alarm would sound.

1.7.2. Future work

C. Turner developing novel manual UAV control methods combining a degree of autonomy with eye-tracking technology. The system aims to be an intuitive flight control system that allows a single user to survey an image provided by the UAV's sensors, and their gaze within the image to control the UAV's motion.

CONCLUSIONS

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. 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 Ian Pawson, Kieran Sloyan, Julian Whiffen, Nigel Fraiser Ker, John Cunningham. We are also grateful to Nigel Fraiser for the photo used in figure 1.

REFERENCES

- “Welcome to the grand challenge“, URL: <http://www.challenge.mod.uk/>
- “Urban Challenge“, URL: <http://www.darpa.mil/GRANDCHALLENGE/>

“Carvec inertial control system”
<http://www.carvec.co.uk/>

‘Core System Installation & Use Manual’, CARVEC, September 2005.

‘dsPIC30F3014/4013 Data Sheet’, Microchip, 2007,
URL: <http://www.microchip.com/>

‘9CAP super / 9CAF super 9CHP super / 9CHF super
9CP super INSTRUCTION MANUAL’, Futaba, 2004.

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, 9th Oct. 2006.

Smith O.J.M. (1959) A controller to overcome deadtime. ISA journal vol 6 pp 28-33 Becerra V. "Time delay compensation" Lecture notes for the course "case studies in control", U. Reading last delivered 2005

All web addresses referred to in this paper were verified on 7th March 2008.