View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Loughborough University Institutional Repository

This item was submitted to Loughborough's Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/

Rapid Prototyping Flight Test Environment for Autonomous Unmanned Aerial Vehicles

Cunjia Liu, Jonathan Clarke, Wen-Hua Chen and John Andrews

Department of Aeronautical and Automotive Engineering
Loughborough University
Loughborough, UK
{C.Liu5, J.H.A.Clarke, W.Chen, J.D.Andrews}@lboro.ac.uk

Abstract: Test facility is essential for most of engineering research activities, from modeling and identification to verification of algorithms/methods and final demonstration. It is well known that flight tests for aerospace vehicles are expensive and quite risky. To overcome this, this paper describes a rapid prototyping platform for autonomous unmanned aerial vehicles (UAV) developed at Loughborough University, where a number of unmanned aerial and ground vehicles can perform various flight and other missions under computer control. Flexibility, maintainability and low expenses are assured by a proper choice of vehicles, sensors and system architecture. Among many other technical challenges, precision navigation of the unmanned vehicles and system integrations of commercial-off-the-shelf components from different vendors with different operational environments are discussed in detail. Matlab/Simulink based software development environment provides a seamless rapid prototyping platform from concept and theoretic developments to numerical simulation and finally flight tests. Finally, two scenarios performed by this test facility are presented to illustrate its capability.

Keywords: UAV; autonomous; rea-ltime; flight control; platform.

Biography notes:

Cunjia Liu received a BEng in Detection, guidance and control technology (2005), a MEng in Guidance, navigation and control (2008) from Beijing University of Aeronautics and Astronautics, Beijing, China. He is currently a PhD student in the Department of Aeronautical and Automotive Engineering at Loughborough University, UK.

Jonathan Clarke Received a MEng in Aeronautical Engineering (2008) from Loughborough University. He is currently a PhD student in the Department of Aeronautical and Automotive Engineering at Loughborough University, UK. His current research interests include parametric modelling of shallow cumulous convection, and autonomous soaring and path planning.

Wen-Hua Chen holds a Senior Lectureship in flight control systems with the Department of Aeronautical and Automotive Engineering, Loughborough University, Leicestershire, U.K. Before this, he was a Lecturer with the Department of Automatic Control, Nanjing University of Aeronautics and Astronautics, Nanjing, China from 1991 to 1996, and he held a research position and then a Lectureship in control engineering with the Centre for Systems and Control, University of Glasgow, Glasgow, U.K from 1997 to 2000. He has published one book and more than 100 papers in journals and conferences. His research interests are the development of advanced system and control strategies and their applications in aerospace engineering.

John Andrews is Professor of Systems Reliability in the Department of Aeronautical and Automotive Engineering at Loughborough University, UK. The prime focus of his

Copyright © 2008 Inderscience Enterprises Ltd.

research has been on methods for predicting system reliability in terms of the component failure probabilities and a representation of the system structure. Much of this work has concentrated on the Fault Tree technique and the use of the Binary Decision Diagrams (BDDs) as an efficient and accurate solution method. He is the author of around 200 research papers on this topic and is joint author, along with Bob Moss, of a text book, Reliability and Risk Assessment, now in its second edition, published by ASME. John is Founding Editor of the Journal of Risk and Reliability (part O of the IMechE Proceedings). He is also a member of the Editorial Boards for Reliability Engineering and System Safety, and Quality and Reliability Engineering International

1 Introduction

Recently increasing research has been devoted to the field of Unmanned Aerial Vehicles (UAVs) since it is widely believed that numerous civil and military applications can be found for UAVs (Department of defence, 2007). To develop and improve intelligence and autonomy of UAVs, advanced methodologies ranged from individual flight control, multi-vehicle control and coordination to mission planning and decision making have being developed. These algorithms need to be evaluated and verified in order to assess their practical performance, and pave the way for inserting them into real engineering practice. For this purpose a unique indoor rapid prototyping platform has been developed at Loughborough University. This article describes the details of the platform and presents some test results to illustrate its capabilities.

It is well known that flight tests are very expensive, impose high risk for personnel and assets, and requires a large airfield and heavy logistic supports. Due to these reasons, most of the research and development works on aerospace vehicles such as aircraft, missiles, and rotorcrafts are still evaluated by numerical simulation. This has been identified as one of the main obstacles for transferring advanced control concepts and methods into real engineering practice. On the other side, UAV has been one of the most active research topics driven by numerous military and civil interests. Although there is significant progress in the research of concepts, there is a lack of test facilities to verify these new methodologies. It is imperative to have a proper test facility to facilitate these research activities and de-risk the new research ideas generated from these activities. To address this issue, a number of attempts have been made recently, including various hardware-in-the-loop simulation and flight testing facilities. At Georgia Institute of Technology, an open system UAV testbed referred to as RTMax was developed to investigate flight control algorithms (Johnson *et al.*, 2004). Researchers in University of California at Berkeley use a platform comprised of a fleet of commercially available rotary-wing and fixed-wing UAVs to study applications such as autonomous exploration in unknown urban areas (Shim *et al.*, 2005). In the Aerospace Controls Laboratory at the Massachusetts Institute of Technology (MIT), an outdoor testbed consisting of a fleet of eight fixed-wing autonomous UAVs provides a platform for evaluating coordination and control algorithms (King *et al.*, 2004).

In general, these outdoor platforms provide most realistic flight tests, however they suffer a number of drawbacks. Firstly, they are expensive and have limitations on how quickly they can perform flight tests due to the constraints on accessing a large airfield. Secondly, most outdoor unmanned aerial vehicles can only be flown in good weather conditions to avoid risks. Thirdly, UAVs typically require a large logistic support team, which makes testing logistically difficult and expensive.

In contrast, indoor testbeds may provide a much more flexible, accessible and cheaper facility for UAVs and for general flight control research. The main constraints for indoor testbeds are confined space and strict requirements on avionic systems. To this end, MIT's RAVEN tesbed is the most promising indoor testbed, where an environment with a number of quad-rotor aircraft has been developed to investigate long duration missions and health management (How *et al.*, 2008). Although most of the hardware components are commercial-off-shelf parts, the software environment Open Control Platform was initially developed and provided by The Boeing Company.

An indoor rapid prototyping platform is described in this paper to perform various flight tests and other multivehicle coordination. It can verify control level and mission level algorithms into the real world in a seamless way and speed up the process from theory to practice. The main features of this testbed are:

- Flexibility and versatile. Almost all the commercial model vehicles and rotorcrafts can be operated in this environment.
- Low costs. It is designed based on the commercial-off-the-shelf concept.
- User friendly. Without being an expert in coding and electronics, researchers using this platform can focus on theoretic methods and algorithms.
- Rapid prototyping. This allows researchers to start from algorithm development to numerical simulations to final real-time flight tests on the corresponding unmanned vehicles.

2 Platform architecture and components

2.1 Design challenges and philosophy

One objective of the rapid prototyping platform is to enable researchers to test a variety of algorithms applicable to UAV coordination and control in nearly real-world scenarios. So vehicles with good handling quality and manoeuvrability are needed in this platform. Another objective is to simplify the operation on demonstrations and allow an individual to carry out the entire process from algorithm design to evaluation. That also means that components that require less modification, have good reliability and are easily maintained. To meet these demands, the platform adopted proper commercial-off-the-shelf equipments and combines them effectively. The architecture of the platform follows a simple hierarchical design. The key constraint of indoor test facility for flight tests is that the operation space is very limited. So only very small unmanned vehicles can be used to perform various realistic flight tests, rather than just taking off and landing. This implies very little payload or no payload can be put on these small aerial vehicles. The core technique in the platform is an object tracking system. That allows a ground station computer to perceive the position and attitude of vehicles in the test area, instead of mounting an onboard computer and a sensor suite on vehicles in a conventional way. The low-cost off-the-shelf radio controlled (R/C) vehicles can be used in the platform without modifications since there is no significant payload requirement on the aerial vehicles. In addition, all these separate components were finally integrated into the Matlab/Simulink environment that is widely used in academia and industry for research and development.

Both high level autonomous algorithms and low level control algorithms are built in the ground station, but control algorithms are modularized to custom to different vehicles. Modularization allows easy addition or removal of different types of vehicles, as needed for different scenarios. Each low level control model has the capability to access hardware directly to enable their functions. The structure of the platform is shown in Fig. 1.

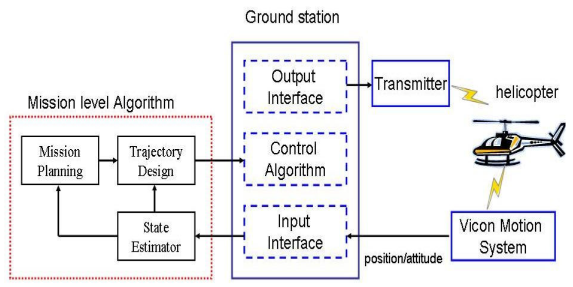

Figure 1 Platform structure

The rapid prototyping platform mainly consists of three components, which are small-scale aerial or ground vehicles, the Vicon motion system and ground station.

2.2 Aerial/ground vehicles

Small-scale vehicles play an important role in indoor tests for emulating the behaviour of real UAVs and setting various scenarios. These vehicles in the platform, especially aerial vehicles, need to be low cost, low risk but be highly flexible and have good maintainability.

As a result a radio controlled helicopter, called Hummingbird (Fig. 2), was chosen as one of the test vehicles for particular attention. It is a fixed pitch electric helicopter with a relatively low rotor tip speed, which means that there is less energy in the main rotor system making the helicopter considerably safe to operate in an indoor environment. Its plastic components can also be easily replaced after a crash. Moreover, the Hummingbird helicopter is not only cheap and commercially available form most R/C model shops, but is the one of best handling '300 class' indoor helicopters. All these reasons make the Hummingbird a good choice as an autonomous helicopter. In addition, the Hummingbird helicopter has been modified to use LiPo batteries to increase the flight times. The alteration of the battery type means a large reduction in weight and a near doubling of the flight times.

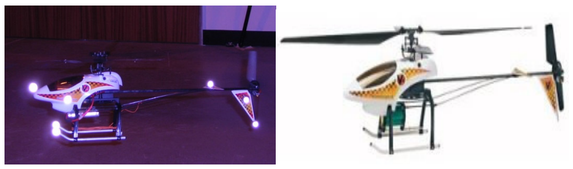

Figure 2 Hunmmingbird Helicopter

In addition to the hummingbird helicopters, there are a number of other helicopters and quod-rotors used as aerial test vehicles. The ground vehicles adopted are Tamiya TT01 cars, which is a type of R/C electric model car. These aerial and ground vehicles enable the user to construct various scenarios such as formation, surveillance, tracking, and so on. It shall be highlighted that the dynamics and control mechanisms of these helicopters and ground vehicles are very much the same to the normal ones except the scale or the change of certain coefficients.

2.3 Vicon motion system

The Vicon motion capture system provides a powerful tracking facility suitable for the indoor environment, which uses MX cameras to sense the lightweight reflective balls or belts in the operating area (Vicon, 2008). Therefore, by attaching some reflective markers to a vehicle, the vehicle can be detected by Vicon cameras. The marker position information is then transmitted via Ethernet using TCP/IP to a computer where Vicon Nexus software calculates the position and orientation of the vehicle. The MX camera and Nexus are shown in Fig. 3.

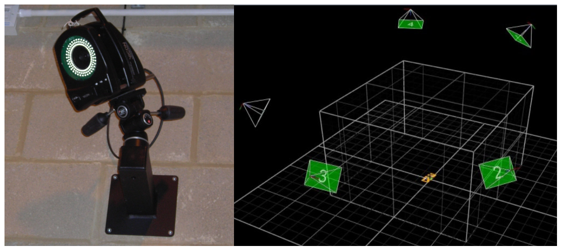

Figure 3 Vicon camera and Nexus software

Currently 5 MX and 3 T10 cameras are used to cover a 5m by 4.5m by 2m testing volume. The static accuracy was assessed by measuring the position of a helicopter sitting on the floor. The result in Fig. 4 shows that the 2 hours drift is less than 0.25 mm. Due to the principle of the motion capture system, this gives a fair indication of the position and attitude accuracy of the motion-capture ystem during flight operations. The Vicon system can capture an object motion with a refresh rate up to 100Hz and can still track the vehicle even if one or two markers on a vehicle are missing. Thereby the Vicon system can be regarded as a high bandwidth and robust navigation system in this platform. It can be considered as an indoor replacement of Global Positioning Systems (GPS) but providing not only the position but also attitude information..

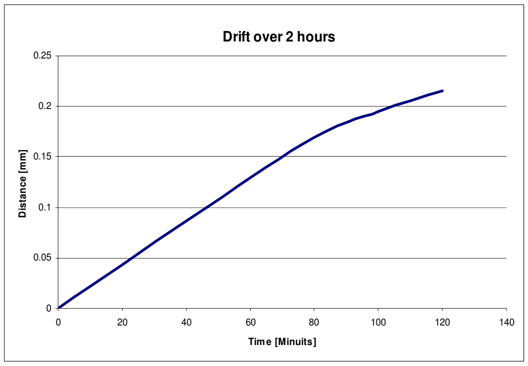

Figure 4 2 hours drift of Vicon system

2.4 Ground station

Ground station consists of several personal computers (PCs) with Intel Core2 6600 CPU at 2.4GHz and some accessories. The ground station acts as the brain of the platform, because the control and commands are sent from the ground station after tall the calculations are performed there.

To control vehicles in the test area, these computers run Vicon Nexus and Matlab applications which provides the position and attitude information of vehicles and calculates corresponding control commands based on autonomous algorithms respectively. In terms of high-level tasks, ground station also manages tasks such as mission planning and trajectory design.

In order to send control signals to the vehicles, the ground station is equipped with JR9X2 computer transmitters. The bridge between the computers and transmitters is an adapter, whose one end connects to the computers via USB port and the other end to the R/C transmitters through trainer port interface as shown in Fig. 5.

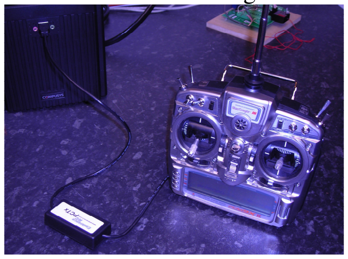

Figure 5 Transmitter and adapter

The proper selection of components is the key to success in building an advanced UAV test platform.

3 System integration based on matlab / simulink

This section describes the details of integrating both hardware and software to construct a rapid prototyping environment based on Matlab/Simulink. Although the commercial-off-the-shelf components provide necessary functionalities for the platform, a considerable system-integration effort is required to link all these components together within a single environment despite they may have software drivers/packages developed by different source codes. Usually this environment should be governed by a real-time operation system or similar software which enables all components to communicate with each other in real-time during tests.

3.1 Initial feasibility analysis

Since multiple subsystems are involved in the platform, it is essential to synchronize the execution among these subsystems in order to guarantee data compatibility, particularly when a realistic helicopter or ground vehicle needs to be controlled.

However, there exists a challenge that each hardware product has its own software or driver and is operated independently. Vicon motion system processes data captured by MX cameras using software, namely Nexus, which contains a real-time engine providing processed position and attitude information of objects. This engine can only be accessed through Vicon real-time Application Programming Interface (API) written in C language. Moreover, the adapter connected to the computer through USB port is driven by C++ style API under Windows operating system (OS). Fortunately the process in which the radio transmitter transmits command signals form a PCTx adapter to a helicopter is straightforward and needs no modification.

To achieve our purpose, Matlab/Simulink is used to build the software environment to manage all the hardware. Matlab/Simulink is a very powerful and convenient tool for control system design and simulation, which also provides a number of communication means and mechanism of integrating with C/C++ language. On the other hand, UAV autonomous algorithms, at the core of the platform are usually developed and implemented by utilizing Matlab/Simulink. This makes it a very promising candidate for the seamless transition from design to numerical simulation and real-time validation in a single software environment.

To overcome the problem that the common Matlab/Simulink programs are executed in computer time rather than real-time, two different real-time implementation environments based on Matlab/Simulink were tested by utilizing different techniques.

3.2 Real-time environment using real-time blockset

An integrated implementation environment requires the capabilities of communication and execution in real-time. Simulink provides a powerful mechanism for extending its capability, namely S-function, which is a computer language description of Simulink block. S-function can be written in C/C++, which can access to the Vicon API receiving vehicle state data and can drive the adapter sending command signals. Thus, two dedicated S-function blocks were developed to communicate with the Vicon system as well as the adapter within Matlab/Simulink environment. Despite using C/C++ during the development progress, the completed blocks can be treated as common Simulink blocks and are easy to use.

After solving communication problems, a real-time block set was added into the Simulink environment to ensure the implementation in real-time (Leonardo, 2008). This block set mainly holds the execution of the Simulink simulation to the real-time. If the cycle time is lower than the simulation step, this block set waits for the time needed to fill the simulation step, leaving the remaining CPU time to all the other Windows Processes or just idle. This concept is very simple but effective.

During the real-time tests, Vicon Nexus and Simulink run in the same PC that connects Vicon MX cameras through Ethernet and the adapter thought USB port respectively. In this manner, the time delay of data transfer can be minimized. The latency of the calculated Vicon data due to the network is less than 1ms, while the latency of sending out control signals is about 5ms on average due to the property of the USB port. Currently the environment is running at the sampling interval of 10ms, and a typical task execution time (TET) with a normal set of control and command algorithms is given in Fig. 6. The lower line represents the delay in the execution of each time interval, while the upper line represents the remaining time for which Simulink waits for the next step (and leave the CPU to remaining Windows applications).

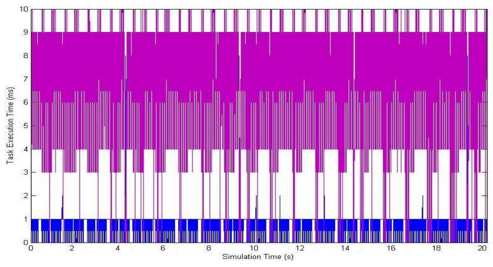

Figure 6 TET of a typical control algorithm

The software environment based on the Simulink can manage the data transfer autonomously in the background. Therefore one can use this platform as if it is a normal Simulink environment. A detailed structure of this environment is shown in Fig. 7.

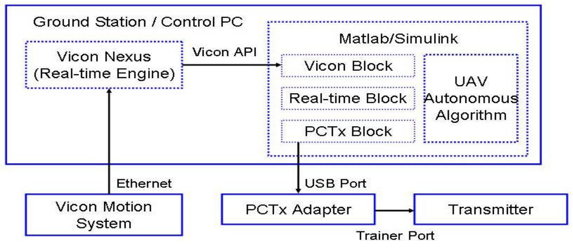

Figure 7 Structure of real-time block set environment

The advantages of this environment are that it provides many powerful toolboxes and other useful built-in resources in Simulink, and it is also very convenient to observe and record signals during the flight tests. Furthermore, it accelerates the development significantly, because there is no obstacle between algorithm development and rapid prototyping. One can implement algorithms into this platform for experiments directly as long as the numerical simulation completed in the Simulink.

Nevertheless, there are some drawbacks, which might have influence on flight tests for some scenarios in the future. For a more complicated mission, due to the heavy computation burden, it might be difficult to complete the calculation within the sampling interval when the RT Block set is used. Furthermore, from the operating system point of view, Matlab/Simulink is running on Windows OS, which is not a true real-time operating system (RTOS). That means that other applications with a higher priority in Windows system may interrupt real-time experiments and possibly cause the loss of helicopters.

To avoid these negative aspects, another real-time environment based on xPC Target was then developed.

3.3 xPC Target environment

xPC Target is a special product in Matlab for prototyping, testing, and deploying real-time systems using standard PC hardware. It is an environment that uses a target PC, which separates from a host PC, for running real-time applications. Theses applications are created from Simulink program on the host PC and downloaded into the target PC through Ethernet or serial connection. xPC Target can significantly enhance the reliability and have the capability of dealing with more complicated algorithms.

The structure of xPC Target environment is different from the previous one. Since the real-time execution is achieved essentially, the synchronising communication is the remaining issue that is of concerned in this structure. xPC Target executes its applications on a real-time kernel, where Vicon Nexus is not compatible and USB port is not supported. Therefore the target PC has to communicate with another server PC that can provide the vehicle's states calculated by Vicon and send command signals calculated by the target PC to transmitters.

There are two basic communication methods built in xPC Target. One is RS232 serial port transport, while another is User Datagram Protocol (UDP) technology. The latter one is chosen in our case, because of its high bandwidth and the ability of talking to multiple nodes in the network. Although

UDP protocol eliminates error check and recovery, it ensures that real-time applications have a maximum chance of succeeding in real-time execution by only using the most recent data. On the other hand, one should locate the target PC, host PC and server PC within a local area network (LAN) to minimize the network latency.

In addition, a C/C++ server program was developed running on the server PC. This program takes charge of the data transmission between target PC and server PC, where Vicon data are converted to UDP packets to send out, meanwhile received UDP packets are decoded into control signals to drive the transmitter. The synchronization of data transfer is implicitly dealt with in a manner that the main application on the target PC calls each communication port at fixed interval during test, whereas the server program receive and send packets passively. The entire structure of this environment is shown in Fig. 8.

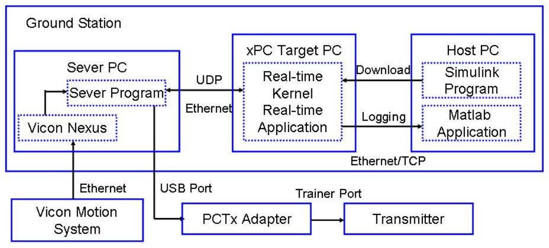

Figure 8 Structure of xPC Target environment

When doing the real-time implementation using this environment, the user needs to configure the network among these PCs first, and then compile Simulink programs into executable real-time applications and download to the target PC. This work is implemented in the Matlab environment. During flight tests, vehicle states can be visualized by the Vicon Nexus on the server PC or can be displayed in a numerical or curve form on the monitor of the target PC. After flight test all the data can be logged back to the host PC for recording or analyzing.

The merits of xPC target environment are that it provides a considerably more reliable environment for implementing various algorithms and has the potential of expansion to meet the real–time requirement for sophisticated algorithms. Although the operation of this environment is not as easy as the real-time block set environment, it is very suitable to demonstrate relatively mature algorithms into a complicated scenario such as multiple vehicle coordination and long duration mission management.

4 Possible useage of the flight test environment

The flight test facility described above is very versatile and flexible, and provides support for many activities as outlined below.

4.1 Research activities

It can provide support for the following research activities:

4.1.1 System identification and modelling

Modelling is always the first step in developing control and other strategies (Sharma, 2009). Various tests of the helicopters and ground vehicles can be performed under human remote control. All the control commands such throttle and the response of the vehicles captured by the optical tracking system can be recorded. This provides an ideal environment for identification and modelling. This function is very much similar to wind tunnels for fixed-wing aircraft.

4.1.2 Flight control

Helicopters have very complicated dynamics, with strong nonlinearities and coupling between different channels. To some extend, control of small scale helicopters is even more challenging than that of conventional helicopters since they are more susceptible to ground effects and the change of structure and propulsion. Various control algorithms can be developed using advanced control methodologies such as nonlinear control and roust control and then evaluated in this flight test environment. The control calculations are performed in Matlab/Simulink in normal PCs, which not only eases the implementation but also provides enough computing power required for complicated algorithms such as model predictive control where one-line optimization is required

4.1.3 Avionic systems

Navigation systems are a very important part of the onboard avionic systems, and provide essential information for aircraft control and positioning (Panzieri *et al.*, 2008; Fravolini *et al.*, 2008). The optical tracking system can be used as a reference system to assess the performance of various new navigation systems. For example, one research topic is to investigate the integration of low cost inertial measurement sensors with computer vision. Together with inertial sensors, a small camera can be installed on the helicopter to perform various flight tests to investigate the performance of these new concepts and algorithms. It can provide support for similar work on vehicle navigation systems.

4.1.4 Autonomous algorithms on path and mission planning

One of the most main motivations for developing this flight test environment is to support research in autonomous algorithms such as mission planning and path planning. In these high level algorithms, the UAVs are treated as a mass point, so the algorithms are largely independent of the platforms. To this end, this test facility provides an environment to verify and de-risk research work on UAV autonomy for various aerial vehicles including fixed wing aircraft.

4.2 Teaching

This flight test environment also provides support for teaching activities. Two modules Flight control Systems and Avionic Systems directly benefit from it by setting the coursework and having experimental tests in this environment. It has also been used to support various final year projects and other group design projects.

4.3 Demonstration

It is important to promote university research activities to the public, the stakeholders and various visitors. The test facility described in this paper is now one of the most popular labs for touring around the Department. It is now well received from student applicants, industrial partners and other visitors. This not only promotes the existing outcomes but also helps to attract potential funds and collaborations for expanding current research activities.

5 Real-time control tests

Before high level algorithms are applied to the platform, it is necessary to complete vehicle stabilization and control functions. This is because effective and reliable control laws are essential for mission-level tasks. In this section, a PID controller for a Hummingbird helicopter is demonstrated using real-time block set environment. Then a simple control test using two cars where they track two circle trajectories simultaneously is presented to illustrate the rapid prototyping ability of the xPC environment for multi-vehicle situations.

5.1 Helicopter flight control

In order to study the characterization of the helicopter and design the controller, a linearized model for hovering status was created, which can be written in the state-space form (Mettler *et al.*, 1999):

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \tag{1}$$

with the state vector $\mathbf{x}$ and the control $\mathbf{u}$ given by:

$$\mathbf{x} = \begin{bmatrix} u & v & p & q & \phi & \theta & a & b & w & r & r_{fb} \end{bmatrix}^T$$
$$\mathbf{u} = \begin{bmatrix} \delta_{lat} & \delta_{lon} & \delta_{ped} & \delta_{thl} \end{bmatrix}^T$$

The state vector includes 11 variables: linear velocities in x, y and z directions that are u, v and w, respectively; attitude variables: roll ϕ and pitch θ; angular rates for roll, pitch and yaw: p, q and r; rotor longitudinal and lateral flapping angle: a and b; yaw rate gyro feedback r_{fb}. In addition, the four control inputs are longitudinal and lateral cyclic δ_{lat} and δ_{lon}, the pedal δ_{ped} and the throttle δ_{thl}, respectively.

The state matrices are also shown below:

$$A = \begin{bmatrix} X_u & 0 & 0 & 0 & 0 & -g & X_a & 0 & 0 & 0 & 0 \\ 0 & Y_v & 0 & 0 & -g & 0 & 0 & Y_b & 0 & 0 & 0 \\ L_u & L_v & 0 & 0 & 0 & 0 & L_a & L_b & 0 & 0 & 0 \\ M_u & M_v & 0 & 0 & 0 & 0 & M_a & M_b & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1/\tau_f & A_b & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & B_a & -1/\tau_f & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & Z_a & Z_b & Z_w & Z_r & 0 \\ 0 & 0 & N_p & 0 & 0 & 0 & 0 & 0 & N_w & N_r & N_{ped} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & K_r & K_{rbf} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ A_{lat} & A_{lon} & 0 & 0 \\ B_{lat} & B_{lon} & 0 & 0 \\ 0 & 0 & 0 & Z_{thl} \\ 0 & 0 & N_{ped} & N_{thl} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

where τ_f is the rotor time constant, g is the acceleration of gravity, the other parameters in matrix A are stability derivatives and parameters in matrix B are control derivatives. The model structure is developed through first principles, and the values of parameters are found by the system identification process (Shim *et al.*, 2000).

Helicopters as described by equation (1) are difficult to control due to their inherent instability and coupling effects. The outer loop of translational velocity u and v are not only decided by the attitude of the fuselage, but influence by the rotor dynamics, which can be reflected by the terms of X_a and X_b. Moreover, the inner loop attitude variables are coupled with each other through terms of L_a and M_b.

Although the real helicopter shows coupling effects among different channels, in the hovering status, the system can be considered as four separate subsystems that are roll, pitch, yaw and heave channels, with the coupling treated as a form of disturbance. Therefore, the transfer function for this multiple SISO system can be derived as following:

$$\mathbf{T}(s) = \begin{bmatrix} \frac{\phi}{\delta_{lat}}(s) & \frac{\theta}{\delta_{lon}}(s) & \frac{\psi}{\delta_{ped}}(s) & \frac{Z}{\delta_{thl}}(s) \end{bmatrix}^T \tag{2}$$

Each channel can be controlled by a conventional PID controller (Castillo *et al.*, 2005). The time-domain equation of it is:

$$\begin{aligned} u(t) &= K_P e(t) + K_I \int e(t)dt + K_D \frac{de(t)}{dt} \\ e(t) &= r(t) - y(t) \end{aligned} \tag{3}$$

where $e(t)$ is the error between the required and the actual signals. K_P, K_I and K_D are the proportional, integral and derivative gains, respectively.

The overall flight control adopts a cascaded structure with two loops. The inner-loop controller is composed of four SISO controllers as described above to make sure the helicopter following the required attitude profiles. The outer-loop controller is used for guidance purpose generating the attitude commands based on the position of the helicopter and the trajectory to track. The controller structure is shown in Fig 9.

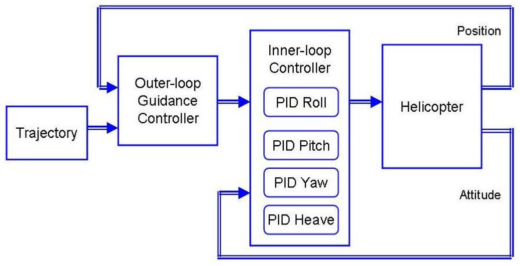

Figure 9 Flight control structure for helicopters

The proper gains for each PID controller that can stabilise the rotational and translational dynamics of the helicopter are needed. First, based on the transfer function of each channel in equation (2), the rough ranges of the gain values were determined by using Matlab model-based control design tools. Then these gains were empirically adjusted during flight tests to get a satisfactory performance.

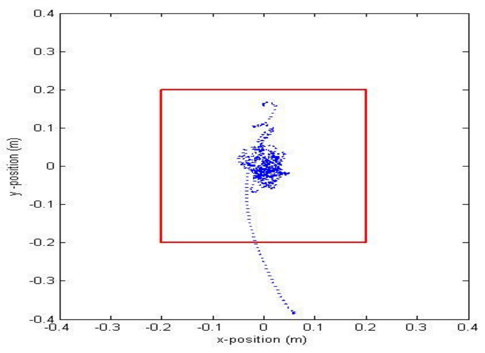

Figure 10 Hovering result of the helicopter

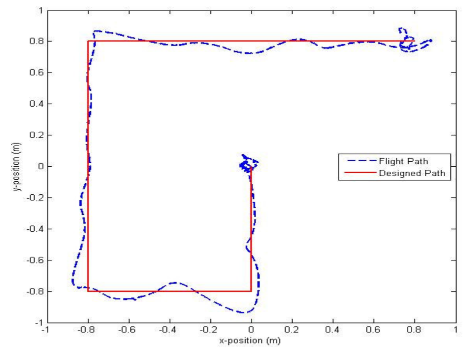

Figure 11 Tracking performance of the helicopter

The helicopter hovering test result and simple trajectory tracking result are given inn Fig. 10 and 11. The hovering result demonstrates that the helicopter can hold it position during flight. In this test, the Hummingbird took off, hovered at 0.5m above the origin and then landing autonomously. The result in Fig.10 shows that the position of the helicopter remains inside the 20-cm box during the hovering except the takeoff stage. In the tracking test, the Hummingbird was controlled to follow the trajectory consisted of 5 waypoints. The result in Fig. 11 shows that the helicopter can track a path at low speeds with maximum position error less than 20 cm.

5.2 Multi-vehicle control

To illustrate that the xPC environment and the ability to demonstrate coordination algorithms, two cars were controlled to track two round trajectories with different diameters.

The configuration of a rear-wheel driving vehicle, as the radio controlled model car used in our lab, is shown in Fig.12.

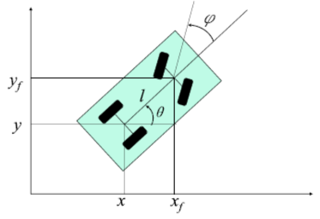

Figure 12 Vehicle configuration

The state variables of the model are $[x \quad y \quad \theta]^T$, where (x, y) are the coordinates of the centre point of the rear axle. θ is the heading angle of the car body with respect to the x axis. Angle φ in Fig. 12 is the steering angle of the front wheel with respect to the vehicle's longitudinal axis, which is a control input. Another parameter of the model is the distance between the front axle and the rear axle, which is l. The kinematical relationship can be described using the following mathematical model:

$$\begin{aligned} \dot{x} &= v \cdot \cos\theta \\ \dot{y} &= v \cdot \sin\theta \\ \dot{\theta} &= v \cdot \frac{\tan\varphi}{l} \end{aligned} \tag{4}$$

where, control input $u = [\varphi \quad v]^T$ is the steering angle and line velocity. Notice that there is a nonholonomic constraint related to a car-like vehicle, which refers to the constraint of rolling without slipping between the rear wheels and ground, and that can be represented as follows:

$$\dot{x}\sin(\theta + \varphi) - \dot{y}\cos(\theta + \varphi) - \dot{\theta}\cos\theta \cdot l = 0 \tag{5}$$

For the demonstration purpose the speed of each car is limited to a constant value, whereas the steering angel is controlled by a PID controller to follow a circle. As shown in Fig. 13, although the start points were not at the expected circles, the controller can make the cars following the trajectories.

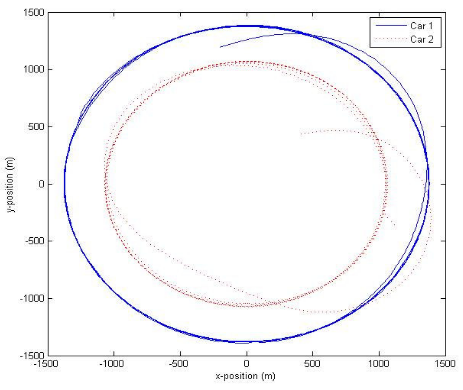

Figure 13 Two cars tracking

6 Conclusion

In this paper, the hardware and software structure of a rapid prototyping flight test environment for autonomous UAVs is discussed in detail. This platform provides a convenient and effective facility for evaluating UAVs and general flight control research activities using real vehicles. This is a very versatile testbed and various scenarios including single aerial vehicle, multi-vehicles, mixed of aerial and ground vehicles can be tested. As outlined in Section 4, it can provides support for a variety of research, teaching and demonstration activities. The adoption of the widely used Matlab/Simulink environment enables researchers to test new research outputs seamlessly on real vehicles. This multifunctional, low cost and flexible indoor flight testbed also enables one researcher to manage/coordinate missions with multi-vehicles, which significantly reduces manpower and logistic supports required for this kind of tasks. Another feature of this platform is that model helicopters are adopted as aerial vehicles. Due to the nature of helicopter dynamics such as hovering, vertical takeoff/landing and low speed cruise, this allows realistic and complicated missions to be simulated in a confined space. Various scenarios such as helicopter hovering and tracking, and two ground vehicles coordination and helicopter tracking a moving ground vehicle have been successfully implemented on this platform to illustrate its capabilities.

References

Castillo, C. L., Alvis, W., Castillo-Effen, M., Moreno, W. and Valavanis, K. (2005) "Small scale helicopter analysis and controller design for non-aggressive flights," *Systems, Man and Cybernetics, IEEE International Conference on*, vol. 4, pp. 3305-3312..

Department of Defence (2005), USA. "Unmanned Aircraft Systems Roadmap:2005-2030. " *Office of the Secretary of Defence*.

Fravolini, M. L., Campa, G. and Napolitano, M. R. (2008) Modelling and performance analysis of a machine vision-based semi-autonomous aerial refuelling. *International Journal of Modelling, Identification and Control* Vol. 3, No.4 pp. 357 – 367.

How, J. P., Bethke, B., Frank, A., Dale, D. and Vian, J. (2008) "Real-time indoor autonomous vehicle test environment," *IEEE Control Systems Magazine*, vol. 28, pp. 51-64.

Johnson, E. N., Schrage, D. P., Prasad, J. V. R. and Vachtsevanos, G. J. (2004), "UAV flight test programs at Georgia Tech." *Proceedings of the AIAA "Unmanned Unlimited" Technical Conference, Workshop, and Exhibit*.

King, E., Kuwata, Y., Alighanbari, M., Bertuccelli, L., and How, J. P. (2004) "Coordination and control experiments on a multi-vehicle testbed," *Proceedings of the American Control Conference.* vol. 6, pp. 5315-5320.

Leonardo, D. (2008) "RT Blockest for Simulink," http://leonardodaga.insyde.it/Simulink/RTBlockset.htm.

Mettler, B., Tischler, MB. and Kanade, T. (1999) "System identification of small-size unmanned helicopter dynamics," *Annual Forum Proceedings of American Helicopter Society*

Panzieri, S., Pascucci, F. and Setola, R. (2008) Simultaneous localisation and mapping of a mobile robot via interlaced extended Kalman filter *International Journal of Modelling, Identification and Control.* Vol. 4, No.1 pp. 68 – 78.

Sharma, S. (2009) Adaptation of modelling and identification: an aeronautical engineering perspective. *International Journal of Modelling, Identification and Control*. Vol. 6, No.2 pp. 104 – 109.

Shim, D. H., Chung, H., Kim, H. J. and Sastry, S. (2005) "Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles," *AIAA GN&C Conference, San Francisco*, pp. 1-8.

Shim, D. H., Kim, H. J., and Sastry, S. (2000) "Control system design for rotorcraft-based unmanned aerial vehicles using time-domain system identification," *Proceedings of the 2000 IEEE International Conference on Control Applications*.

Vicon, (2008) "Vicon Motion Capture Systems," http://www.vicon.com/products/viconmx.html.

Xi, W. & Baras, J.S. (2007), "MPC based motion control of car-like vehicle swarms", *Mediterranean Conference on Control & Automation*.