Preprints of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, July 3-6, 2012

Coaxial UAV Helicopter Control Laboratory Design

Bernardo Morcego

Abstract— **This paper describes the design of a low cost laboratory platform to perform control related exercises with an UAV (Unmanned Aerial Vehicle). All the elements this platform is made up of are very common laboratory instruments or of the shelf toys. The types of experiments one can perform are SISO and MIMO UAV modeling and identification as well as control.**

I. Introduction

THE laboratory described in this paper is part of the project titled 'μUAVs Control Project'. This project is developed by members of the department of Automatic Control at the Universitat Politècnica de Catalunya, all belonging to the Advanced Control Systems research group. The project begun in September 2007 and it intends to study control algorithms and strategies using electric indoor toy micro helicopters. In order to do so, a cheap and very easy to use platform (or laboratory) was to be designed and developed. That platform should be useful for research and educational purposes.

The aim of this project is to study the application of advanced control algorithms on hard-to-control dynamic systems. The chosen systems are small scale helicopters which are examples of nonlinear, unstable, multivariable and coupled systems.

During the development of the experimental platform several questions arose. They are now part of the sub-objectives of this project, but not all of them are dealt with in this paper. Those sub-objectives are:

- Identification of unstable systems
- Identification of closed loop systems
- Control structure for a helicopter autopilot
- Adaptation of controller algorithms to changing flying modes
- Fast computer vision localization of a moving object
- Fast computer vision orientation determination of a moving target

The platform is currently in a mature full production stage and can be used for research purposes and control-related exercises in all kinds of engineering curricula. Its primary use, though, is master's thesis.

Manuscript received January 31, 2012. This work was supported in part by the Spanish Exploration and Innovation Ministry under project DPI2008-00403 (TEOMAPIN).

B. Morcego is with the Advanced Control Group, Universitat Politècnica de Catalunya, Spain (phone: +3493 7398144; fax: +3493 7398628; e-mail: bernardo.morcego@upc.edu).

Before undertaking the design of the platform a thorough review of similar projects was made. There is an increasingly large number of laboratories and people working in the development or control of unmanned aerial vehicles. Most of them, however, use outdoor platforms mainly (see, for instance, the Helicopter Lab at Carnegie Mellon Robotics Institute, the Unmanned Systems Laboratory at University of South Florida, The UAV Research Facility at Georgia Tech, the UAV group at Aalborg University, the Autonomous Flying Vehicle Project at University of Southern California, the Stanford University Autonomous Helicopter). Some laboratories and people also work with indoor platforms, but they are essentially of quadrotor configuration (see, for instance, the Robotics and Intelligent Systems Lab at City College of New York). The number of coaxial indoor projects or facilities is low, to the best of our knowledge.

The UAV Research Facility at Georgia Tech developed a coaxial indoor UAV, [1]. This aircraft is a commercial toy helicopter modified to accommodate sensors, computing resources, communications, etc. The result is an autonomous helicopter, capable of flying indoors. One of the project's constraints is weight because commercial toys are designed to carry their original weight, not more. The resulting helicopter is 50% heavier than the original one, 410gr initially and 600gr after adding the instrumentation.

There are two projects, muFly (see [2]) and μFR (see [3]), in which the main objective is to design a helicopter rather than to design a laboratory platform. muFly is a larger project and they put a lot of effort in the design of most of the parts of the helicopter. The μFR also has a special steering mechanism and the experimental setup they use involves a second commercial helicopter, apart from three computers and one camera platform to follow the aircraft.

Other projects more closely related to the one described in this paper are [4], [5] and [6]. They all do the control computing in a ground control station, rather than onboard like the projects explained before. Paper [4] is especially relevant because they use markers to track the 3D positions of a commercial Mosquito helicopter. That paper was known in the beginnings of the present project but by that time one of our main concerns was not to modify the toy helicopter in any way. Therefore their solution seemed unviable to us. However, the solution explained here, developed independently, is similar to the one explained there. Paper [5] describes the use of an onboard IMU and a bluetooth connection to send all the information to the controller base.

Finally, paper [6] describes a very original solution that uses leds mounted on the blades to localize and identify the helicopter.

This paper describes in detail the hardware and software involved in the platform. The second section gives a summary of the main components of the laboratory. Sections III and IV are more in depth descriptions of the hardware adjustments and software developments, respectively. Section V gives experimental results and the last section concludes the paper.

I. Brief Description of the Platform

The experimental platform consists on a small helicopter flying indoors in a 2m cubic space. The helicopter carries no instrumentation at all, therefore its position and attitude measurements are obtained with external sensors and the control commands are sent through the original remote control transmitter commanded by the control computer.

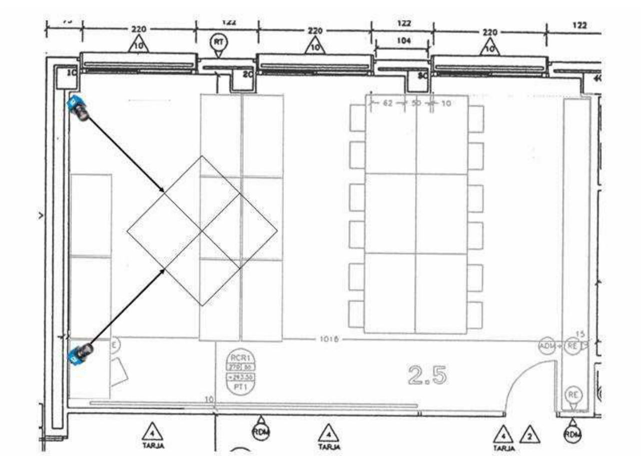

Fig. 1. The Advanced Control Laboratory floor plan, with a representation of student tables and the µUAV control space on the center, left hand side.

The control space is an imaginary cube of 2m side in the Advanced Control Laboratory of the group. This space, shown in figure 1, is watched by two cameras situated at 2.5m from its external surface and situated at 90º angle with respect to each other. Two different computer applications are involved in the development of this platform, they are: the sensor application and the control loop application.

The sensor application is a computer vision application, capable of detecting a moving irregular object and calculating its position (X, Y, Z coordinates) and attitude (*pitch*, *yaw*, and *roll* angles) in the control space. The sensor, a stand alone WinXP application, serves its measures every 1/30th of a second in TCP port 20248 at demand of any application which may need them.

The control loop application is a Matlab script that uses timers and `pnet` sockets to retrieve the measurements from the sensor. After that, the calculated control actions are issued through the actuator. This application is not a controller, it is a wrapper for any control algorithm the user may want to test. It takes care of the interface between the controller and the data input/output and the real time execution issues. The Matlab script allows the user to set three different types of interaction with the actuator: simply read in the joystick positions the human pilot is sending to the helicopter; send the helicopter those readings plus any calculated value (for identification purposes, for instance); or act as a traditional closed loop controller in which the human pilot has no control over the helicopter.

The actuator is a hardware arrangement. The original RC transmitter was modified in order to allow external signals to be taken as the signals given by its own joysticks. Consequently, using two standard Advantech PCI input/output cards, RC transmitter joystick positions can be read and set from Matlab. The RC transmitter also has a safety switch which disables any external connection and only pays attention to the human pilot commands.

II. Hardware System

The hardware arrangements for this laboratory platform to work are not very complex. They are needed for the sensor and the actuator applications to work properly. The overall hardware needed follows:

- one Intel Dual core 2 laptop
- two The Imaging Source DMK 31AU03 cameras (1024 x 768 pixels, monochromatic, Y800, 30fps)
- two Edmund Optics UV/VIS-Cut Filter R-72 Mounted M30.5 x 0.5mm
- input/output card Advantech PCI – 1711U (16 A/D channels/2 D/A channels)
- output card Advantech PCI - 1720U (4 D/A channels)
- one Belkin PCI Express FireWire 800 card (F5U602ea)
- several toy helicopters with their radio controllers. Currently using Walkera 5G4 or Walkera 5#10 models.
- 9 OSRAM SFH 4250Z leds
- one 3.7V 400mAh LiPo battery
- cables, soldering material

The electronic cards are all plug and play and the rest of the hardware needs no specific configuration. Only the cameras need to be calibrated (both extrinsic and intrinsic parameters) but this is operation needs to be done only once; for more information on this topic see the Camera Calibration Toolbox for Matlab and references therein. The computer, because it runs on top of a general purpose operating system, needs to have installed only the essential applications.

A. *Sensor hardware*

The sensor is purely a software piece. It will be explained in depth in the following section. However, in order to make it work properly, it is necessary to use infrared leds on the helicopter and infrared filters in the cameras.

Figure 2 (left) shows the actual led distribution. They are placed in the middle of the longitudinal axis of the helicopter making a right-angled triangle. Each led position needs in fact 3 leds, mounted in a turret and forming 120º with each other, to be detected by the camera in any possible orientation of the helicopter.

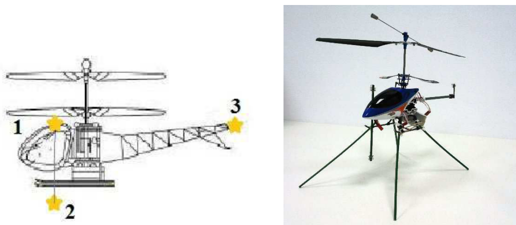

Fig. 2. Distribution of led turrets in the helicopter (left) and picture of the modified helicopter (right).

B. Actuator hardware

The actuator is the original RC transmitter sold with the toy helicopter. Some modifications were made to enable it as a proper interface between the human pilot, the helicopter and the computer.

The transmitter is a Walkera WK-2401 model. It is a four channel PPM 2.4GHz device. There are several helicopters compatible with this transmitter, in fact any helicopter with receivers Walkera RX2410 or RX2401 are compatible. This transmitter offers a four channel flight which is enough to maneuver a helicopter confidently.

The objective is to be able to read in the commands given by the pilot in the computer, send computer commands to the transmitter and resend them through the transmitter to the helicopter, as shown in figure 3.

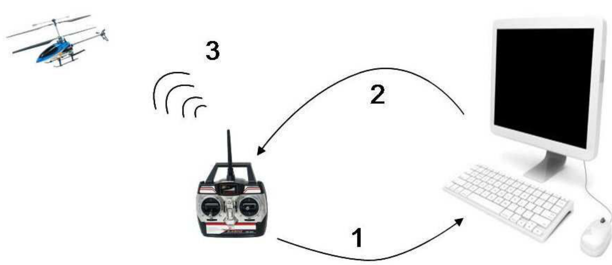

Fig. 3. Conceptual diagram of the actuator capabilities. Input and output interface with the computer and output interface with the helicopter.

One A/D card is needed to read in the four signals from the joystick positions of the transmitters and another D/A card is needed to write the computer commands. Advantech PCI – 1711U and PCI – 1720U cards are used.

The commands of the pilot are collected through mechanical joysticks bonded to two precision potentiometers each. The middle connector of each potentiometer carries a tension (between 1V and 4V) proportional to the joystick position. This connector, which is originally soldered to the transmitter board, is isolated from the board and its signal is taken out to a manual selector, following the connections of figure 4. The selector's middle pin is connected to the transmitter board, where the potentiometer was previously connected.

This way it is possible to manually select the signal that the board reads from the internal source of the transmitter or an external source. Moreover, the action commands given by the pilot can be read from outside the RC transmitter and can also be fed back to it. This way, the actuator application is fail safe.

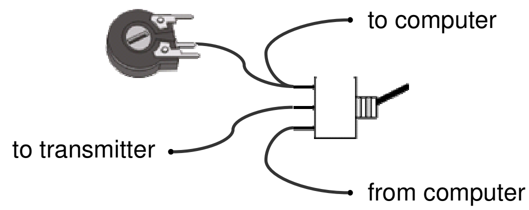

Fig. 4. Diagram of the joystick potentiometer connections. The middle terminal of the potentiometer is connected to the computer A/D card and to the selector. The manual selector sends the transmitter either the signal from the computer D/A card or the signal from the potentiometer.

III. SOFTWARE APPLICATIONS

The two software applications interacting in this laboratory platform are the sensor application and the control loop application. The sensor is developed in C++, using the standard libraries of Windows console applications and the OpenCV library, for computer vision specific classes. The control loop is a collection of Matlab scripts. Nevertheless, the control loop application could have been programmed with any platform capable of reading from standard TCP/IP ports and reading/writing to Advantech PCI cards.

A. Sensor application

This application is a Windows XP console program. Its objective is to obtain the position and attitude of a flying object in the control space and make them available to other applications.

The measurements are calculated from the images obtained from the cameras in the laboratory. Each image is delivered to the application independently at a rate of 30fps. The XYZ coordinates and pitch, yaw, roll angles of the helicopter are transmitted to any other application that might need them through TCP port 20248.

The structure of the sensor application was designed in the beginnings of this project but its features have evolved through time. The structure of the application is shown in figure 5.

It is a multithreaded application with four threads executing concurrently. Thread 1 and 2 are the image processing threads. Each one of them carries out exactly the same processing on the left or right images obtained from the cameras. The data they produce are sent to the triangulation thread, which calculates the 3D position and attitude. After that, thread 4 sends through TCP port 20248 a string of text with a prespecified tagged format, containing the tick number, the reliability number, the *x*, *y* and *z* positions, the *pitch*, *yaw*, and *roll* angles and the time stamp. The format and meaning of those values are discussed in the next section.

The processing carried out by threads 3 and 4 is rather standard. Thread 3 calculates position and attitude by 3D reconstruction following typical computations; see [7]. Thread 4 is a TCP/IP server using sockets. It can only attend one connection at a time. Conversely, threads 1 and 2 contain the most challenging procedure, which is obtaining the position and attitude of an irregular object moving in an image. This procedure is explained next in detail.

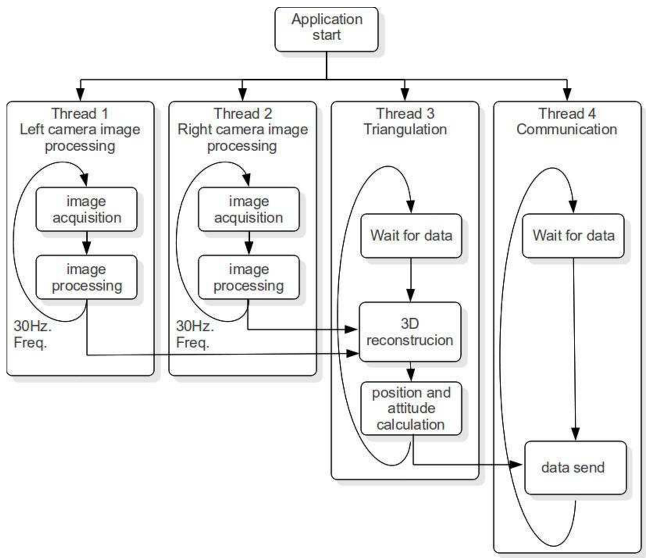

Fig. 5. The sensor application structure. Two threads are devoted to image acquisition and processing and the remaining threads perform the geometrical calculations and the data server tasks.

The processing carried out to detect the helicopter has evolved over time. First attempts worked with background subtracted images. The remaining object could be easily segmented and it was blobbed to extract its moments. The center of gravity was useful to approximate the position of the helicopter. Nevertheless, the shape variability of the blob depending on the attitude of the helicopter made it too difficult to obtain any further information from it.

The current sensor application is much more accurate and simple due to the use of infrared leds on the helicopter and filters in the cameras. The images thus obtained are roughly binarized and the remaining connected pixels correspond to the leds of the helicopter. The images in figure 6 show the binarization process.

The only problem left is the identification of each led, *i.e.* the correct labeling of leds 1, 2 and 3 frame after frame. In order to be able to make an accurate identification, the helicopter needs to be initially placed so that led 2 is under led 1 (which is a natural starting position).

Fig. 6. The left image is the scene the left camera sees, it was taken without infrared filter. The right image is the same one taken with the infrared filter mounted on the camera and binarized. The three white points correspond to the helicopter leds.

After that, images are acquired continuously and three different cases are considered. The first case is when 3 leds are detected. In this case each led is labeled according to the shortest distance from the labeling in the previous image (see figure 7).

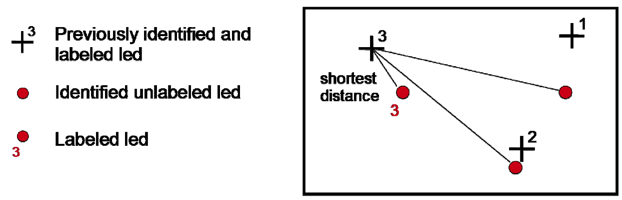

Fig. 7. Labeling process when three leds are detected. The labeling obtained previously is taken into account. Red dot 3 is labeled according to the distance to black crosses. The closest cross is number 3.

The second case is when 2 leds are detected and one is missing due to occlusions or alignments. In this case, a first order approximation of the motion of leds is considered and they are labeled according to the shortest distance from their estimated position (see figure 8).

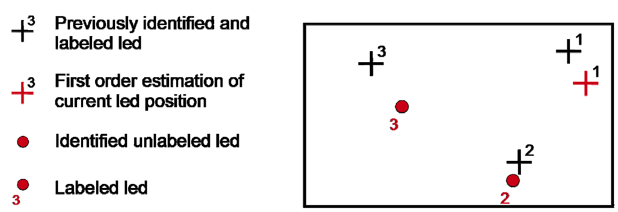

Fig. 8. Labeling process when two leds are detected. The labeling obtained previously is taken into account. Red dots 2 and 3 are labeled according to the distance to black crosses. The position of led 1 is estimated.

The third case is when one led or no leds are detected at all. This case is rare in normal operation. It usually happens only when the helicopter gets completely outside of the control space. When this situation occurs, the identification process is restarted and the program waits until three leds are detected, then led 2 is supposed to be situated underneath led 1and the first labeling is thus achieved.

B. Control loop application

The control loop can be any application capable of reading TCP sockets and reading/writing to Advantage PCI 1711U/1720U cards. The present control loop is implemented in Matlab. It uses `pnet` sockets, matlab timers and DAQ analog input/output objects.

This application is implemented to be generic enough to allow manual control, identification experiments and control experiments. The entry point is a Matlab script which initializes everything and starts a timer object. This timer executes every 30th of a second. It takes care of reading and writing all the data and calling the user function, the controller strictly speaking. Figure 9 depicts the structure and main procedures of this application.

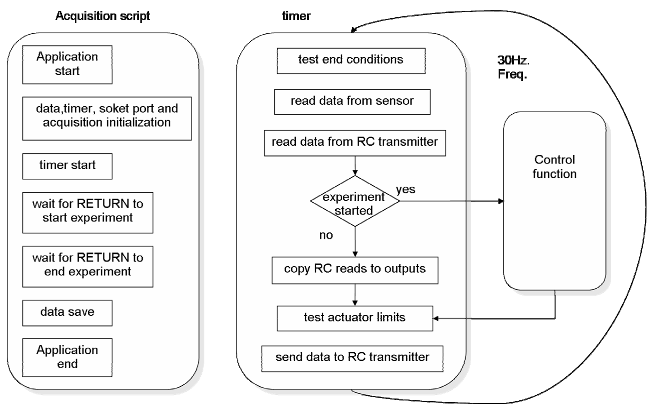

Fig. 9. Control loop application structure. The real processing is carried out in the timer function. The user control function allows interacting with the helicopter in a very flexible way.

The data read by the timer are the instant values collected from the RC transmitter and the values obtained from the sensor. The first group of data are throttle, antitorque, longitudinal cyclic and lateral cyclic. The second group contains the tick number, the reliability number, the x, y and z positions, the *pitch*, *yaw*, and *roll* angles and the time stamp.

The tick number is used to check data packet losses. Those values should be consecutive. The reliability number is used to assess the validity of measurements. It is 1 when the sensor was able to detect all three leds. It is 0 when two or more leds could not be detected and 1 / (frames + 1) when one led was not detected, indicating the number of frames this condition holds. The time stamp is important for resampling tasks or time related calculations because the operating system is not reliable regarding the execution time of processes.

All those data is passed to the control function, which has to calculate the actual throttle, antitorque, longitudinal cyclic and lateral cyclic commands.

IV. EXPERIMENTAL RESULTS

This section demonstrates the use of the laboratory platform for identification purposes. A linear model is sought to relate the yaw of the helicopter with the input commands, particularly antitorque.

Three types of input signals are investigated: short pulses, a pseudorandom binary signal and a sum of sine waves. The pulses are used to detect delays and main traits of the response. The other signals, see figure 10, are used for identification and validation. Those signals were generated using `idinput` Matlab command, trying to maximize the power spectrum in the band between 0 Hz and 12 Hz.

The data collecting procedure is quite straightforward. First the sensor application must be started. When the helicopter is correctly identified by the sensor the user confirms the online processing of images. After that, Matlab control loop application is started. In this case, the user function's role is to add to the pilot's command the desired signal (short pulses, the pseudorandom signal or the sum of sines). The control loop application allows the pilot to start flying the helicopter and when the user feels confident with the flight then he (or she) confirms the specific processing of the user function. The experiment finishes after a time set in advance or when the user decides to abort it, pressing a key in the Matlab console.

The main result of antitorque pulse insertion experiment is shown in figure 10. Pulses are 1/3 of a second long and their amplitudes vary from -1.25 to 1.25. The figure shows how the input signal contains both the pulses and the pilot's commands, which were kept as minimum as possible.

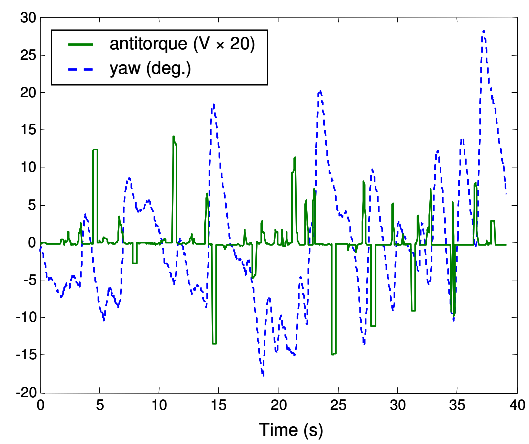

Fig. 10. Main result of the short pulses experiment. The solid (green) line is the input (antitorque) multiplied by 20 and subtracted its mean (-51.5º). The dashed (blue) line is the output (yaw) subtracted its mean.

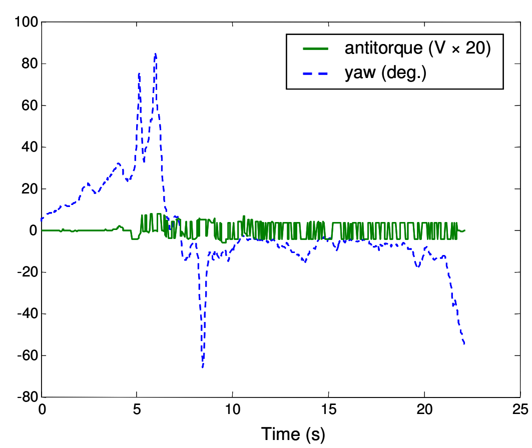

Fig. 11. Main result of the pseudorandom binary signal experiment. The solid (green) line is the input (antitorque) multiplied by 20 and subtracted its mean (2.55º). The dashed (blue) line is the output (yaw) subtracted its mean.

The results of the antitorque pseudorandom binary signal experiment are shown in figure 11. Figures 12 and 13 show the pitch and roll inputs and outputs and the trajectory (x, y, z variables) recorded during the experiment. All those signals can be used afterwards for identification purposes.

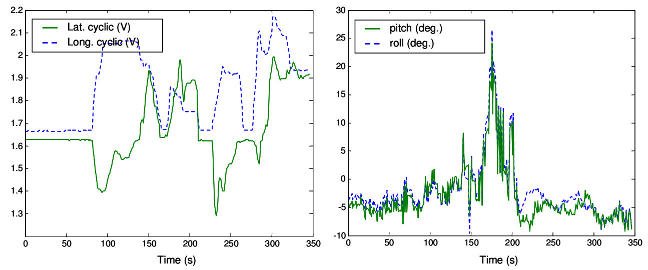

Fig. 12. Other signals of the pseudorandom binary signal experiment. Left shows the lateral cyclic input (solid green) and longitudinal cyclic input (dashed blue). Right shows the pitch (solid green) and roll (dashed blue) signals.

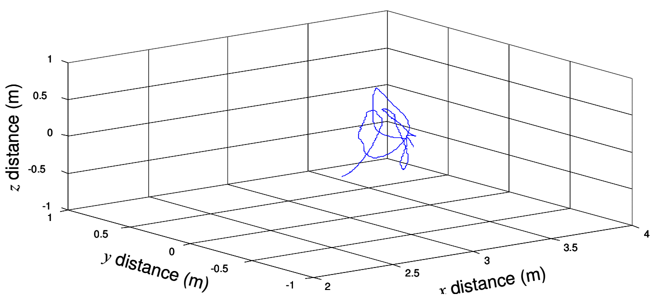

Fig. 13. Trajectory followed by the helicopter during the pseuderandom binary experiment. This is a composition of x,y,z measurements.

V. Conclusion

This paper describes the details and main components of a UAV control laboratory. A toy helicopter is modified with the addition of infrared leds to localize and identify it easily. Two CCD cameras obtain images of the laboratory control space. Those images are processed by a computer vision algorithm that calculates the position and attitude of the helicopter. A Matlab script is able to retrieve those measures and operate with them so as to obtain the control actions and send them via the helicopter's remote control transmitter.

Acknowledgment

This project was carried out fundamentally by the author with the help of students (Daniele Laccetti, Gabriel Portell, Alex Vargas, Sebastià Roca, David Heredia, Rubèn Acedo, Albert Jordà) that made their MSc or BSc final projects under his tuition. Josep Cugueró also collaborated tutoring some projects.

References

[1] D.M. Sobers, Jr., G. Chowdhary, E.N. Johnson, "Indoor Navigation for Unmanned Aerial Vehicles", *AIAA Guidance, Navigation, and Control Conference*, Chicago, Illinois, 2009.

[2] D. Schafroth, C.Bermes, S.Bouabdallah, R.Siegwart, "Modeling, system identification and robust control of a coaxial microhelicopter", *Control Engineering Practice* 18, 2010, 700–711

[3] Wei Wang; Gang Song; Nonami, K.; Hirata, M.; Miyazawa, O. , "Autonomous Control for Micro-Flying Robot and Small Wireless Helicopter X.R.B," *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on* , pp.2906-2911, 2006.

[4] C. Tisse, H.F. Durrant-Whyte. "A micro aerial vehicle motion capture system". *Proceedings of the International Conference on Sensing Technologies* (ICST'05), 2005.

[5] D. Neamtu, R. Deac, R. De Keyser, C. Ionescu, I. Nascu, "Identification and control of a miniature rotorcraft Unmanned Aerial Vehicle (UAV)" *Automation Quality and Testing Robotics (AQTR), 2010 IEEE International Conference on*, pp.1-6, 2010.

[6] L.C. Mak, M. Whitty, T. Furukawa, "A localisation system for an indoor rotary-wing MAV using blade mounted LEDs", Sensor Review, Vol. 28 Iss: 2, pp.125 – 131, 2008.

[7] Introductory techniques for 3D computer vision. By Emanuele Trucco, Alessandro Verri,