International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 6 (2015) pp. 15377-15387

http://www.ripublication.com

Autonomous Indoor Quadrotor

Shreyaank Raje Urs R[1] and Oliver S Nesaraj[2]

[1] PG Student, Robotics, Department of Mechanical engineering,
SRM University, Chennai
[2]Assistant Professor (Sr.G), Department of Mechanical Engineering,
SRM University, Chennai

Abstract

Recently, there has been increased interest in the development of low cost autonomous flying vehicles. However, as most of the proposed approaches are suitable for outdoors, only a few techniques have been designed for indoor environments, where the systems cannot rely on the Global Positioning System (GPS) and therefore, they haveto use their exteroceptive sensors for navigation. In this paper, we present a general navigation system that enables a small-sized quadrotor toperforma complete navigation that approaches different aspects of localization, mapping, path-planning, height estimation, control and obstacle avoidance aided by its on-board sensors. Algorithms are described and presented with a broad set of experiments, which illustrate that they enable a quadrotor robot to reliably and autonomously navigate in indoor environments. This quadrotor is mainly applicable in GPS denied environments i.e. indoor environments.

Keywords: Navigation, quadrotor, simultaneous localization and mapping (SLAM), unmanned aerial vehicle (UAV).

1. Introduction

In recent years, the robotics community has shown an increasing interest in autonomous aerial vehicles, especially quadrotors. Low-cost and small-size flying platforms are becoming widely available, and some of these platforms are able to lift relatively high payloads and provide an increasingly broad set of basic functionalities and applications. This directly raises the question of how to equip them with autonomous navigation abilities. However, as most of the proposed approaches for autonomous flying, focus on systems that are for outdoor environments, vehicles that can autonomously operate in indoor environments are envisioned to be useful for a variety of applications that include surveillance and search and rescue. Mobility of the flying vehicles is much higher compared to ground vehicles. As for ground vehicles, the main task for an autonomous flying robot consists in reaching a desired location in an unsupervised manner, that is, without human interaction. In the literature, this task is known as navigation or guidance. To address the general task of navigation one is required to tackle a set of problems that range from state estimation to trajectory planning. Several effective systems for indoor and outdoor navigation of ground vehicles are currently available.

In this paper we present the technology that enables a fully functional autonomous indoor quadrotor unmanned aerial vehicle to hover in its own right in accordance to the operating environment. This includes responding to unexpected conditions such as wind buffets, obstacles and maintaining stability. The ability to fly and respond to adverse conditions implies that the vehicles must have the full range of motion on all three axes. The vehicle must also respond to input from a pilot from the controller or base station. The UAV must be capable of landing autonomously. However, as a flying vehicle moves in 3-D indoors, there is usually enough structure to describe the environment with 2-D representations. Instead of using a full 3-D representation we rely on a 2-D one for the walls augmented with the elevation of the floor. The advantage of this choice compared with the full 3-D representation is that we can operate in a large class of indoor environments by using efficient variants of 2-D algorithms that work on dense grid maps instead of space and time consuming 3-D methods. Having these functionalities adapted for the 3-D case would be either too slow or not accurate enough given the limited time constraints to make the system stable.

2. System Design

Hardware requirements:

- Microcontroller
- Ultrasonic sensor
- Zigbee
- Rs232
- Rf transmitter and receiver

Software requirements:

- Flash programmer
- Embedded c

The construction of the frame as shown in figure 1 was done after making an FDM prototype, it was determined that the rapid prototype parts were suitable for use on the prototype helicopter. A strength test showed that the frame could support 1.5kg at the centre of the hub while being supported from each motor mount. Also, the entire frame including spars and motor mounts had a mass of 39g, compared to 39.9g for just the aluminium hub on the previous version of the quadrotor. The FDM parts did have drawbacks however. Because the resolution of the machine is about .02”, there were issues getting parts to fit.

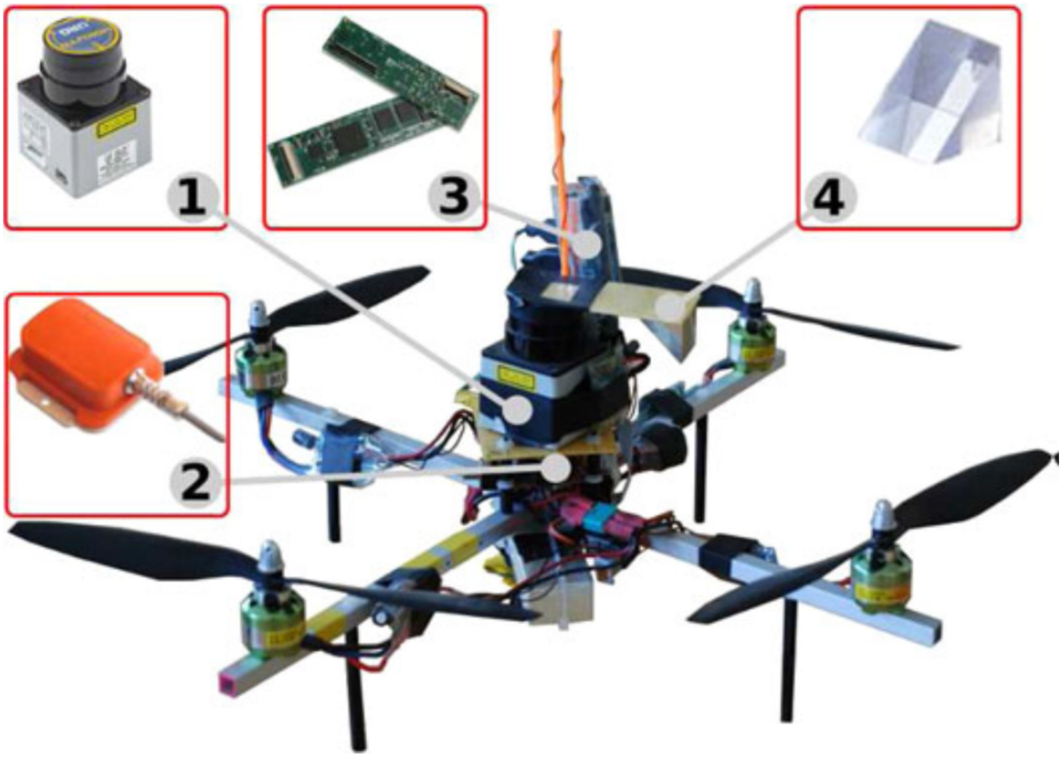


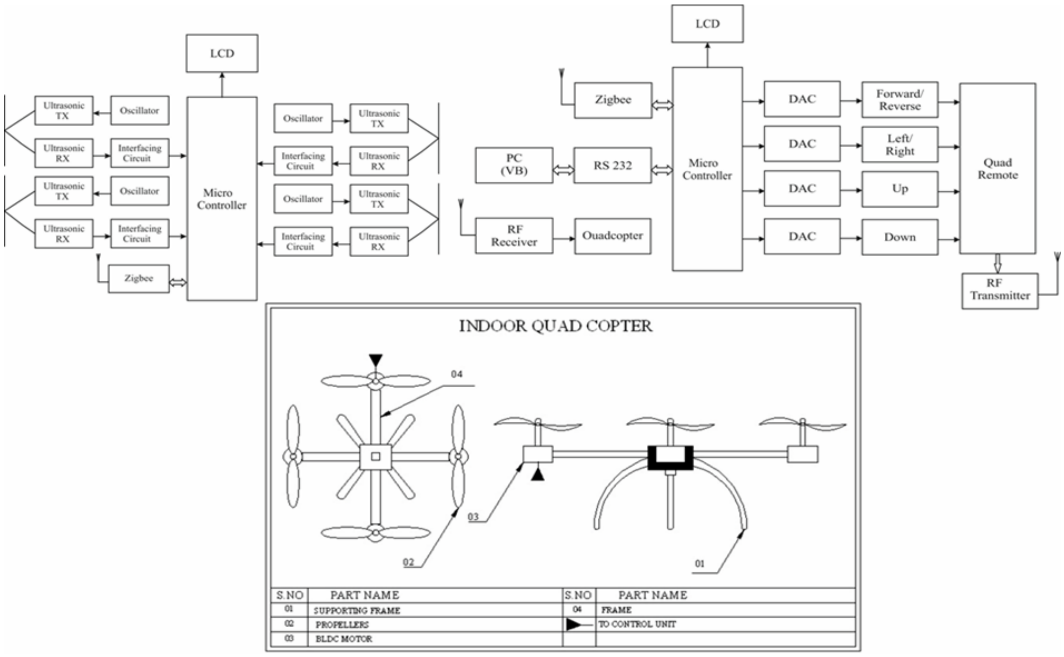
Fig. 1 Overview of the Quadrotor

2.1 Architecture of the Indoor quadrotor

The complete process has been split into different modules starting from design of the UAV, selection of materials and the fabrication of the frame. The system for the overall control of the motors and for the stability of the aircraft in-flight is viewed. Furthermore the systems and algorithm are designed for autonomous flight and for implementing SLAM principles.

Figure 2 describes the architecture of the proposed system for the indoor quadrotor, we can have the capability of automatically controlling the remote system with the aid of a personal computer or laptop. Object detection is also possible by using four ultrasonic sensors on the quadrotor. Sensor values are sent to ZIGBEE receiver, and it receives the value to display in the form of an image. This image will have the indoor map of the surrounding and thus our objectives of Simultaneous Localization and Mapping can be met. The data communication is done through 2.4GHz wifi link to ground station using a wireless NIC (Network Interface Card) onboard operating on 802.11n mode. The ground station node communicates among them through a separate adhoc wifi link. For safety pilot/ kill switch a separate 2.4 GHz RC link is used with an onboard receiver interfaced with flight controller. The system uses a four cell Lithium polymer Ion battery. It provides power to both motors and to the onboard processor. The battery has power rating of 2200mAh with a full charge voltage of

16.2V and nominal voltage of 14.8V. The flight controller has an on board battery eliminator circuit (BEC). The BEC takes care of controlled descent in the event of low battery. A ground station based on the ROS (Robot operating System) environment will continuously monitor the vehicle and display health and status information during the flight. The vehicle will continuously send information about its pose, velocity, health status, environment (real time images), etc. to the ground station (over a 2.4GHz wireless LAN data link) preconfigured with ROS making it easier to interpret the data received and present it in an informative way. When a certain force is applied any of the natural modes can be excited which in turn may lead to catastrophic failure of the system. This in turn leads to the importance of studying the resonance frequency of the quadrotor. The resonance frequencies of a system are the frequencies at which the system will be "excited"; therefore, it is imperative to determine the correct resonant frequencies of the quadrotor in order to ensure that the natural modes of the system will not be disturbed. To do these we have determined the natural frequencies of the propellers, which we then relate to the resonance frequencies of the quadrotor to ensure the stability of the system.



S.NO	PART NAME	S.NO	PART NAME
01	SUPPORTING FRAME	04	FRAME
02	PROPELLERS	▶	TO CONTROL UNIT
03	BLDC MOTOR		

Fig.2: Architecture of the Quadrotor

2.2 Mechanical Design and Material Selection

The mechanical design of the quadrotor is relatively simple and a solved problem. The primary consideration for this project is cost, accessibility of material and the machining equipment as this project is mainly exploring the capabilities of autonomous flight and not on the design.

When designing an autonomous quadrotor, there are several material options which must be considered. Any design must consider different materials based on durability, machinability, and price. When dealing with a machine capable of flight, then one must consider weight a major factor. The materials in consideration for our design include aluminium, plastic, and carbon fibre, their properties are classified in the Table 1. Aluminium, historically, has been the material of choice for RC helicopters. Aluminium is light and strong, dissipates heat well, and is relatively inexpensive in comparison to some of the other possibilities. The negative for aluminium is that it tends to be too heavy for small aircraft models. Also, aluminium can develop cracks over time from vibrations. Plastic absorbs vibration much better the previously mentioned aluminium. Also, it is fairly durable and will return to its original shape if bent. Plastic is also very inexpensive, light, and can be machined very easily. Various types of plastics were explored, including Nylon, polypropylene, Delrin, Ultem, polyethylene, and ABS. The loads placed on these parts are well within the yield strength of the materials, so ultimately the decision came down to price, and raw Nylon was the cheapest plastic available.

Another plastic option presented itself, and that was rapid prototyping of ABS using fused-deposition modelling (FDM). FDM uses a 3-D printer to lay down thin lines of plastic. The printer builds pieces in layers, and this must be taken into consideration in the design. Holes which are perpendicular to the horizontal plane will be smooth, but holes whose centre axis is parallel to horizontal will be built in “steps”, and thus aren’t suitable for applications where smoothness or threads are required. The resolution of the machine is the thickness of lines, which is about .2cm. This means that machined pieces will not be close to the accuracy of CNC machining of metal. Finally, certain thin features aren’t possible because two lines of fused plastic aren’t as strong as a similar size of machined ABS. The main selling point for FDM is its extreme simplicity to use (export directly from Solid Works into Catalyst, which then determines proper orientation and prints the part), its cost (60-80 Rs. per cubic inch), and speed of manufacture (about 2 hours for all of this quadrotor’s parts). Carbon Fibre is currently the best material available for RC helicopters. It is stronger and lighter than aluminium and absorbs vibration better than plastic. It can be moulded to be super stiff in one direction and flexible in the other. But, it is also much more expansive than other materials. Also, it is difficult to machine so it would require an outside source to manufacture the required parts.

Table 1- Material Comparison

Material	Modulus of Elasticity (GPa)	Tensile Strength (MPa)	Density (g/cm^3)
Nylon 6.6	2.61	82.8	1.14
ABS	.001	29.0	1.02
Ultem	3.45	114	1.28
Delrin	2.55	52.4	1.42
Carbon Fibre	220	760	1.7
Stainless Steel 404	200	1790	7.80
Aluminium 7075	71	572	2.80

2.3 Indoor navigation of an autonomous flying quadrotor

To autonomously reach a desired location, a mobile robot has to be able to determine a collision-free path connecting the starting and the goal destinations. This task is known as path planning and requires a map of the environment to be created and aware of. This map is acquired by the quad by processing the sensor measurements that are obtained during an exploration mission. This task of generation of the map is known as simultaneous localization and mapping (SLAM). For most of the applications it is sufficient to perform SLAM off-line on a recorded sequence of measurements. To follow the path with a sufficient accuracy, the robot needs to be aware of its position in the environment at any point in time. This task is known as localization. A further fundamental component of a navigation system is the control module that aims to move the vehicle along the trajectory, given the pose estimated by the localization. Because of the increased risk of damaging the flying platform during testing, the user should have the possibility of taking over the control of the platform at any point in time. Finally, the more complex dynamics of a flying platform poses substantially higher requirements on the accuracy of the state estimation process than for typical ground-based vehicles. Although in outdoors scenarios, positioning errors upto1m might be acceptable, they are not indoors, as the free-space around the robot is substantially more confined.

2.4 Navigation system

Our navigation system is based on a modular architecture in which different modules communicate via the network by using a publish–subscribe mechanism. In our current system, all device drivers are executed on-board, while the more computationally over wireless with the platform. Since roll (φ) and pitch (θ) measured by the IMU are in general accurate up to 1°, we can directly use this information within our navigation system. This allows us to reduce the localization problem from 6 degrees of freedom (DOF) namely (x,y,z,φ,θ,ψ) to 4 DOF, consisting of the 3-D position(x,y,z) and the yaw angle ψ. The only sensor that is used to estimate these 4DOF and detecting obstacles is the laser range scanner. Based on known initial calibration parameters and on the current attitude (φ,θ) that is estimated by the IMU, we project the endpoints of the laser into the global coordinate frame. Given the projected laser beams, we estimate the (x,y,z,ψ) of the vehicle in a 2-D map containing multiple levels per cell. To compensate for the lack of odometry measurements we estimate the incremental movements in (x,y,ψ) by 2-D laser scan matching. Finally, we control the altitude of the vehicle and simultaneously estimate the elevation of the underlying surface by fusing the IMU accelerometers and the distance from the ground measured by the laser. Accordingly, we track and map multiple levels within an environment, which enables our robot to correctly maintain its height, even when it flies over obstacles such as tables or chairs.

2.5. Localization and SLAM

If a map of the environment is known a prior, pure localization (in contrast with SLAM) is sufficient for the estimation of the remaining 4DOF of the quadrotor. The 2D position of the quad is estimated with the aid of Monte carlo localisation.. The idea is to use a particle filter to track the position of the robot. Here, we sample the next generation of particles according to the given relative movement estimated by the scan matcher and evaluate the current particle by using likelihood fields [12]. Our system can acquire models of unknown environments during autonomous or manual flights by simultaneously localizing and mapping the environment. The goal of a SLAM algorithm is to estimate both the vehicle position and the map of the environment by processing a sequence of measurements acquired, while moving in the environment. Even when a map is known a priori, a local map is needed until the robot is localized if the robot is running autonomously. In our system, we use a popular graph-based SLAM algorithm. The idea of these types of algorithms is to construct a graph from the measurements of the vehicle. Each node in the graph represents a position of the vehicle in the environment and a measurement taken at that position. Measurements are connected by pairwise constraints encoding the spatial relations between nearby robot poses. These relations are determined by matching pairs of measurements acquired at nearby locations. Whenever the robot re-enters a known region after traveling for long time in an unknown area, the errors accumulated along the trajectory become evident. These errors are modelled by constraints connecting parts of the environment that have been observed during distant time intervals and are known in the SLAM community as loop closures. To recover a consistent map we use a stochastic gradient descent optimization algorithm that finds the position of the nodes that maximizes the likelihood of the edges. The optimization approach is discussed in detail in [6], and an open source version is available on Open SLAM [4]. Again, we restrict our estimation problem to 4DOF, since the attitude provided by the IMU is sufficiently accurate for ourmappingpurposes.Furthermore,weassumethatthevehicle flies over a piecewise constant surface and that the indoor environment is characterized by vertical structures, such as walls, doors, and so on. Although trash bins, office tools on a table or the table itself are violating this assumption by the usage of a 2-Dmapisstillsufficientforaccuratemappingandlocalization. This arises from the fact that clutter in general is only visible in a small portion of the current measurement, whereas mapping, forexample,thedeskimproveslocalizationsincethereisaclear difference in x-y between a desk and a nearby wall. Thus, we restrictourapproach toestimatea2-Dmapanda2-Drobottrajectory spanning over 3DOF, (x,y,ψ), i.e., we map all objects if they had an infinite extend. The estimate of the trajectory is the projection of the 6DOF robot motion on the ground plane, along the z-axis. We estimate the altitude of the platform once the 2-D position and the attitude are known, which is based on the procedure described in the next section.

2.6 Altitude Estimation

Estimating the altitude of the vehicle in an indoor environment means determining the global height with respect to a fixed reference frame. Since the vehicle can move over an on flat ground, we cannot directlyusethebeamshdeflectedbythemirror. Our approach therefore concurrently estimates the altitude ofthevehicleandtheelevationofthegroundundertherobot.In our estimation process, we assume that the (x,y,ψ) position of the robot in the environment is known from the SLAM module described earlier. We furthermore assume that the elevation of thesurfaceundertherobotisapiecewiseconstant.Wecalleach of these connected surface regions having constant altitude a “level.” The extent of each level is represented as a set of cells in a 2-D grid sharing the same altitude. Sinceoursystemlacksglobalaltitudesensorssuchasbarometers or GPS to determine the altitude of the vehicle, we track the altitude of the vehicle over the ground and map different elevations by a two-staged system of Kalman filters. Algorithm 1 describes our approach in an abstract manner. In the first stage, a Kalman filter is used to track the altitude z and the vertical velocity vz of the vehicle by the combination of inertial measurements, altitude measurements, and already mapped levels under the robot. In the second stage, a set of Kalman filters is used to estimate the elevation of the levels

2.7 Path Planning and Obstacle Avoidance

In this section, we present an experiment that demonstrates our algorithms for path planning and dynamic obstacle avoidance. The quadrotor was given a goal point approximately 5 m in front of it. A person was standing on the left enter the corridor when the quadrotor moved to its desired goal. The second image shows the situation when the person is completely blocking the robot’s path. In this case, the quadrotor hovered around the last valid way point since there was no longer a valid plan to the goal. When the person moved to the left again, the quadrotor was able to follow a detour. Note, that the snapshots show the endpoints of the laser only. Although it looks like the quadrotor might have the space to fly around the person in the second image, there is no valid plan because of the safety margins around the walls.

3. Conclusion and Future work

In this paper, a navigation system for autonomous indoor flying utilizing an open-hardware quadrotor platform has been presented. A complete navigation solution that approaches different aspects of localization, mapping, path-planning, height estimation, and control has been described. Since we do not rely on special characteristics of the flying platform like the dynamics model, we believe that our system can easily be adapted to different flying vehicles. All modules in our system run online. However, because of the relatively highcomputationalcost ofsomealgorithmsonlyapartofthesoftwarerunson-boardon the ARM processor, whereas the other part runs off-board on a laptop computer. Some preliminary tests make us confident thatthewhole systemcan runon-board bytheusageofthenext generation of embedded computers that are based on the Intel Atom processor. We provided a wide range of experiments and some videos that highlight the effectiveness of our system. In future work, we plan to add a time of flight camera into our system. It has been believed that this technology can be effectively integrated and will allow us to relax the assumption that the vehicle moves over a piecewise planar surface.

References

1 Annaswamy, A., Dydek, Z. and Lavretsky, E. (2013) ‘Adaptive Control Of Quadrotor UAV’s: A Design Trade Study With Flight Evaluation’, IEEE Transactions On Control Systems Technology, Vol. 21, No.4, pp.1400-1406.

2. Bi, Y. and Duana, H. (2012) ‘Implementation of Autonomous Visual Tracking and Landing for a low-cost Quadrotor’, Optik, Vol.124, pp.3296–3300.

3. Go, T. and Zhao, W. (2014) ‘Quadrotor Formation Flight Control Combining MPC and Robust Feedback Linearization’, Journal of the Franklin Institute, Vol.351, pp.1335–1355.

4. Beji, L., Maaref, H. and Zemalache, K. (2007) ‘Self-tunable Fuzzy Inference System: A Comparative Study for a Drone’, Analysis and Design of Intelligent Systems using Soft Computing Techniques Advances in Soft Computing, Vol.41, pp.780-789.

5. Chee, K.Y. and Zhong, Z.W. (2013) ‘Control, Navigation and Collision Avoidance for an Unmanned Aerial Vehicle’, Sensors and Actuators, Vol.190, pp.66– 76.

6. Blažič, S., Bošnak, M. and Matko, D. (2012) ‘Quadrotorrocopter Control Using an On-Board Video System with Off-Board Processing’, Robotics and Autonomous Systems, Vol.60, pp.657–667.

7. Ortega, M., Raffo, G. and Rubio, F. (2011) ‘Path Tracking of a UAV via an Underactuated H∞ Control Strategy’, European Journal of Control, Vol.2, pp.194–213.

8. Pei, H. and Zhao, Y. (2012) ‘An Improved Vision-Based Algorithm for Unmanned Aerial Vehicles Autonomous Landing’, Physics Procedia, Vol.33, pp.935 – 941.

9. Efe, M., Eresen, A. and Imamoglu, N. (2012) ‘Autonomous Quadrotor Flight with Vision-Based Obstacle Avoidance in Virtual Environment’, Expert Systems with Applications, Vol.39, pp.894–905.

10. Alexis, K., Nikolakopoulos, G. and Tzes, A. (2011) ‘ Switching Model Predictive Attitude Control For A Quadrotor Helicopter Subject To Atmospheric Disturbances’, Control Engineering Practice, Vol.19, pp.1195–1207.

11. Lamberti, F., Manuri, F., Paravati, G. and Sanna, A. (2013) ‘A Kinect-Based Natural Interface For Quadrotor Control’, Entertainment Computing, Vol.4, pp. 179–186.

12. Akbar, F., Azman, A. and Mustafah, Y. (2012) ‘Indoor UAV Positioning Using Stereo Vision Sensor’, Procedia Engineering, Vol.41, pp.575 – 579.

13. Lange, S., Protzel, P. and Sunderhauf, N. (2009) ‘A Vision Based Onboard Approach For Landing And Position Control Of An Autonomous Multirotor UAV In GPS-Denied Environments’, International Conference on Advanced Robotics, pp.22-26.

14. Fantoni, I., Kendoula, F. and Onamib, K. (2009) ‘Optic Flow-Based Vision System For Autonomous 3D Localization And Control Of Small Aerial Vehicles’, Robotics and Autonomous Systems, Vol.57, pp.591-602.

15. Abdul, Q. and Moona, M. (2012) ‘Autonomous Navigated Global Positioning System Based Surveillance Unmanned Aerial Vehicle’, International Journal of Information Technology and Electrical Engineering, Vol.1, Iss.1, pp.25-29.

16. Altug, E., Mahony, R. and Ostrowski, J.P. (2002) ‘Control of a Quadrotor Helicopter Using Visual Feedback’, IEEE International Conference on Robotics and Automation, Vol.1, pp.72–77.

17. Abbass, V. and Amir, M.Y. (2008) ‘Modelling Of Quadrotor Dynamics’. International Conference on Smart Manufacturing Application, pp.100 –105

18. Altug, E. and Erginer, B. (2007) ‘Modelling and PD Control of a Quadrotor VTOL Vehicle’, IEEE Intelligent Vehicles Symposium, pp.894 –899.

19. Alpen, M., Frick, K. and Horn, J. (2009) ‘Nonlinear Modelling and Position Control of an Industrial Quadrotor with On-Board Attitude Control’, IEEE International Conference on Control and Automation, pp.2329 –2334