IJCTA, 10(9), 2017, pp. 803-819 ISSN: 0974-5572
© International Science Press

Model Based Design of Vision Guided Vertical Landing System for Quadrotor

R. Senthilnathan*, S. Nikhil*, R. Sasikala and Anirudh Maruvada****

ABSTRACT

Aerial vehicles are seeing an increase in role in the field of mission planning which involves unmanned regions which last for short duration. This paper presents the research work where a computer vision system is aiding landing of a quadrotor by identifying the markers in the region of landing for a safe landing. The quadrotor modelling realizes the attitude and motion control of the system which issignaled by a secondary controller. The vision system is developed to identify shape based feature extraction of markers in the landing region under day light condition. The image analysis is performed continuously to extract the shape feature and to identify the coordinates on the marker. With the help of vision feed of the marker the quadrotor aims for a vertical landing above the region of the marker. This paper presents the control algorithm for descent of the quadrotor guided by computer vision system to achieve a safe landing.

Keywords: Quadrotor, Vision Guidance, Mathematical Modelling, Vision in the Loop Control, Simulation, Vertical Landing

1. INTRODUCTION

UAV is a complex system which integrates different hardware components, such as camera, Global Positioning System (GPS), Inertial Management Unit (IMU), controller, and different software components to make autonomous flights [1]. The idea of using four rotors was realized as a full-scale helicopter in the early 1920s [1] and it has been a topic of research for military applications since 1950s [2]. Unmanned Aerial Vehicles (UAVs) have the potential to revolutionize current working practices for many military and civilian applications such as assisting in search-and-rescue missions, environmental monitoring, traffic surveillance and other transportation operations [2] [3]. UAVs are classified as either rotary-wing or fixed-wing [2]. Quadrotor is a type of rotary-winged craft that consists of four rotors and two pairs of counter rotating, fixed-pitch blades located at four corners of the body [1]. Quadrotors simplify the design and maintenance of the vehicle because of their simple mechanical structure and dynamics. Furthermore, the use of four rotors allows each individual rotor to have a smaller diameter. In this way, the damage that may be caused by the rotors in case of undesirable situations can be greatly reduced [1].

Recent researches have enhanced its visual perception, visual processing, redundant sytems for landing and manoeuvring in structured and unstructered environments and alternative methods for charging the quadrotor. Different algorithms have been proposed and developed to address current challenges [1][3][5][4][6]. A novel multiple view algorithm to improve the motion and structure estimation for vision-based landing of UAV is proposed in [1]. A quadrotor which uses on board processing for localization and mapping with stereo vision to manoeuvre it in In-door and Out-door GPS denied environment is presented in [4]. Although Interstitial/GPS approach was used for the quadcopter to perceive the environment, it suffers from two drawbacks: necessity to receive GPS signals and lack of precision in position [6].

* Department of Mechatronics Engineering, SRM University Kattankulathur, India, *Email: senthilnathan.r@ktr.srmuniv.ac.in*

** Department of Electronics and Instrumentation Engineering, SRM University, Kattankulathur, India.

Vision based aerial vehicles provide the users with complete information of the environment in which they are located from time to time with the ability to live stream the information of their surroundings, and in the recent past this has helped the user to manoeuvre the aerial vehicle as per their need, operated from a remote location. Vision sensor could also be used to estimate the altitude by itself or combined with other inertial measurements [7]. The pseudo roll and pitch can be decided from the onboard video or image streams [8]. Experiments on only vision based navigation and obstacle avoidance have been achieved on small rotary wing UAVs [9]. The vision system itself faces a lot of challenges which can be due to the camera, surrounding environment or to a certain extent the aerial vehicle itself due to its body dynamics.

With the increasing applications of aerial vehicles, the challenges faced by them also are increasing. The aerial range is one of the major concerns, which is mainly dependent on the onboard battery. Current battery technology only enables ûights of 8–15 min (depending on vehicle, payload, and wind conditions) for a typical quadrotor [10], which constrains the mission and experimental capabilities of these small-scale UAVs. Establishing ground landing stations has addressed the range issue which affects the time of flight but research is still going on to make charging possible in the landing station.

The landing is based on the concept of vertical take-off or landing on board from the landing station. The most important aspect is the interaction between the landing-pad and the drone for the automatic take-off and landing. A optical flow algorithm for hovering flight and vertical landing control of the quadrotorhas been developed at French Atomic Energy Commission [1]. In some cases light based markers are created on the landing pad to make it easier for the drone to recognize the landing station while the ultrasonic sensors aid them to detect obstacles prior to the initation of the landing sequence. The LED in the landing-field guides the drone's accurate landing. When the drone locates the landing-pad and approaches above it, both the drone and the pad proceed through the landing process. If any obstacle is recognized in the landing area, it then stops the progress and waits in a hovering state. On completion of landing as expected, the landing-pad will turn off the LED, recognizing the drone by the ultrasonic sensor [3].

2. MODELLING OF QUADROTOR

The modelling of quadrotor presented in the following sections is completely based on [11] and the various references presented in the book. For developing the SIMULINK model, the necessary mathematical equations are described below. The generic frame assignment illustrating the thrusts is shown in Fig. 1.

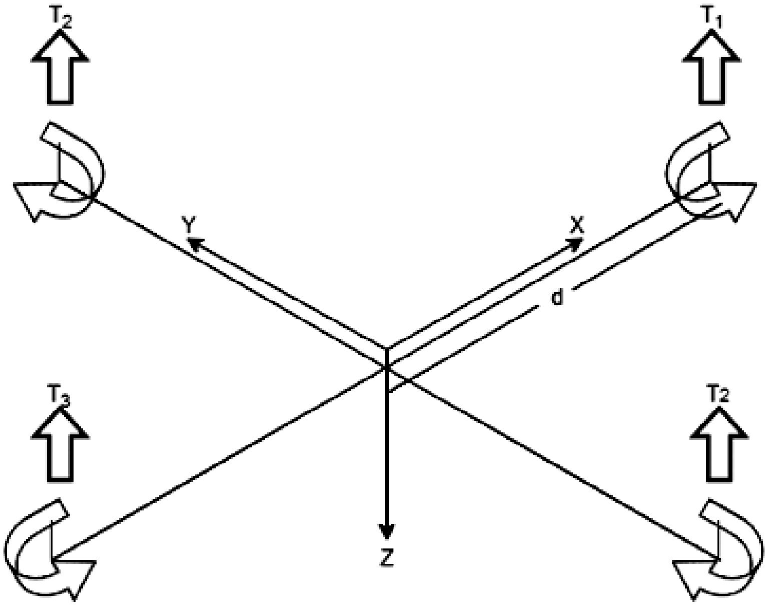

Figure 1: Quadrotor Frame Assignment

2.1. Forward Kinematics of Quadrotor

The position vectors, linear velocity vectors, angular position vectors and angular velocity vectors are defined as:

$$V_B = \begin{bmatrix} \upsilon_x, B \\ \upsilon_y, B \\ \upsilon_z, B \end{bmatrix}, \nu = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{1}$$

The transformation matrix for angular velocities from the inertial frame to the body frame is described below:

$$\nu = W_\eta \dot{\eta} \tag{2}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -S_\theta \\ 0 & C_\phi & C_\theta S_\phi \\ 0 & S_\theta & C_\theta C_\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{3}$$

Where: C_ϕ, C_θ, S_ϕ and S_θ are sine and cosine angles of angular position vectors.

2.2. Forces

The quadrotor is driven be four motors producing angular velocity which generates thrust. These thrusts are represented as follows.

$$T_i = b\omega_i \tag{4}$$

Where i = 1, 2, 3 and 4. $b > 0$ is the lift constant of the quadrotor. ω_i is the angular velocity of rotors.Therefore the total thrust generated is given by:

$$T = \sum_{i=1}^{i=4} T_i \tag{5}$$

$$T = db(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \tag{6}$$

Where T is total thrust generated by the motors. d is the distance between the motor and the centre of gravity of the quadrotor. Due to the rotation of the motors, it creates torque about x, y and z axis. The torque about vehicle's x axis is the rolling torque and is given by

$$\tau_\phi = db(\omega_4^2 - \omega_2^2) \tag{7}$$

Similarly, the pitching torque is given

$$\tau_\theta = db(\omega_1^2 - \omega_3^2) \tag{8}$$

The torque applied to each propeller is opposed by aerodynamic drag. This torque exerts a reaction torque on airframe which acts to rotate the airframe about the propeller shaft given by

$$\tau_\psi = k(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \tag{9}$$

Where k is drag constant and depends on similar factors as b. Therefore the total torque exerted by the motors is given by

$$\tau = \begin{bmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} db(\omega_1^2 - \omega_3^2) \\ db(\omega_2^2 - \omega_4^2) \\ k(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} \tag{10}$$

Where τ is the total torque exerted by the propellers along *x*, *y* and *z* axis.

2.3. Equations of Motion

From newton's second law of motion we can infer translational dynamics of quadrotor in world frame.

$$m\ddot{\xi} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RT_B \tag{11}$$

Where T_B is the total thrust vector represented by:

$$T_B = \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} \tag{12}$$

The rotational acceleration of the airframe is given by Euler's equation of motion:

$$I\dot{\nu} + \nu \times (I\nu) = \tau \tag{13}$$

Where I is 3 × 3 inertial matrix. Assuming the mass distribution to be symmetric:

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{14}$$

Where I_{xx}, I_{yy} and I_{zz} are moment of inertia about *x*, *y* and *z* axis. By solving the equation, we get

$$\dot{\nu} = \begin{pmatrix} \tau_\phi I_{xx}^{-1} \\ \tau_\theta I_{yy}^{-1} \\ \tau_\psi I_{zz}^{-1} \end{pmatrix} - \begin{pmatrix} \dfrac{\left(I_{yy} - I_{zz}\right)qr}{I_{XX}} \\ \dfrac{\left(I_{zz} - I_{xx}\right)pr}{I_{yy}} \\ \dfrac{\left(I_{xx} - I_{yy}\right)pq}{I_{zz}} \end{pmatrix} \tag{15}$$

By combining equations (6) and (10) we get

$$\begin{pmatrix} T \\ \tau \end{pmatrix} = \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} -b & -b & -b & -b \\ 0 & -db & 0 & db \\ db & 0 & -db & 0 \\ k & -k & k & -k \end{bmatrix} = A \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{16}$$

Where A is of full rank if b, k, $d > 0$ and be represented as

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = A^{-1} \begin{bmatrix} T \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} \tag{17}$$

3. MODELLING OF MOTION CONTROL OF QUADROTOR

To control the vehicle we will employ a nested control structure. The rotational dynamics has second order transfer function of

$$\frac{\eta(s)}{\tau(s)} = \frac{1}{Is^2 + Bs} \tag{18}$$

Where B is aerodynamic damping constant and is generally low. Therefore we use Proportional-Derivative controller for the system.

$$\tau_\theta = K_{\tau p}(\theta^* - \theta) - K_{\tau d}(\dot{\theta}^* - \dot{\theta}) \tag{19}$$

Considering a coordinate frame $\{V\}$ attached to the vehicle and with the same origin as $\{B\}$ but with its x and y-axes parallel to the ground. To move the vehicle in the V_x direction we pitch the nose down which generates a force.

$$f = R_\theta(\theta)\begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = \begin{bmatrix} T\sin\theta \\ 0 \\ T\cos\theta \end{bmatrix} \tag{20}$$

Therefore for small angles we can assume the component

$$f_x = T\sin\theta \approx T\theta \tag{21}$$

This component accelerates vehicle in V_x direction. It can be controlled using proportional control law

$$f_x^* = mK_f({}^V v_x^* - {}^V v_x) \tag{22}$$

Combining with equations (21) and (22) we get

$$\theta^* = \frac{m}{t} K_f({}^V v_x^* - {}^V v_x) \tag{23}$$

Similarly for roll angle we also get:

$$\phi^* = \frac{m}{t} K_f({}^V v_y^* - {}^V v_y) \tag{24}$$

Considering position of the vehicle in x-y plane is p, then desired velocity is given by:

$$v^* = K_p(p^* - p) \tag{25}$$

Based on error between desired and actual position, desired velocity in frame $\{V\}$ is a function of yaw angle given by

$$\begin{pmatrix} {}^V v_x \\ {}^V v_y \end{pmatrix} = {}^0R_V^T(\psi)\begin{pmatrix} v_x \\ v_y \end{pmatrix} \tag{26}$$

The Fig. 2 represents SIMULINK model of the closed loop control of quadrotor. All the mathematics described are built in the SIMULINK platform.

The input for the quadrotor block are speeds from four rotors that can be derived from equation (27). And the output block of the quadrotor returns state vector

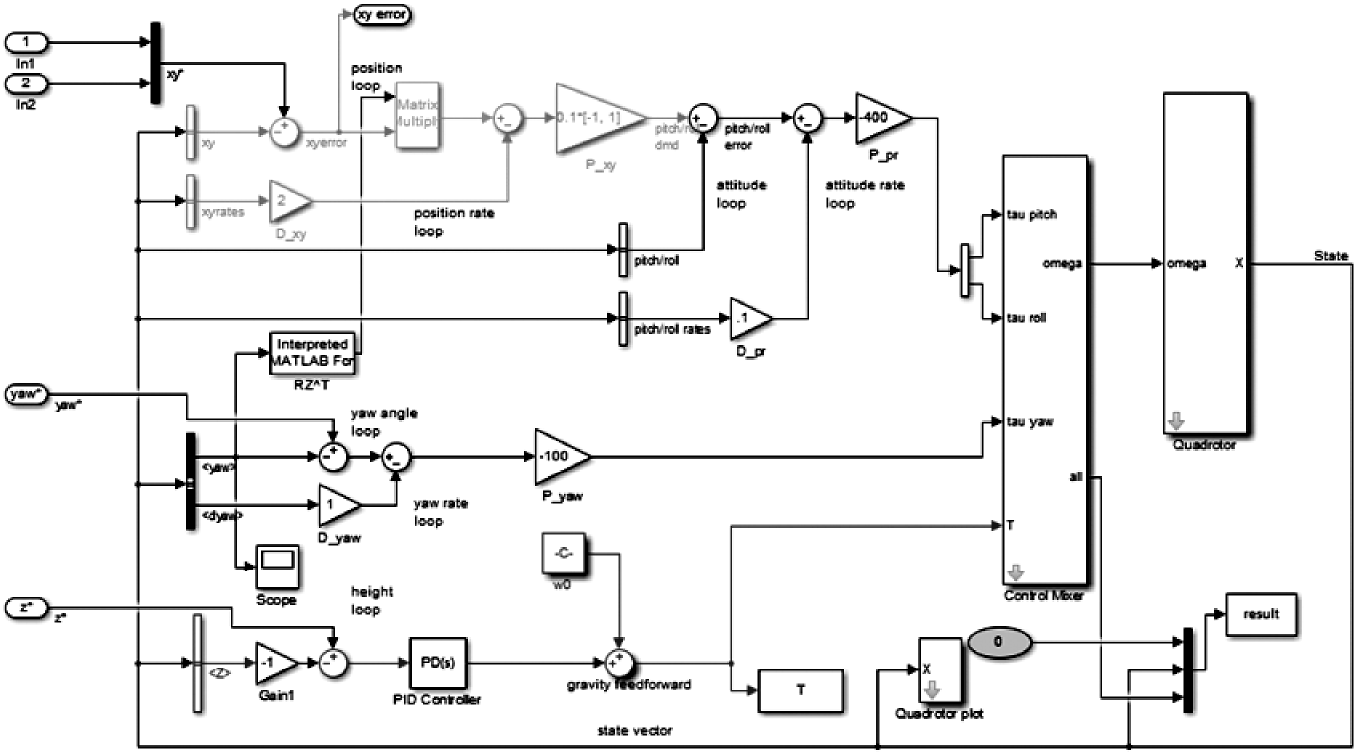

Figure 2: SIMULINK Model of Quadrotor

$$x = (x, y, z, \dot{x}, \dot{y}, \dot{z}, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}) \tag{28}$$

Vehicles position control loops. The innermost loop controls the pitch and yaw of vehicle and its inputs are actual and desired roll and pitch angles as well as pitch and roll rate to provide damping. The outer loop controls the position of the vehicle by requesting change roll and pitch for the vehicle. The equation for pitch and roll with combing the equation of velocity returns

$$\theta^* = K_1({}^V p_x^* - {}^V p_x - K_2 {}^V v_x) \tag{29}$$

And for the roll we get

$$\phi^* = K_3({}^V p_y^* - {}^V p_y - K_4 {}^V v_y) \tag{30}$$

The yaw and altitude is controlled by proportional derivative controller given by

$$\tau_z = K_p(\psi^* - \psi) + K_d(\dot{\psi}^* - \dot{\psi}) \tag{31}$$

$$T = K_p(z^* - z) + K_d(\dot{z}^* - \dot{z}) + \omega_0 \tag{32}$$

The additive term ω_0 is responsible for adding necessary rotor speed to generate a thrust equal to weight of vehicle and is given by

$$\omega_0 = \sqrt{\frac{mg}{4b}} \tag{33}$$

For the decent control, the above mentioned algorithm is enacted to generate control signals. But in-order to achieve descent without going out of scope of correction of x, y error, the signals are checked after reaching 1m to make the quadrotor hover at its position until the centre of the vision sensor lies inside the error tolerant region.

4. QUADROTOR SPECIFICATIONS

The quadrotor used for the experiment is a controlled in X configuration with high strength glass fiber material chassis. It is controlled through a primary flight controller which is controlled through a master controller passing it positional error values through camera sensor as a visual feedback. The specifications for the quadrotor are explained below.

4.1. Chassis Specification

Quadrotor used for this experiment is made up of Skydom X525 frame with 1045 propeller blades. The frame is maneuvered through four 980kV Brushless DC (BLDC) motors.The Fig. 3. Shows the quadrotor with all the specifications mentioned above.

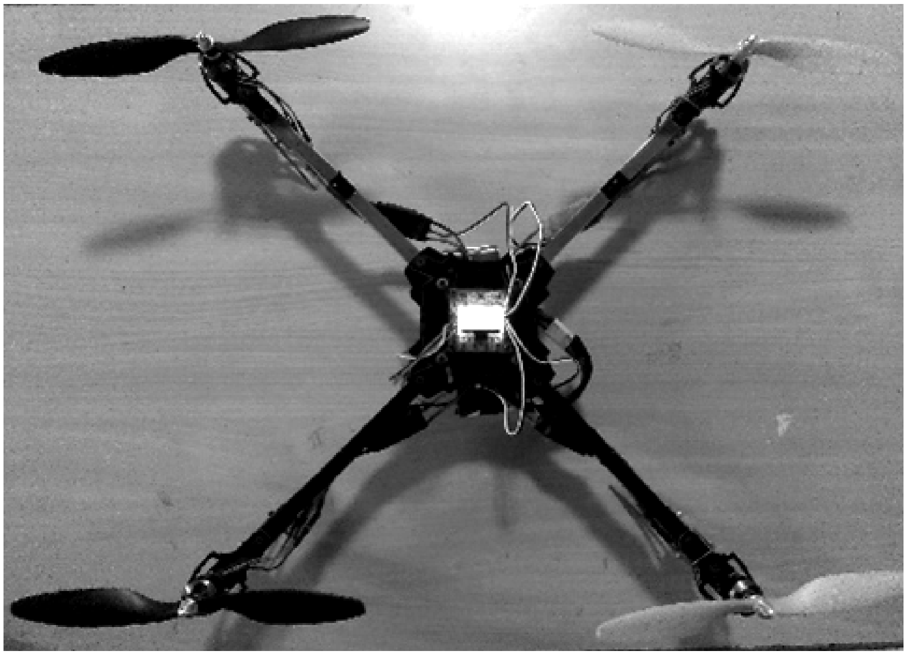

Figure 3: Photograph of the Quadrotor

4.1.1. Frame Specifications

The Skydom X525 is afour-axis rack cut from high-strength glass fiber material CNC, which has high accuracy.The configuration of the quadrotor is of × type, designed for mounting four motors on the frame.The frame has a base plate for the flight controller, power distribution board and the battery. The power distribution board and battery is mounted below the base plate, above the base plate area the flight controller is mounted.

4.1.2. Motor Specifications

Quadrotor used for this experiment uses SunnySky Angel 2212 980 KV BLDC motor for propelling the propeller blades.The details are listed in Table 1.

Table 1
Motor Specifications

Parameter	*Specification*
RPM per Volt	980 KV
Configuration	12N14P
Stator diameter	22 mm

(contd...Table 1)

Parameter	*Specification*
Stator length	12 mm
Shaft diameter	3 mm
Motor dimensions	27 × 30 mm
Weight	50 g
Number of cells	2 – 3 S
Maximum continuous Current	14.5 A
Maximum continuous Power	160 W
Maximum efficiency current	(4–9 A) > 80%
Internal resistance	120 mΩ

4.1.3. Propeller

The propeller blades user is made of strong plastic material. The specifications of the propeller blades are mentioned in the Table 2.

Table 2
Propeller Blade Specifications

Parameters	*Specification*
Propeller Length	10 inches
Propeller Pitch	4.5 inches
Propeller Adapter	3 mm

4.2. Drive Circuit and Power Supply

BLDC motors for the quadrotors, are controlled each by 30 Ampere Electronic speed controller (ESC). The three series cell Lithium Polymer (LiPo) battery is used to power up the quadrotor. The battery and the four motors are directly connected to a power distribution board.This board distributes the power to the individual motor uniformly. The power to the motor is fed from the ESC which is connected between the power distribution board and the motor.

4.2.1. ESC

The Electronic Speed Controller (ESC) is an electrical drive circuit to control BLDC motors for the quadrotor. The ESC mainly converts the DC supply to a 3 phase supply to the BLDC motor. These ESCs can be electronically controlled. The quadrotor has four ESCs connected individually to each BLDC motor. The details of the ESC are listed in Table 3.

Table 3
Specifications of the ESC

Parameter	*Specification*
Input Voltage	11.1 V 3 cell LiPo
Drive Current	30 A
Size	50×23×08 mm
Battery Elimination Circuit	5V / 3A
Weight	25 g
Motor frequency	16 KHz

4.2.2. Power Board and Battery

The Quadrotor is powered through a Lithium Polymer pack for maximum weight to power ratio and maximum discharge capacity required to generate enough thrust to lift off. This battery is connected to the power distribution board which supplies the power for the Motors via the ESC.The details of the battery are listed in Table 4.

Table 4
Battery Specification

Parameters	Specification
Minimum capacity	5200 mAh
Configuration	3S1P / 11.1 V / 3 Cell
Constant discharge	25 C
Peak discharge	35 C
Pack weight	417 g
Pack size	160 × 62 × 32mm
Charge plug	JST-XH
Discharge Plug	XT60

4.3. Micro Controller Unit

The Quadrotor for the application of vision in loop autonomous landing requires two different set of controllers used for specific tasks. KK 2.1 is used as the flight controller and Raspberry Pi is used as a master to control the flight controller.

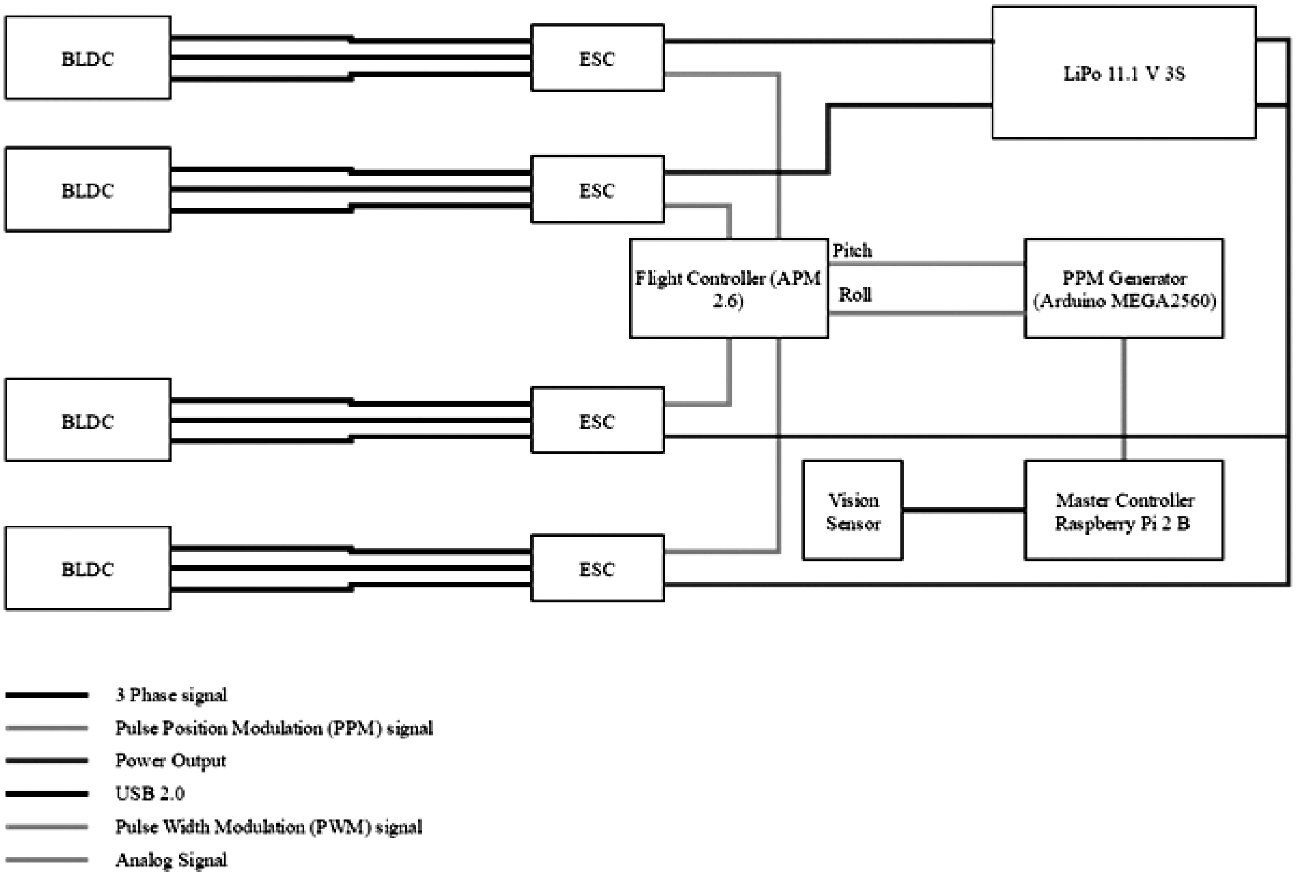
BLDC
ESC
LiPo 11.1 V 3S
BLDC
ESC
Flight Controller (APM 2.6)
Pitch
Roll
PPM Generator (Arduino MEGA2560)
BLDC
ESC
Vision Sensor
Master Controller Raspberry Pi 2 B
BLDC
ESC
3 Phase signal
Pulse Position Modulation (PPM) signal
Power Output
USB 2.0
Pulse Width Modulation (PWM) signal
Analog Signal

Figure 4: Signaling Block diagram

From the Fig. 4 stated above, the vision sensor is used for x, y and z positional feedback whose signals are generated using Raspberry Pi which does so through image processing and then converting information of position into thrust, yaw, pitch and roll signals. These Pulse Position Modulation (PPM) signals are read by flight controller which in turn updates its output PWM signals accordingly using PID control algorithm. Using PWM signals, the ESC then accordingly generates 3 phase power output to control the BLDC motor of quadrotor.

4.3.1. Flight Controller APM 2.6.

The flight controller in theQuadrotor is used to control the dynamics of the system. Using real time feedback of attitude provided by Inertial Motion Unit (IMU), the flight controller generates error signal which are provided to the ESC to control BLDC motors.The Flight Controller details are mentioned in Table 5.

4.3.2. Raspberry Pi

The Raspberry Pi is used as a master controller which acquires images from image sensors and performs Image processing to provide a vision feedback of the quadrotor's position to the flight controller to control its position. The Raspberry Pi Hardware, Software specifications are mentioned in Table 6.

Table 5
Flight Controller Specifications

Parameter	*Specification*
Size	70.5 × 45 × 13.5mm
Weight	31 g
Microcontroller	Atmega2560 and Atmega32 U2
Embedded sensor	MPU 6000 Inertial Motion Unit (IMU)
Auto-Level	Yes
Input Voltage	4.8 – 6.0 V
AVR Interface	6 pin USBasp AVR programming interface
Signal from receiver	1520 μs Pulse Position Modulation 5 Channel
Signal to ESC	1520 μs Pulse Position Modulation

Table 6
Raspberry Pi Hardware and Software Specifications

Parameter	*Specification*
SoC	Broadcom BCM2836 (CPU, GPU, DSP, SDRAM)
CPU	900 MHz quad-core ARM Cortex A7
GPU	Broadcom VideoCore IV @ 250 MHz
Memory	1 GB (shared with GPU)
USB Ports	4
Video input	15-pin MIPI camera interface (CSI) connector
Video outputs	HDMI, composite video via 3.5 mm jack
Audio input	I2S
Audio outputs	Analog via 3.5 mm jack; digital via HDMI and I^{2}S
Storage	MicroSD
Network	10/100 Mbps Ethernet
Peripherals	17 GPIO plus specific functions and HAT ID bus
Power rating	800 mA (4.0 W)
Power Source	5V via MicroUSB or GPIO header
Size	85.60×56.5 mm
Weight	45 g
OS Platform	Raspbian
Platform	OpenCV
Language	Python 2.7.2
Packages	cv2, fswebcam, numpy, Matplotlib, RPi.GPIO, os

5. VISION ALGORITHM

The main theme of the proposed work is about incorporating a vision system for precise landing of the quadrotor. A camera is intended to be placed on the quadrotor facing the ground below. In simulation environment the images acquired from different standoff distances are used. The following section of the paper describes the various steps involved in processing of the images to extract the marker information which helps in the guidance of the quadrotor.

5.1. Scene Constraints

The marker used for the research work which is the reference information for the drone to land, is shown in the following Fig. 5. The marker consists of a dark triangular background which contains a white circular region whose centre is considered as the landing location of the quadrotor. The main assumption in the proposed is the constancy of the background with the relative pose between the quadrotor and the marker as a variable. The entire demonstration arena is a hangar with concrete flooring.

One of the inherent challenges in any vision application is the ability of the algorithm to be robust for illumination changes. The proposed algorithm operates on binary images of the scene and hence choosing

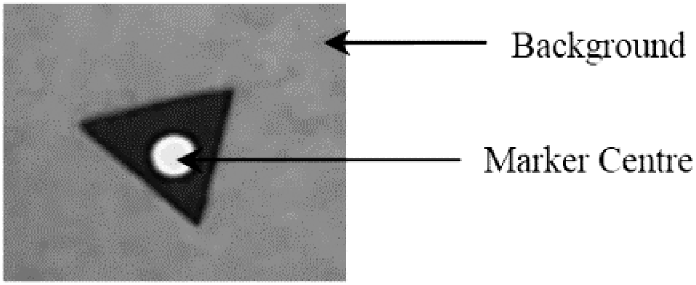

Figure 5: Marker for Landing Location

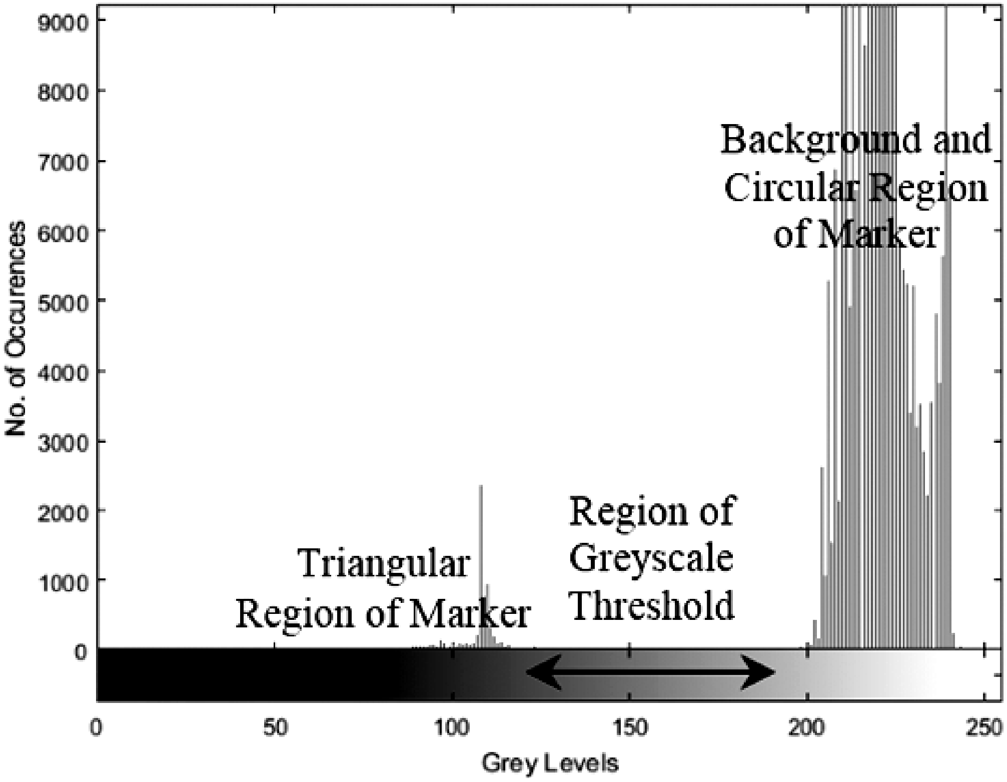

Figure 6: Sample Histogram of the Marker Image

the right threshold becomes a very important step, since the value of the threshold must be suitable for a broad variation in the ambient illumination conditions. A sample histogram of the marker image acquired from a standoff distance of 150 inches is shown in Fig. 6. The darker section of the histogram is the marker and the brighter section of the marker is the background. The sufficiency of the gap is the main attribute desired by the image processing algorithm for the possible variations in the ambient illumination. In the proposed research work, in order to make the threshold value for binarization of the grayscale images, the marker image is acquired under different ambient illumination conditions.

The Fig. 7 shows the images of the marker acquired under various ambient illumination conditions. The images also show the range of thresholds possible for the binarization. It may be observed that the poor illumination eliminates the contrast between the object and the background which causes the failure of the image processing algorithm. This experiment also helps in identifying the range of illumination under which the system will perform. The range of threshold that may be used for binarization is also obtained from this experiment.

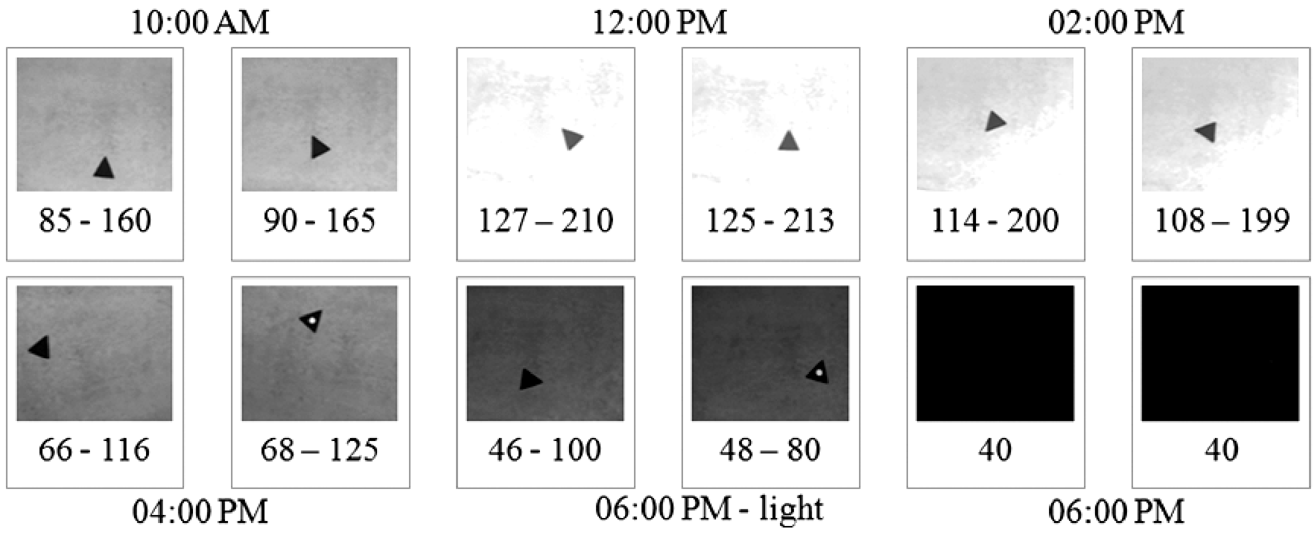

Figure 7: Marker Images Under Varying Ambient Illumination

5.2. Image Acquisition

The various specifications of the image system are presented in Table 7. The image acquisition device onboard the quadrotor is a low cost USB camera which is meant to communicate with the flight controller. For the simulation purpose, the image of the marker is acquired from various standoff distances.

Table 7
Imaging Specifications

Parameter	*Specification*
Light Source	Ambient Sun Light (Indoor Environment)
Lighting Model	Partially diffused bright field incident lighting
Camera Make &Model	Logitech Webcam C170
Interface	USB 2.0
Resolution	1024 × 768
Operating Frame Rate	30 Frames Per Second (fps)
Mode of operation	Mono8 mode
Trigger Type	Software trigger
Image Acquisition Time	~150ms
Processor	Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz (4 CPUs), ~2.0GHz
Memory	8192MB, 1600MHz

5.3. Camera Modelling

The camera modelling process relates the metric real world information to information in image space. For the current work, a scaled orthographic projection type of camera model would suit the purpose since the marker is a planar geometry and the relative motion between the camera and the marker is a linear path along the optical axis. If a point $P(X, Y, Z)$ on the real world with reference to the camera is mapped to the point $I(x, y)$ on the image plane then the relation as per a scale orthographic projection may be defines as:

$$x = s.\,X \;\&\, y = s\,.Y \tag{34}$$

Where s is the scaling factor which is a function of the standoff distance, Z. The distance tool in the Image Tool GUI in MATLAB is used to measure the initial scaling factor at a known distance of the camera from the object. A linear interpolation is assumed for the images acquired from various distances.

5.4. Image Processing

For the simulation study of the vision guidance, pre-acquired images of the marker from different standoff distances, are saved in the hard-disk of the PC. The grayscale images of the marker are subjected to the thresholding process where the grayscale threshold is chosen based on Fig. 7. Since the triangular region has a lower gray value compared to the background, the binary image is complemented to ensure the convention of object and background is intact. The raw grayscale image, thresholded binary image and its complemented version are shown in Fig. 8.

5.5. Image Analysis

The main image analysis component in the proposed method is the blob analysis for detecting centroid of the marker which is the primary information for the quadrotor for landing. Before estimating the centroid of the marker, the triangular region is subjected to a morphological closing operation to estimate the centroid. This approach is intentionally chosen so that when the standoff distance is very high the spatial resolution of the image is poor hence the estimate of the lateral relative position error of the quadrotor with respect to the marker may be inaccurate. In order to tackle this, a two stage vision-in-the-loop control strategy is planned which is explained in the next section of the paper.

During the first stage control the centroid of the triangular region of the marker is taken. And when the quadrotor has descended down to a height where the magnification is very high and the complete triangular marker is no more visible, the control switches to the second stage. In the second stage the centroid of the circular region of the marker alone is considered. This switching is estimated by monitoring the values of corner pixels of the binary image of the scene acquired while descending down. The switching to the second stage of control happens exactly when the boundary pixels of the complemented binary image starts to have binary 1. During the second control stage the morphological closing is not carried out since the circular white region of the marker is the feature of interest. The SIMULINK model of the image processing part is shown in Fig. 9.

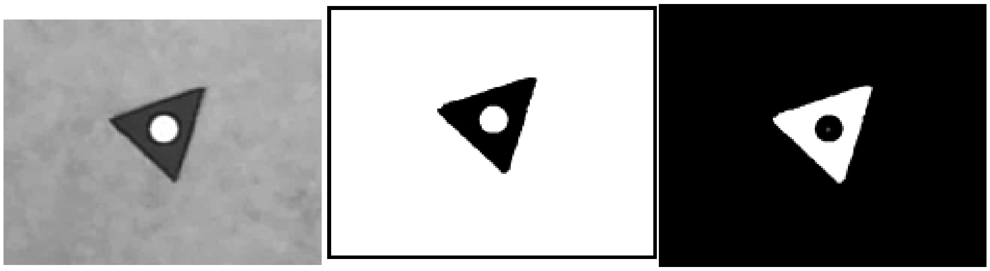

Figure 8: Images during Various Stages of Processing

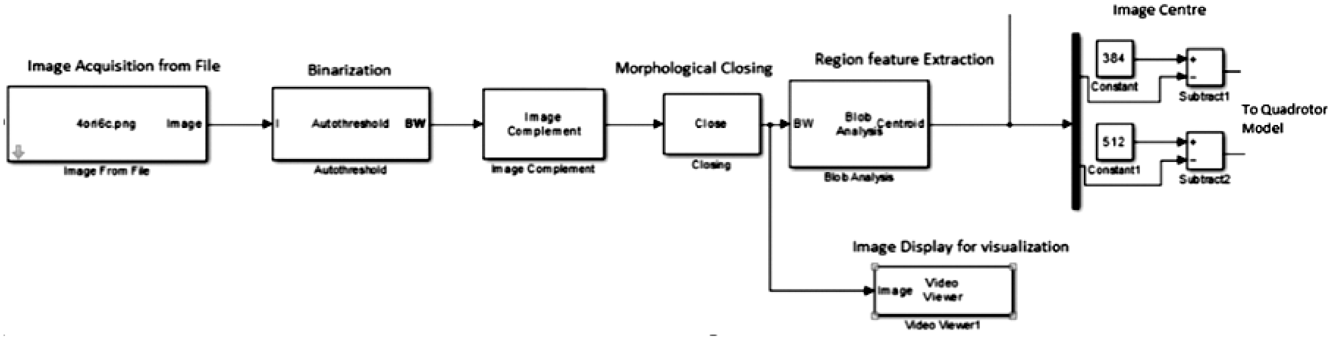

Figure 9: SIMULINK Model of the Vision Algorithm

6. SIMULATION OF QUADROTOR LANDING

The quadrotor is expected to land precisely on the coordinates defined on the marker. The vision sensor fetches complete information of the marker as it descends. With the marker being within the field of view of the camera, it is simpler for the quadrotor to locate it. While designing the marker, the centroid of the triangle and the circle are made the same for a purpose that the quadrotor need not change its path while the circle acts as a marker and gives it a vision feedback. With the help of image centre and marker centroid, the quadrotor calculates its position error $P_E(x, y)$. The image acquired is of 1024 × 768 pixels size.

$$x = 512 - dx \;\&\; y = 384 - dy \tag{35}$$

Where dx & dy give the difference in the centroid positions which is calculated between the image centre and the marker centroid. Landing of quadrotor refers to controlling of position $P_E(x, y)$ and angular position i.e. pitch and roll. The initial condition of the quadrotor is such that the marker is in the field of view of vision sensor. The angular positions, ϕ and θ are controlled by the outer loop controlling positional error through vision. The altitude of the quadrotor is controlled using mapping of magnification of the marker to the altitude. The quadrotor starts descending only after finding the centroid at the first instance. Following this, the descent of the quadrotor is independent of the position error until it reaches a height of 1 m above the marker.At this point, the quadrotor hovers above the marker until the position error is corrected to a minimal value of 0.01 m in both x & y coordinates. The block diagram in Fig. 10 illustrates the quadrotor control.

6.1. Simulation Environment

The simulation of the quadrotor landing system using a vision sensor is ignoring the aerodynamic parameters like wind, air drag and other external conditions. The simulation demonstrates the landing of the quadrotor when the control loop is given the position error and also aims to link the image data which gives the position error with the quadcopter model. The outer controller is responsible for taking in vision feedback from vision sensor and generating output.The inner control loop is directly responsible for controlling the

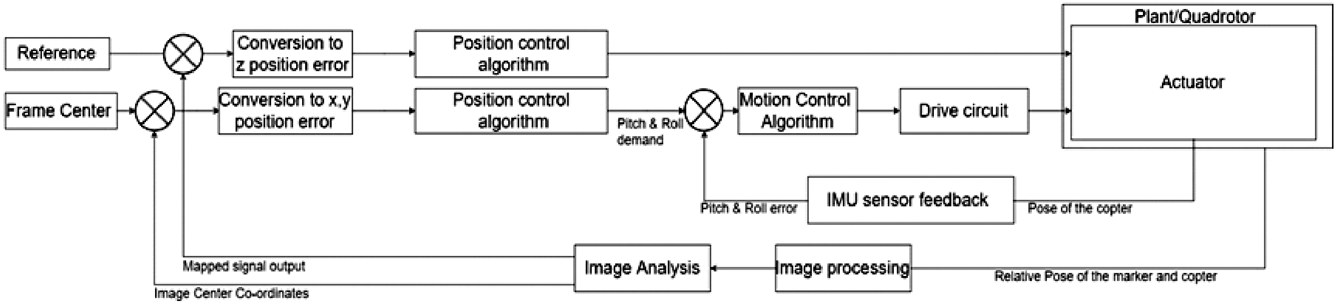

Figure 10: Vision in the Loop Control for Quadrotor Landing

actuators i.e. BLDC motors by taking in the motion and orientation feedback from the quadcopter. The simulation is performed in a cartesian space whose size is given as follows:

$$\begin{aligned} -3m \le x \le 3m \\ -3m \le y \le 3m \\ 0 \le z \le 6m \end{aligned} \tag{36}$$

The descent of the quadrotor begins from a height of 6m considering the fact that the marker is fully visible in the cameras' field of view with well-defined boundaries within this region. The descent rate is independent of the poistion error $P_E(x, y)$, i.e. that descent continues as long as the quadrotor reaches the ground. Fig. 11. shows the two frames from the simulation of the landing of the quadrotor.

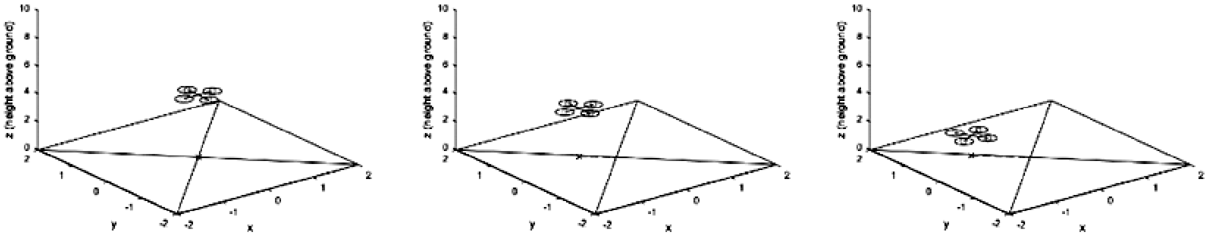

Figure 11: Simulation of Quadrotor Landing

6.2. Descend Control Simulation

The descent control of the quadrotor depends on three control loops. The control system block diagram, Fig. 9 represents the control loops in the quadrotor. These loops are X,Y position control loop, Pitch and roll control loop and the Z error control loop. Each of the control loop are aimed to work with the information of the position $P_E(x, y)$,the orientation in terms of angular position which is the pitch and roll of the quadrotor and the descent above the z-axis.

6.2.1. X, Y position control loop

The position control loop depends on the instantaneous position error $P_E(x, y)$ which is calculated by keeping Vision In loop so that the error signal is generated in the form of pixels. It then converts it into position error which is fed to the controller. As per Fig. 1, image is acquired through the camera mounted on the quadrotor and the image is analyzed to detect the properties of the marker. The position of the marker in pixels is then fed as feedback to the x, y position control loop. The error signal is then converted to metric unit which is then fed to position controller. The position controller generates output that is further fed to control the pitch and roll of vehicle.

6.2.2. Pitch and roll control loop

Motion control loop is the inner loop of the system which is responsible for controlling the Pitch and roll of quadrotor. As per Fig. 10, the pose related information is given as the feed to the loop through inertial motion unit (IMU). The pitch and roll signal demand signals generated through the outer X, Y position control loop is compared to the feed signals from the IMU. The generated error signal is fed to the motion controller. The output of motion controller is then converted to motor signals through drive circuit and is used for actuation of rotors in the system.

6.2.3. Z position control loop

Z position control loop is responsible for controlling the altitude of the quadrotor. As per Fig. 10, image is acquired through the camera mounted on the quadrotor and analyzed to detect the properties of the marker.

The function of magnification of marker and altitude information is pre-fed to the vehicle. Based on this information, the image analysis block generates mapped signal output which is compared with the reference. The error signal generated is then fed to position controller. The position controller generates thrust control signals. These signals control the altitude of the quadrotor.The vision sensor feed helps in the landing of the quadrotor only till the circular portion of the marker is in the field of view of the vision system. Beyond 0.20m height the quadrotor will be unable to view the inner circular marker as it starts losing the complete image of the circle, post which the quadrotor descends on its own without vision in the loop. This is when the quadrotor purely relies on the previously set or defined position errors and continues to descend assuming no further changes in the coordinates of landing. The descent beyond this is initiated only after the position error is corrected to lie within 0.01m.

6.2. Docking in Real World

The concept of the docking is to achieve a precise landing of the quadrotor over the coordinate. A simple mechanism is designed, where in two parts mate each other to dock the quadrotor over the landing station. A mechanism with a spherical surface makes it easier by providing multiple contact points for the quadrotor to slide down on to the dock. As it touches the docking surface, it tries to automatically align itself along the spherical surface Fig. 12 describes the docking mechanism. The onboard IMU ensures that the quadrotor base is along one plane as it lands.

The mating part is mounted at the centre below the camera of the quadrotor. A vent is created on the mating part through which the camera module gets its vision. There is no mechanical contact between the camera module and the mating part.

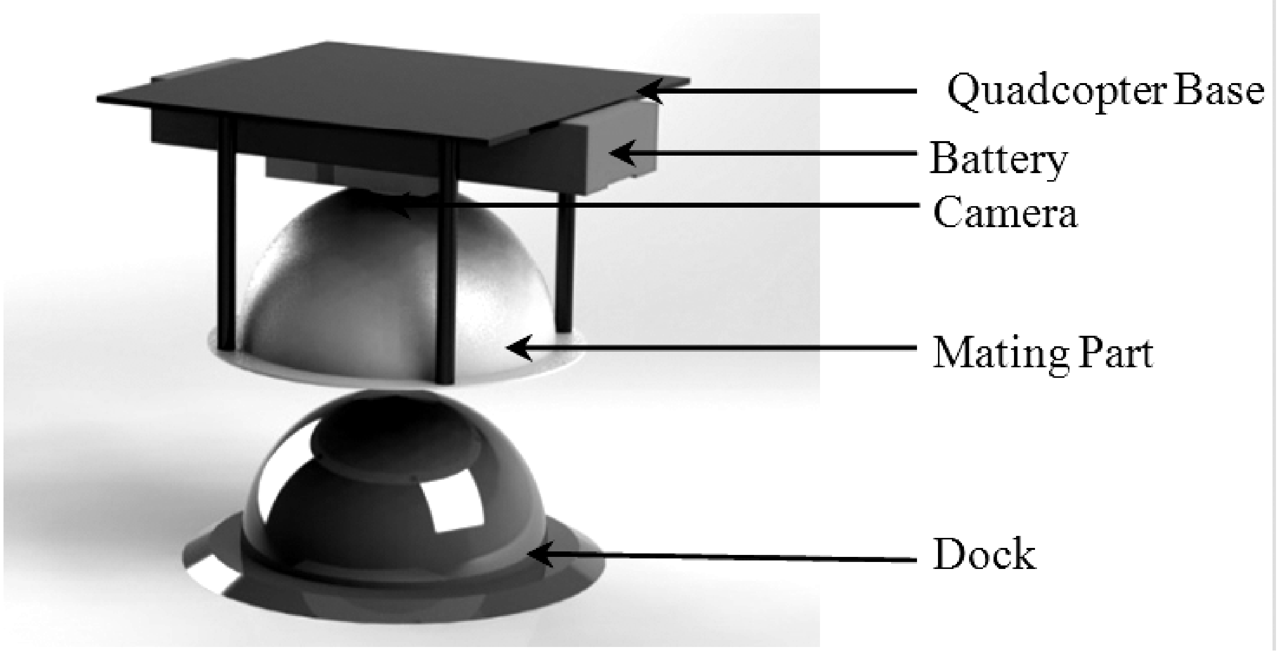

Figure 12: Docking System

6.3. Concluding Remarks

This paper presented the issue of quadrotors landing method in a simulated environment using MATLAB and SIMULINK and the information of the hardware usage with their role. The explanations can be applied to any existing quadrotor, as the model is based on the simulated results which can be altered as per the requirement. The experiments were conducted in a closed environment which excludes the illumination parameters as the light inside is diffused in nature.This offers further scope for the algorithm to exhibit illumination-invariance properties. The increase in the upgradation of aerial vehicles at a fast rate demands landing stations for various applications like charging, establishing communication with different landing station and the ground station, instant take-off and landing.This paper gives an idea to implement landing in real time which gives a platform to extend to any of the above mentioned tasks. The description of the various challenges presented in the paper would make significant contribution to the open literature in the context of vision guided aerial vehicles which is a rapidly evolving trend in aerial robotics.

REFERENCES

[1] Bi Yingcai, and Haibin Duan.: Implementation of autonomous visual tracking and landing for a low-cost quadrotor, Optik-International Journal for Light and Electron Optics vol. 124.18, pp. 3296-3300(2013).

[2] Puri Anuj.: A survey of unmanned aerial vehicles (UAV) for traffic surveillance, Department of computer science and engineering, University of South Florida (2005).

[3] Meier Lorenz, Petri Tanskanen, Lionel Heng, Gim Hee Lee, Friedrich Fraundorfer and Marc Pollefeys.: Pixhawk: A system for autonomous flight using onboard computer vision, Robotics and automation (ICRA), 2011 IEEE international conference on. IEEE, (2011).

[4] H. Chae, J. Park, H. Song, Y. Kim andH. Jeong: The IoT based automate landing system of a drone for the round-the-clock surveillance solution.Advanced Intelligent Mechatronics (AIM), IEEE International Conference on. IEEE, (2015).

[5] Timothy Pattersona, Sally McCleana, Philip Morrowa, Gerard Parra, Chunbo Luob.: "Timely autonomous identification of UAV safe landing zones." Image and Vision Computing vol. 32.9, pp. 568-578 (2014).

[6] Blösch, Michael, Vision based MAV navigation in unknown and unstructured environments, Robotics and automation (ICRA), 2010 IEEE international conference on. IEEE, (2010).

[7] Roberts, Peter J., Rodney A. Walker, and Peter O'Shea, Fixed wing UAV navigation and control through integrated GNSS and vision, AIAA Guidance, Navigation, and Control Conference and Exhibit, (2005).

[8] Dusha, Damien, Wageeh W. Boles, and Rodney Walker.: Fixed-wing attitude estimation using computer vision based horizon detection, pp. 1-19 (2007).

[9] Calise, Anthony J.: Applications of adaptive neural-network control to unmanned aerial vehicles, AIAA/ICAS International Air and Space Symposium and Exposition: The Next.vol. 100 (2003).

[10] How, Jonathan P.: Real-time indoor autonomous vehicle test environment, Control Systems, IEEE vol. 28.2, pp. 51-64 (2008).

[11] Peter Corke.: Robotics, Vision and Control: Fundamental Algorithms in MATLAB, Springer Tacts in Advanced Robotics (2011).