ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 3, Issue 11, November 2016

Autonomous Agile Flying Robot (Quad rotor)

[1] Kshthija C Reddy [2] Likitha B A [3] Ramya N [4] Shubha N [5] Dr. B Shadaksharappa

Department of Computer Science and Engineering

Sri SaiRam College of Engineering

***Abstract:* -- An unmanned aerial vehicle also known as UAV is anunpiloted aircraft which can either be remotely operated or flown autonomously based on pre-programmed flight plans. Usually these types of vehicles are used in military applications for missions that are too dangerous for manned aircraft. They are also used in a growing number of civil applications such as aerial photography and the transport of various goods. The last decade has seen rapid progress in micro aerial robots, autonomous aerial vehicles that are smaller than 1 meter in scale and 1 kg or less in mass. Winged aircrafts can range from fixed-wing vehicles to flapping-wing vehicles , the latter mostly inspired by insect flight. Rotor crafts, including helicopters, coaxial rotor crafts,ducted fans, quadrotors and hexarotors, have proved to be more mature with quadrotors being the most commonly used aerial platform in robotics research labs.This paper focused on the development of a family of trajectories defined as a sequence of segments each with a controller parameterized by a goal state.This approach permits the development of trajectories and continous enabling aggressive maneuvers such as flying through narrow,vertical gaps and perching on inverted surfaces with high precision and repeatability.This design describes a flying quadrotor prototype of 750gms,38 cm diameter with on board altitude estimation and control that operates autonomously.The bot is designed to be controlled by PCB circuits with usage of microcontroller, actuators and sensors. The controlling programs are to be stimulated by MAT LAB/Arduino.This work will be useful in the field of Defence,Navigation,Automation and also could be a multi utility product.**

I. INTRODUCTION

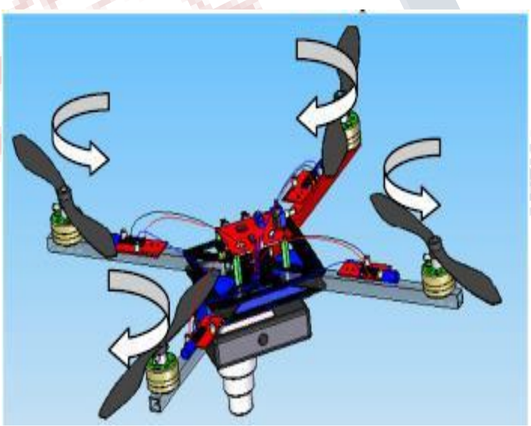

Fig. 1 Flying robot

The last decade has seen rapid progress in micro aerial robots, autonomous aerial vehicles that are smaller than 1 meter in scale and 1 kg or less in mass.

Winged aircrafts can range from fixed-wing vehicles to flapping-wing vehicles, the latter mostly inspired by insect flight.Rotorcrafts, including helicopters, coaxial rotor crafts,ducted fans, quadrotors and hexarotors, have proved to be more mature with quadrotors being the most commonly used aerial platform in robotics research labs. In this class, the Hummingbird quadrotor sold by Ascending Technologies, GmbH, with a tip-to-tip wingspan of 55 cm, a height of 8 cm, mass of about 500 grams including a Lithium Polymer battery and consuming about 75 Watts is a remarkably capable and robust platform. Of course micro aerial robots have a fundamental payload limitation that is difficult to overcome in many practical applications. However larger payloads can be manipulated and transported by multiple UAVs either using grippers or cables. Applications such as surveillance or search and rescue that require overage of large areas or imagery from multiple sensors can be addressed by coordinating multiple UAVs, each with different sensors. Our interest in this paper is scaling down the quadrotor platform to develop a truly small micro UAV. The most important and obvious benefit of scaling down in size is the ability of the quadrotor to operate in tightly constrained environments in tight formations.While the payload capacity of the quadrotor falls dramatically, it is possible to deploy multiple quadrotors that cooperate to overcome this limitation. Again, the small size benefits us because smaller vehicles can operate in closer proximity than large

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 11, November 2016

vehicles. Another interesting benefit of scaling down is agility. As argued later and illustrated with experimental results, smaller quadrotors exhibit higher accelerations allowing more rapid adaptation to disturbances and higher stability.

II. COMPONENTS

Onboard controller hardware: Following

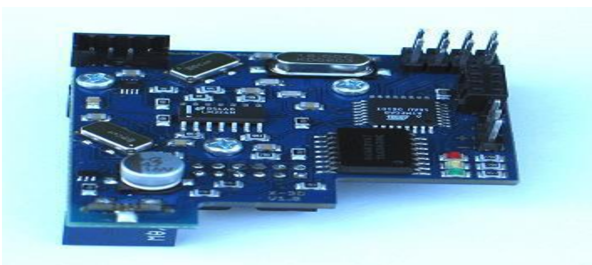

Fig. 2 Controller hardware

a minimalistic approach, the central controller board was kept as simple as possible in order to reduce cost and failure rate. It consists of three low-cost piezo gyroscopes, an 8-bit digital to analog converter (DAC) and an AVR microcontroller.

Table 1: Component Specifications

S.No.	Components	Specifications
1	Brushless DC motor	Speed-1200rpm/v Voltage-11.1 V
2	ESC	Max.current- 30 A Cut-off voltage-4V
3	Propeller	Size- 10"x 4.5"
4	Li-po Battery	Voltage : 11.1 Capacity : 1800mAph
5	Indiuno	Op.voltage-5V
6	LM35 Temperature sensor	Range- (-55 to + 150˚C)
7	Ultrasonic Distance sensor HC SR04	Range- (2-500 cm)

Despite this very lean design, this controller is very capable due to efficient control algorithms. The central controller board is used to read sensor-data, compute angular velocities and angles in all axes and to run independent control loops for each axis. Thus, the highest accuracy can be achieved. All processing is done with a control loop frequency of 1kHz. The main consequence of high frequency control is a low drift rate of the relative angles, as errors arising from time discrete integration are small, and a very stable flight because of very short deadtimes in the control loop. Furthermore, the high update rate facilitates FIR filtering sensor data in software without generating big delays. This capability reduces vibrations and shakiness during the flight.

Onboard Controller Structure:

The onboard controllers are three independent PD loops, one for each rotational axes (roll, pitch and yaw). Angular velocities measured by the gyroscopes and computed relative angles are used as inputs. The angles are derived by integrating the sum of the output of one gyroscope and an external control input for the respective axis. Without an external input signal the calculated integral represents the angle the flying robot has turned in the respective axis. Looking at the closed loop and disregarding measurement noise and integration errors, this means that the robot will always keep its current orientation. The integrated angles can be shifted by an external control input. As a result, the robot‟s orientation changes proportionally to the input. Its movements are controlled by steering it to a certain orientation and keeping this orientation for a certain time. Due to measurement noise and discrete integration the integrated angles drift about ±3 degrees per minute. However, this drift can be easily compensated by a human pilot or an autonomous external position control such as a motion capture system. Figure shows the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 11, November 2016

principal structure of our onboard controllers commonly referred to as ”heading-lock”. The controller implementations have been optimized for shortest possible execution time and robustness in almost every flight situation. Three controllers are running in parallel on an 8-bit AVR microcontroller (ATMega8). The loop is interrupt triggered, which enables stable time constants for integration and filtering. By using the AVR‟s internal ADCs at a high sampling rate, fixpointarithmetics only, runtime optimized FIR filter implementations and interrupt driven I2C communication to update the motor speeds, we achieved a system running at a control frequency of 1kHz.

b) Basic Structure Of The Onboard Control-Loops:

All controller parameters have been set empirically and optimized experimentally over several months. Our central controller board including the controllers is compatible to the Silverlit X-UFO, which is available on the international toy market. From January to September 2006 we had 35 people beta-testing the hardware and optimizing parameters within hundreds of hours of human controlled flight. During this period both, hardware and software, have been optimized as far as possible. The result is a very reliable hardware revision of the central controller board, as well as a set of controller parameters capable of reliable control during slow movements as well as during fast maneuvers, even including loops where the robot is inverted for short periods.

c) Brushless Motors And Rotors:

The brushless outrunner motors used in our flying robot are a special design for low rpm applications. The stator diameter is 22.5mm, the stator height 5mm. The windings result in a motor constant of 1000rpm/V . The weight of the complete motor is 19g. The rotor was designed to fit directly to the left and right turning propellers from the Silverlit X-UFO. Those propellers are available very cheap as spare parts of the X-UFO and offer good performance with excellent safety as they are very flexible.

III.AUTONOMOUS FLIGHT

We implement autonomous flight by using an external sensor system (i.e. motion capture system) to compute the position, height and yaw for the robot. The sensor system can be GPS or DGPS for outdoor applications, or any kind of indoor tracking systems like a sensor node network, an ultra sonic position measurement or an optical motion tracker.

IV.AUTONOMOUS FLIGHT USING A MOTION CAPTURES SYSTEM:

We have performed hundreds of hours of human controlled flights with our platform. Those experiments demonstrate the robustness, stability and endurance of our platform. In this section we focus on autonomously controlling the robotindoors. We use an external sensor system that is reliable indoors motion capture system that uses a system of cameras to compute position information.

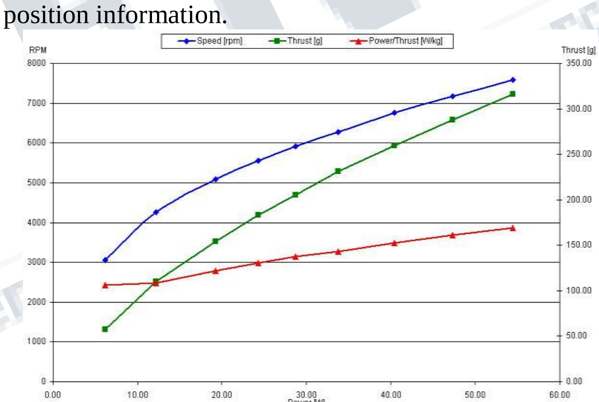

Fig. 3 Variation of RPM with power

a) Experimental Setup: The autonomous flight controlexperiments were performed in the “Holodeck” lab at MIT.

This lab is equipped with an indoor motion tracking system by VICON that can measure the position vector of specific points on the body of the robot. These points are marked by incorporating small tracking balls on the body of the robot at the desired locations. We measure the robot‟s position vector

$$\underline{x} = \begin{pmatrix} X \\ Y \\ Z \\ \varphi \end{pmatrix}$$

where X, Y and Z are the Cartesian coordinates relative to the motiontracker‟s origin and is the orientation in yaw. To get reliable measurements of this vector we used three markers tracked by the motion tracking system and arranged them in the configuration of an isosceles triangle. We attached one marker to the front of the flying platform, one to its right, and one to its left hand side. Given the Cartesian coordinates of each marker, the robot‟s position and orientation can be determined using simple geometry. The markers‟ positions are transmitted via a TCP/IP-Link to a computer running the position control algorithms. After identifying the markers by mapping them to a model of the robot, the robot‟s orientation and position is calculated and provided as real-time input to the controllers. The update frequency of the position controllers is set and limited to 50Hz due to the limitations of the R/C transmitter used for sending commands to the flying robot.

The performance and stability of the onboard electronics make this external control loop frequency of 50Hz adequate for achieving stable flight. In our experiments we observed that frequencies as low as 5Hz result in stable performance. However, a higher frequency enables higher position accuracy, especially during fast maneuvers. The system diagram is shown in Figure. The transmitter we used is a standard model helicopter R/C. However, we had to modify the internal electronics using another AVR microcontroller to connect it to the laptop. The protocol of the serial interface allows us to select a source independently for each of the channels. The source can either be the joystick for human control or the PC-software. This system has a user interface for developing the position controllers which enables debugging, testing and optimization step by step.

b) ***Position control***: The laptop receives the datastreamfrom the motion tracking system and outputs data to the transmitter. There are four independent controllers running on the laptop computer. They are implemented using a customized C++ software module. The control loops are timer triggered to enable a precise 50Hz update rate. The Yaw-Controller was implemented as a PD loop. Inputs for the controller are the measured yaw angle, its FIR lowpass filtered derivative, and the desired yaw angle (heading). The height controller is non linear and was implemented using an

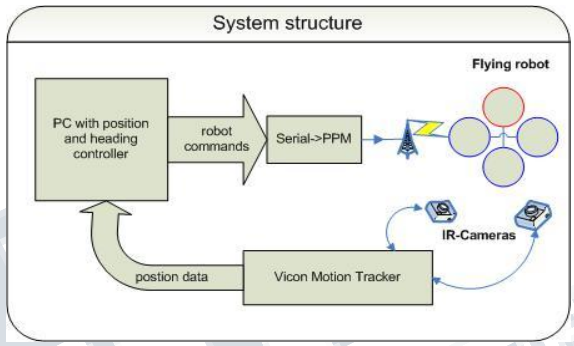

Fig. 4 System Structure

accumulator. The idea is to maintain a mean value for the total thrust required to get the robot hovering. This mean value has to adopt to battery voltage drop and to compensate for payloads. Adaptation is achieved by an accumulator that counts up whenever the robot is below its desired height, and down otherwise. In addition to this controller we use a second controller that is capable of fast response to compensate for sudden changes like turbulence and wind. The second controller is implemented as a standard PD-loop. Figure shows the structure of the height controller.

The X-axis and Y-axis controllers are identical and were more challenging to derive. The system is harder to control in these degrees of freedom since there is no proportional behavior response. The inputs of the onboard controllers are proportional to the rotational velocity in pitch and roll, but they are not directly proportional to horizontal speed. For this reasons we designed a cascaded controller system. The inner controller cascade is a horizontal speed controller that uses horizontal speed and acceleration as inputs Controller structure of the height controller weighting the accelerations, we achieve ”predictive” behavior in this controller, much like a human pilot controlling this

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 11, November 2016

system would have. The outer controller cascade is a PD-controller whose output is the desired speed for travel to the desired position. Figure shows the structure of the X and Y position controllers.

V.CONTROLLER STRUCTURE OF THE X AND Y POSITION CONTROLLERS:

All controller parameters have been determined empirically and tuned experimentally. Finding parameters was easy. We believe this is due to the good stability properties of the robot and its high-rate update.

a) Hovering Accuracy:

In the first experiment the flying robot was commanded tomaintain its flight position at

$$\underline{x_0} = \begin{pmatrix} x = 0mm \\ y = 0mm \\ z = 1000mm \\ \varphi = 0 \end{pmatrix}$$

The following figures show the achieved position accuracy while hovering for 150 seconds. The data in figures 7, 8, and 9 show that the flying robot‟s deviation from its desired position is less than ± 10cm in X. Probability for X/Y-Positions trying to stay at X = Y = 0m.

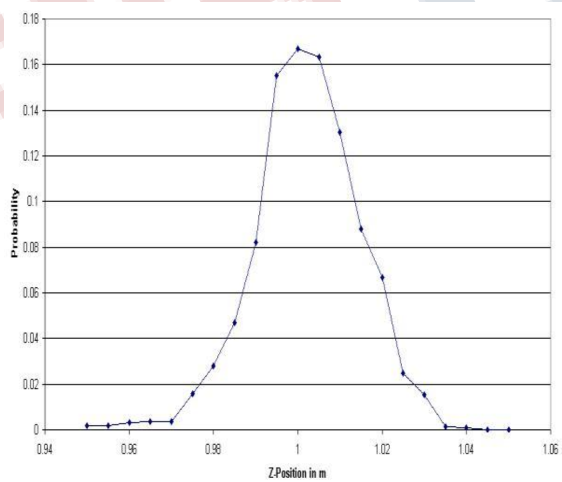

Fig. 5 Probability for X&Y positions Probability for an actual height Z at desired Z = 1m.and Y

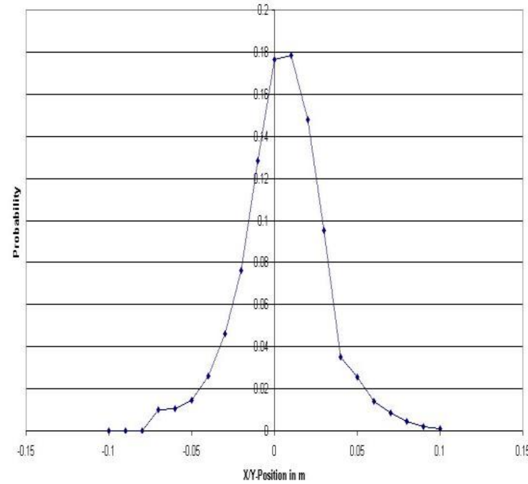

Fig. 6 Variation of x & y with z axes and ± 4cm in Z axis and is within ±1 degree in

VI.FOLLOWING A TRAJECTORY:

In the second set of experiments the robot was controlled to follow a trajectory including auto takeoff and landing. The robot was commanded to start at the center of a square with a side length of 1.2m. After a successful auto takeoff to a height of 1.0m the robot was required to travel to one of the corners, then to follow the perimeter of the square, and finally to return to the center of the square and execute an autonomous landing maneuver. This experiment was repeated 10 times. Figure shows the results of this experiment. The desired trajectory is marked in red. The measured trajectory is marked in blue. The entire maneuver (including autonomous takeoff and landing) takes 55 seconds to complete. The maximum deviation to the desired square was 0.1m, which is consistent with the hovering results. Probability for an actual heading at desired heading = 0.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 3, Issue 11, November 2016

VII.FLYING ROBOT FOLLOWING A TRAJECTORY

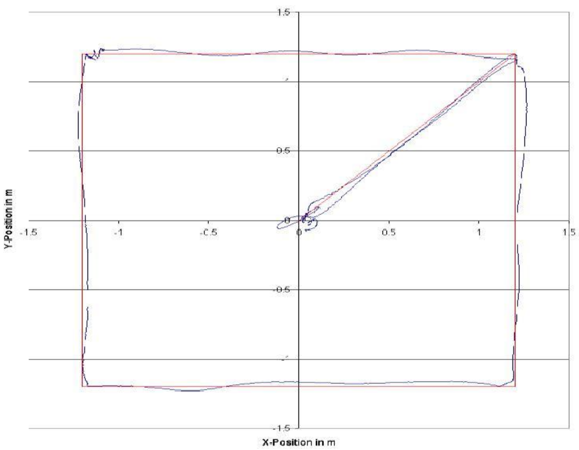

Fig. 7 Flying Trajectory

VIII.CONCLUSIONS:

In this paper we presented a reliable and efficient solution for a UAV. Our solution is simple, stable, and inexpensive. The key innovation is a platform capable of very high update rates and the development of simple, adaptive, and highly optimized controllers. Our plans for the future include testing the platform in combination with acceleration sensors for dynamic and acrobatic maneuvers. We also plan to continue our work with a second generation platform offering even longer flight times and larger payload capabilities. Ultimately, we wish to see this platform used as a mobile node in mobilesensor networks that use cameras for mapping, monitoring, and tracking. We have already done some preliminary experiments in which our smaller platform was controlled to fly indoors and outdoors while carrying a video camera. These preliminary experiments show promise for using our approach in the development of a practical aerial mobile sensor networks.

REFERENCES:

[1]Hawkins electrical guide, pg no.359 – 362

[2]Khurmi.r.s, guptha.j.k., „ a textbook of machine design‟ s.chand and co, newdelhi

[3]Metal finishing by feldstein.n,vol.81 (8),(1983),35-41

[4]Engineering materials and metallurgy by v.jayakumar ,a.r.s. Publications