2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Energy-efficient Autonomous Four-rotor Flying Robot Controlled at 1 kHz

Daniel Gurdan, Jan Stumpf, Michael Achtelik, Klaus-Michael Doth, Gerd Hirzinger, Daniela Rus

Abstract— **We describe an efficient, reliable, and robust four-rotor flying platform for indoor and outdoor navigation. Currently, similar platforms are controlled at low frequencies due to hardware and software limitations. This causes uncertainty in position control and instable behavior during fast maneuvers. Our flying platform offers a 1 kHz control frequency and motor update rate, in combination with powerful brushless DC motors in a light-weight package. Following a minimalistic design approach this system is based on a small number of low-cost components. Its robust performance is achieved by using simple but reliable highly optimized algorithms. The robot is small, light, and can carry payloads of up to 350g.**

I. Introduction

The goal of this project is to create a small, robust and highly maneuverable autonomous flying robot that can be used both indoors and outdoors under any weather conditions. We believe that the key to achieving this goal is to build minimalist platforms that are light-weight and controllable at very high frequencies, e.g. 1 kHz. This approach is in contrast with existing commercial and research platforms where control is done with update rates around 50 to 100Hz. Control at very high frequencies enables very fast response to changing environmental conditions such as strong, choppy winds, and also allows extreme acrobatic maneuvers. The challenges to achieving this kind of control are both on the hardware and the software front. From a hardware point of view we need light-weight low-cost Inertial Measurement Units (IMU) capable of fast responses. From a software point of view, robust control algorithms that are tightly coupled to the hardware are needed. In this paper we describe a four-rotor autonomous robot we developed in response to these challenges.

One of the main design goals was to obtain a high controlling frequency of 1 kHz throughout the system. To support this, our platform features a custom built onboard high-speed sensing system which consists of three gyroscopes to give relative measurements for the robot's angles. High control frequency precludes the use of commercially available brushless motor controllers, such as those found in model aircrafts, as they only allow motor speed update rates of 50 Hz. We designed a new brushless controller capable of a 1 kHz update rate with an I2C interface. This controller has very low deadtimes and supports very dynamic movements. Intensive manual acrobatic flights with loops, flips, spins, sharp turns and combined maneuvers proofed the stability of the controller in extreme situations.

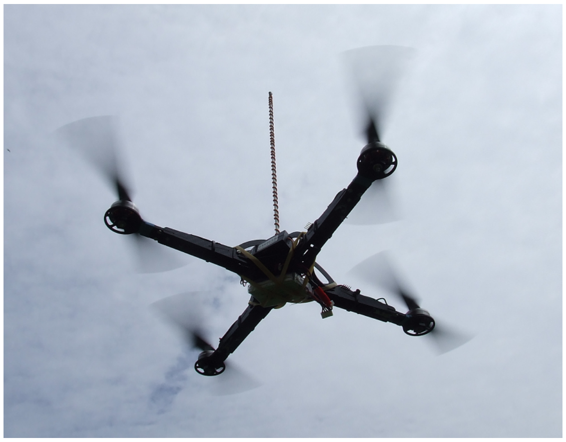


Fig. 1. Our four-rotor flying robot

Having such a high control frequency allows us to create an extremely stable platform, even with payloads of up to 350g. Many applications for such a platform exist. The outstanding stability of the platform makes the integration of onboard and offboard position tracking system possible. At the end of this paper we demonstrate the performance of the system using an external motion tracking system to provide closed loop position control. Cameras mounted on the platform also benefit from a stable image.

II. Related Work

We are inspired by very exciting new results and strides in developing autonomous four motor flying robots. Valetti, Bethke et al. [1] describe a platform based on the RcToys Draganflyer used for experiments at Aerospace Controls Laboratory, MIT. This platform is controlled autonomously using a motion capture system. The control updates are at 50Hz. Robustness in the controller was achieved by relying on software. The platform was used as the basic component in a Multi UAV system. Tasking tools for use by one operator commanding several UAVs on semi autonomous missions were also developed.

The X-4 Flyer described by Pounds, R.Maloy et al. in [2] and [3] was developed at the Australian National University.

Daniel Gurdan, daniel@gurdan.de,
Technical University of Munich, visting at CSAIL, MIT
Jan Stumpf, jan@stumpf.eu,
Technical University of Munich, visting at CSAIL, MIT
Michael Achtelik, Technical University of Munich
Klaus-Michael Doth, University of Erlangen
Gerd Hirzinger,
Head of the Institute of Robotics and Mechatronics, German Aerospace Center, Oberpfaffenhofen
Daniela Rus
Professor at CSAIL, MIT and Director of the Lab of Distributed Robotics

1-4244-0602-1/07/$20.00 ©2007 IEEE.

Hoffmann, Rajnarayan et al. [4] developed STARMAC and STARMAC II at Standford University. The Stanford platform was also used for Multi-UAV Experiments. STARMAC II and the new version of OS4 recently switched to brushless motors to enhance the efficiency. Hanford et al. [5] describe a four rotor helicopter developed at the The Pensylvania State University.

Our work is part of a broader context of developing robust stable control for autonomous helicopters. Many important strides have been made in previous work which has inspired our approach [9], [7], [8], [6], [1], [10], [11]. Our abilities to control the robot differ from other work in the flying robot community as others are bound to commercially available flying platforms and IMUs with update rates between $50Hz$ and $120Hz$, which is an order of magnitude lower than what we use. Our work differs from other hardware platforms in that we have taken a minimalistic approach with a focus on high update rates. This approach has led to a very reliable hardware and control system.

III. The Four-Rotor Hardware

A. General design

Our flying robot has a classical four rotor design with two counter rotating pairs of propellers arranged in a square and connected to the cross of the diagonals. The controller board, including the sensors, is mounted in the middle of the cross together with the battery. The brushless controllers are mounted on top of the booms. Figure I shows a photograph of the flying robot. The weight without battery is $219g$. The flight time depends on the payload and the battery. With a 3 cell $1800mAh$ LiPo battery and no payload the flight time is 30 minutes. We measured the thrust with a fully charged 3 cell LiPo (12.6V) at $330g$ per motor. With four motors the maximum available thrust is $1320g$. Since the controllers need a certain margin to stabilize the robot also in extreme situations, not all the available thrust can be used for carrying payload. In addition, efficiency drops and as a consequence flight time decreases rapidly with a payload much larger than 350g. Because of this we rate our robot for a maximum payload of 350g.

With a 350g payload, a flight time of up to twelve minutes can be achieved. The maximum diameter of the robot without the propellers is $36.5cm$. The propellers have a diameter of $19.8cm$ each. The sensors used to stabilize the robot are very small and robust piezo gyros ENC-03R from Murata [14]. The second design iteration of this robot is already functional but not fully tested and characterized experimentally. This second version additionally has a three axial accelerometer and relies on datafusion algorithms, still running at $1kHz$, to obtain absolute angles in pitch and roll.

B. Components

1) Onboard controller hardware: Following 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. 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. In addition, the control-outputs are combined to compute a desired speed for each motor, which is then transmitted to the respective motor controller. As piezo gyros suffer a high temperature drift, an 8-bit DAC is used to compensate the sensors' drift *before* amplifying the outputs. 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.

TABLE I
General Data

Size (Diameter)	36.5 cm
Propellersize	19.8 cm
Weight	219g (without battery)
Max. Thrust	1320g @ 12.6V
Payload	up to 350g
Flight time	up to 30 min. (without Payload, 1.8Ah battery)
Sensors	three gyroscopes (Murata ENC-03R) optional: acceleration sensors

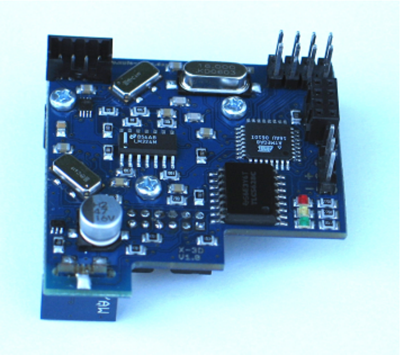


Fig. 2. Central controller board

2) 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

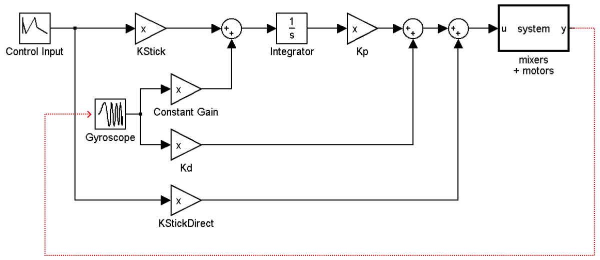


Fig. 3. Basic structure of the onboard control-loops.

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 3 shows the 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, fixpoint arithmetics 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 $1kHz$. 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.

3) Sensorless Brushless Controllers: Our robot uses brushless DC motors. Unlike brushed DC motors, brushless motors are commutated electronically rather than electro-mechanically. The common brushless motors use three phases. Current is always floating through two of these phases, while the third phase floats free and is used to measure the angle of the motor. Then the controllers commutates electronically with a three phase H-bridge. The time when the third phase crosses $Vdd/2$ is called zero-crossing point and triggers the next commutation step after a certain time. The three phases are driven in a semi-sine mode, where the phase difference between any pair of phases is always 120 degree.

Most commercially available sensorless brushless controllers are sold in the model aircraft market. These are controlled using servo impulses with a $50Hz$ update rate. To achieve better stability in every situation, we wish to reduce the deadtime in the control-loop from the gyro measurement to the torque change of the motor. We designed the system for a control-loop frequency of $1kHz$. In order to update the motor speeds fast enough, we use an I2C-Bus connected to four custom built brushless controllers. The microcontroller used on the brushless controller is an Atmel ATMega168 [13]. This microcontroller was selected because it offers all required hardware features like timers, PWM generators and a I2C communication interface with $400kHz$.

A challenge with sensorless commutation is the necessary minimum speed to detect the position of the motor. The start up is done open loop in a stepping mode to accelerate the motor. The loop is closed afterwards to control the commutation. As soon as the motor runs in the closed loop mode the current is regulated by the PWM that is commanded by the central controller board. The commutation loop maintains a synchronization between the motor and the electrical commutation.

The I2C Routine and the PWM-Update run on lower priority than the commutation, which is very time critical task. In the worst case, the PWM is updated latest on every commutation. Our motors run between 2000 and 8000 rpm. The motors have two electrical commutations per mechanical commutation. Thus the worst case deadtime from the sensor measurement to the torque change is:

$$t_{wc} = 1ms + \frac{1}{2*2000}s = 1250\mu s$$

The other advantage of our brushless controllers is the optimization for the low-rpm optimized brushless outrunner motors.

4) 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. In figure 4 you can see measurements of voltage, current, RPM and power per thrust with our motor and the X-UFO propeller.

IV. 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.

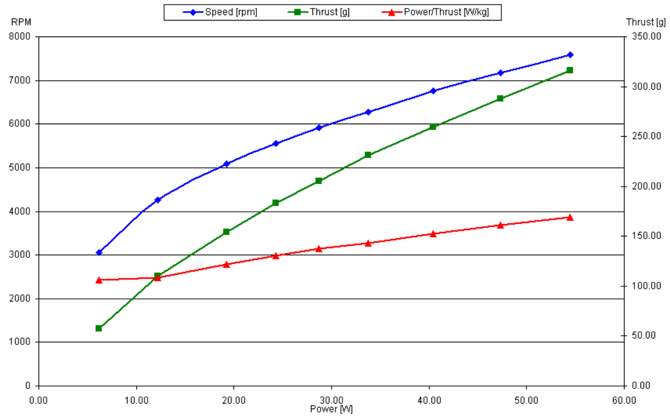
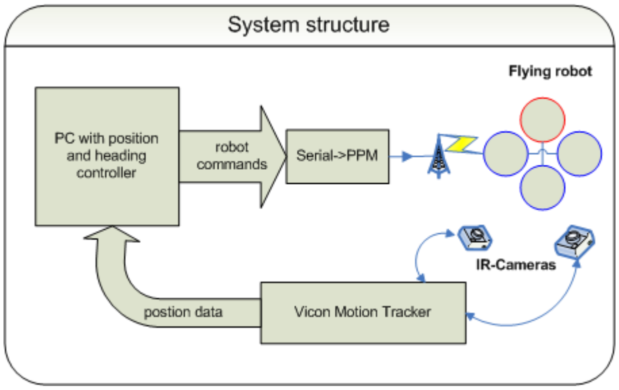


Fig. 4. Motor measurements



A. Autonomous flight using a motion capture 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 robot indoors. We use an external sensor system that is reliable indoors—a motion capture system that uses a system of cameras to compute position information.

1) Experimental Setup: The autonomous flight control experiments 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 motion tracker’s origin and φ 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 5. 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.

2) Position control: The laptop receives the datastream from 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 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 5 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. By highly

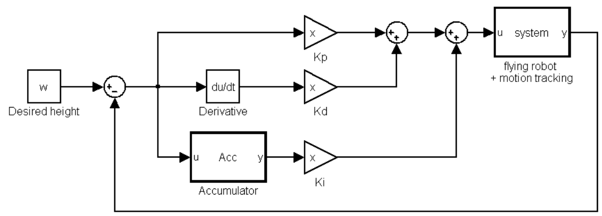


Fig. 5. Controller structure of the height controller

weighting the accelerations, we achieve ”predictive” behavior in this controller, much like a human pilot controlling this system would have. The outer controller cascade is a PD-controller whose output is the desired speed for travel to the desired position. Figure 6 shows the structure of the X and Y position controllers.

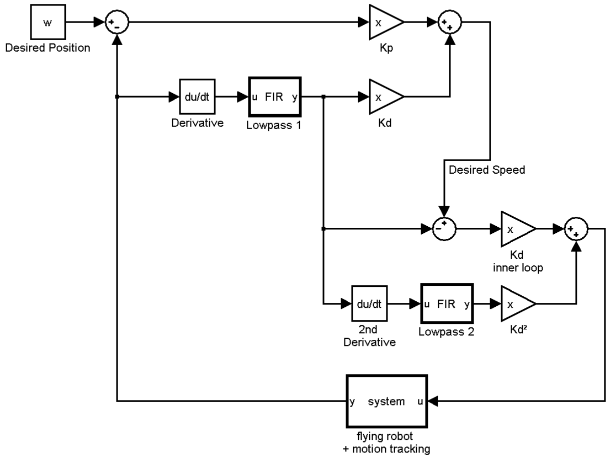


Fig. 6. Controller structure of the X and Y position controllers

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.

B. Results

To demonstrate the performance of our system we collected data from several flights using the motion capture system.

1) Hovering accuracy: In the first experiment the flying robot was commanded to maintain 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 and Y axes and ± 4cm in Z axis and is within ± 1 degree in φ.

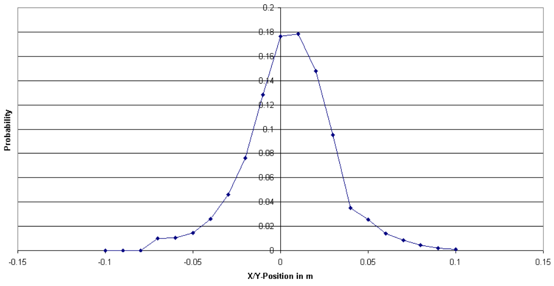


Fig. 7. Probability for X/Y-Positions trying to stay at X = Y = 0m.

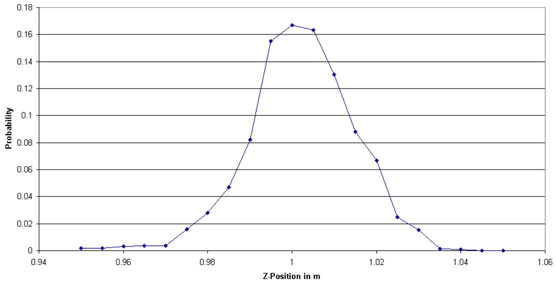


Fig. 8. Probability for an actual height Z at desired Z = 1m.

2) 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 10 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.

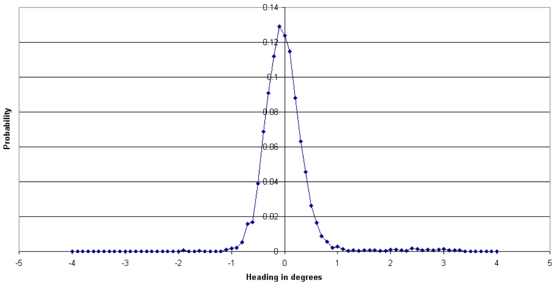


Fig. 9. Probability for an actual heading φ at desired heading $\varphi = 0$.

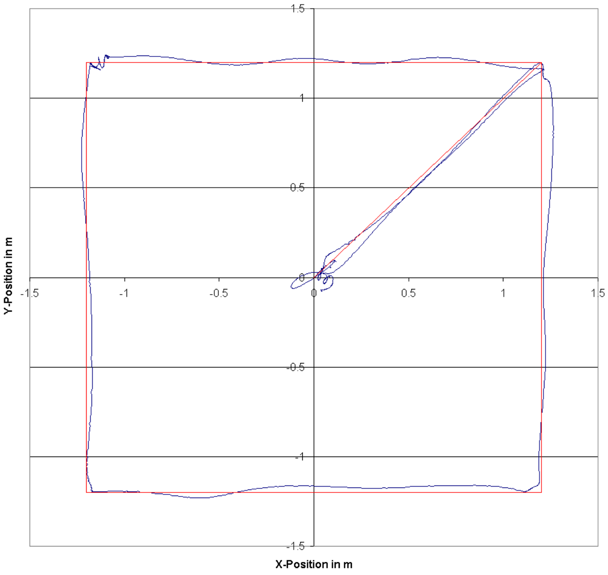
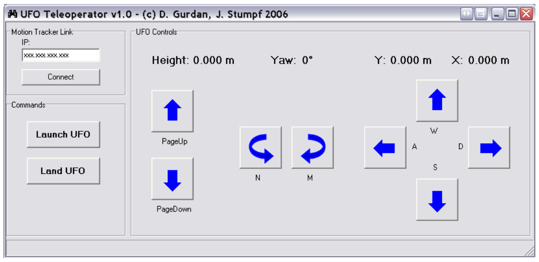


Fig. 10. Flying robot following a trajectory.

3) Telepresence experiments: We implemented a UDP client to control the flying robot over the Internet. A webcam and videophone software were used for visual feedback. Figure 11 shows the remote control software. The remote pilot is able to command the desired location of the flying robot within the volume of a cube of $2.4m \times 2.4m \times 1.2m$. The remote pilot may also set an arbitrary heading. The X and Y position controllers are mapped to the robot's pitch and roll axes so that the pilot does not have to consider the robot's current heading. The robot will always travel to the right hand side of the webcam image if the pilot presses the right arrow. Five different test pilots located in Germany controlled the flight of this robot in the Holodeck Lab at MIT. Since all potentially unstable movements and positions of the robot are prohibited by software, there is nothing the pilot can do wrong. The delay caused by the internet transmission was short enough to not be considered disturbing by any of the remote pilots.


UFO Teleoperator v1.0 - (c) D. Gurdan, J. Stumpf 2006

Motion Tracker Link

IP: xxx.xxx.xxx.xxx

UFO Controls

Height: 0.000 m Yaw: 0° Y: 0.000 m X: 0.000 m

Fig. 11. Internet based control software for the autonomous flying robot.

We further tested the adaptation ability of this controller. We displaced the hovering robot by 1 meter by pulling on a rope attached to the platform. The robot returned to its hovering position after overshooting just once.

V. Conclusions and Future Work

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 mobile sensor 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.

VI. Acknowledgements

We are grateful to Prof. Jovan Popovic and Eugene Hsu for their support with using the motion capture system. We are also grateful to Iuliu Vasilescu and Carrick Detweiler for technical support during the development and experiments. Support for this work has been provided in part by a MURI grant.

References

[1] M. Valenti, B. Bethke, G. Fiore, J. P. How and E. Feron, *Indoor Multi-Vehicle Flight Testbed for Fault Detection, Isolation, and Recovery*, AIAA 2006

[2] P. Pounds, R. Maloy, P. Hynes and J. Roberts, *Design of a Four-Rotor Aerial Robot*, Australian Conference on Robotics and Automation, 2002

[3] P. Pounds, R. Mahony, J. Gresham, P. Corke, and J. Roberts, *Towards Dynamically-Favourable Quad-Rotor Aerial Robots*, Australian Conference on Robotics and Automation, 2004

[4] G. Hoffmann, D. G. Rajnarayan, S. L. Waslander, D. Dostal, J. S. Jang, C. J. Tomlin, *The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (STARMAC)*, DASC 2003

[5] S. D. Hanford, L. N. Long, and J. F. Horn,*A Small Semi-Autonomous Rotary-Wing Unmanned Air Vehicle (UAV)*, AIAA 2005

[6] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger and E. Liang, *Inverted autonomous helicopter flight via reinforcement learning*,International Symposium on Experimental Robotics, 2004

[7] K. Harbick, J. F. Montgomery, G. S. Sukhatme, *Planar Spline Trajectory Following for an Autonomous Helicopter*, Journal of Advanced Computational Intelligence - Computational Intelligence in Robotics and Automation, Vol. 8, No. 3, pp. 237-242, May 2004.

[8] M. Jun, S. I. Roumeliotis, G. S. Sukhatme, *State Estimation of an Autonomous Helicopter using Kalman Filtering* IROS, 1999.

[9] G. Buskey, J. Roberts, P. Corke, G. Wyeth,*Helicopter Automation using Low-Cost Sensing System*, Australian Conference on Robotics and Automation, 2003

[10] V. Gavrilets, *Autonomous aerobatic maneuvering of miniature helicopters.*, Ph.D. thesis, Massachusetts Institute of Technology, Boston, MA, 2003

[11] M. La Civita, *Integrated modeling and robust control for full-envelope flight of robotic helicopters.*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, 2003

[12] Draganfly Innovations, RcToys, *http://www.rctoys.com*

[13] Atmel Corperation, *http://www.atmel.com*

[14] Murata Manufacturing Co.,Ltd., *http://www.murata.com*