2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

On-Board Object Tracking Control of a Quadcopter with Monocular Vision

Alex G. Kendall, Nishaad N. Salvapantula, Karl A. Stol, *Member, IEEE*

Abstract— **The development of an object tracking controller for a quadcopter using an on-board vision system is presented. Using low-cost components, a novel system is introduced that operates entirely on board the quadcopter, without external localization sensors or GPS. A low-frequency monocular computer vision algorithm is applied in closed-loop control to track an object of known color. Parallel PID controllers for aircraft bearing, relative height and range are implemented with feedback from object offset and size in the image frame. The noise exaggerated by measuring range from object pixel area is mitigated with a Kalman filter. Stable closed-loop tracking is demonstrated experimentally for all three control axes when tested individually and coupled together. Individual settling times were under 10 seconds and coupled control settling times under 25 seconds.**

I. Background and Motivation

Unmanned Aerial Vehicles (UAVs) have gone through a global boom in recent years due to their increasing accessibility. Quadcopters in particular, with four rotors, have favorable characteristics for many applications such as infrastructure inspection, asset monitoring and surveillance, due to their ability to hover and take-off and land vertically. Events such as the 2011 Christchurch earthquake have benefited from their use inside dangerous seismic locations.

However, UAV localization remains a significant challenge, especially in confined or GPS denied locations. Positioning the aircraft relative to another object is a common task that is difficult to achieve. Here, an autonomous quadcopter controller is presented that is able to track and maintain a desired relative location to a static object. This is achieved with a monocular vision system with the entire computational processing completed on board the UAV. As a consequence, the UAV is designed to operate in infrastructure free environments, providing increased efficiency, autonomy and safety.

Visual control of small-scale UAVs in the past has been limited by real-time computer processing performance. Much research has therefore focused on off-board processing by a ground station with wireless control of the UAV. One example is the quadcopter presented in [1] that relies on visual markers and wireless transmission of camera images to a ground computer. The computer then remotely controls the UAV. Ground station control of quadcopters with other sensors such as a laser scanner has also been demonstrated [2].

On-board visual closed-loop control was initially explored with flight stabilization objectives. Optical flow techniques such as [3] apply themselves well to stabilization as they measure relative motion and are computationally lightweight. Other monocular visual stabilization techniques are demonstrated in [4] and [5]. An overview of hardware requirements for on-board image processing with emphasis on stabilization is given in [6].

Subsequent research has achieved on-board visual tracking of objects, but restricted to overhead translational control. [7] and [8] present overhead tracking, with an application to assist landing and take-off actions. On-board monocular visual systems have also been used on UAVs for simultaneous localization and mapping (SLAM) with a helicopter [9].

Similar research has also been conducted with stereo camera systems which have a more significant computational requirement. Initial off-board ground mounted stereo cameras were used in [10] and [11], focusing on a quadcopter to remotely control it from a ground computer. In [12], a forward-facing stereo camera is mounted on a quadcopter to achieve closed-loop control, again with processing on a ground station.

Only very recently, UAVs have emerged which are able to perform stereo processing on board. The UAV presented in [13] features a forward-facing stereo camera for obstacle avoidance, however still requiring visual markers for navigation. The UAV in [14] has two sets of stereo cameras for robust on-board control.

The approach used in this paper is unique in the fact that it presents closed-loop object tracking control with a low-cost on-board monocular vision system and a simply defined target object. The paper demonstrates the feasibility of this approach and issues relating to dynamic coupling.

Alex Kendall is an undergraduate student in the Department of Mechanical Engineering, University of Auckland (email: alexgkendall@gmail.com).

Nishaad Salvapantula is an undergraduate student in the Department of Mechanical Engineering, University of Auckland (email: nishaad.sn@gmail.com).

Dr. Karl Stol is a senior lecturer with the Department of Mechanical Engineering, University of Auckland (email: k.stol@auckland.ac.nz).

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

II. Object Tracking Methodology

The task of object tracking for a quadcopter requires regulation of relative position from the object. We assume the object is stationary and unique; detecting and tracking a dynamic object (i.e. object following) is the natural extension of this work.

The coordinates used to define motion of the quadcopter in flight are illustrated in Fig. 1. The coordinates $\phi,\ \theta,$ and ψ are the roll, pitch, and yaw angles respectively of the body-fixed frame $\{x,\ y,\ z\}$ with respect to the inertial frame $\{x_o,\ y_o,\ z_o\}$. The x-axis is aligned with the optical axis of a forward-facing camera. The features of the object that are extracted from the image are its centroid coordinates (x_{offset} and y_{offset}) and pixel area. Components of the object's relative position, $\boldsymbol{r}$, in the quadcopter reference frame can then be determined as follows:

- Object yaw error, ψ_r (degrees), from the horizontal offset of the object in the image frame, x_{offset} (pixels);
- Object height error, r_z (m), from the vertical offset of the object in the image frame, y_{offset} (pixels);
- Object range error, r_x (m), from the object's size in the image frame (pixels^2).

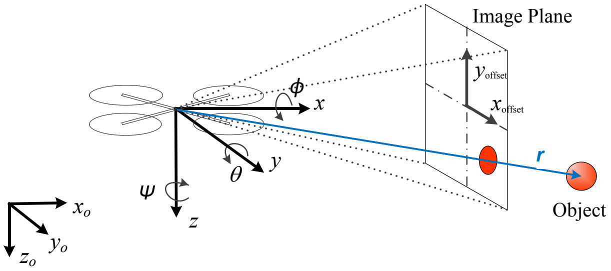

Fig. 1. Quadcopter coordinates.

III. Aircraft Design

The control system was tested on a quadcopter shown in Fig. 2 with characteristics in Table 1.

TABLE 1
QUADCOPTER SPECIFICATIONS

Diameter	800mm
Weight	1.7kg
Max Thrust	45N
Endurance	10 minutes
Total Cost	US$350

Closed-loop flight control was achieved with two processors. A Raspberry Pi microcomputer was used to implement the computer vision algorithm. The Raspberry Pi is a low-cost, credit-card sized computer with a processing speed of 700MHz. Flight control was achieved using an APM2.5 board with an ATMega2560 processor [15].

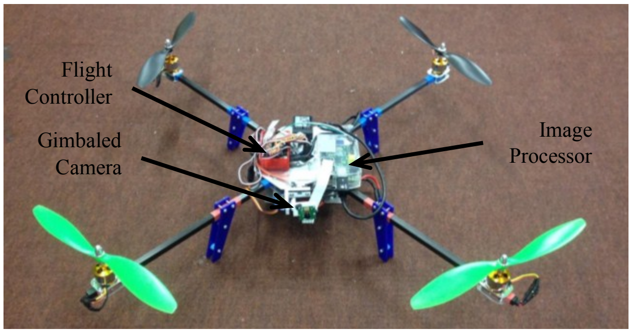

Fig. 2. Quadcopter UAV and control system hardware.

A two degree-of-freedom camera gimbal was designed to stabilize the camera in pitch and roll axes. If the camera was mounted rigidly to the quadcopter, the projected image plane would be affected as the quadcopter rolls or pitches. This would result in false measurements of the object's location relative to the aircraft. Therefore, it was essential to compensate for the quadcopter's attitude. Mechanical compensation, with a camera gimbal, was used over software compensation as it was both faster and retained the entire field of view of the camera pointing forward. As the camera was mounted at an offset to the quadcopter's center of gravity then there will be a small translation from attitude changes.

IV. On-board Vision System

A Raspberry Pi native camera sensor was used. The image was down scaled from $1024\ \times\ 768$ to 320×240 pixels to decrease computation cost, which was found to be an effective resolution within the designed operating range. The field of view was experimentally measured to be 31 degrees horizontally and 24.2 degrees vertically. The camera has no significant image distortion such as barrel or fisheye distortion. From measured optical properties, the relationship between pixel area and object distance for the chosen object was found to be $r_x = \frac{82}{\sqrt{a}}$ where a is the area (in pixels^2) of the object.

The Connected Component Labeling algorithm was used for object detection, which is a very efficient blob finding algorithm. It was implemented on the Raspberry Pi using the OpenCV library [16]. The output of the function is the coordinates of the blob centroid and the blob area. A bright orange 20cm diameter ball was chosen as the tracked object, having a uniform color, distinct in the test environment. Furthermore, a spherical object maintains its frontal area with change in perspective. HSV color coordinates were used as they are robust to lighting changes, Table 2.

TABLE 2
TUNED HSV THRESHOLDS FOR ORANGE BALL

	Minimum	Maximum
Hue	0%	8.2%
Saturation	81.6%	100%
Value	56.6%	100%

The frame rate of the processing was recorded in different configurations and compared in Table 3. There is some variability due to the difference in computational requirements between images. Increased complexity and number of other objects in the scene increases processing time. In all stationary tests, the image processor had a 100% object detection rate with a non-conflicting background and a frame rate variability of approximately 5%. This demonstrates the cost of computation on an on-board low cost system achieving 48% performance when compared to a laptop.

TABLE 3
IMAGE PROCESSOR AVERAGE FRAME RATES

Processor	Speed
Raspberry Pi Command Line Operation	8.1 Hz
Raspberry Pi full OS and GUI	4 Hz
Laptop - Intel I5 Processor	17 Hz

The latency in the transportation delay due to the camera interface was also measured and was found to be 0.054 seconds. Therefore the total delay of the 8.1Hz operation and transportation delay combined was 0.18 seconds.

The vision system's robustness was tested under three different lighting conditions with the luminosity measured for each situation: sunlight, bright indoor lighting and dim indoor light. The vision system area measurement results are shown in Fig. 3 with a representative example of each lighting condition shown. From the results in Fig. 3 we conclude that our vision system is robust and can be expected to operate in indoor lighting environments, and outside in sunlight with similar results. Only when the light becomes very dim will the performance start to decrease.

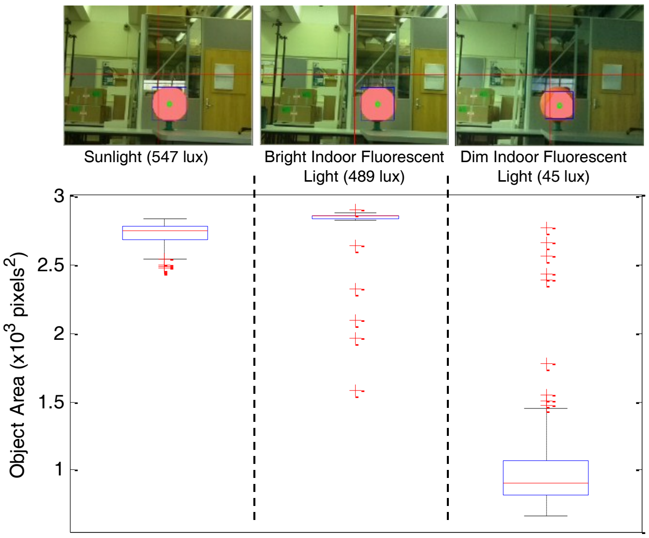

Fig. 3. Object area at 5m under varying light conditions (100 samples).

V. CONTROLLER DESIGN

The object tracking controller was designed with two nested loops to incorporate feedback from both the vision system and inertial measurement unit (IMU). It was found experimentally with computer vision feedback alone at 8Hz, the system was unstable due to the transportation delay. Stability is achieved with the control system in Fig. 4 operating at 100Hz. The quadcopter plant consists of a control signal sent to electronic speed controllers, which regulate the voltage to the four rotors. The PID controllers were implemented in discrete-time form (1), where $T = 0.01$ seconds.

$$u(t_k) = K_p e(t_k) + K_i \sum_{j=0}^{j=k} T\, e(t_j) + K_d \frac{\left(e(t_k) - e(t_{k-1})\right)}{T} \qquad (1)$$

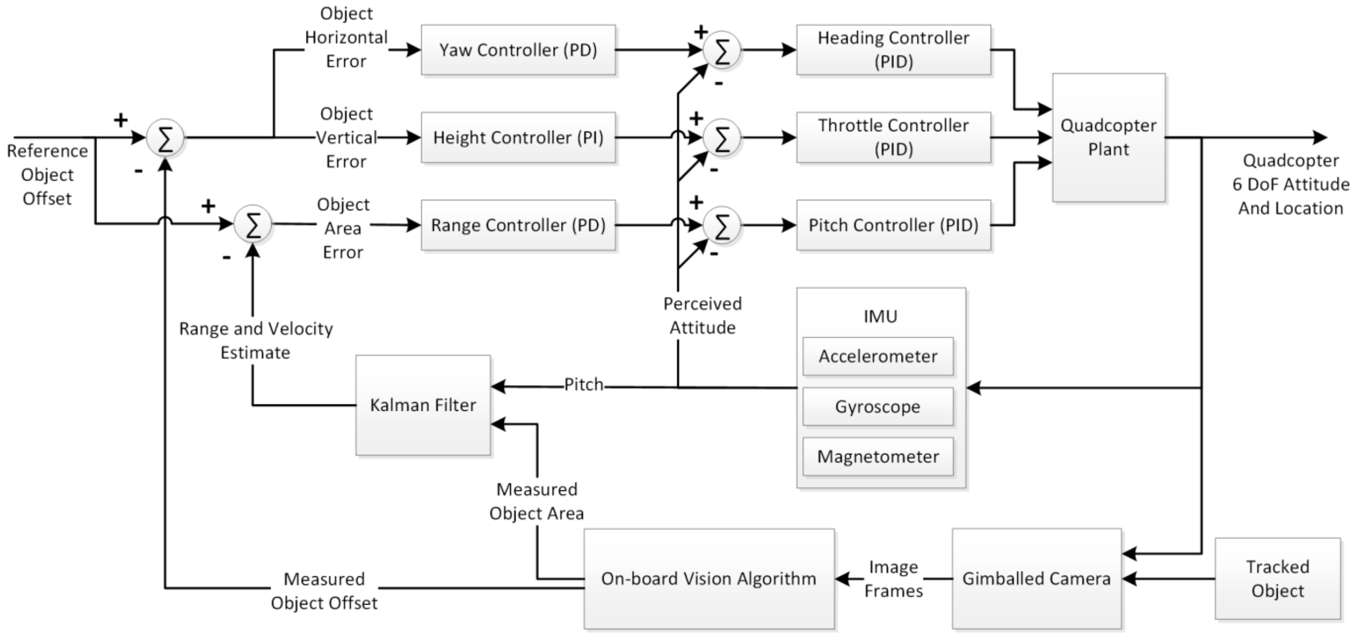

Fig. 4. Object tracking control system.

A. *Ground Truth System*

A NaturalPoint OptiTrack external tracking system was used to measure the quadcopter's ground truth motion. The system comprises six infrared cameras placed surrounding the flight testing area as shown in Fig. 6. The camera system was able to ascertain the quadcopter's position to 0.5mm accuracy. The cameras were arranged to give an effective corridor for testing of approximately 7×3×2m. The following conditions were observed for all testing.

- The quadcopter remained above ground effect (approximately 0.5m altitude) and commenced each run with negligible velocity.
- The object (orange ball) remained stationary.
- The background contained no significant orange objects to interfere with vision tracking system.
- Each test was completed five times.

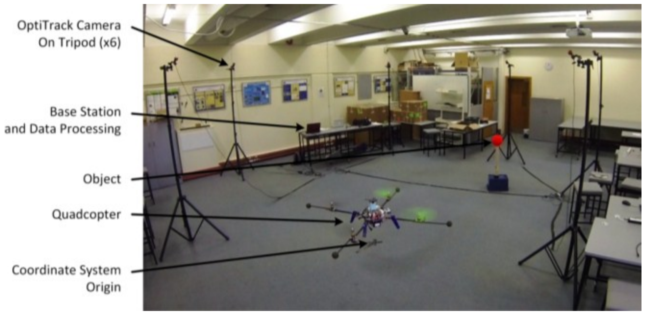
OptiTrack Camera On Tripod (x6)
Base Station and Data Processing
Object
Quadcopter
Coordinate System Origin

Fig. 6. External camera system experimental setup.

B. *Range Controller*

Fig. 5 shows the raw range measurement from the computer vision system. The ground truth measurement is also plotted for comparison. It shows that the on-board camera system provides very noisy raw data, which is not suitable for direct feedback.

A Kalman filter was designed to optimally fuse data from the IMU and computer vision system to improve range and velocity measurements. By obtaining better measurements of position and velocity a proportional-derivative (PD) controller was able to be implemented. An integral term was not introduced as it was assumed that disturbances were not large or persistent. This assumption would need to be reconsidered if outdoor flight was attempted in the future. The Kalman filter assumes the target object is stationary. The system was modeled with linear aerodynamic damping and linearized around small pitch angles in (2).

$$\ddot{r_x} = k\theta - c\dot{r_x} \tag{2}$$

Noise is introduced to the system through process noise, v, in measurement of pitch attitude from the IMU, and measurement noise, w, from the inaccuracies of the vision system. Therefore the system can be modelled with (3) and (4) with state vector $\underline{x} = [r_x \quad \dot{r_x}]^T$, input as pitch angle $u = \theta$, and output $y = r_x$.

$$\underline{\dot{x}} = \begin{bmatrix} 0 & 1 \\ 0 & -c \end{bmatrix} \underline{x} + \begin{bmatrix} 0 \\ k \end{bmatrix} u + \begin{bmatrix} 0 \\ k \end{bmatrix} v \tag{3}$$

$$y = [1 \quad 0]\underline{x} + w \tag{4}$$

It was assumed that the properties of the system noise was constant; v and w were obtained from statistical analysis of experimental data. The time-invariant Kalman filter was discretized with period $T = 0.01$ seconds and implemented with (5).

$$\underline{\hat{x}}_{k+1} = (A_d - K_e C)\underline{\hat{x}}_k + B_d u_k + K_e y \tag{5}$$

The IMU data is available at 100Hz, however the camera sensor measurement data is only available at 8Hz. A multi-rate recursive Kalman filter was explored in simulation but was found to offer negligible improvement in performance.

The filter was tuned in simulation off line using experimental data logged from the aircraft. It was found to be very sensitive to model parameters but less sensitive to changes in covariance matrices. The filter's performance was quantified using RMS error measurements, as well as considering the proportion of the time that the speed estimate had the correct sign.

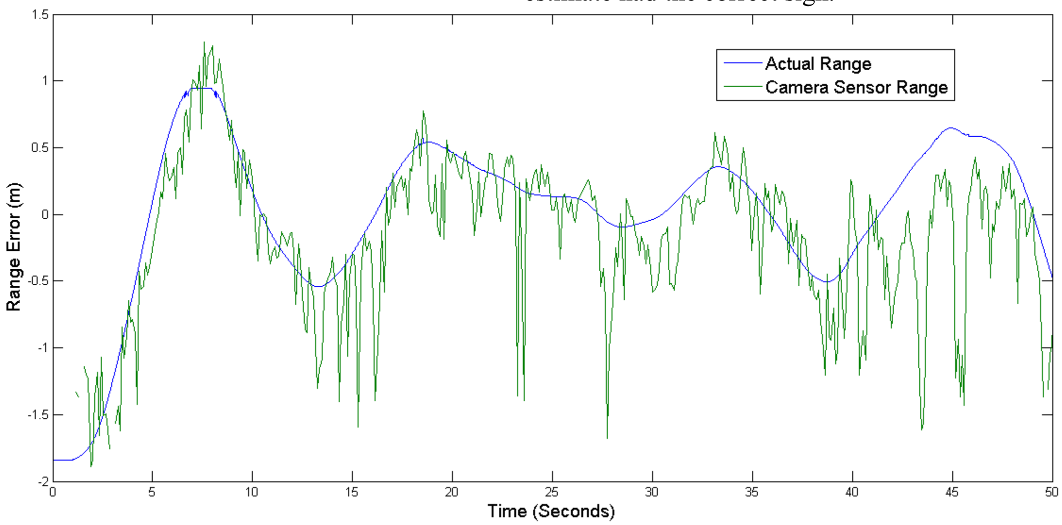

Fig. 5. Range data to illustrate sensor noise.

The filter was relatively robust to changes in angular bias on the IMU measurement with a limit of +/- 1.5 degrees. Biases greater than these limits significantly reduced the filter's performance. These limits are of the same magnitude of what is to be expected in a quadcopter system and therefore acceptable.

The Kalman filter was finally tuned in open-loop in flight. A typical response is shown in Fig. 7. Compared with a low-pass filter on the raw camera measurement, the Kalman filter output has less lag in position and is more accurate in speed.

C. *Range Controller Performance*

The range controller was designed to maintain a desired distance from the object and a PD control law was initially tested. An issue was identified when the object was located at the extremities of the camera's field of view due to transient movement of the quadcopter. In this situation, the image border would intersect the image. Horizontal and vertical offset readings are still valid however the area measurements become compromised. Partial area of the object may be out of frame, giving a false reading that the object is smaller than reality. This will result in an incorrect control effort to move towards the object. Therefore, should the object intersect the edge of the camera's frame, the last area reading was used by the controller.

For experimental testing, a safe reference distance of $r_{x_ref} = 4$m from the object was chosen. The quadcopter's initial relative position was $r_x(0) = 7$m. A sample transient response is shown in Fig. 8 and averaged performance metrics (over 5 runs) are listed in Table 4.

TABLE 4
RANGE CONTROLLER PERFORMANCE

Steady State Oscillation Amplitude	0.30m
Steady State Error	0.18m
Settling Time within Steady State Oscillation	7.6s
Overshoot	8%

The range controller exhibits steady state error of 0.18m. This is because small angular biases exist in the IMU measurement causing the pitch controller to not be perfectly trimmed. However, this error is small and within the control objectives, therefore an integral term was not introduced.

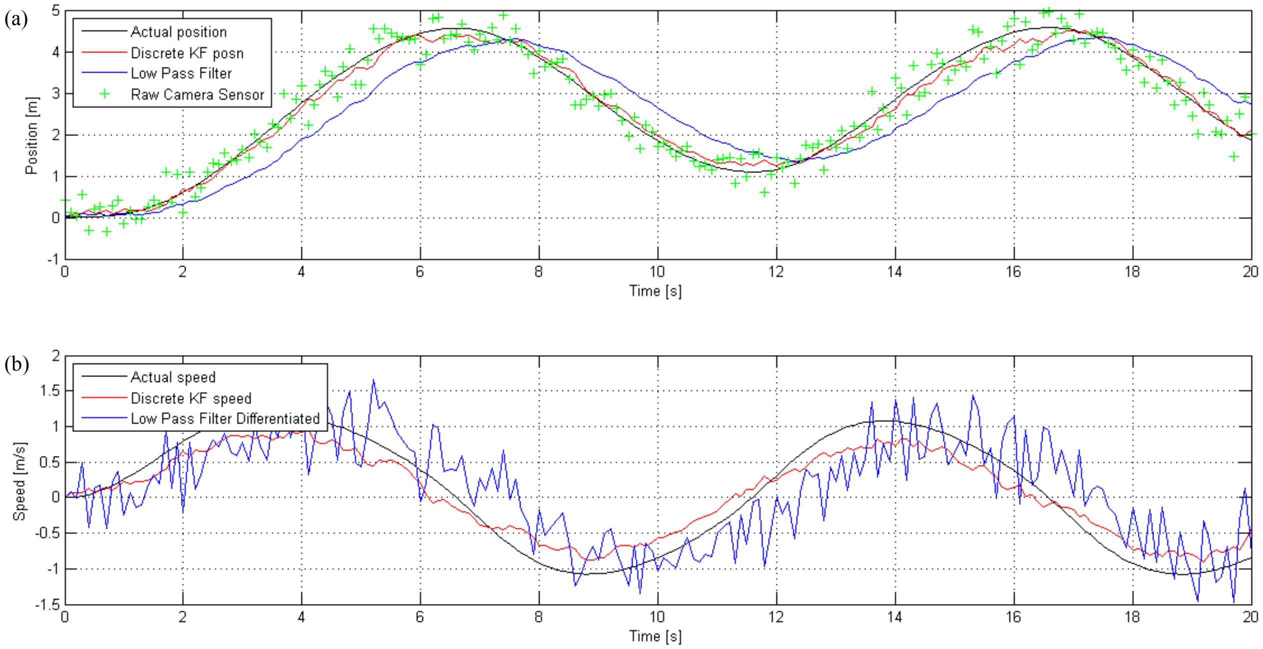

Fig. 7. Range (a) and closure speed (b) data while flying manually.

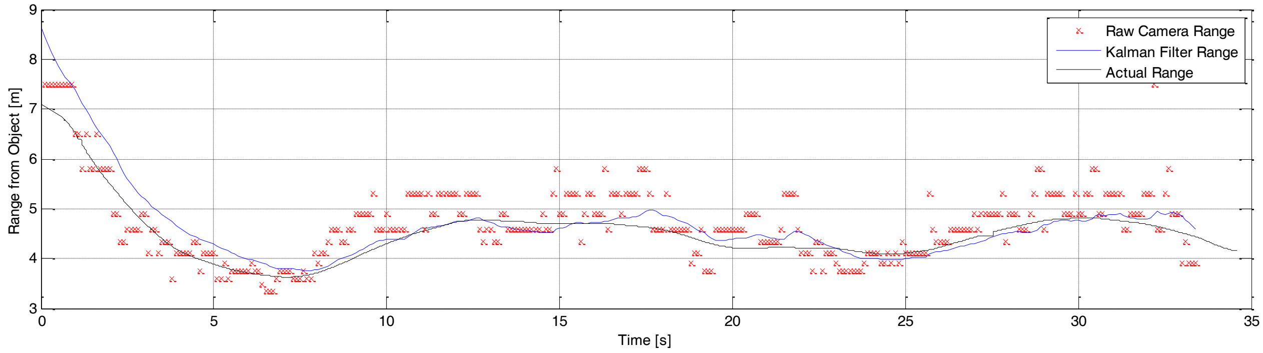

Fig. 8. Typical range controller response.

Variations in performance that were observed across all tests were due to the following observations. Firstly, the tests were completed at a variety of states of battery levels. The lithium polymer batteries operate at nominally 12.6V, decreasing to 10.5V at low charge. This variation in voltage will affect the actuator dynamics of each control system, and therefore the overall performance. Secondly, variation was caused by IMU and vision sensor noise and perception of the environment. Finally, small variations in initial conditions existed.

D. Yaw Controller

The yaw controller regulates the quadcopter's bearing (yaw error) from the target object by setting yaw rate commands. A PD control law was selected; no integral term was necessary because in yaw, the control system is Type I, i.e. zero steady-state error expected to step inputs.

The horizontal object offset signal from the computer vision sensor is converted to yaw angle and differentiated to find yaw rate. The sensor exhibited low noise content, as seen in Fig. 9, and therefore did not require any filtering.

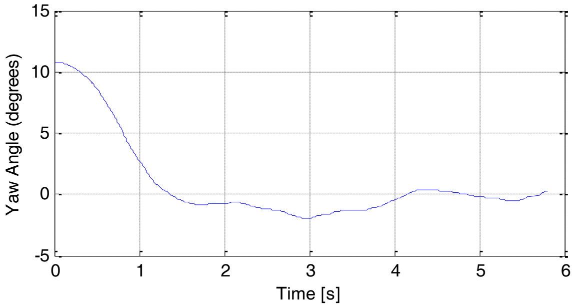

Fig. 9. Typical yaw data (object bearing) during flight.

E. Yaw Controller Performance

The yaw controller was tuned experimentally with the range and height controllers in open-loop; pitch and thrust were manually controlled. The controller was tested with an initial bearing on $\psi_r(0) = 12°$.

Typical transient responses are shown in Fig. 10 for three sets of control gains. Adding a derivative term halved the average settling time from 7 seconds to 3 seconds. The non-zero steady-state error for $\{Kp = 2, Kd = 0\}$ can be attributed to a small bias when calibrating the external tracking system and is insignificant. The gain set $\{Kp = 2, Kd = 0.5\}$ was found to be most suitable due to the resulting low amplitude oscillation and settling time. Average performance measures for this set are given in Table 5.

TABLE 5
YAW CONTROLLER PERFORMANCE

Steady State Oscillation Amplitude	1.51°
Steady State Error	0°
Settling Time within Steady State Oscillation	1.6s
Overshoot	11.6%

F. Height Controller

The height controller provides regulation of the quadcopter's relative height from the target object by controlling its vertical thrust (throttle setting). The vertical offset measurement from the computer vision sensor was converted to relative height from the object.

A proportional-integral (PI) controller was selected, allowing the integral term to account for the thrust required to balance weight, as this will change over time due to changing battery voltage. The integral term was initialized to the current throttle setting when the controller was activated. Assuming the quadcopter was in a hover then this is a good estimate of the equilibrium point thrust. The height data from the vision system is similar to the yaw data shown in Fig. 9 and is relatively noise free. A derivative term could have been added, however, this was not explored.

G. Height Controller Performance

The height controller was tested similarly to the yaw and range controller with non-zero initial conditions. The yaw and range controllers remained in open-loop; roll and pitch

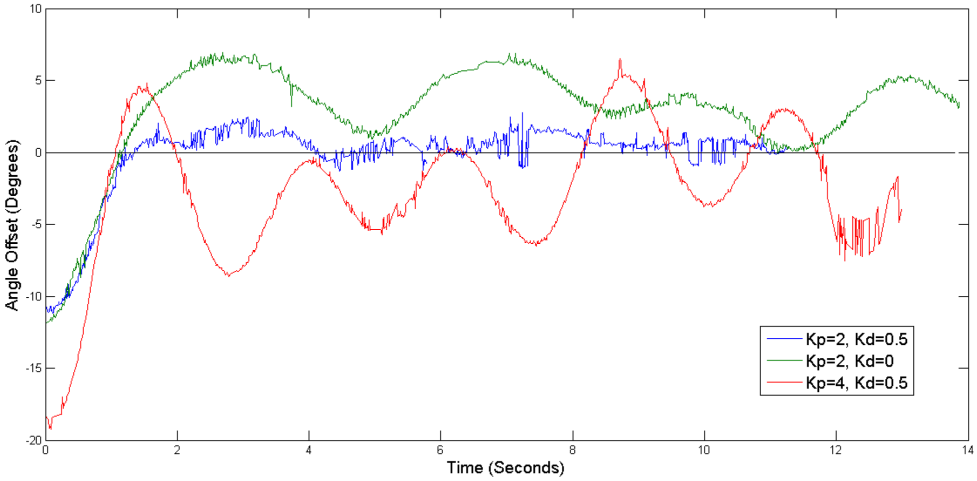

Fig. 10. Typical yaw control response for tested gains.

were manually controlled. Two sets of initial conditions were tested to determine if symmetry exists in the system and whether different dynamics can be expected between rising and falling height control responses: $r_z(0) = 1\text{m}$ and $r_z(0) = -0.75\text{m}$. The controller appeared to be symmetrical as results for rising and falling responses were statistically similar. Combined results are given in Table 6.

TABLE 6
HEIGHT CONTROLLER PERFORMANCE

Steady State Oscillation Amplitude	0.1m
Steady State Error	0.05m
Settling Time within Steady State Oscillation	10s
Overshoot	0%

VI. COMBINED YAW, RANGE AND HEIGHT CONTROLLER PERFORMANCE

In this final case, all three controllers were implemented simultaneously to regulate yaw, range and height using the independently tuned gains. Initial conditions were:

$$\psi_r(0) = 16°,$$
$$r_x(0) = 7\text{m}, \quad r_{x\_ref} = 4\text{m},$$
$$r_z(0) = 1\text{m}.$$

For all 5 trials, no unstable behavior was observed. Fig. 11 shows typical transient responses and average performance results are summarized in Table 7. From basic flight dynamics, it can be deduced that regulating the range by varying the pitch attitude will have an effect on the vertical component of the thrust force. Therefore, when the range and height controllers are coupled together, they act as an extra disturbance to each other. This dynamic coupling causes the decrease in control performance in the coupled situation.

There is also a coupling effect in yaw, observed from the decrease in control performance. This is because when yaw acceleration occurs, the nominal motor thrust level changes. The thrust relationship is non-linear, and assumed to be linearized around a constant operating point. By changing this operating point it acts as an extra disturbance to the other controllers causing different behavior and ultimately longer settling time due to this extra disturbance regulation. On average, coupled settling time increased by 240% compared to individual axis control, illustrated in Fig. 12. Due to the dynamic coupling demonstrated between control axes, the controller could be improved by using a multivariable approach or by further tuning of the coupled system.

The combined controller was also tested to follow a dynamic object and was observed to be successful in tracking at a walking pace of approximately 1m/s. The controller is therefore robust to a level of uncertainty but could be extended to accommodate a dynamic object in outdoor settings.

TABLE 7
YAW, RANGE AND HEIGHT COUPLED CONTROLLER RESULTS

Control Axis	Yaw	Range	Height
Steady State Oscillation Amplitude	1.48°	0.6m	0.18m
Steady State Error	0.81°	0.48m	0.27m
Settling Time within Steady State Oscillation	3.1s	23.7s	21s

VII. CONCLUSIONS

A low-cost, real-time object tracking system was successfully implemented on board the quadcopter using separate processors for both flight control and image processing. The on-board, monocular vision system was used to measure object offset and size in the image frame to

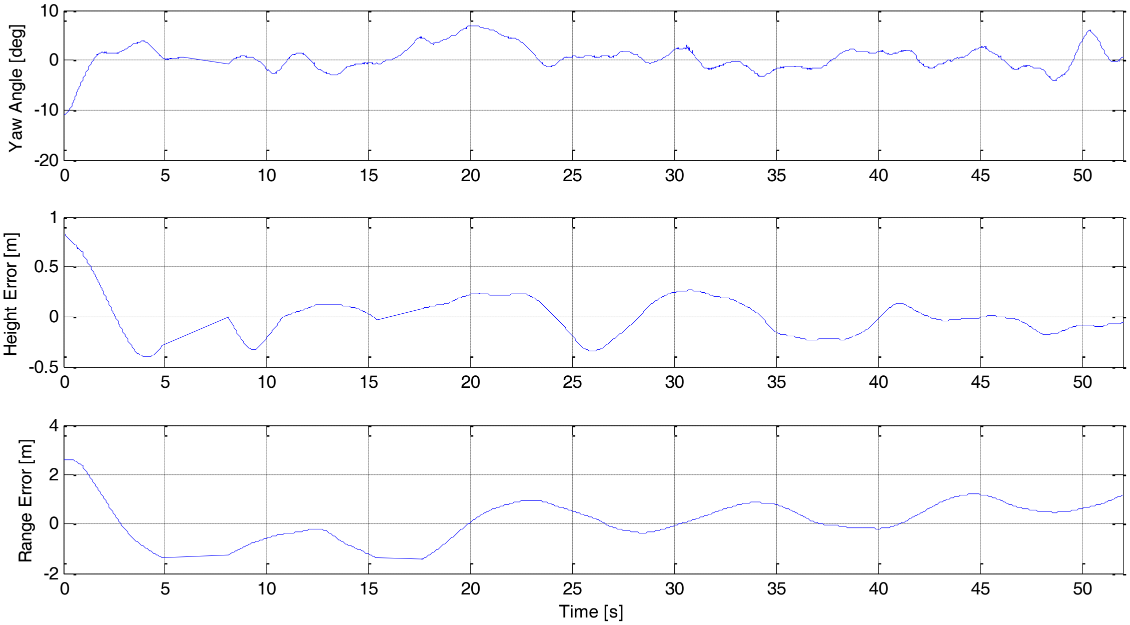

Fig. 11. Typical time responses of the yaw, height and range controllers when combined.

ascertain relative location from a static object in a range of lighting conditions.

Control of all flight axes was achieved with parallel PID loops. Range control required a Kalman filter in order to add damping. The quadcopter control system gains were tuned to achieve state regulation at steady-state within 0.18m for height; 0.6m for range; 1.5° for yaw. There was a statistically longer response time for the controllers when combined together by an average factor of 240%, suggesting a multivariable control design may yield a better response. An extension of the controller is being developed for dynamic object tracking.

REFERENCES

[1] Achtelik, Markus, Abraham Bachrach, Ruijie He, Samuel Prentice, and Nicholas Roy. "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments." SPIE Defense, Security, and Sensing. International Society for Optics and Photonics, 2009.

[2] Sa, Inkyu, and Peter Corke. "System identification, estimation and control for a cost effective open-source quadcopter." IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2012.

[3] Grabe, Volker, H. H. Bulthoff, and Paolo Robuffo Giordano. "On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow." IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2012.

[4] Yigit, Cihat Bora, and Erdinc Altug. "Visual attitude stabilization of a unmanned helicopter in unknown environments with an embedded single-board computer." IEEE International Symposium on Robotic and Sensors Environments (ROSE). IEEE, 2012.

[5] Fahimi, Farbod, and Karansingh Thakur. "An alternative closed-loop vision-based control approach for Unmanned Aircraft Systems with application to a quadrotor." International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, 2013.

[6] Konomura, Ryo, and Koichi Hori. "Designing hardware and software systems toward very compact and fully autonomous quadrotors." IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, 2013.

[7] Deng, Haibo, Xiaoguang Zhao, and Z. G. How. "A vision-based ground target tracking system for a small-scale autonomous helicopter." Fifth International Conference on. IEEE Image and Graphics (ICIG'09), 2009.

[8] Yang, Shaowu, Sebastian A. Scherer, and Andreas Zell. "An onboard monocular vision system for autonomous takeoff, hovering and landing of a micro aerial vehicle." Journal of Intelligent & Robotic Systems 69.1-4 (2013): 499-515.

[9] Çelik, Koray, Soon-Jo Chung, Matthew Clausman, and Arun K. Somani. "Monocular vision SLAM for indoor aerial vehicles." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2009.

[10] Achtelik, Markus, Tianguang Zhang, Kolja Kuhnlenz, and Martin Buss. "Visual tracking and control of a quadcopter using a stereo camera system and inertial sensors." International Conference on Mechatronics and Automation (ICMA). IEEE, 2009.

[11] Herisse, Bruno, F-X. Russotto, Tarek Hamel, and Robert Mahony. "Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2008.

[12] Carrillo, Luis Rodolfo García, Alejandro Enrique Dzul López, Rogelio Lozano, and Claude Pégard. "Combining stereo vision and inertial navigation system for a quad-rotor UAV." Journal of Intelligent & Robotic Systems 65.1-4 (2012): 373-387.

[13] Heng, Lionel, Lorenz Meier, Petri Tanskanen, Friedrich Fraundorfer, and Marc Pollefeys. "Autonomous obstacle avoidance and maneuvering on a vision-guided MAV using on-board processing." IEEE international conference on Robotics and automation (ICRA). IEEE, 2011.

[14] Schauwecker, Konstantin, and Andreas Zell. "On-Board Dual-Stereo-Vision for the Navigation of an Autonomous MAV." Journal of Intelligent & Robotic Systems (2014): 1-16.

[15] APM 2.5 Flight Control Board, 3D Robotics, http://3drobotics.com/

[16] OpenCV Library, http://opencv.org/

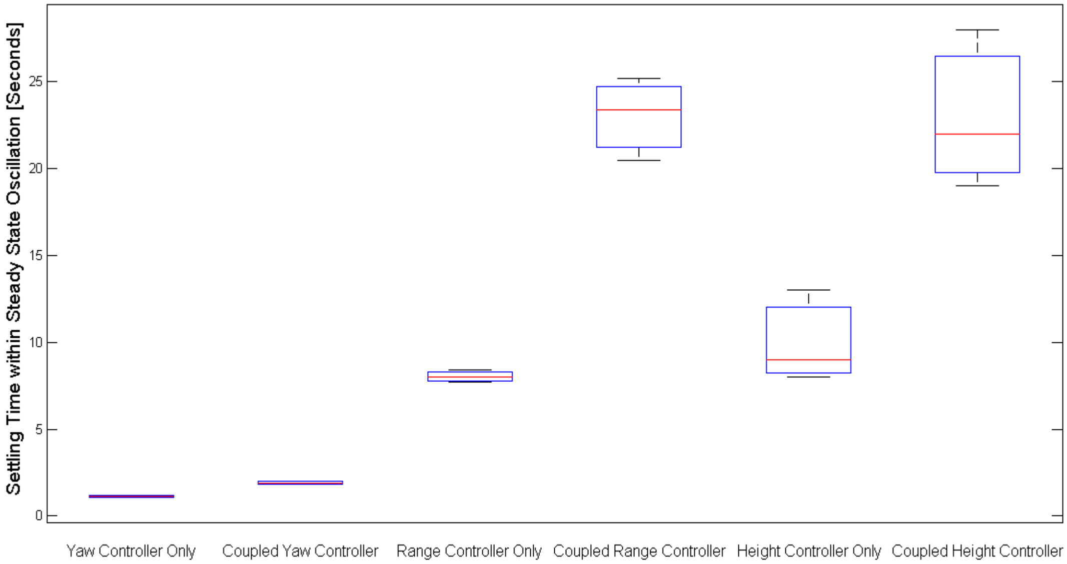

Fig. 12. Individual and coupled controller settling times for identical initial conditions.