System Identification, Estimation and Control for a Cost Effective Open-Source Quadcopter

Inkyu Sa and Peter Corke

***Abstract*— This paper describes system identification, estimation and control of translational motion and heading angle for a cost effective open-source quadcopter — the MikroKopter. The dynamics of its built-in sensors, roll and pitch attitude controller, and system latencies are determined and used to design a computationally inexpensive multi-rate velocity estimator that fuses data from the built-in inertial sensors and a low-rate onboard laser range finder. Control is performed using a nested loop structure that is also computationally inexpensive and incorporates different sensors. Experimental results for the estimator and closed-loop positioning are presented and compared with ground truth from a motion capture system.**

I. INTRODUCTION

Quadcopters are compact rotor craft air vehicles that can be applied to indoor and outdoor inspection and exploration tasks. Like a conventional helicopter they can hover, however, they have other significant advantages such as mechanical simplicity, vertical take off and hover capabilities, and dynamic manoeuvrability. Although quadcopters have become very popular in recent times, the first quadcopters were built in the 1920s and were utilised as passenger aircraft. One of the earliest robotic quadcopters was built by Pounds et al. [1] [2] at a time when it was necessary to build a vehicle from scratch due to the lack of available resources and components. In recent times there has been an explosive growth of interest in quadcopters [3]–[5] spurred by the availability of research-grade platforms.

From a control perspective [6] [7], the quadcopter is an under-actuated force-controlled vehicle with 6 degrees of freedom. Force actuation implies that rotational and translational motion can be approximated as a double integrator system from command to attitude angle or horizontal position. The key to stable control of such systems is to provide artificial damping through feedback of rotational and translational velocity. In classical control system terms, we add a zero to the system dynamics in order to move the system poles into the left-hand half of the s-plane. For rotational motion, this is conveniently achieved by using the angular rates measured by onboard gyroscopes, and this is typically performed at a high rate (500–1000 Hz) in the onboard embedded flight controller. Translational damping is more problematic since there is no translational velocity sensor. One effective approach is differentiating high-rate position information obtained by a motion capture system (eg. Vicon) and this has led to impressive demonstrations of acrobatics, formation control, juggling, manipulation and assembly [8]–[10]. However when the position measurements occur at a low rate, for example from a light-weight onboard laser scanner or camera, the resulting lag in the velocity estimate is highly destabilising.

The authors are with the CyPhy Laboratory, School of Engineering Systems, Queensland University of Technology, Australia. i.sa@qut.edu.au, peter.corke@qut.edu.au

Fig. 1. The Cyphy Lab MikroKopter research platform.

In order for quadcopters to become more pervasive, they must become lower in cost, and to become more agile they must become smaller. Both these trends require a reduction in sensor weight and cost (typically at the expense of sensor performance), and also in computational cost which would allow for a smaller and less power hungry onboard computer.

Toward these goals, the contributions of this paper are as follows. Firstly, we develop a multi-rate and multi-sensor velocity state estimator which compared with others such as [11] is computationally cheap, easy to tune and effective. Secondly, we describe a computationally inexpensive nested control structure. Thirdly we show that a cost-effective high-performance amateur-grade quadcopter, the MikroKopter, can be used for serious robotics research. Using just the built-in low-cost inertial sensors, a 10 Hz laser scanner, and computationally cheap state estimator and control, we show that its hovering accuracy is comparable with other recent published results using platforms that cost more than five times as much. To achieve this performance, it is vital to understand the characteristics of the quadcopter — its sensors and onboard flight control system—which we achieved through system identification and reverse engineering. Full details of this platform: the software, tutorial and technical documents are available online at `https://wiki.qut.edu.au/display/cyphy/RosMikroKopter` [12].

Preprint submitted to 2012 IEEE International Conference on Robotics and Automation. Received September 16, 2011.

The remainder of this paper is organised as follows. Section I-A reviews related work on quadcopter state estimation and control, and Section I-B describes the characteristics of the low-cost MikroKopter platform we use in our experiments. The dynamics of the quadcopter for translational motion are summarised in Section II to motivate the need for translational velocity estimation, and our proposed estimation and control approach is described in Section III. We present our experimental results in Section IV, important technological trends in Section V and conclusions in Section VI.

A. Related work

In the past 3 years, many robotics researchers have demonstrated impressive navigation and control achievements with quadrotor platforms in indoor and outdoor environments. Pounds et al. presented fundamental dynamics analysis and control approaches through the design of a large-size quadrotor with a total weight of 4 kg and capable of lifting a 1 kg payload which was deemed necessary for the computers and sensors of the time [13], [14]. Hoffmann et al. exhibited a model-based algorithm for an autonomous flying vehicle, the Stanford Testbed of Autonomous Rotorcarft for MultiAgent Control (STARMAC) [4]. This platform has been used to demonstrate GPS based path-planning and obstacle avoidance in outdoor environments. Bouabdallah and Siegwart acheived impressive results in control and state estimation with a quadcopter platform and a ground station. Image data was sent to the ground station, processed, and commands were transmitted back to the flying vehicle over a radio communications link [15]. Kemp presented a similar configuration system in an office environment using an amateur quadrotor platform. Image data was transmitted over radio and processed at the ground station where edge features were extracted [16]. However, there are significant issues in transmitting video data over a wireless connection including latency and interference. All of these systems have been closed source with software not available to other researchers.

To overcome the challenges of video communications many researchers have turned to motion capture systems (such as Vicon) to estimate vehicle state. Such systems provide accurate and fast estimates of state using multiple fixed infrared cameras observing retroflective targets affixed to the vehicle. Using these external sensors several research groups have demonstrated significant achievements, for example challenging maneuvers such as flipping or flying through a thrown hula-hoop [17] [18]. Although quadrotor vehicles are able to perform these dynamic aggressive maneuvers the required high precision and high sample rate state estimates can only currently be achieved within a Vicon environment. This approach is impractical for autonomous navigation in large scale indoor or outdoor environments.

Self-contained quadrotors with on-board-computing and sensing for GPS-denied and navigation infrastructure free environments have also been demonstrated. The core idea of these systems is the application of well known 2D laser-range-finder based SLAM [19] [11] [20] to a flying vehicle operating at a nominal height.

B. The MikroKopter platform

In the last couple of years a new intermediate class of vehicle has emerged — the serious amateur class — which is exemplified by the MikroKopter project (`http://mikrokopter.de`). This is an open-source project where the mechanical, electronic and software design is available and it can be purchased in component form, as a kit, or as a ready-to-fly set.

1) Hardware: The MikroKopter quadrotor consists of four brushless-DC motors each with its own speed controller. These communicate via I^2C bus to the central flight controller board. Four 10-inch diameter and 4.5-inch slope propellers are used for propulsion. The flight control board version 2.1 is based on an Atmega 1284 processor running at 20 MHz which implements the attitude estimator, control loops, decodes the pulse stream from the radio control receiver, and also receives commands over a serial data port and transmits status information (at 20 Hz). The flight control board holds a triaxial MEMS accelerometer and gyroscope, and a barometric pressure sensor. A magnetic compass can also be fitted, however we do not use one due to interference from the magnetic fields generated by the motors. Multi-channel input from a Futaba transmitter is read via a digital input pin. For safety, the transmitter must be active to enable the quadrotor to fly. The flight controller has a serial port which can be used to receive commands or transmit status information. This is connected to a Zigbee module which allows commands and status to be communicated wirelessly. The MikroKopter has a mass of 0.65 kg and an endurance of 18 minutes (with 2200 mAh battery). The nominal payload was 0.25 kg, however, it has been tested to carry a payload of 0.85 kg with a reduction in endurance to 8.3 minutes.

For robotics research, this platform offers a lower cost and higher payload than the more well known research-grade vehicles. However, it has been disadvantaged by a lack of available scientific resources, such as documentation, technical details and source code. Grzonka et al. presented a system which had several similarities to the proposed approach in that the MikroKopter platform was used with a laser-range-finder and the range data was transmitted to a ground station for processing. However, attitude angles were measured using an extra IMU sensor and only the bearing and altitude were autonomously controlled [21].

The great advantage of the MikroKopter is that the flight control software is open source. The non-trivial disadvantages are that the code is written in German; it is poorly documented, particularly the serial data input and output protocol; and the state estimation algorithm is not a familiar type of filter. These factors do not impose limitations on somebody who simply wants to fly, nonetheless, they are a problem for roboticists who want to control the vehicle via external commands. For example, the specifications of whether the roll command is an angle or rate and the unit it is measured is unspecified. The biggest disadvantage of this platform is the lack of documentation and technical details

Preprint submitted to 2012 IEEE International Conference on Robotics and Automation. Received September 16, 2011.

that would satisfy a researcher. Details of the identification procedures and the results are given in more detail in [12].

II. The Quadcopter Control Problem

From a control perspective [6], [7] the quadcopter is an under-actuated force-controlled flying vehicle. Force actuation implies that rotational and translational motion can be modeled as a double integrator from command to attitude angle or horizontal position. The key to stable control of such systems is providing artificial damping through feedback of rotational and translational velocity. In classical control system terms we can consider rotational and translational motion as double integrators. Velocity feedback is required to move the system poles into the left-hand plane.

A. Attitude dynamics

Rotational motion is typically handled by roll and pitch attitude proportional-derivative (PD) control loops running at a high-sample rate. Attitude (the P term) and attitude rate (the D term) are derived from the gravity vector direction (accelerometers) and gyroscope measurements respectively. The MikroKopter flight controller runs PD controllers at 500 Hz for pitch and roll angle. We logged pilot commands and MikroKopter attitude estimates, see Figure 2, for manual flight. Using recursive least squares we fit an autoregressive moving average models with exogenous inputs model (ARMAX) to this time series data giving a linear discrete-time (at 50 ms) model of the closed-loop attitude controller

$$F(z)_{pitch} = \frac{0.148}{z-0.7639}$$
$$F(z)_{roll} = \frac{0.145}{z-0.7704}$$

which corresponds to a rise time of ≈ 0.4 s. The response of these models to the pilot input is compared to the measured response in Figure 2. We have experimented with adjusting the PD parameters in the flight controller firmware and can increase the control bandwidth; however this causes human piloting to become more challenging.

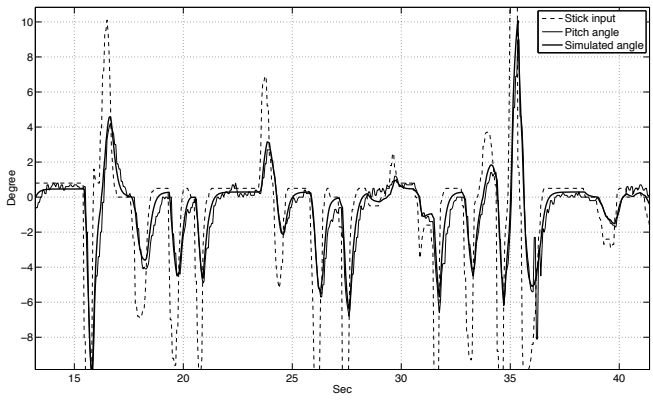


Fig. 2. Measured and predicted pitch angle for manual flight. Green denotes the commanded stick input, blue line the measured angle from the MikroKopter and green represents the model response.

B. Equations of motion

We define three coordinate frames: the world frame $\{o\}$, the body-fixed frame $\{b\}$, and the frame $\{Q\}$ which has the same origin as $\{b\}$ where x_Q and y_Q are the projections of x_b and y_b onto the world horizontal plane.

In the x_Q-direction the equation of motion is

$$m\ddot{x}_Q = T\sin\theta \tag{1}$$

where T is the total rotor thrust and θ is the pitch angle. In vertical equilibrium $T = mg$ and the small angle approximation $\sin\theta \approx \theta$ allows us to write this as a linear equation

$$\ddot{x}_Q = g\theta \tag{2}$$

which is a classical double-integrator plant. This ignores aerodynamic drag which would add a small damping term proportional to $\dot{x}_Q$, and out of balance forces due to poor trim which would add a constant acceleration.

The open-loop dynamics (from angle command to horizontal position) therefore include two free integrators and the first-order attitude response. If we assume an instantaneous position sensor and a one sample time delay for control, the z-plane poles are +1, +1, 0.7639 and 0 which is not stable for any finite gain as shown Figure. 3. Using a low-cost light-weight laser scanner with a 10 Hz sample rate would result in at least two delay states for the position sensor making stability even more unachievable. Introducing a system zero would would lead to closed-loop stability, however a *realisable* lead compensator requires an additional pole at the origin which again exacerbates the stability problem. Feedback of vehicle translational velocity, if it could be measured, would effectively introduce a pure zero and stabilise the system.

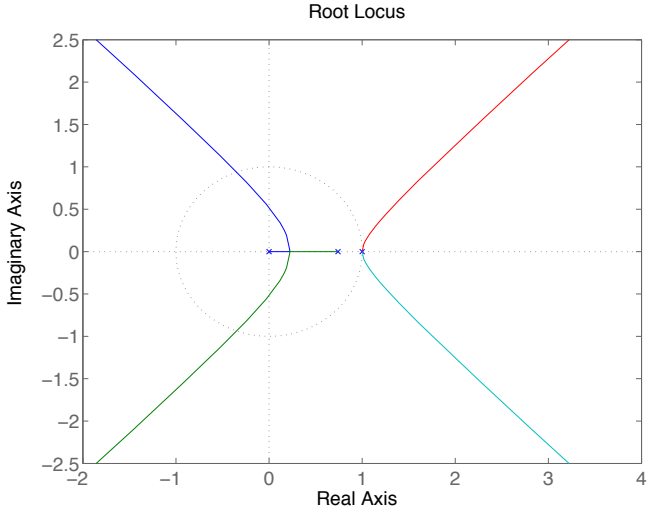


Fig. 3. z-plane root locus diagram of the MikroKopter x-axis open-loop dynamics $\frac{0.148}{z(z-0.7639)(z-1)^2}$ from pitch-angle command to x-position.

Preprint submitted to 2012 IEEE International Conference on Robotics and Automation. Received September 16, 2011.

III. Velocity estimation and control architecture

A. Complementary filter

To increase damping we require a high quality velocity estimate: smooth, high update rate with low latency. Differentiation of the position from the laser scanner and pose estimator results in velocity at 10 Hz with a latency of 150 ms or 3 control loop time intervals. This significantly limits the gain that can be applied when used for closed-loop velocity control. Instead we use the MikroKopter acceleration measurements (`AccRoll` and `AccNick`) which we read at 20 Hz with low latency and integrate them to create a velocity estimate. We subtract the acceleration due to gravity using the MikroKopter's estimated roll and pitch angles

$$\ddot{x}_Q = \frac{a_x + g\sin\theta}{\cos\theta} \tag{3}$$

$$\ddot{y}_Q = \frac{a_y - g\sin\phi}{\cos\phi} \tag{4}$$

where a_x, a_y are the measured acceleration from the flight control board converted to our coordinate system, and θ, ϕ denote the pitch and roll angles respectively. Acceleration and attitude are returned together in the flight-controller status message at 20 Hz.

Any estimator that relies on integration is subject to substantial errors due to drift, even over quite short time intervals. We therefore fuse these two estimates using a simple discrete-time complementary filter [22]

$$\hat{v}_x(t+1) = \hat{v}_x(t) + \ddot{x}(t)_Q + K(\tilde{v}_x(t) - \hat{v}_x(t))\Delta_t \tag{5}$$

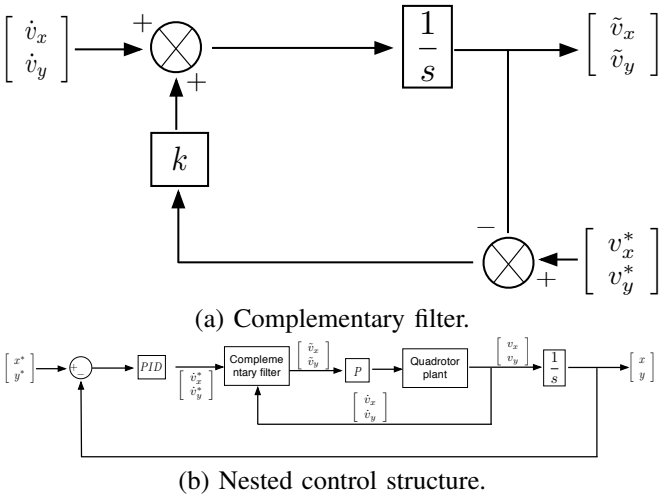
where $\hat{v}_x$ is estimated velocity, $\tilde{v}_x$ is obtained from differentiation of the laser-based pose estimate and is computed at a slower rate than $\ddot{x}_Q$ so the filter takes the most recent value, and K is a gain

Compared to a Kalman filter the computation is simple, and there is only one tuning parameter. Considering the estimator in the frequency domain, K controls the cross-over frequency: below this $\tilde{v}_x$ dominates and above it $\ddot{x}_Q$ dominates. Complementary filters have been used previously for UAV velocity estimation, such as to fuse velocity from low-rate optical flow with high-rate inertial data [23].

We do not explicitly deal with accelerometer bias in this estimator since the bias is estimated and corrected in the MikroKopter flight controller. We could however deal with it quite simply by introducing an integral term to the complementary filter.

B. Control

1) xy-translational control: The block diagram of our nested controller is shown in Figure 4(b). The inner-loop is a velocity controller with proportional and integral control with feedback of estimated velocity from the complementary filter (5). The outer loop is a position controller with proportional control. This structure is equivalent to a proportional-integral-derivative (PID controller and is similar to the back-stepping controller of [24], however the nested structure decouples the different sampling rates of the position sensor and the velocity sensor. The inner-loop runs at 20 Hz and the outer-loop at 10 Hz.



(a) Complementary filter.

(b) Nested control structure.

Fig. 4. Velocity estimator and control structure for translational motion. The K_P for the velocity loop is 27 and K_P=0.8, K_I=0.1 and K_D=0.7 for the position PID controller.

2) z-axis (altitude) control: Several researchers have demonstrated impressive altitude control approaches [24] [25], but there are 3 challenges for the altitude control of the MikroKopter with our sensor configuration. The z-axis on-board accelerometer and the laser height measurement are significantly prone to vibrations, thus, it is difficult to estimate the z-axis velocity. Further, the thrust control signal is highly quantised in the range $0 \sim 255$. We use a standard PID controller to compute thrust demand

$$u_h(t) = K_p e_h(t) + K_i \int_0^t e_h(\tau)d\tau + K_d \frac{d}{dt} e_h(t) + u_{off}(t) \tag{6}$$

where $e_h(t) = z^* - z$ and z^* denotes the desired height, z is the current measurement and u_{off} is a feedforward term ($K_p = 21.7,\ K_i = 5,\ K_d = 13.3$). The height measurement from the laser sensor is coupled with pitch and roll angle and the true height (z) is

$$z = \cos\theta\cos\phi\ \hat{z} \tag{7}$$

where $\hat{z}$ is the sensor measurement.

Significant thrust is required to lift the vehicle, typically a command value of approximately 120 (on a scale of 0 – 255), is required for hover. This can be considered as a constant disturbance force which will be rejected by the integral term, but this typically takes many seconds to ramp up to the required level. We can preset the integrator to a nominal value or provide a feedforward term u_{off}. We observe that the delivered thrust is a function of both the command value and the battery terminal voltage. The measured nominal thrust command value for hover versus battery voltage is shown in Figure 5 and is well described by a fitted cubic polynomial. The Lithium-Polymer battery also has a non-linear discharge curve. The voltage drops significantly at the start and end of discharge and continuously decreases at the mid-zone.

Preprint submitted to 2012 IEEE International Conference on Robotics and Automation. Received September 16, 2011.

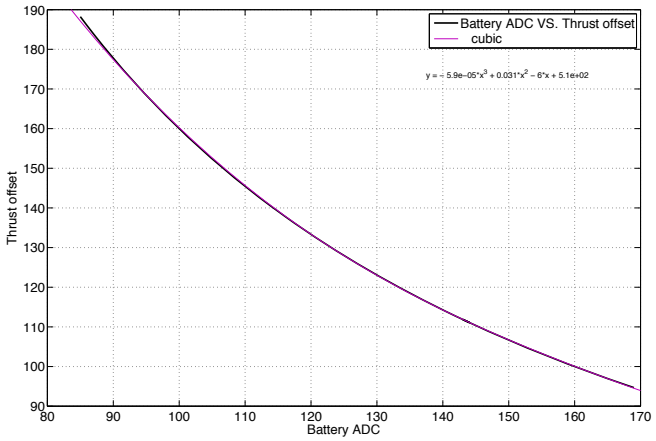


Fig. 5. Measured relationship between thrust offset, $u_{off}(t)$, and battery ADC reading. The coefficients of the fitted polynomial are shown.

3) Yaw control: The yaw (heading angle) of the quadrotor is generally a less critical degree of freedom and it can be independently controlled using a PID controller

$$u_{yaw}(t) = K_p e_{yaw}(t) + K_i \int_0^t e_{yaw}(\tau)d\tau + K_d \frac{d}{dt} e_{yaw}(t) \quad (8)$$

where $e_{yaw}(t) = \psi^* - \psi$, ψ^* is the desired angle and ψ is the current measurement ($K_p = 6$, $K_i = 1$, $K_d = 2$). Figure 6 shows the yaw angle control result when a pilot introduces yaw disturbances.

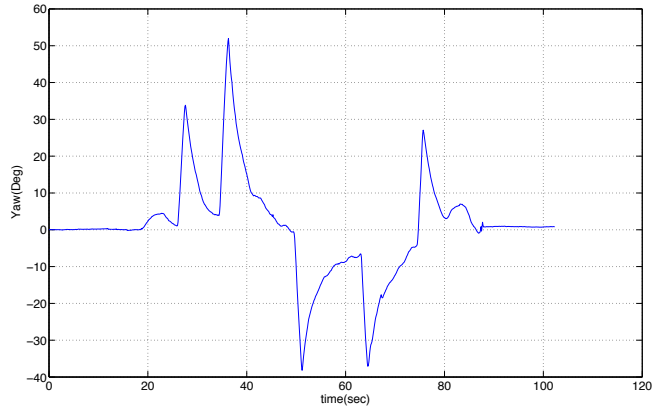


Fig. 6. Yaw controller disturbance rejection. The pilot manually imposes disturbances and the vehicle returns to the desired heading angle of 0°. Yaw angle is measured with a laser scan matcher.

IV. Experimental and Results

In this section, we present results of estimator performance evaluation while hovering which includes velocity, position and orientation estimation and hovering performance comparison.

A. Implementation

1) Hardware Setup: Our MikroKopter L4-ME quadcopter carries an Overo Gumstix which runs Ubuntu Linux and ROS [26]. An Hokuyo model URG-04LX laser scanner

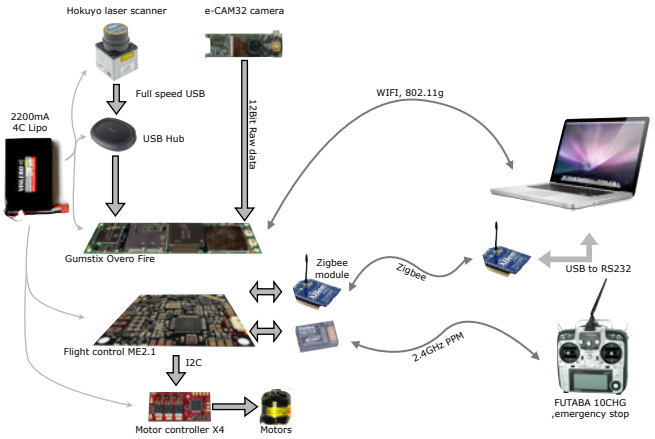


Fig. 7. System architecture. The laser scanner is attached to a USB Hub since the Overo Gumstix USB host only supports High Speed USB. The Zigbee module is used to transmit IMU data to the ground station and receive commands. The WiFi connection connects the ROS nodes on the Gumstix to the ground station. For safety a manual pilot transmitter is linked to the quadrotor system.

(10 Hz and 4 m range) scans in the horizontal plane and the "laser hat" from the City College of NewYork [27] povides altitude as well. The total payload mass is 0.18 kg and the communication links are WiFi for transferring laser data and ZigBee for status monitoring and command sending. A Lipo pack (4cells, 2200 mAh), provides the system power. The advantage of the MikroKopter is a competitive price. This platform is ***6.4*** times more cost effective [12] than the similar level platform of Ascending Technologies-the Pelican [28].

2) Software Architecture: The ROS framework is used to integrate modules (see Figure. 8), where dark grey boxes denote the ROS nodes which are individual processes. The Overo Gumstix runs the standard ROS laser scanner node and publishes the topic */scan* over WiFi to the base station every laser scan interval (100 ms). The ROS canonical scan matcher subscribes to this topic, and estimates 2D pose (x,y,θ) using an ICP algorithm [29] which it publishes as topic */pose2D* . The ROS serial node runs on the base station and communicates with the MikroKopter flight control board over the ZigBee link. Every 50 ms it requests a `DebugOut` packet which it receives and the inertial data (converted to SI units). is published as the */mikoImu* topic. This node also subscribes to the */mikoCmd* topic and transmits the command over the ZigBee uplink to the flight controller. These topics can be recorded in a log file (ROS bag format) and later replayed (using `rosbag`) to test the state estimator and controller. Note that the overall software system latency is about 170 ms and the system response delay is about 200 ms.

B. Estimator performance

We evaluate the performance of the estimators for velocity, position and attitude while hovering.

1) Velocity estimation: The performance evaluation of the velocity estimator is performed by comparing the measured

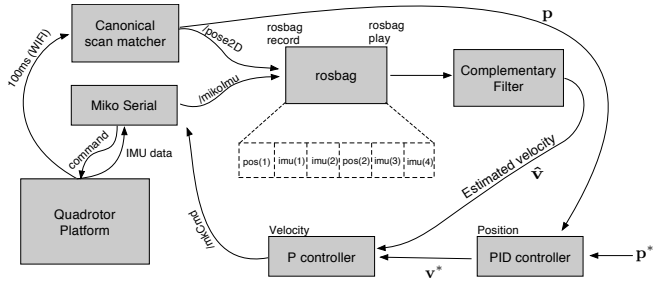


Fig. 8. Software implementation using ROS platform where dark grey boxes represent ROS nodes and the prefix '/' denotes a ROS topic. p is position and * denotes demand.

velocities with the ground truth — a sub-millimetre accuracy g-speak motion capture system [30]. The ground truth velocities are obtained by calculating first derivative of the position and the estimated velocities are generated by the proposed complementary filter (5). Note that during takeoff, the quadrotor moves a little horizontally due to poor trim but returns quickly to the desired hovering position. Figure. 9 shows the standard deviations of the velocities errors compared to the g-speak and the standard deviation values are $\{\sigma_{vx}\,\sigma_{vy}\} = \{0.051, 0.054\}$ m. Note that these values are calculated over the flight interval between $t = 20$ s (takeoff) and $t = 60$ s (landing).

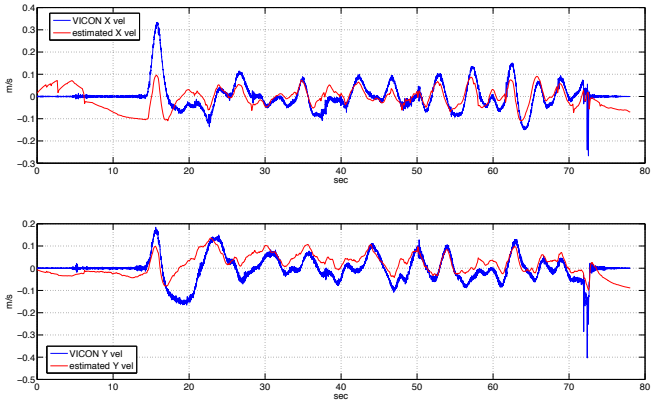


Fig. 9. The horizontal velocities estimation results with the ground truth while hovering. Blue denotes the ground truth and red indicates the complimentary filter output.

2) Position estimation: In this study the vehicle position was estimated using the laser-range-finder and scan matcher and ground truth came from the g-speak system. The goal position of the vehicle is $(0, 0, 0.4)$ m. The standard deviations of the ground truth position are $\{\sigma_x\,\sigma_y, \sigma_z\} = \{0.082, 0.092, 0.055\}$ m. These are again computed over the flight interval.

3) Angle estimation: Figure 11 compares the MikroKopter angle estimation with the ground truth. Pitch and roll angle estimations, (from the onboard embedded flight controller) are very close to the ground truth. However for the yaw angle estimation, there is rapid drifting due to pure integration of the gyroscope

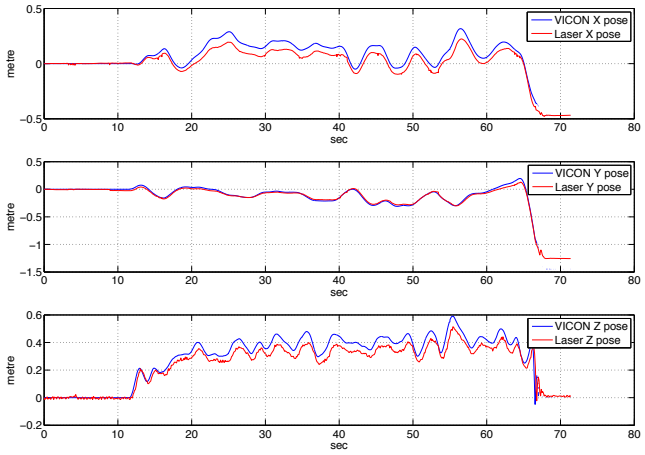


Fig. 10. x,y and z position estimation while hovering with the ground truth.

measurement. Therefore we use the yaw angle obtained from the laser scan matcher as a synthetic compass. Note that the noise on the pitch and roll angle ground truth is due to a small baseline (0.03 m) of g-speak infrared markers.

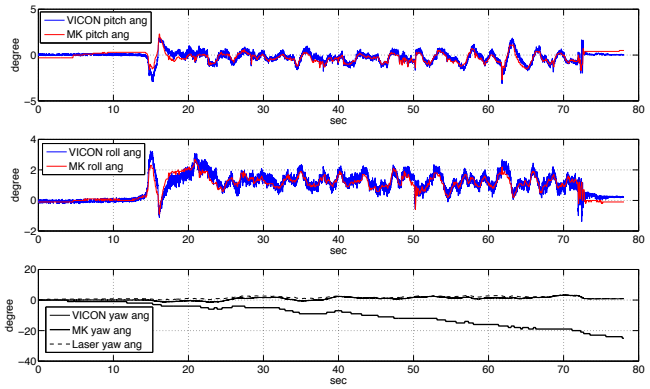


Fig. 11. Pitch, roll and yaw angle estimation while hovering compared with the ground truth. Blue indicates the ground truth and red is the MikroKopter angle estimation.

C. Hovering performance comparison

Hover performance is a quantitative performance metric that can be used compare estimation and control approaches. The performance of our system is compared with performance data from [20], [21], [25], [31], [32]. Note that [20], [31] both used the same Ascending technology platform (Pelican) and the Hokuyo 03LX laser range finder; [25], [32] used the Pelican and the Hummingbird platform, respectively, with a monocular camera; and [21] used a MikroKopter platform but with an additional sensor for attitude estimation. Our system shows highly competitive results particularly when taking into account the lower sample rates of the sensors and the computational simplicity of the estimation and control algorithms. Performance is sufficient for *close-quarters* navigation.

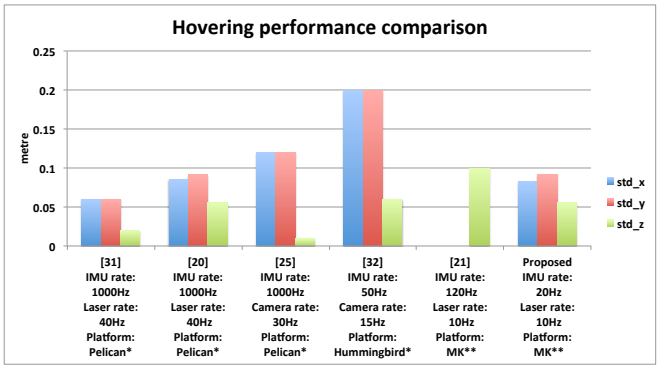


Fig. 12. Hovering performance comparing the proposed system with the state-of-the-art achievements. Note that the data is the standard deviations and is collected from the cited papers and a lower value denotes a better performance. * Ascending Technologies, ** MikroKopter.

V. Smaller and cheaper quadcopters

To date much research has leveraged well known sensors, such as laser range finders, and classical techniques such as 2D SLAM [33], [34] from the mobile ground robot field. These 2D sensors suffer a coupling from vehicle attitude to obstacle distance which can be problematic. Other extroceptive sensors such as, monocular vision, stereo vision or 3D cameras do not have this coupling problem and have advantages in terms of lower weight and higher sample rate. The relevant characteristics of some quadcopter sensors is summarized on Table I. The Kinect sensor has significant advantages for indoor close-range oeprations due to its high sample rate and ability to sense full 3D, rather than a 2D cross-section.

VI. Conclusions

We have described the crucial importance of translational velocity estimation for quadcopter control and navigation. In particular we have presented computationally efficient state estimation and control algorithms which will allow for smaller onboard computers in future. We have demonstrated ground-truthed performance comparable to state-of-the-art with a quadcopter systems, done with a platform of less than one fifth the cost and with a laser scanner that scans four times more slowly. We used an amateur-class quadcopter, and to achieve this level of performance required understanding the dynamics of the quadrotor through system identification and reverse engineering. This platform has many advantages such as cost efficiency, high payload, open source firmware and a large user community. Our learnings about this platform are returned to the community through open documentation and software available online.

We have a large program of ongoing work. Firstly we are currently extending our approach to the z-axis using a complementary filter to combine vertical acceleration, laser altimetry and barometric pressure and a nested controller. Secondly we are exploring changes to the flight control firmware to improve the performance of the attitude control, and to increase the frequency with which inertial state is reported over the serial link. Thirdly we will migrate the pose estimator to the onboard Gumstix processor which eliminates the complexity, limited range and unreliability of the communications link. Finally, we are investigating other extroceptive sensors such as cameras with wide-angle fields of view and the Kinect range camera.

TABLE I

Comparison of extroceptive sensors for quadcopters. Notes: * used in this work, ** outer case removed.

Property Property	**Hokuyo URG-04LX ***	**Hokuyo UTM-30LX**	**Microsoft Kinect**
Cost (USD)	2375	5590	150
Weight(g)	140	370	250 **
Range(m)	0.02∼4	0.1∼30	0.5∼3.5
Framerate(Hz)	10	40	30
Field of view(Degree)	H=240	H=270	H=57, V=43
Power consumption(W)	2.5∼4	8.4∼12	3.4

References

[1] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a quad-rotor robot," in *Australasian Conference on Robotics and Automation (ACRA)*, Dec 2006.

[2] P. Pounds, R. Mahony, J. Gresham, P. Corke, and J. Roberts, "Towards dynamically-favourable quad-rotor aerial robots," in *Australasian Conference on Robotics and Automation (ACRA)*, Dec 2004.

[3] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pp. 2451 – 2456, 2004.

[4] H. Huang, G. Hoffmann, S. Waslander, and C. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3277 –3282, 2009.

[5] J. X. William Morris, Ivan Dryanovski, "3D Indoor Mapping for Micro-UAVs Using Hybrid Range Finders and Multi-Volume Occupancy Grids," in *Proceedings of Robotics: Science and Systems (RSS) Workshop on RGB-D Cameras*, Jun 2010.

[6] P. Corke, *Robotics,Vision and Control Fundamental algorithms in MATLAB.*, vol. 73. Springer, 2011.

[7] R. Mahony, S.-H. Cha, , and T. Hamel, "A coupled estimation and control analysis for attitude stabilisation of mini aerial vehicles," in *Australasian Conference on Robotics and Automation (ACRA)*, Dec 2006.

[8] S. Lupashin, A. Schöandllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1642 –1648, May 2010.

[9] D. Mellinger, M. Shomin, and V. Kumar, "Control of quadrotors for robust perching and landing," in *International Powered Lift Conference*, (Philadelphia, PA), Oct 2010.

[10] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The GRASP Multiple Micro-UAV Testbed," *IEEE Robotics and Automation Magazine*, vol. 17, pp. 56–65, Sept. 2010.

[11] A. Bachrach, A. de Winter, R. He, G. Hemann, S. Prentice, and N. Roy, "RANGE - robust autonomous navigation in GPS-denied environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1096 –1097, May 2010.

[12] "ROS MikroKopter tutorial and technical documentation." `https://wiki.qut.edu.au/display/cyphy/RosMikroKopter`.

[13] P. Pounds, R. Mahony, and P. Corke, "Modelling and Control of a Large Quadrotor Robot," in *Control Engineering Practice*, vol. 18, pp. 691 – 699, 2010.

[14] P. Pounds and R. Mahony, "Design principles of large quadrotors for practical applications," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3265 –3270, May 2009.

[15] R. S. Samir Bouabdallah, Pierpaolo Murrieri *Autonomous Robots*, vol. 18, Mar 2005.

[16] C. Kemp, *Visual Control of a Miniature Quad-Rotor Helicopter*. PhD thesis, University of Cambridge, 2006.
[17] D. Mellinger and V. Kumar, "Minimum Snap Trajectory Generation and Control for Quadrotors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.
[18] S. Lupashin, A. Schollig, M. Hehn, and R. D'Andrea, "The Flying Machine Arena as of 2010," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2970 –2971, May 2011.
[19] R. He, S. Prentice, and N. Roy, "Planning in information space for a quadrotor helicopter in a GPS-denied environment," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1814 –1820, 2008.
[20] S. Shen, N. Michael, and V. Kumar, "Autonomous Multi-Floor Indoor Navigation with a Computationally Constrained MAV," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.
[21] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2878 – 2883, May 2009.
[22] J. M. Roberts, P. I. Corke, and G. Buskey, "Low-Cost Flight Control System for a Small Autonomous Helicopter," in *Australasian Conference on Robotics and Automation (ACRA)*, Dec 2002.
[23] P. Corke, "An Inertial and Visual Sensing System for a Small Autonomous Helicopter," *Journal of Robotic Systems*, vol. 21, p. 9, Feb 2004.
[24] S. Bouabdallah and R. Siegwart, "Full control of a quadrotor," in *Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pp. 153–138, Nov 2007.
[25] M. Achtelik, S. Weiss, and R. Siegwar, "Onboard IMU and Monocular Vision Based Control for MAVs in Unknown In- and Outdoor Environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.
[26] "Robot Operating System." `http://ros.org`.
[27] "CCNY Robotics Lab." `http://robotics.ccny.cuny.edu`.
[28] "Ascending Technologies." `http://www.asctec.de/`.
[29] A. Censi, "An ICP variant using a point-to-line metric," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 19 –25, May 2008.
[30] "Oblong, g-speak motion capture platform." `http://oblong.com`.
[31] A. G. Bachrach, "Autonomous Flight in Unstructured and Unknown Indoor Environments," Master's thesis, MIT, 2009.
[32] F. Bourgeois, L. Kneip, S. Weiss, and R. Siegwart, "Delay and Dropout Tolerant State Estimation for MAVs," in *Proceedings of The 12th International Symposium on Experimental Robotics (ISER)*, 2010.
[33] I. Dryanovski, W. Morris, and J. Xiao, "An Open-Source Pose Estimation System for Micro-Air Vehicles," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.
[34] C. Hertzberg, R. Wagner, O. Birbach, T. Hammer, and U. Frese, "Experiences in Building a Visual SLAM System from Open Source Components," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

Preprint submitted to 2012 IEEE International Conference on Robotics and Automation. Received September 16, 2011.