Vertical Infrastructure Inspection using a Quadcopter and Shared Autonomy Control

Inkyu Sa and Peter Corke

Abstract This paper presents a shared autonomy control scheme for a quadcopter that is suited for inspection of vertical infrastructure — tall man-made structures such as streetlights, electricity poles or the exterior surfaces of buildings. Current approaches to inspection of such structures is slow, expensive, and potentially hazardous. Low-cost aerial platforms with an ability to hover now have sufficient payload and endurance for this kind of task, but require significant human skill to fly.
We develop a control architecture that enables synergy between the ground-based operator and the aerial inspection robot. An unskilled operator is assisted by onboard sensing and partial autonomy to safely fly the robot in close proximity to the structure. The operator uses their domain knowledge and problem solving skills to guide the robot in difficult to reach locations to inspect and assess the condition of the infrastructure. The operator commands the robot in a local task coordinate frame with limited degrees of freedom (DOF). For instance: up/down, left/right, toward/away with respect to the infrastructure. We therefore avoid problems of global mapping and navigation while providing an intuitive interface to the operator.
We describe algorithms for pole detection, robot velocity estimation with respect to the pole, and position estimation in 3D space as well as the control algorithms and overall system architecture. We present initial results of shared autonomy of a quadrotor with respect to a vertical pole and robot performance is evaluated by comparing with motion capture data.

Inkyu Sa
Queensland University of Technology, Brisbane, Australia, e-mail: i.sa@qut.edu.au

Peter Corke
Queensland University of Technology, Brisbane, Australia e-mail: peter.corke@qut.edu.au

1 Introduction

The options for inspecting locations above the ground are quite limited, and all are currently cumbersome. Ladders can be used up to a height of 10–15 metres but are quite dangerous: each year 160 people are killed and 170,000 injured in falls from ladders in the United States[1]. A person can be lifted in the basket of a cherry picker up to a height of 15 m but vehicle access is required and the setup time is significant. Beyond that height a person either climbs up the structure or rappels down from the top, both of which are slow and hazardous. Inspection from manned rotorcraft is possible but is expensive and only suitable in non-urban environments. In recent years we have seen significant advances in small VTOL platforms, in particular quadrotors, driven by advances in power electronics, MEMS sensors and microcontrollers. These systems are low-cost and have sufficient payload and endurance for useful inspection missions. They are also low-weight which reduces the hazard due to their deployment.

This paper presents a shared autonomy system for inspection of vertical infrastructure — tall man-made structures such as streetlights, electricity poles or the exterior surfaces of buildings — using a vertical take-off and landing (VTOL) robot platform. Shared autonomy indicates that the major fraction of control is accomplished by the onboard computer. The operator provides "high level" commands in a reduced DOF task space, while the robot is responsible for stable flight, disturbance rejection and collision avoidance. This allows an unskilled operator to easily and safely control a quadrotor to examine locations that are otherwise difficult to reach.

(a) (b) (c)

Fig. 1 (a) The Cyphy Lab MikroKopter research platform. The pole can be seen on the left of the image. (b) A dangerous situation to inspect or repair a street light[2]. (c) Sufficient space is required for vehicle access and it is a time consuming process to setup operation.[3]

[1] May 2009 Consumer Reports magazine.
http://www.consumerreports.org/cro/magazine-archive/may-2009/may-2009-toc.htm

[2] Baltimore museum of industry. http://www.thebmi.org/

[3] Facelift. http://www.facelift.co.uk/

The presented VTOL flying robot has functionalities of pole detection and task-space operator command input. Implicit in the inspection task is the requirement to fly close to structures with which a collision would signficantly damage the vehicle. Air flow around tall structures results in eddies that induce disturbances on the vehicle which must be robustly rejected to ensure safety and task performance. This requires accurate and fast velocity and position estimation and an appropriate control methodology.

This paper is organised as follows: Section 2 presents relevant research on quadrotor and bio-inspired climbing robots suitable for inspection. Section 3 explains the methodologies: system modeling and identification, velocity estimation and nested controllers, pole detection algorithm, shared control scheme. We present our experimental results in Section 4, and important technological trends and conclusions in Section 5.

2 Related work

Robotics and mechatronics researchers have been demonstrated a variety of climbing robots for vertical infrastructure inspection. Typically, these robots are inspired by reptiles, mammals and insects and their type of movement varies between sliding, swinging, extension and jumping. The MATS robot has 5 DOF and a symmetrical mechanism that showed good mobility features for travel, however, it requires a special docking stations to hold itself [1]. A bio-mimicking robot, StickyBot, has a hierarchical adhesive structure under its toes to hold itself on any kind of surfaces [2]. RiSE V3, a legged locomotion climbing robot, is designed for high-speed climbing of a uniformly convex cylindrical structure, such as a telephone or electricity pole [3]. A bridge cable inspection robot [4] has wheels held against the cable to create a contact force required to move along the cable. These types of robots could not only replace a worker undertaking risky tasks in a hazardous environment but also increase the efficiency of such tasks. However, they require complex mechanical designs, special materials and complicated dynamics analysis. Their applications are limited to specific type of structures, such as cylindrical-shaped poles. VTOL platforms are a feasible alternative to achieving the same goals as climbing robots and involve a much simpler mechanism. Recently, [5] demonstrates embedded stereo camera based egomotion estimation for structures inspections such as a boiler and general indoor scenarios. Although IMU guided feature matching and stereo based camera pose estimation show impressive real-time achievements, it might need integration of control theory to fly in close quarters.

3 Methodologies

This section describes the key approaches of our system: shared control; modeling and system identification; pole detection; velocity estimation and nested controllers.

3.1 Shared Control and Task Frame

Sheridan. [6] introduced a spectrum of approaches for remote operation of a system. At one end is "conventional manual control" (system 1) where the system is fully controlled by a human operator and there is no computer-aided functionalities. At the other end is "fully autonomous system" (system 5) where a human operator can observe but cannot intervene in the process. Our proposed system is modelled on Sheridan's "Supervisory Control" architecture, specifically system 4, in which the control loop is closed through a computer but there are still human interventions. This approach allows the high-bandwidth flight control and obstacle avoidance loops to be closed on board the robot with the "high level" commands from the human being treated as requests that will implemented if safe to do so.

A task frame (TF) refers to a coordinate frame that can be attached to an object in the workspace [7]. There is a geometric transformation between the world coordinate and TF. The advantages of a TF is that actions which are difficult to express in the world coordinate can be easily specified in the TF. For an inspection task the TF is associated with the operator's current view of the infrastructure and provides an intuitive control framework to the user in which to express desired motion commands. Figure 3(a) shows the world coordinate $\boldsymbol{W}$ and the task frame $\boldsymbol{T}$.

A VTOL platform has four DOF (roll, pitch and yaw angles, and throttle) and significant operator skill is required to control position in 3-dimensional Cartesian space. One aspect of this skill is that the roll and pitch angles induce forces on the vehicle, and with relatively little aerodynamic damping these inputs are effectively Cartesian accelerations. The level of skill required is greatly increased when flying next to a large and unforgiving structure in the presence of wind-induced force disturbance. Manual piloting also requires the vehicle to be in the pilot's visual field of the pilot and sufficiently close that its orientation in space can be determined.

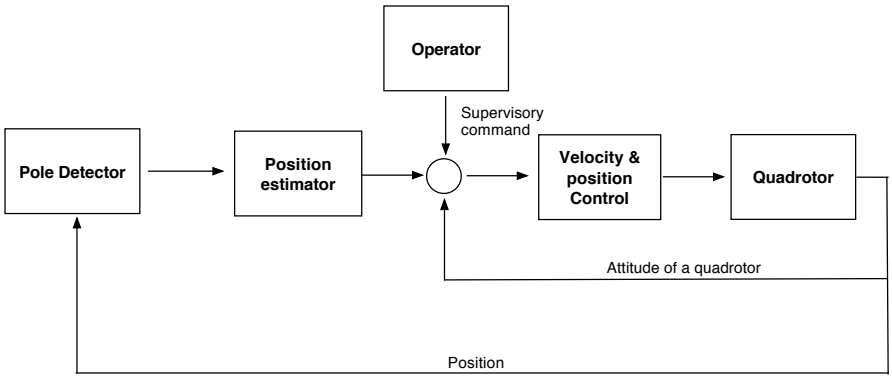
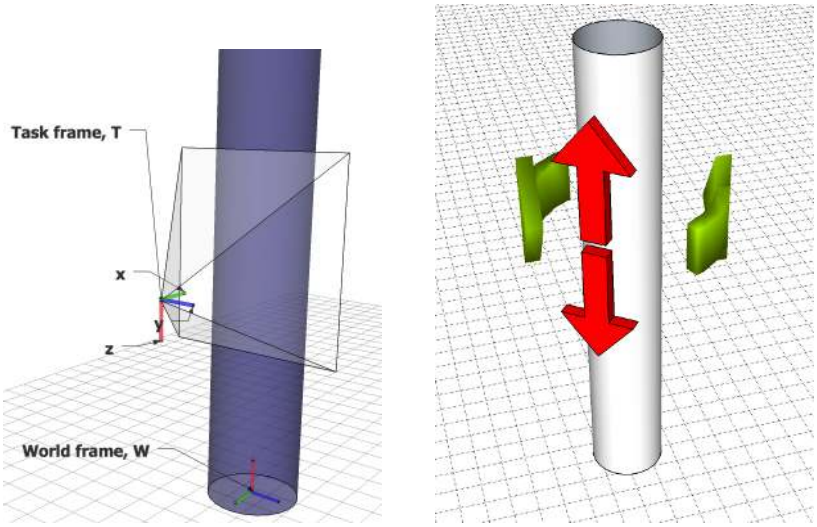


Fig. 2 Hierarchical multi-loop shared control architecture. The inner loop receives a desired goal by the outer loop. Control, Position estimator loops have different update rates for a purpose. Arrows indicate data flow directions and specify inputs.



(a) (b)

Fig. 3 (a) World frame ***W*** and the task frame ***T***. ***W*** is the centre of the pole and ***T*** denotes a camera coordinate which is equal to a user's point of view. An unskilled operator can easily control the robot because it can localize with respect to the pole.(b) Reduced controllable task degree of freedom(DOF).

To allow use by an unskilled operator we need to reduce the number of DOF that must be controlled and make the DOF intuitive and task specific. As shown in Figure 3(b), for a pole inspection task, the operator controls only 2 DOF: distance along the pole and angle around the pole. This is sufficient for inspection of the entire pole area and easy to control. Small supervisory commands forward and backward is possible however it is subtle compared to the height and yaw commands.

3.2 Modeling and System Identification

The quadcopter is an under-actuated force-controlled flying vehicle. This force actuation implies that rotational and translational motion can be modeled as a double integrator from command to attitude angle or horizontal position [8],[9]. In our work we use the MikroKopter open-source quadrotor[4] for which there is little engineering documentation or published dynamic models [11]. The vehicle has an onboard attitude controller which uses rate and angle feedback from gyroscopes and accelerometers. We identified the dynamics of the closed-loop attitude by recording pilot commands and MikroKopter attitude estimates, for manual flight. We fit

[4] MikroKopter. http://www.mikrokopter.de/

an autoregressive moving average model with exogenous inputs model (ARMAX) using recursive least squares to this time series data giving a linear discrete-time (at 50 ms) first-order model

$$F(z)_{pitch} = \frac{0.148}{z-0.7639},\ F(z)_{roll} = \frac{0.145}{z-0.7704} \tag{1}$$

as the angle response to angle demand.

Translational motion is driven by the thrust force component in the horizontal plane and can be modelled as a double integrator. There is relatively little translational aerodynamic damping, though blade flapping does add some damping [10]. For stability additional damping is required and this necessitates velocity estimation.

3.3 Velocity Estimation and Nested Controllers

The key to stable control of such systems is providing artificial damping through feedback of rotational and translational velocity. In order to introduce damping we require a high quality velocity estimate: smooth, high update rate with low latency. Computing velocity using differentiation of the position from the pole detection and pose estimator results in velocity at 10 Hz with a latency of 100 ms. 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

$$\ddot{x}_Q = \frac{a_x + g\sin\theta}{\cos\theta},\ \ddot{y}_Q = \frac{a_y - g\sin\phi}{\cos\phi} \tag{2}$$

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. $\{Q\}$ is a coordinate frame centred on the vehicle with axes parallel to the world frame. Acceleration and attitude are returned together in the flight-controller status message at 20 Hz.

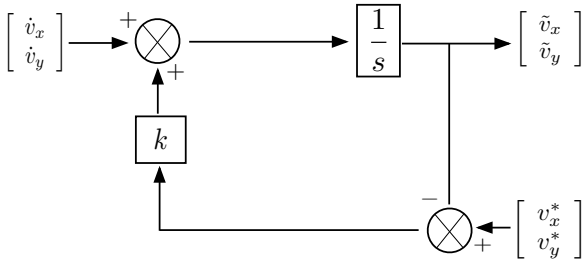


Fig. 4 Complementary filter for velocity estimation. Compared to a Kalman filter the computation is simple, and there is only one tuning parameter, K. $\dot{v}_x$ and $\dot{v}_y$ are obtained from a onboard IMU sensor. v_x^* and v_y^* are from a laser range finder.

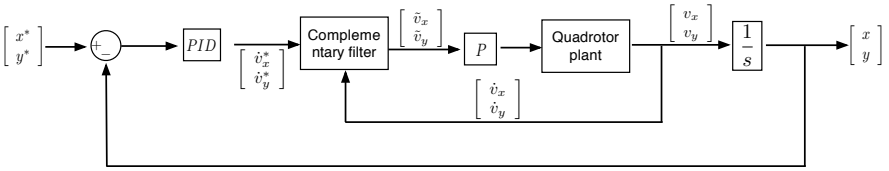


Fig. 5 Velocity estimator and control structure for translational motion. The K_P for the velocity loop is 27 and K_P=0.8, K_I=0.1 and K_D=0.7 for the position PID controller.

As 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 [12] as shown in Figure 4 and described by

$$\hat{v}_x(t+1) = \hat{v}_x(t) + \ddot{x}(t)_Q + K(\tilde{v}_x(t) - \hat{v}_x(t))\Delta_t \tag{3}$$

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. 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 [14].

The block diagram of our nested controller is shown in Figure 5. The inner-loop is a velocity controller with proportional and integral control with feedback of estimated velocity from the complementary filter, Equation (3). The outer loop is a position controller with proportional control. This structure is equivalent to a proportional-integral-derivative, 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. As we showed in [11] this simple control architecture gives performance that is comparable with other published results that are using 40 Hz laser scanners and 1 kHz IMU sample rates.

3.4 Pole Detection

We use an Hokuyo model URG-04LX laser range finder (10 Hz and 4 m range) to detect the pole. As shown in Figure 6 the laser detects the 15 cm radius pole as a straight line rather than a circlular arc, and we believe this is an artifact of filtering firmware in the laser range finder. We use a Split-Merge line extraction [15] routine on the raw laser data, followed by target discrimination (see Algorithm 1), tracking and filtering to estimate the range and bearing of the pole with respect to the robot.

We score each candidate using a previous detected averaged position.

$$S_k = dist(\bar{P}, \tilde{P}_k) \tag{4}$$

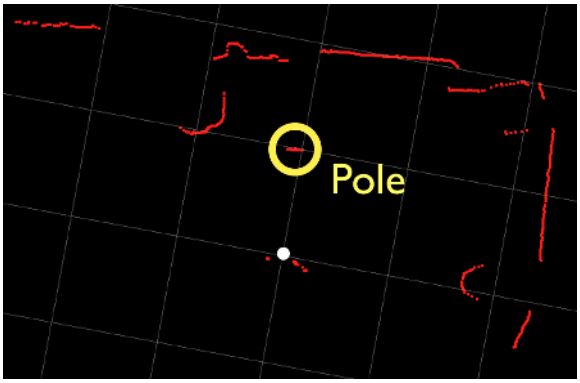


Fig. 6 Top view. Red dots are raw laser scan data and the yellow circle denotes the scan data corresponding to the pole. The white dot is the centre of the sensor.

where $\tilde{P}_k \in \mathbb{R}^2$ is the kth candidate position, and $\bar{P} \in \mathbb{R}^2$ is the average position. All candidates are sorted by decreasing score and the one with the maximum score is selected. For bootstrapping, we assume that a pole, $\mathbf{P}$, is located within discoverable boundary ($\mathbf{P} < \alpha,\beta,\gamma$) at system startup (see Algorithm 1) .

4 Experimental Results

In this section, software and hardware implementation are described in depth. We also present results of estimator performance evaluation while hovering which includes velocity, position and ground-truthed circle trajectory around a pole.

4.1 Software and Hardware Implementation

The ROS framework is used to integrate modules (see Figure. 7), where blue boxes denote the ROS nodes which are individual processes. The onboard 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 pole detector subscribes to this topic, and estimates 2D pose (x,y) which it publishes as topic */pole pose2D* . The ROS serial node 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. Note that the overall software system latency is about 170 ms and the system response delay is about 200 ms. Technical documentation and this software are available online[5].

[5] ROS QUT Cyphy wiki page http://www.ros.org/wiki/MikroKopter/Tutorials

Algorithm 1: Pole detection algorithm

```
while ! (Find a pole) do
    if (l.length[i] <α) && (l.distance[i] < β)&&(l.angle[i] <γ) then
        l[i] is the pole;
        Find a pole = TRUE;
    else
        i=i+1;
    end
end
Continuous : find the best candidate satisfying less strict conditions.
while ! (Find the best candidate) do
    if (c.length[j] <δ) && (c.distance[j] <ε)&&(c.angle[j] <ζ) then
        Put c[j] in the candidate list;
    else
        j=j+1;
    end
    Calculate scores using S_k = dist(P̄, P̃_k); //Equation 4
    Ascending sorting of the candidate list and pick the best score,c;
    if c > ξ then
        pole=c;
        Find the best candidate=TRUE;
    end
end
Note that constant parameters α < δ, β < ε and γ < ζ.
ξ denotes the score threshold.
```

Our MikroKopter L4-ME quadcopter carries an Overo Gumstix which runs Ubuntu Linux and ROS[6]. An Hokuyo model URG-04LX laser scanner (10 Hz and 4 m range) scans in the horizontal plane and the "laser hat" from the City College of NewYork[7] provides altitude as well. The total payload mass is 0.18 kg and 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 than the similar level "Pelican " platform[8].

4.2 Estimation and Control

The performance evaluation of the velocity estimator is performed by comparing the measured velocities with the ground truth — a sub-millimetre accuracy g-speak/VICON motion capture system[9]. The ground truth velocities are obtained by calculating the first derivative of the position and the estimated velocities are generated by the proposed complementary filter, Equation (3). Note that during

[6] Robot Operating System, http://www.ros.org/wiki/

[7] City College of NewYork Robotics Lab, http://robotics.ccny.cuny.edu/blog/

[8] Ascending Technologies, http://www.asctec.de/

[9] Oblong,g-speak motion capture platform. http://www.oblong.com

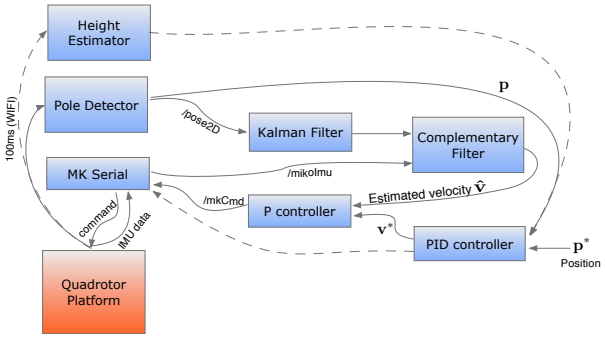


Fig. 7 Software implementation using ROS platform where blue boxes represent ROS nodes running on the ground station in real time and the orange box is the quadrotor platform. The prefix '/' denotes a ROS topic. $\hat{p}$ and p^* are estimated and desired position respectively. v denotes velocity and notation are same as position.

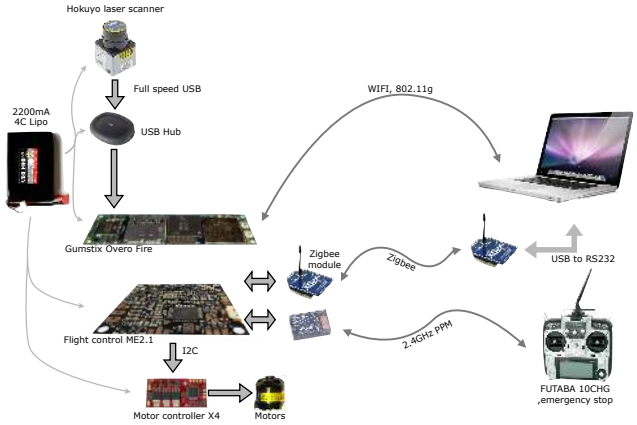


Fig. 8 Hardware integration. The laser scanner is attached to a USB Hub since the Overo Gumstix USB host only supports High Speed USB. The Zigbee module is used to transmit IMU data to the ground station and receive commands. The WiFi connection connects the ROS nodes on the Gumstix to the ground station. For safety a manual pilot transmitter is linked to the quadrotor system.

takeoff, the quadrotor moves a little horizontally due to poor trim but returns quickly to the desired hovering position. Figure. 9 shows the estimated horizontal velocities compared to the ground truth. The standard deviation values are $\{\sigma_{vx}\,\sigma_{vy}\} = \{0.0495, 0.0375\}\,\mathrm{m/s}$. Note that these values are calculated over the flight interval between $t = 30\,\mathrm{s}$ (takeoff) and $t = 70\,\mathrm{s}$ (landing).

The vehicle position was estimated using the laser-range-finder, pole detector and Kalman filter and used in a PID controller to maintain the pole at a fixed range and bearing angle — hovering with respect to the pole. Ground truth data obtained from

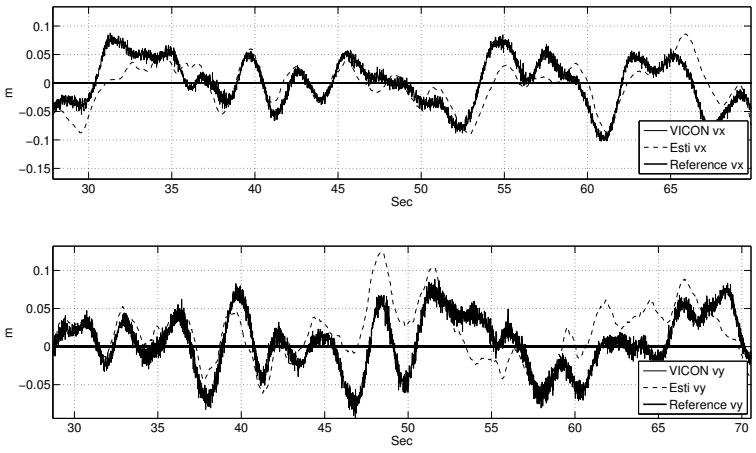


Fig. 9 The lateral velocities estimation results with respect to the pole while hovering. Solid line denotes the ground truth and dash indicates the complimentary filter velocity estimation output. Thick solid line is the reference.

the g-speak system is shown in Figure 10. The reference position of the vehicle is $(0,0,0.6)$ m. The standard deviations of the ground truth position are $\{\sigma_x\,\sigma_y,\sigma_z\} = \{0.0483,0.0455,0.0609\}$ m. These are again computed over the flight interval.

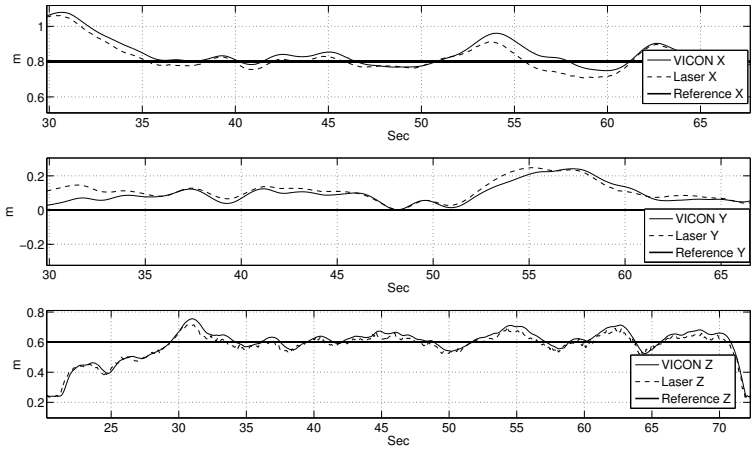
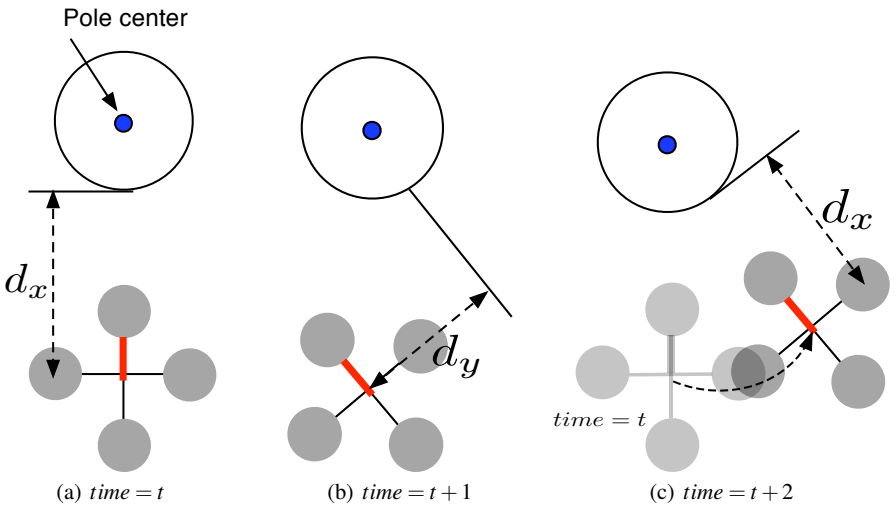


Fig. 10 x,y position estimation with respect to the pole while hovering with the ground truth. Solid line denotes the ground truth and dash indicates Kalman filter position estimation. Thick solid line is the reference. Median filter is used to estimate z position estimation.

If we yaw the vehicle while maintaining the pole at a fixed bearing, the result is motion around the pole as shown in Figure 11. Figure 12 shows the ground truth circle trajectory with the proposed shared control. A current limitation is that yaw angle is estimated from the vertical gyro and drifts with time. A video demonstration is available on our YouTube channel[10].



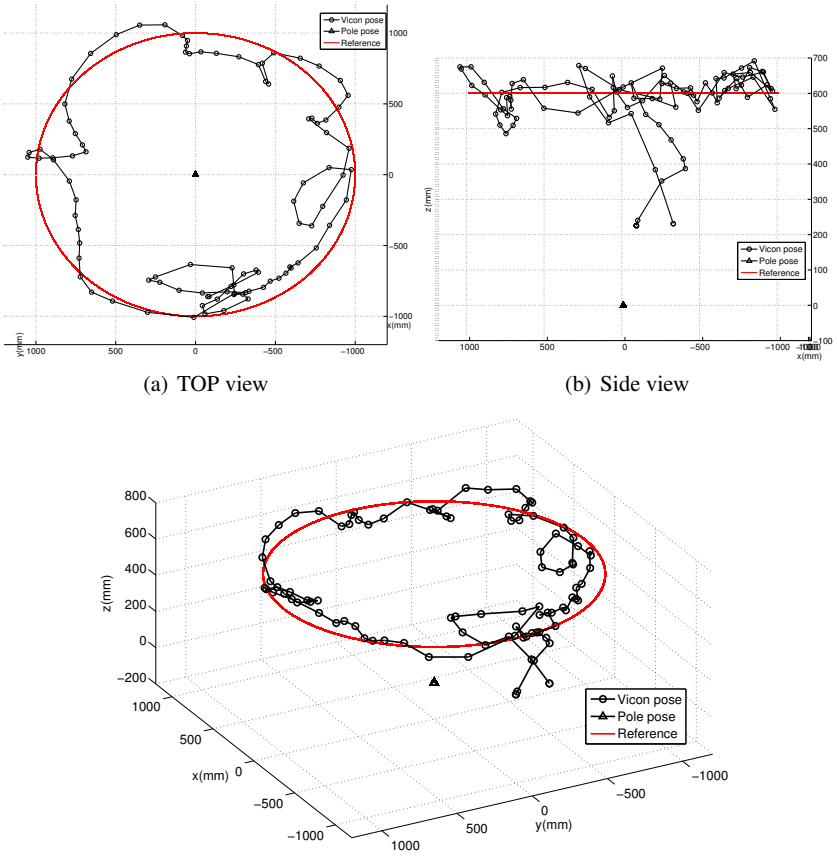
(a) *time* $= t$ (b) *time* $= t+1$ (c) *time* $= t+2$

Fig. 11 Changing yaw angle makes the quadrotor circle around the pole (red bar indicates the *front* rotor. References for x,y position controllers are d_x and 0 respectively. The robot hovers by keeping d_x distance at *time* $= t$. (b) An operator sends yaw command and it introduces d_y distance at *time* $= t+1$. (c) The robot moves to right to eliminate d_y and keeps d_x distance at *time* $= t+2$.

5 Conclusion and Future work

We have described our progress toward a shared control scheme that allows an unskilled operator to control a quadrotor easily and safely for a useful class of tasks. Translational velocity estimation is crucially important for quadcopter control and we have presented computationally efficient state estimation and control algorithms which allow for smaller onboard computers. We have demonstrated ground-truthed comparison of lateral velocity, position estimation while hovering and presented circle movement around a pole, done with a platform of less than one fifth the cost and with a laser scanner that scans four times more slowly than other comparable results in the literature.

[10] YouTube QUT Cyphy channel. http://youtu.be/F1vljjPIglg



(a) TOP view (b) Side view

(c) Perspective view

Fig. 12 The ground truth trajectory with shared control. An operator sends only yaw commands using a joystick and the quadrotor keeps the desired distance, d_x, d_y, d_z =[1, 0, 0.6] in metre, with the pole. Red denotes the reference. Note that only the ground truth trajectory is presented due to difficulty in estimating yaw angle with a low performance gyroscope.

We used an amateur-class quadcopter, and to achieve a high 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 knowledge about this platform are returned to the community through open documentation and software available online[11].

We have a large program of ongoing work. We are augmenting gyro-based yaw angle estimation with a magnetic compass and a visual compass. We are moving

[11] ROS QUT Cyphy wiki page http://www.ros.org/wiki/MikroKopter/Tutorials

to a higher performance onboard computer which allows us to move computational processes to the robot and eliminate the complexity, limited range and unreliability of the communications link. We are investigating upward looking sensors so the robot can manoeuvre around pole-top structures. Finally, we are investigating high update rate monocular camera (up to 125 Hz) with wide-angle field of view for fast estimation of robot and task-relative state.

Acknowledgements We would like to thank Liz Murphy for providing Split-Merge source code, and Timothy Gurnett for assistance while using the VICON system at the QUT Cube Lab.

References

1. Balaguer, C., Gimenez, A., Jardon, A. (2005) Climbing Robots' Mobility for Inspection and Maintenance of 3D Complex Environments, Autonomous Robots. **18**, 157–169
2. Sangbae K., and Spenko, M., Trujillo, S., Heyneman, B., Santos, D., Cutkosky, M.R. (2008) Smooth Vertical Surface Climbing With Directional Adhesion, Robotics, IEEE Transactions on. **24**, 65 –74
3. G. C. Haynes, Alex K., Goran L., Jon A., Aaron S., Alfred A. R., D. E. K., (2009) Rapid Pole Climbing with a Quadrupedal Robot, IEEE International Conference on Robotics and Automation (ICRA). 2767-2772
4. Xu, F., Wang, X., Wang, L., (2011) Cable inspection robot for cable-stayed bridges: Design, analysis, and application, Journal of Field Robotics. **28**, 441–459
5. Voigt, R., Nikolic, J., and Hurzeler, C., Weiss, S., Kneip, L., Siegwart, R. (2011) Robust Embedded Egomotion Estimation, International Conference on Intelligent Robots and Systems (IROS). 2694 -2699
6. Sheridan, Thomas B. (1992) Telerobotics, automation, and human supervisory control, The MIT Press.
7. Baeten, J., Bruyninckx, H. and De Schutter, J. (2002) Shared control in hybrid vision/force robotic servoing using the task frame , International Conference on Intelligent Robots and Systems (IROS). 2128–2133
8. Corke, P. (2011) Robotics,Vision and Control Fundamental algorithms in MATLAB. , Springer.
9. Pounds, P. and Mahony, R. (2009) Design principles of large quadrotors for practical applications. IEEE International Conference on Robotics and Automation (ICRA). 3265 –3270
10. Abeywardena, D., Kodagoda, S., Munasinghe R. and Dissanayake G. (2011) A Virtual Odometry for a Quadrotor Micro Aerial Vehicle. Australasian Conference on Robotics and Automation.
11. Sa, I. and Corke, P., (2011) System Identification, Estimation and Control for a Cost Effective Open-Source Quadcopter, IEEE International Conference on Robotics and Automation (ICRA) (Accepted).
12. Roberts, M., Corke, P. and Buskey, G. (2003) Low-cost flight control system for a small autonomous helicopter. IEEE International Conference on Robotics and Automation (ICRA). 546–551.
13. R. E. Kalman., (1960) A New Approach to Linear Filtering and Prediction Problems. , Transactions of the ASME Journal of Basic Engineering. **82**, 35–45
14. Corke, P. (2004) An Inertial and Visual Sensing System for a Small Autonomous Helicopter. J. Robot. Syst. **21**, 9
15. Borges, G.A., Aldon, M.-J. (2000) A split-and-merge segmentation algorithm for line extraction in 2D range images . International Conference on Pattern Recognition. 441–444
16. Bouabdallah, S., Siegwart, R., (2007) Full control of a quadrotor, International Conference on Intelligent Robots and Systems (IROS). 153-138