Omnidirectional Vision based Surveillance with Spincopter

Tomislav Haus, Matko Orsag, and Stjepan Bogdan
Faculty of Electrical Engineering and Computing
University of Zagreb
Croatia
Email: tomislav.haus@fer.hr, matko.orsag@fer.hr, stjepan.bogdan@fer.hr

***Abstract*—The goal of the Spincopter project was to design an energy efficient aircraft capable of autorotation glide that could be used for both indoor and outdoor surveillance. This paper demonstrates how to utilize the self-rotating capability of the spincopter in order to acquire a 3D image of its environment. The omnidirectional capabilities of such an UAV extends the otherwise very limited field of view of conventional cameras usually used for aerial photography and surveillance. The paper presents the theoretical background and the actual implementation of such a system. It presents the resulting images and offers a brief survey of the image quality.**

I. Introduction

In the course of the past several years we have designed, simulated and experimentally verified several prototype versions of a novel unmanned aerial vehicle design, called *The Spincopter*. The goal of this project was to design an energy efficient aircraft capable of autorotation glide that could be used for both indoor and outdoor surveillance. The end result was a low-cost, self-rotating UAV that weighs around 1kg.

This paper demonstrates how to utilize the self-rotating capability of the *Spincopter* in order to acquire a 360° view of its environment.The self-rotation of the aircraft makes a single, conventional camera attached to its body an omnidirectional camera. There exists a wide range of potential use for such a vehicle with imaging capabilities: monitoring of traffic, crops and pollutions; crime and border surveillance; coastguard, fisheries and environment protection; security of national assets, transport & communications, etc.

The omnidirectional vision has, for a long time, been the topic of interest in the robotics community [1]–[3]. It extends the otherwise very limited field of view of conventional cameras. The most common implementation uses the so called Fish-Eye lenses [4]. With its short focal length, such a lens enables hemisphere view for the camera. Though it has been proven to work, this approach introduces to much distortion in an image. Another commonly used approach, dubbed Catadioptric vision, uses reflecting surfaces to achieve a 360° field of vision [5]. Aside from introducing distortions in an image, these cameras are usually very difficult to adjust and tune.

Stitching the images from a rotating camera is not a novel approach. It has been proposed and tested by several research groups [6], [7]. The main drawback of this imaging system is that it requires accurate positioning and a relatively fast camera. As the technology progresses, such a system becomes ever more feasible. In the paper we present a mid-cost imaging system based on the self-rotating camera.

The paper is structured in two parts. First, in Sec. II a *Spincopter* review is given, implemented control structure for both vertical and horizontal motion is described (Sec. II-A), as well as a system for vehicle's estimation that maximally utilizes onboard hardware (Sec. II-B). The emphasis is put on achieving fast attitude estimation since its rate directly influences horizontal motion performance and image acquisition. In Sec. II-C experimental results of *Spincopter's* vertical and horizontal motion control are presented. In second part, Sec. III presents a panoramic vision system that utilizes vehicle's self-rotation. A camera choice is analysed (Sec. III-A), experimental testbed presented (Sec. III-B) and acquired images shown. Finally, a comparison of images taken with rotating and still camera is conducted (Sec. III-C) and conclusion given.

II. Spincopter Review

In this paragraph a review of an unmanned aerial vehicle (UAV) called *The Spincopter* is given. It is a relatively novel concept first presented in [8]. It extends the basic ideas of a concept called *Samara*. Comparison between them is given in [9]. Unlike *Samara*, it is comprised of two wings and two propulsion motors placed symmetrically against vehicle's *yaw* axle, alongside two other main principal axis. Similar to *Samara*, there is also central fuselage with microcontroller unit, power electronics boards and battery placed inside. For a surveillance task we plan to mount a conventional camera and a wireless unit for image transfer. First we present implemented control structure and Kalman filter, which are essential for vehicle's motion performance.

A. Implemented control structure

Dominant motion for this type of vehicle is rotation around its *yaw* axle. Corresponding momentum is a result of thrust forces produced by propulsion motors, acting on some radius from the main rotational axle. Due to the spinning, lift forces

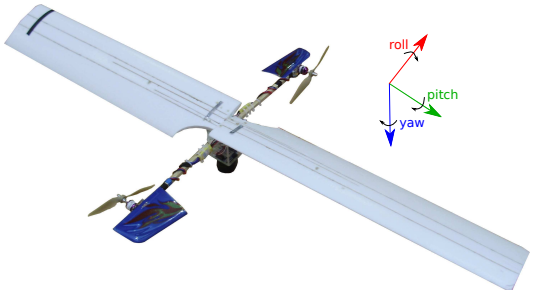

Fig. 1: Constructed vehicle

TABLE I: *Spincopter's* specifications

Control Board	*ArduPilotMega 2.0*
Additional Sensors	Sonar *MB1000 LV*
Communication	*XBee*
Battery	*Parrot* 1000mAh
Span (cm)	180
Weight (g)	1000

Motor

Type	*jDrones* BDLC
ESC	*jDrones* 20A
Voltage (V)	7.5 - 15
Max Power (W)	140
Max Thrust (g)	1100

Wing

Length (cm)	90
Width (cm)	20
AoA (°)	14

are generated along the wings. These forces cause vertical thrust and consequently vertical motion. Horizontal motion is produced by generating differences between the two motor thrusts during the specific periods of spinning cycles. As it can be seen, the vehicle is underactuated which results in coupling between horizontal and vertical motion. Basic nonlinear mathematical model of this concept and simulation results are given in [8], while more detailed control structure and results from both simulations and experiments are presented in [9]. Constructed vehicle is depicted in Fig. 1 while its specifications are given in Table I.

While in previous work inherent stability has been assumed ([8]) and proved in [9], real experiments have shown some problems. Although stability through these experiments has been confirmed, issues have been faced with undesired rotations in *roll* and *pitch* axis. Perfect symmetrical construction is assumed in early mathematical models but real system shows some symmetry deficiencies, resulting in change of rotational axis orientation. The main problem is the difference between total lift forces generated on each wing which is mainly the result of slight differences between the angles of attack. This difference produces dynamic *roll* and *pitch* momentum which eventually vanishes in undesired stable equilibrium state. In this state a significant amount of lift force is lost in lateral direction thus producing undesired oscillations in horizontal plane while decreased total vertical thrust fails to lift the vehicle beyond the *ground effect* area.

Problem was solved by introducing an additional control loop which utilises unidirectional coupling between vehicle's *pitch* and *roll* angles - a small change in *pitch* produces difference in wing's angle of attack thus increasing the total lift force of one wing and decreasing of the other. This finally results in *roll* angle change. To control vehicle's *pitch* angle, propulsion motors are inclined with respect to the main rotational plane thus creating additional vertical forces and *pitch* momentum. Introducing a simple PID controller with estimated *roll* angle as a feedback signal we are able to stabilize *roll* and *pitch* within $\pm 2^\circ$ [9]. After that, a control loop for vertical motion is implemented together with an algorithm for horizontal motion. Autonomous horizontal movement has not yet been achieved due to the lack of adequate sensors. All experiments have been conducted indoors so GPS can't be used, while some other sensors, such as a optical flow sensor, can't be utilised with this type of vehicle. Complete control structure implemented so far is shown in Fig 2. While in [9] the experimental results of attitude stabilization have been shown, here we present a Kalman filter used for vehicle's attitude and position estimation along with the results of vehicle's position control.

B. Attitude and position estimation

Quality attitude estimation is important for this type of vehicle. As stated earlier, estimated *roll* signal is required for stabilization control loop, while *yaw* signal is essential for horizontal movement algorithm and camera triggering as shown in Sec. III. For these tasks, estimator rate is critical for good performance.

Typical sensors used for attitude determination are gyroscope, accelerometer and magnetometer. Usually they are integrated on a single board called *Inertial Measurement Unit* (*IMU*). All signals are measured in a vehicle's moving coordinate frame so adequate mathematical model describing attitude change must be introduced. To optimally utilise each measurement a process called sensor fusion is undertaken. It usually includes a Kalman filter as an optimal stochastic estimator over mathematical model of sensors. Basic idea is to exploit complementary sensor features. For example, angle gained from integrating gyroscope measurements has a high drift component (low frequency error), while magnetometer measurement suffers mostly from high frequency noise [10].

In this project we use *ArduPilotMega 2.0* (*APM*) board with *Invense*® *MPU6000* chip. This chip integrates gyroscope, accelerometer and *Digital Motion Processing Unit*™ which calculates and outputs attitude in the form of quaternion at the maximum rate 200 Hz. Chip doesn't include magnetometer which results in *yaw* angle drifting. Magnetometer is integrated on *APM* board, outside the chip, so we introduce an additional Kalman filter for *yaw* estimation. In the actual implementation we use estimated *roll* and *pitch* outputs from *MPU6000* chip, using these estimates transform gyroscope and magnetometer measurements in main rotational plane and run algorithm described in Sec II-B1. This procedure enables estimates at frequencies close to 200 Hz. On the other hand, Kalman filter for complete attitude estimation

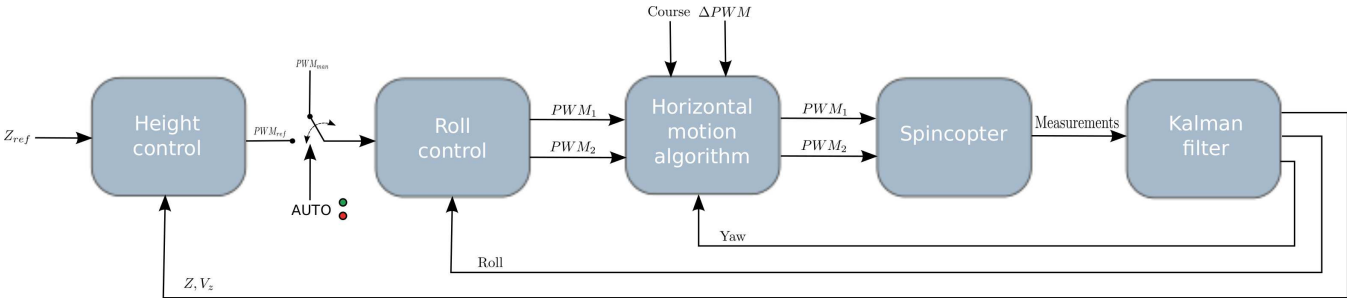

Fig. 2: Spincopter's implemented control structure

could be used, but it would result in much lower rate, mostly the consequence of 6×6 matrix algebra [11] [12].

1) Kalman filter for yaw estimation: Attitude estimation is usually performed using a Kalman filter over mathematical models of sensors rather than mathematical model of vehicles [12]. The indirect feedback filter version is utilized [10].

The first step in *yaw* estimation is transformation of gyroscope measurements in *yaw* ax, knowing vehicle's *roll* and *pitch*:

$$\omega_m = -s\phi \cdot g_x + s\phi c\theta \cdot g_y + c\phi c\theta \cdot g_z \tag{1}$$

where ω_m is *yaw* rate, g_x, g_y, g_z are scaled gyroscope measurements, s and c represent sine and cosine functions, ϕ and θ are *roll* and *pitch* angles. Calculated value is approximated as a standard gyroscope measurement with a certain level of high frequent noise and systematic error (bias), which is included in the model as a state:

$$\omega_m = \omega_t + b_t + n_w = \dot{\psi}_t + b_t + n_\omega, \tag{2}$$

where ω_m, ω_t are the measured and the true rotational speed, θ_t is the true *yaw* angle, b_t is the true *bias*, n_ω is the noise modelled as Gaussian white noise with expected value zero and some variance.
Bias is modelled as a constant value with some noise:

$$\dot{b}_t = 0 + n_b \tag{3}$$

where n_b is the modelled *bias* noise which represents estimate uncertainty. It has an important role in estimation as it enabled escape from the undesired equilibrium state.
Written in the state model form, we get:

$$\frac{d}{dt}\begin{bmatrix}\psi_t \\ b_t\end{bmatrix} = \begin{bmatrix}0 & -1 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\psi_t \\ b_t\end{bmatrix} + \begin{bmatrix}1 \\ 0\end{bmatrix}\begin{bmatrix}\omega_m\end{bmatrix} + \begin{bmatrix}n_\omega \\ n_b\end{bmatrix}. \tag{4}$$

Here we ignore negative noise sign as it has no influence on statistical properties. Noise modelled as Gaussian white noise can't be predicted. An inertial sensor model is [10]:

$$\frac{d}{dt}\begin{bmatrix}\psi_i \\ b_i\end{bmatrix} = \begin{bmatrix}0 & -1 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\psi_i \\ b_i\end{bmatrix} + \begin{bmatrix}1 \\ 0\end{bmatrix}\begin{bmatrix}\omega_m\end{bmatrix}. \tag{5}$$

The inertial model represents integration process with known *bias* value assumed.

In order to get the desired indirect feedback Kalman filter, we substract (5) from (4) and get:

$$\frac{d}{dt}\begin{bmatrix}\Delta\hat{\psi} \\ \Delta\hat{b}_i\end{bmatrix} = \begin{bmatrix}0 & -1 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\hat{\psi} \\ \Delta\hat{b}_i\end{bmatrix} + \begin{bmatrix}n_\omega \\ n_b\end{bmatrix}, \tag{6}$$

Difference between the measured *yaw* value ψ_m and the value calculated after integration ψ_i is a measurement signal for this model:

$$\Delta z = \psi_m - \psi_i = \psi_t + n_\theta - \psi_i = \Delta\psi + n_\psi, \tag{7}$$

where n_ψ is the noise modelled again as a Gaussian white noise.
Written in the state model form:

$$\Delta z = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\Delta\hat{\psi} \\ \Delta\hat{b}\end{bmatrix} + n_\psi \tag{8}$$

Estimated values are obtained with a standard Kalman filter procedure - first we calculate prediction values followed by measurement correction:

$$\frac{d}{dt}\begin{bmatrix}\Delta\hat{\psi} \\ \Delta\hat{b}\end{bmatrix} = \begin{bmatrix}0 & -1 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\Delta\hat{\psi} \\ \Delta\hat{b}\end{bmatrix} + \begin{bmatrix}K_1 \\ K_2\end{bmatrix}(\Delta z - \Delta\hat{\psi}), \tag{9}$$

where K_1 and K_2 are Kalman gains calculated according to the well known formulas. To get a compact form of indirect Kalman feedback version, we insert $\Delta\hat{\psi} = \hat{\psi} - \psi_i$ in Equ. (9). Using Equ. (4) and Equ. (5), we get:

$$\frac{d}{dt}\begin{bmatrix}\hat{\psi} \\ \hat{b}\end{bmatrix} = \begin{bmatrix}0 & -1 \\ 0 & 0\end{bmatrix}\begin{bmatrix}\hat{\psi} \\ \hat{b}\end{bmatrix} + \begin{bmatrix}1 \\ 0\end{bmatrix}\omega_m + \begin{bmatrix}K_1 \\ K_2\end{bmatrix}(\psi_m - \hat{\psi}). \tag{10}$$

Measured *yaw* value is determined from magnetometer measurements as follows:

$$\begin{bmatrix}M_x \\ M_y\end{bmatrix} = \begin{bmatrix}c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\theta s\psi \\ c\theta s\psi & s\phi s\theta s\psi - c\phi c\psi & c\phi s\theta s\psi - s\theta c\psi\end{bmatrix}\begin{bmatrix}m_x \\ m_y \\ m_z\end{bmatrix}$$
$$\theta_m = \text{atan}(\tfrac{M_y}{M_x}) \tag{11}$$

where s and c again represent sine and cosine functions, ϕ, θ are known *roll* and *pitch* angles, ψ is yaw angle calculated from inertial model, m_x, m_y, m_z are magnetometer measurements. Dynamic changes due to the dependency on estimated angles are neglected [10].

Since the Kalman filter should be executed on microcontroller, it is required to derive expressions in discrete form. We assume constant sample time T. System transition matrix,

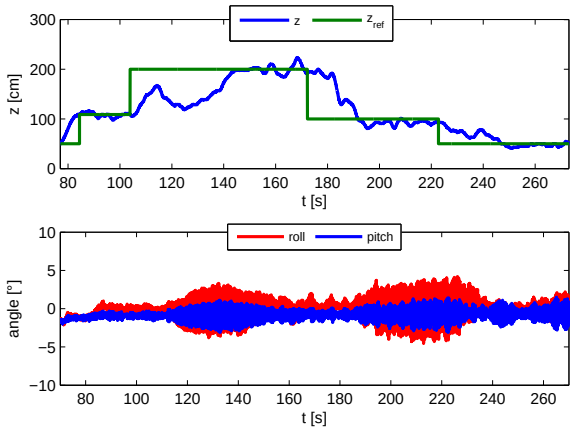

Fig. 3: Height control responses

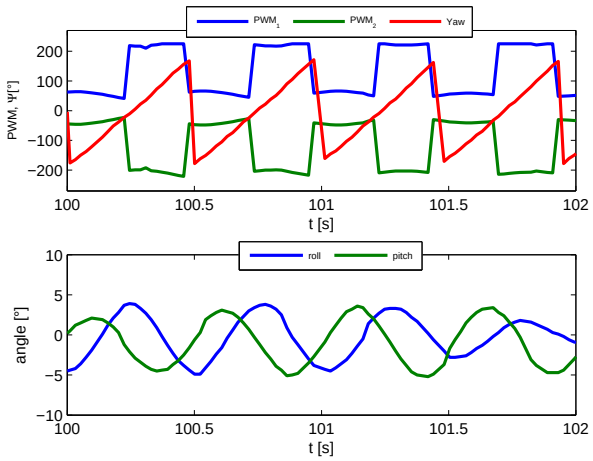

Fig. 4: PWMs and Euler angles during horizontal movement

corresponding to the state matrix $\mathbf{A}$ in the continuous form (Equ. (6)), is:

$$\Phi = e^{\mathbf{A}T} = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix}. \tag{12}$$

System input matrix Γ is [13]:

$$\Gamma = \int_0^T e^{\mathbf{A}\zeta} d\zeta \mathbf{B} = \begin{bmatrix} T \\ 0 \end{bmatrix}. \tag{13}$$

These matrices are derived by development of the exponential function in a Taylor series. It can be shown that the second exponent of the state matrix $\mathbf{A}$ equals null due to the nillpotent property [14].

Output matrix remains the same as in the continuous form (Equ. (8)):

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}. \tag{14}$$

2) Kalman filter for position estimation: Model for this filter basically follows the same rules as the one for *yaw* angle. It is based on integration model, however accelerometer, sonar, barometer and/or GPS are used instead of a gyroscope and a magnetometer (accelerometer measurements are transformed in static coordinate frame). Furthermore, dynamic dependency upon estimated angle is neglected [10]. A cascade of Kalman filters for orientation and position is obtained as a final result. It can be shown that this cascade is stable if every single filter is stable [15].

C. Experimental results

We have tested *Spincopter* height control with a classical cascade structure comprised of PI controller in inner and P controller in outer loop (Fig. 2). PWM value calculated in outer loop enters *roll* control algorithm, which calculates PWM differences and adds it to the output of the height controller. The result is the successful control of vehicle's height, which can be seen in Fig. 3. In the first plot, vehicle's measured height response to step reference change, is shown. In the second, transient response of vehicle's *roll* and *pitch* angles are presented.

Horizontal control algorithm, described in details in [9], is based on motor thrust change during the specific periods of spinning cycle. We call this process pulsation. Since vertical components of motor thrusts are used for stabilization, the idea is to keep the mean thrust values constant during one spinning period. To achieve that, PWM value is increased for one motor and decreased for the other during the first half of a rotation period and opposite is done during the second half. The function, which describes dependency of the pulsation period beginning in terms of *yaw* angle with respect to the desired direction, has been experimentally determined in [9]. In the first plot of Fig. 4 scaled PWM values together with estimated *yaw* angle are shown during pulsation process. In the second plot, *roll* and *pitch* angles are shown during this process. It can be seen that their values stay inside $\pm 5°$. Video clip, showing *Spincopter's* vertical and horizontal motion, can be seen in [**?**].

III. Omnidirectional vision with Spincopter

In this section we analyse the possibility of mounting a camera on the vehicle. This means the on-board camera would spin at vehicle's rotational speed (around 125 rpm). There are several projects dealing with a rotating camera but in these projects the camera is first rotated for some angle and then images are taken while the camera is held still [18]–[20]. In our approach, images are taken while the camera is spinning. Hence, there is no need for additional actuators for camera movement. However, depending on the type of camera this could cause various motion effects.

A. Camera analysis

There are two types of the camera sensors on the market: *complementary metaloxidesemiconductor* (CMOS) and *charge-coupled device* (CCD). In case of *Spincopter* application, their sensitivity to relative motion between camera and the scene is of the most importance. This sensitivity is related to the exposition, i.e. the type of a shutter used in a camera.

There are two main types of the shutter: rolling and global. A camera with the rolling shutter exposes different parts of the single frame at different points in time. Hence, if there is a relative motion between the camera and the objects in the scene, motion effects such as *wobble*, *skew* and *partial exposure* appear. On the other hand, a camera with the global shutter exposes a whole frame at the same time thus providing an image without these effects. There are few effects that are specific for CCD camera images like *smear*, but their appearance largely depends upon scene properties [21].

The most important feature for image taken with a moving camera is the level of motion blur. There are many motion deblurring algorithms developed, however they are computationally too slow for our application [22]. Hence, the main idea is to increase the shutter speed to avoid initial creation of blurry image. In general, both CMOS and CCD cameras can have adjustable shutter speed, but most CMOS cameras found on the market have a rolling shutter.

Having all that in mind we have chosen *Smartek® GC1621M* ethernet camera with CCD sensor. The most important camera feature is adjustable shutter speed in range $10\,\mu\text{s} - 10\,\text{s}$. With shutter speed decreased down to $500\,\mu\text{s}$, camera rotational speed $125\,\text{rpm}$ and object distance $5\,\text{m}$, relative camera displacement during the sensor exposure is $3.5\,\text{mm}$. In the following sections we will show that this setup ensures almost blurriless image. Another useful feature this camera offers is image triggering (software and hardware). Due to the limited bandwidth, the idea is to trigger camera at desired azimuths. Taking into account camera's field of view (FOV), we can capture $\lceil \frac{360}{FOV} \rceil$ pictures and combine them to get a 360° view. For the chosen camera and optics, FOV is about 45° which requires at least 8 pictures. To achieve a larger overlapping area, which might be required for image stitching algorithm, 10 pictures per rotation should be taken.

B. Experimental testbed

In order to confirm feasibility of capturing 360° view before setting the camera on-board the vehicle, we have built the an experimental testbed, depicted in Fig. 5. The testbed contains speed controlled DC motor with mounted camera, ethernet-to-wifi bridge for image transfer and *APM* board with *XBee* wifi module.

Kalman filter and software for camera triggering are executed on *APM* board. There is a two channel communication with the ground station computer: XBee for attitude data and standard IEEE 802.11a for image data. Software modules for the ground station computer are implemented on *Robot Operating System (ROS)* framework. Developed architecture is depicted in Fig. 6.

C. Image quality assessment

Developed GUI, showing captured 360° view represented with 10 images, is depicted in Fig. 7. The experimental testbed in action has been recorded with another camera and can be seen in [**?**]. Two main image features are discussed: quality in terms of motion blur level and frame rate.

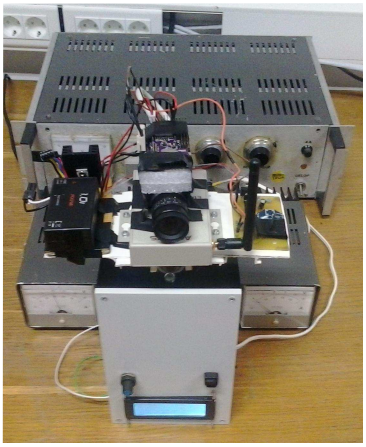

Fig. 5: Experimental testbed

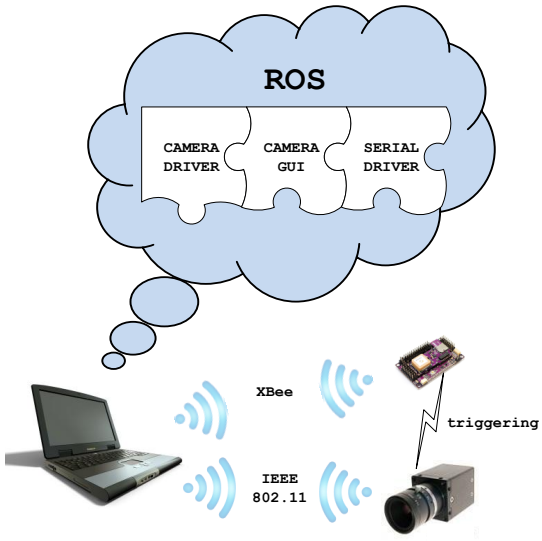

Fig. 6: Hardware and software architecture

From the presented images and video we can see very small level of blur in images taken with rotating camera. Such small blur level is neglectable for image processing algorithms. To confirm that we have carried out and compared the fundamental contour detection algorithm in images taken with rotating camera and camera holding still. The algorithms are implemented using OpenCV functions. First we use Canny function for edge detection followed by contour finding algorithm. The comparison is carried out over 10 images presenting 360° view. In Fig. 8 we present results as percentage difference of pixels number detected as edges, number of detected contours and their total length and area. The biggest difference appears in the total number of contours detected and their total area (up to 30% and 40% in worst case, respectively). In these cases, due to the motion blur, there are more breakups in detected contours which

Fig. 7: 360° view taken with rotating camera

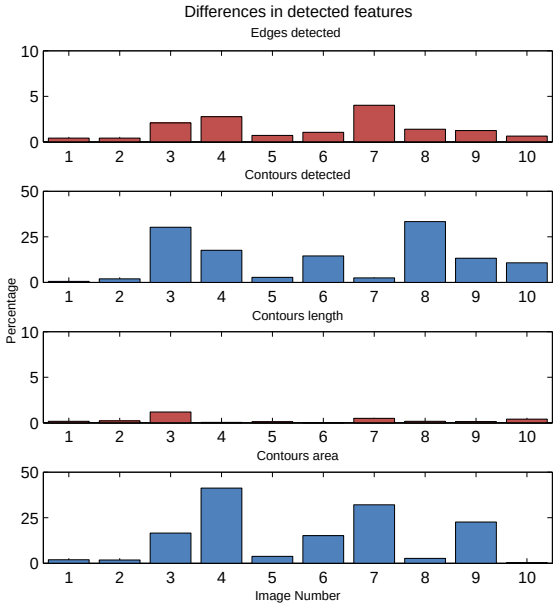

Fig. 8: Feature comparison of images taken with rotating and still camera

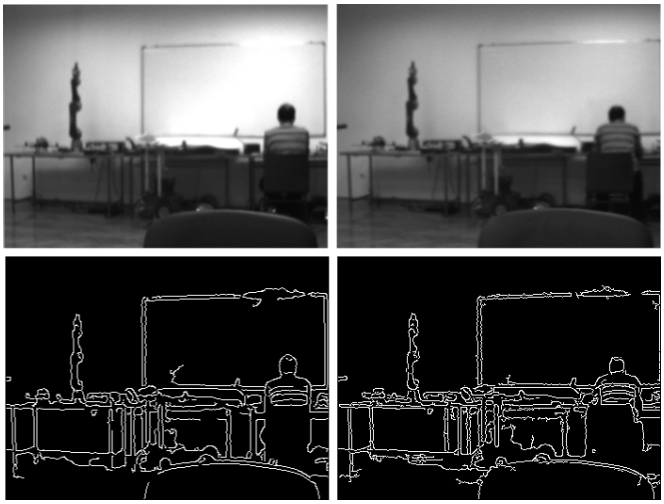

Fig. 9: Comparision of images taken with rotating camera (right) and camera holding still

results in their larger total number and smaller areas they encompass.

Next we present average case of difference between the two images, one taken with rotating and the other with still camera. Detected contours of both images are shown in the second row in Fig 9. Next, we present contours' area and length histograms as image distortion measure, shown in Fig. 10. Total area and length values are shown in Table II.

TABLE II: Contour comparison data

	Edges detected	Contours detected	Contour area	Contour length
Still camera	5261	74	6359	8352
Rotating camera	5707	123	6197	7584

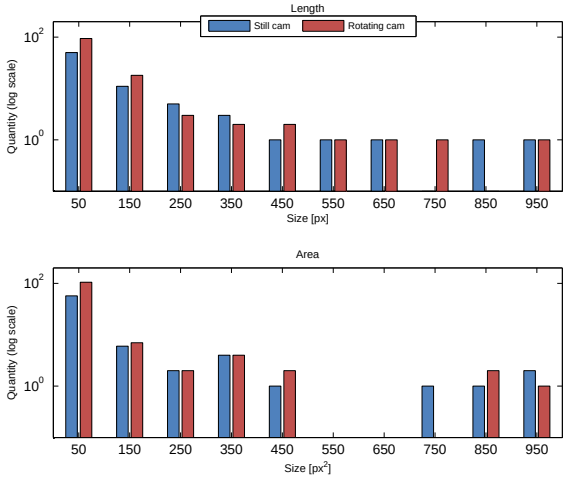

Fig. 10: Contours' area and length histograms

In the presented case, there are more pixels detected as edges in rotating camera image (difference 8%) which results in a larger number of contours detected. However, from the presented histograms (Fig. 10) we see that majority of these overdetected contours are of small lengths and areas and can be further filtered with increased Canny threshold. Total contour area and length detected with rotating camera are 2% and 9% less respectively, which represent the average difference levels.

Finally, we discuss frame rate of this panoramic view which equals *Spincopter's* rotational frequency, i.e. at operational velocity of 125 rpm, the rate is around 2 fps. This might be too low for highly dynamic environments but, as an example, for surveillance of fire caught areas is high enough. Higher rotational velocity, which we expect due to the added mass after camera is mounted on the vehicle, would increase frame rate.

IV. Conclusion

In this paper we presented a development of the UAV concept called *The Spincopter*. Constructed vehicle together with the implemented control and estimation algorithms are described. Fast attitude estimation is based on synergy of special hardware unit (*MPU6000* chip) and custom Kalman filter for *yaw* estimation. Control structure is comprised of classical cascade height controller, *roll* angle stabilization loop and horizontal motion controller. The concept of vertical and horizontal motion control, based on motors' speed pulsation, is experimentally confirmed and presented. Next, we have shown how to utilize vehicle's self-rotation to obtain an omnidirectional vision system with a conventional camera. We have chosen the CCD camera with controllable trigger speed and have shown that the chosen setup ensures almost bluriless images. Our subjective image quality assessment is confirmed with comparison of fundamental algorithms on images taken with rotating and still camera. Future work includes mounting the camera on the vehicle, adding additional board with higher computational power for on-board image compression and processing and utilizing visual feedback for horizontal position estimation. This would finally make this vehicle fully autonomous.

Acknowledgements

This project was supported by the European Community Seventh Framework Program under grant No. 285939 (ACROSS).

References

[1] K. Daniilidis and C. Geyer, "Omnidirectional vision: theory and algorithms," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 1, 2000, pp. 89 –96 vol.1.

[2] C. Drocourt, L. Delahoche, C. Pegard, and A. Clerentin, "Mobile robot localization based on an omnidirectional stereoscopic vision perception system," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 2, 1999, pp. 1329 –1334 vol.2.

[3] S. K. Nayar, "Omnidirectional vision," in *In Proc. of ISRR1997*, 1997, pp. 1–11.

[4] S. J. Oh and E. L. Hall, "Guidance of a mobile robot using an omnidirectional vision navigation system," pp. 288–300, 1987. [Online]. Available: + http://dx.doi.org/10.1117/12.968260

[5] J. Murphy, "Application of panospheric imaging to a teleoperated lunar rover," in *Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference on*, vol. 4, oct 1995, pp. 3117 –3121 vol.4.

[6] A. Krishnan and N. Ahuja, "Panoramic image acquisition," in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, jun 1996, pp. 379 –384.

[7] J. Zheng and S. Tsuji, "Panoramic representation of scenes for route understanding," in *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, vol. i, jun 1990, pp. 161 –167 vol.1.

[8] M. Orsag, S. Bogdan, T. Haus, M. Bunic, and A. Krnjak, "Modeling, simulation and control of a spincopter," in *IEEE International Conference on Robotics and Automation, ICRA 2011, Shanghai, China, 9-13 May 2011*. IEEE, 2011, pp. 2998–3003.

[9] M. Orsag, J. Cesic, T. Haus, and S. Bogdan, "Spincopter wing design and flight control," *Journal of Intelligent & Robotic Systems*, vol. 70, pp. 165–179, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10846-012-9725-2

[10] P. Maybeck, *Stochastic models, estimation and control*. Academic Pr, 1979, vol. 1.

[11] E. Lefferts and F. Markley, "Dynamic modeling for attitude determination," in *AIAA Guidance and Control, 1976. Proceedings. 1976 AIAA Conference*. AIAA, 1976.

[12] R. S.I., G. Sukhatme, and G. Bekey, "Circumventing dynamic modeling: Evaluation of the error-state kalman filter applied to mobile robot localization," in *Robotics and Automation, 1999 IEEE International Conference on*, vol. 2. IEEE, 1999, pp. 1656–1663.

[13] B. Kuo, *Automatic control systems*. Prentice-Hall, 1982.

[14] G. Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2003.

[15] J. Ryu, "State and parameter estimation for vehicle dynamics control using gps," Ph.D. dissertation, Stanford University, December 2004.

[16] T. Haus, "Spincopter height stabilization video," June 2012. [Online]. Available: http://vimeo.com/45385562

[17] ——, "Spincopter horizontal movement video," June 2012. [Online]. Available: http://vimeo.com/45385215

[18] H. Ishiguro, M. Yamamoto, and S. Tsuji, "Omni-directional stereo," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 257 –262, feb 1992.

[19] K. Sarachik, "Characterising an indoor environment with a mobile robot and uncalibrated stereo," in *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, may 1989, pp. 984 –989 vol.2.

[20] R. Benosman and J. Devars, "Panoramic stereo vision sensor," in *Pattern Recognition, 1998. Proceedings. Fourteenth International Conference on*, vol. 1, aug 1998, pp. 767 –769 vol.1.

[21] B. Green, "Sensor artifacts and cmos rolling shutter," October 2007. [Online]. Available: http://dvxuser.com/jason/CMOS-CCD/

[22] Z. Al-Ameen, G. Sulong, M. Johar, N. Verma, R. Kumar, M. Dachyar, M. Alkhawlani, A. Mohsen, H. Singh, S. Singh *et al.*, "A comprehensive study on fast image deblurring techniques," *focus*, vol. 44, 2012.