

# Position Based Visual Control of the Hovering Quadcopter

Atulya Shivam Shree, Radhe Shyam Sharma,  
Laxmidhar Behera<sup>(✉)</sup>, and K.S. Venkatesh

Indian Institute of Technology Kanpur, Kanpur 208016, UP, India  
{atulya,sharmars,lbehera,venkats}@iitk.ac.in

**Abstract.** Autonomous navigation of quadcopters in unstructured indoor environments is a major problem due to the difficulty of reliable position sensing. While outdoor applications can use GPS for reliable localization, working indoors will require the use of either laser range finders or some other sensors. If the indoor scene is unknown to a robot, the task of mapping new areas also becomes a necessity. The two processes are combined and run together in a framework of Simultaneous Localization and Mapping (SLAM). Our work is focused on using onboard cameras for the task of SLAM in an indoor scenario. Vision based techniques that do not use time of flight methods like laser range finders, have the potential to provide a low cost alternative framework for navigation. In this work, localization using a monocular SLAM framework on an unknown and unstructured scene, a cascaded position controller along with a Luenberger observer which can combine the data of Inertial sensors and vision based position to generate a complete velocity feedback for the system have been used. Sensor data fusion using EKF (Extended Kalman Filter) have been performed for scale estimation. The localization algorithm has been implemented on a quadcopter. Finally hovering experiment has been performed in an indoor lab based environment.

**Keywords:** Quadcopter · Autonomous · Navigation · Vision · VSLAM

## 1 Introduction

In recent years, quadcopters have gained a lot of popularity. There are numerous commercial products already developed and being sold in the market. The highly agile dynamics of a quadcopter allow it to easily takeoff and fly in any indoor or outdoor scenario. People are using them commercially for photography and for recording videos by mounting a camera on them. One of the major challenges in the field of micro aerial vehicles is to make them completely autonomous: this requires localization. In an outdoor environment it is possible to use GPS based position feedback to fly over a particular trajectory. But GPS does not work indoors and so while flying indoors, if no correction is provided by a manual operator the quadcopter will drift in any random direction due to the absence

of precise location information. This calls for using vision or lasers for localizing the robot in an indoor environment.

Passive systems like VICON or active systems system like phoenix are very popular choices for lab based environments. It has been shown in [1] that even highly complex maneuvers and trajectory tracking can be done provided we have a highly accurate position feedback system. However system relies on external tracking hardware and is therefore limited to a lab environment.

For complete autonomy in indoor operations, it is necessary to implement SLAM techniques onboard the system. This can also allow it to be able to navigate through initially unknown terrains. Highly accurate SLAM with dense mapping of the scene has been demonstrated in [2] where the authors have used laser along with vision to generate a pointcloud of the environment. While laser range finders provide highly accurate depth data they are generally heavy and expensive. Hence there is a need for developing the SLAM framework by using cameras as the primary sensors without using laser range finders. The research on vision based SLAM is particularly important because it is believed that this is how we humans explore our surroundings. Vision based sensors are now extremely cheap, and hence developing a good algorithm for state estimation can make a product commercially viable to the general public.

In order to stabilize a robot over any particular scene two methods have been used IBVS (Image Based Visual Servoing) and PBVS (Position Based Visual Servoing). IBVS has been used in [3], and it uses spherical image based error in between the desired image and current image to directly generate control signals. However, it requires knowledge of the desired scene beforehand and hence is restricted to use with pre-defined markers and visual patterns. PBVS on the other hand generates the current position feedback of the camera by tracking certain inherent salient features in the scene. This method continuously generates a sparse feature point map of the surroundings and simultaneously performs localization with respect to it. To implement complete vision based SLAM, Parallel Tracking and Mapping (PTAM) [4] which uses a single camera has been the most popular choice. In the work in [5] the authors have used an AR Drone platform for sending the camera stream wirelessly over a WiFi network and implementing PTAM on an offboard laptop computer. In spite of delays in transmission the authors have been able to successfully stabilize the Micro Aerial Vehicle (MAV). The major disadvantage of using monocular camera based algorithms is that the pose measurements are scaled with respect to the metric measurements. To solve this problem work has been done on using stereo SLAM for application to MAVs. A hybrid approach using a slow stereo camera set along with fast PTAM based odometry has been presented in [6]. While PTAM uses only one camera, with a stereo configuration it is also possible to obtain accurate scale corrections and depth map estimation. In [7] the authors use a bottom facing camera for velocity estimation using optical flow and a forward facing stereo configuration for performing complete mapping and exploration. In [8] two sets of stereo cameras have been used, in the front and the bottom direction to provide complete state metric localization and mapping. Another

technique for scale correction is to fuse the readings of vision with that of Inertial sensors. In [9] the readings of vision based monocular sensor have been combined with the readings of an IMU and Barometer to provide complete state feedback to the system.

In more advanced work [10–12] the authors have demonstrated complete state estimation by fusing the sensor readings of vision and IMU. The method does not use any other sensor and can provide metric state feedback by estimating the scale of the monocular SLAM system. However the system is very sensitive to initial values of scale and can fail to converge if the initial estimate is outside a certain bound. More recent work developed in 2014 include SVO (Semi-direct Visual Odometry) [13] which is a highly efficient and fast open source monocular SLAM algorithm. This algorithm has the ability to work at almost 60 fps on a computationally restricted onboard computer and even faster on consumer grade laptops. It achieves this by directly operating on pixel intensities for matching patches within different frames. This reduces the computational burden of extracting feature points at every new frame making the visual system more robust. This paper is mainly concerned with implementation of a vision based hovering framework on a quadcopter. A Luenberger observer has been designed which predicts the velocity using the accelerometer and the position feedback. For localization, monocular SLAM algorithm SVO has been used which is an open source package released for Robot Operating System (ROS) by [13]. This is followed by experimental demonstration of hovering over particular waypoints in 3D space.

The paper is organized as follows: sensors, observer & EKF are described in Sect. 2. Experimental results are provided in Sect. 3. Section 4 concludes the paper.

## 2 Localization and Position Control in Unstructured Environment

### 2.1 Hardware Description

Nayan quadcopter is used to perform experiment which is shown in Fig. 1. Description of its components are mentioned in Table 1.

**Table 1.** Description of Nayan hardware platform

Component	Description
Flight controller	ARM Cortex M4 32 Bit 168 MHz CPU
Onboard computer	Odroid
Camera	CMOS, Bluefox, $752 \times 480$
Ultrasonic sensors	Px4Flow
Battery	LiPo, 5200 mAh
Propeller	$11'' \times 4.5''$
Weight	1.9 Kg



Fig. 1. Nayan quadcopter

## 2.2 Sensor Description

**Inertial Measurement Unit.** The IMU is the most crucial element of the quadrotor and is used by the attitude controller for maintaining a desired orientation. It consists of a gyroscope which measures the angular velocity in the body frame and an accelerometer which measures acceleration in the body frame. The Rotation matrix at any time is obtained by fusing the data of accelerometer with that of the gyro. Ideally a gyro measures the angular velocity of the system and is sufficient to estimate the orientation angles  $\phi, \theta, \psi$ , given the initial estimate. However there is a small bias in the angular velocity measurements which leads to a drift in the attitude, thus calling the need for fusion. More details on the working of Inertial navigation systems can be found at [14]. For this experiment it is assumed that the attitude fusion has already been done in the LLP (Low Level Processor) and we have been provided with the roll, pitch and yaw angles in High Level Processor (HLP) where the custom user code runs. This is utilized for obtaining the acceleration in the NED-b frame from the body frame. The data of accelerometer is governed by the following equation:

$$z = R_g^b(a_I - g) + b + n_a \quad (1)$$

where,  $z$  is the data measured by the accelerometer,  $R_g^b$  is the rotation matrix from Inertial to Body frame,  $a_I$  is the acceleration in the inertial frame,  $b$  is a bias in the accelerometer,  $g$  is the acceleration of gravity,  $n_a$  is noise in the sensor readings. A very important assumption taken to simplify all sensor fusion steps is that the yaw angle has been considered to be fixed. Under this assumption it is necessary that the position feedback to the observer is the NED-body fixed frame. This also leads to a simplified  $R_g^b$  matrix as the yaw term is not considered. Hence after these assumptions the final equation for acceleration in the NED-B frame is:

$$a_{nedb} = \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \cos \phi \sin \theta \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} z_x \\ z_y \\ z_z \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 9.81 \end{bmatrix} \quad (2)$$

where,  $\phi$  &  $\theta$  are roll and pitch angle respectively. Note however that the above acceleration is not bias free and needs to be corrected before being further used.

**Ultrasonic Distance Sensor.** An ultrasonic sensor calculates the distance to an object by using time of flight data for a pulsed echo signal. In this experiment a Px4Flow board has been used comprising of an inbuilt ultrasonic sensor which provides direct metric measurements of the height of MAV from the ground. The data is sampled at a frequency of 10 Hz from the sensor.

### 2.3 Luenberger Observer

The output from the ultrasonic sensors is a depth reading available at 10 Hz. The VSLAM algorithm will also be able to give position coordinates at upto 25 Hz. The two are combined on the main computer and finally a metric position update is provided to the position controller running on the HLP of the flight controller. We also have the acceleration data coming in from the onboard IMU at a high frequency. With these available sensor data a luenberger observer has been designed similar to that used in [9, 15]. Each of the  $x, y, z$  coordinates are assumed to be an independent system with an order 3 state for each.

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ b_x \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ b_x \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} a_{nedb_x} \quad (3)$$

$$Y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ b_x \end{bmatrix} \quad (4)$$

Here,  $a_{nedb_x}$  is the acceleration in  $x$  axis which is taken as an input to this system, while  $(x, \dot{x}, b_x)$  are the states.  $Y$  is taken as the system output which in our case is available to the flight controller from the main computer. The states can similarly be defined for  $y, z$  and  $a_{nedb}$  is obtained from Eq. 2. Based on the above formulation a Luenberger observer is designed. For a system defined as:

$$\dot{x} = Ax + Bu, \quad Y = Cx$$

we have an observer,

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \quad (5)$$

The error  $e = x - \hat{x}$  converges to zero provided the eigen values of the matrix  $A - LC$  are all negative. Hence the values of the observer  $L$  are chosen appropriately.

### 2.4 Kalman Filter for Metric Position Feedback

**Position Feedback from SVO.** The visual subsystem can be thought of as a black box unit which ultimately gives the current pose as the feedback. In this

experiment, only the position data has been used for controls while the IMU which is more reliable for roll and pitch angles, is used for the attitudes.

$$\vec{r}_v = \frac{1}{\lambda}(\vec{r} - \vec{r}_0) \quad (6)$$

where,  $\vec{r}_v$  is the feedback coming from SVO,  $\vec{r}$  is the actual 3D coordinate,  $r_0$  is position at which the visual system was initialized,  $\lambda$  is some unknown scale factor. This output is with respect to an inertial frame which is aligned with the ENU-body frame at the initial point. At any time after initialization four states from the VSLAM are taken  $(x_{vi}, y_{vi}, z_{vi}, \psi_{vi})$ . These are first transformed to get the position in the gravity aligned body fixed frame. In the final step the position is transformed from the ENU (East North Up) coordinate system of SVO to NED (North East Down) for the NED-Body-fixed frame in which the control algorithm primarily works.

$$\vec{r}_{vb} = R_{vi}^{vb} \vec{r}_{vi}, \quad R_{vi}^{vb} = R_z(\psi) = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where,  $\psi$  is the yaw angle.

**EKF Framework for Estimation of Static States.** Since there are multiple sensory inputs for the height of the quadcopter, a fusion step using EKF is performed so as to estimate the static states  $\lambda, z_0$ . The different equations are:

$$z_v = \left( \frac{z - z_0}{\lambda} \right) + n_v, \quad z_u = z + n_u, \quad \ddot{z} = a_{nedbz} + b + n_a \quad (7)$$

where,  $z_v$  is the depth from vision sensors,  $z_u$  is the depth from ultrasonic sensors,  $z$  is the position of the quadcopter on  $z$  axis.  $\lambda, z_0$  are the static states for the VSLAM system,  $b$  is the static bias of the IMU.  $n_v, n_a, n_u$  are Gaussian noise with 0 mean and fixed variance. The state of the system at instant  $k$  is defined as

$$x_{k|k} = [z \ \dot{z} \ b \ \lambda \ z_0]^T \quad (8)$$

## Prediction

$$x_{k+1|k} = F_k x_{k|k} + B u, \quad P_{k+1|k} = F_k P_{k|k} F_k^T + V_k$$

where,

$$F_k = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B_k = [0 \ \Delta t \ 0 \ 0 \ 0]^T$$

### Update for VSLAM

$$\begin{aligned}\hat{z}_v &= \frac{1}{x_{k+1|k}(3)}(x_{k+1|k}(0) - x_{k+1|k}(4)), \quad v = z_v - \hat{z}_v \\ H_{v,k+1} &= \begin{bmatrix} \frac{1}{x_{k+1|k}(3)} \\ 0 \\ 0 \\ -\frac{(x_{k+1|k}(0) - x_{k+1|k}(4))}{x_{k+1|k}^2(3)} \\ -\frac{1}{x_{k+1|k}(3)} \end{bmatrix}^T \\ S &= H_{u,k+1}P_{k+1|k}H_{u,k+1}^T + W_u, \quad R = P_{k+1|k}H_{u,k+1}S^{-1} \\ x_{k+1|k+1} &= x_{k+1|k} + Rv, \quad P_{k+1|k+1} = P_{k+1|k} - RH_{u,k+1}P_{k+1|k}\end{aligned}$$

### Update for Ultrasonic Sensor

$$\begin{aligned}v &= z_u - x_{k+1|k} \\ H_{u,k+1} &= [1 \ 0 \ 0 \ 0 \ 0] \\ S &= H_{u,k+1}P_{k+1|k}H_{u,k+1}^T + W_u, \quad R = P_{k+1|k}H_{u,k+1}S^{-1} \\ x_{k+1|k+1} &= x_{k+1|k} + Rv, \quad P_{k+1|k+1} = P_{k+1|k} - RH_{u,k+1}P_{k+1|k}\end{aligned}$$

where,

$$P_{0|0} = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0.4 \end{bmatrix}, \quad V_k = \begin{bmatrix} 0.3^2 & 0 & 0 & 0 & 0 \\ 0 & 0.6^2 & 0 & 0 & 0 \\ 0 & 0 & 0.04^2 & 0 & 0 \\ 0 & 0 & 0 & 0.003\Delta t & 0 \\ 0 & 0 & 0 & 0 & 0.001\Delta t \end{bmatrix} \Delta t$$

here, standard notations are used for EKF. Symbol hat is used to represent the estimated parameter.  $W_u$  is taken as 0.02,  $W_v$  is taken as 0.1,  $\Delta t$  is taken as 0.02s which corresponds to the fact that the IMU update works at 50 Hz on the system.

## 2.5 Velocity Feedback Based Controller

This controller assumes that the position feedback it receives is in the NED-body frame. The controller used in this experiment is a cascaded PID controller. It comprises of a velocity feedback loop and a position feedback. The velocity feedback loop obtains the current velocity using the observer designed in Eq. 5. The position feedback runs at a frequency of 50 Hz and it mainly generates the desired velocity command for the inner velocity control loop.

$$\begin{aligned}e_x &= x_{ref} - \hat{x}, \quad e_{intgx} = \int e_x dt \\ v_{xref} &= K_{px}e_x - K_{dx}v_x + K_{ix}e_{intgx} \\ e_{vx} &= v_{xref} - \hat{v}_x, \quad e_{intgvx} = \int e_{vx} dt \\ a_{xref} &= K_{pvx}e_{vx} - K_{dvx}a_{nedbx} + K_{ivx}e_{intgvx}\end{aligned}$$

where,  $e_x$  is the position error in  $x$ ,  $e_{vx}$  is the velocity error in  $x$ . In subscript ref stands for reference, whereas hat is used to denote estimated parameter.  $a_{xref}$  is the acceleration in  $x$ . The final output of this controller is a desired  $a_{xref}$ .  $a_{yref}$  and  $a_{zref}$  are obtained in a similar fashion. The acceleration now needs to be converted to a desired attitude angle and thrust. The small angle assumption is taken

$$\ddot{x} = -g\theta, \quad \ddot{y} = g\phi, \quad \ddot{z} = g - \frac{U_1}{m}$$

where  $U_1$  is the thrust. Hence the equations for desired attitude angles can now be computed according to the inverse of the above relation.

$$\phi_d = \frac{a_{yref}}{g}, \quad \theta_d = -\frac{a_{xref}}{g}, \quad T_d = T_0 - a_{zref}$$

where

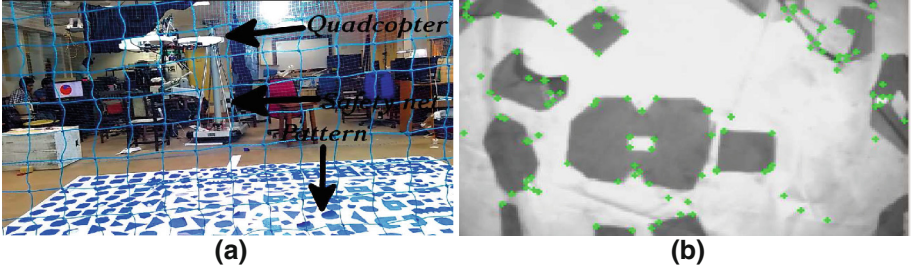
$$U_1 = b_{tc}(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$$

$b_{tc}$  is the thrust coefficient,  $\Omega$  denotes the angular velocity of the respective rotor.  $\ddot{x}$ ,  $\ddot{y}$  and  $\ddot{z}$  denote the accelerations in  $x$ ,  $y$  and in  $z$  direction respectively.  $m$  is the mass of the system.  $T_d$  and  $T_0$  represent the desired thrust and hovering throttle respectively.

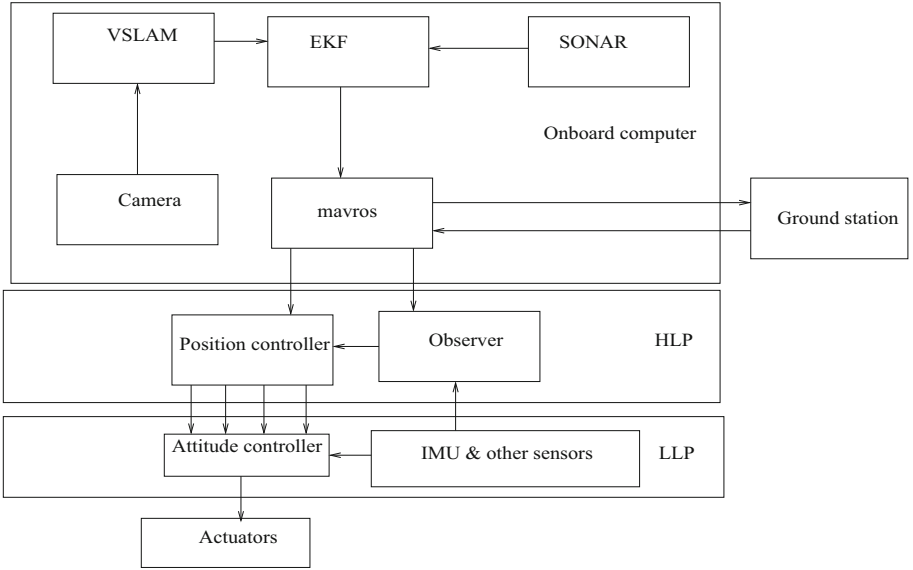
### 3 Experimental Results

In Fig. 2(a), we have included a picture of a real time experiment in which the quadcopter, safety net & pattern are marked. For experimenting indoors a pattern was placed on the ground. The pattern is used to localize quadcopter in a precise and robust manner using SVO. Features tracking using SVO is shown in Fig. 2(b). Natural features may or may not contain enough number of strong point features, which can potentially lead to tracking failure. That is why we have chosen an artificial pattern that is sufficiently rich in strong point features. A safety net exists around the operational area to keep other people safe in case of an accident. The entire code for this experiment runs on the onboard systems. We have chosen the values of Kalman filter parameters heuristically. Visual SLAM and Kalman state estimation run on the onboard computer whereas the Luenberger observer and the position controller run on the onboard HLP. The mavros driver is used to perform all required two way communication between the main onboard computer and the flight controller. In order to initiate the different ROS nodes running on the onboard computer, the user remote logs into the system via a WiFi network. The different nodes are then executed using ROS. There are five nodes running on the main onboard computer while one node runs on the desktop computer and is used for viewing the debug output of the VSLAM (Visual SLAM). The complete workflow and details of various nodes on ROS network are given in Fig. 3 and Table 2 respectively.





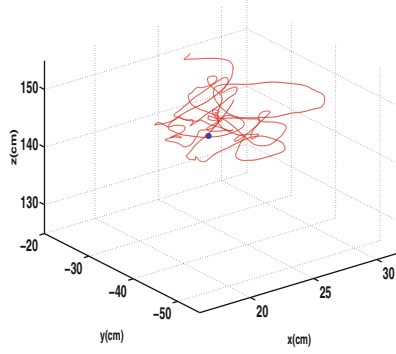
**Fig. 2.** (a) Indoor lab setup with hovering experiment in autonomous mode. (b) Feature tracking using SVO.



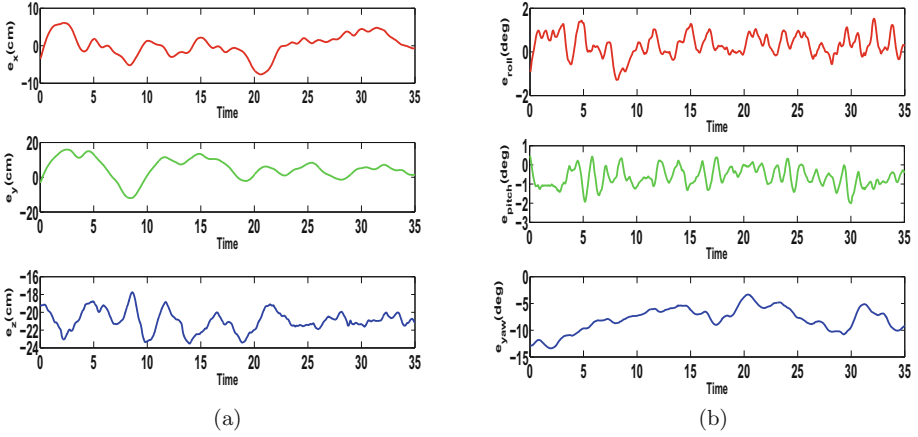
**Fig. 3.** System overview & complete work flow.

**Table 2.** Various nodes on the ROS network.

Node	Purpose
Mavros	For communicating with the onboard flight controller
Cam_driver	For extracting and publishing camera data
Px4flow	For extracting and publishing data acquired from the px4flow sensor
SVO	For running the VSLAM algorithm
Sensor_fusion	Fusing depth data from visual and ultrasonic sensors
Image_view	For viewing the debug output of the visual SLAM



**Fig. 4.** Hovering at fixed point  $(23, -37, 144)$  in 3D space, blue point indicates the target location. (Color figure online)



**Fig. 5.** (a) Position error plots. (b) Attitude error plots.

We used Bluefox (CMOS,  $752 \times 480$ ) camera in our experiment. Localization using SVO is performed after camera calibration. For an experiment, the manual operator performs the takeoff and the landing procedures. Even while in autonomous flight, the operator must remain alert to take back control in case of any random behavior or loss of track. Hovering at the fixed point is shown in Fig. 4 while position and attitude error are given in Fig. 5(a) and (b) respectively. Error plots in Fig. 5(a) and (b) are taken from the controller inputs and outputs respectively. The central objective to obtain autonomous hovering without the help of any external positioning system like VICON. The video of the experiment can be seen in <https://youtu.be/L7imtpw8mU>.

## 4 Conclusion

In this paper, vision based localization and mapping techniques have been explored with application to a micro aerial vehicle. The SLAM based technique has been found to perform good localization in unknown scenes. We have tested this framework in indoor environment using vision based feedback to achieve a hovering task. This work is useful to achieve waypoint navigation. Experimental results have confirmed the potential of this framework.

## References

1. Mellinger, D., Shomin, M., Kumar, V.: Control of quadrotors for robust perching and landing. In: International Powered Lift Conference, pp. 205–225 (2010)
2. Shen, S., Mulgaonkar, Y., Michael, N., Kumar, V.: Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 4974–4981 (2014)
3. Hamel, T., Mahony, R.: Visual servoing of an under-actuated dynamic rigid-body system: an image-based approach. *IEEE Trans. Robot. Autom.* **18**(2), 187–198 (2002)
4. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, pp. 225–234 (2007)
5. Engel, J., Sturm, R., Cremers, D.: Accurate figure flying with a quadrocopter using onboard visual and inertial sensing. *IMU* **320**, 240 (2012)
6. Schauwecker, K., Ke, N.R., Scherer, S.A., Zell, A.: Markerless visual control of a quad-rotor micro aerial vehicle by means of on-board stereo processing. In: Levi, P., Zweigle, O., Häußermann, K., Eckstein, B. (eds.) *Autonomous Mobile Systems 2012. Informatik aktuell*, pp. 11–20. Springer, Berlin (2012)
7. Fraundorfer, F., Heng, L., Honegger, D., Lee, G.H., Meier, L., Tanskanen, P., Pollefeys, M.: Vision-based autonomous mapping and exploration using a quadrotor mav. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4557–4564 (2012)
8. Yang, S., Scherer, S., Zell, A., et al.: Visual SLAM for autonomous MAVs with dual cameras. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 5227–5232 (2014)
9. Achtelik, M., Achtelik, M., Weiss, S., Siegwart, R.: Onboard IMU and monocular vision based control for MAVs in unknown in-and outdoor environments. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 3056–3063 (2011)
10. Nützi, G., Weiss, S., Scaramuzza, D., Siegwart, R.: Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *J. Intell. Robot. Syst.* **61**(1–4), 287–299 (2011)
11. Weiss, S., Achtelik, M.W., Lynen, S., Chli, M., Siegwart, R.: Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 957–964 (2012)

12. Weiss, S., Siegwart, R.: Real-time metric state estimation for modular vision-inertial systems. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 4531–4537 (2011)
13. Forster, C., Pizzoli, M., Scaramuzza, D.: SVO: fast semi-direct monocular visual odometry. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 15–22 (2014)
14. Woodman, O.J.: An introduction to inertial navigation. University of Cambridge, Computer Laboratory, Technical report. UCAMCL-TR-696, vol. 14, p. 15 (2007)
15. Shree, A.S.: Vision based navigation of a quadcopter using a single camera. Master’s thesis, Indian Institute of Technology, Kanpur, Kanpur, India (2015)

Intelligent Human Computer Interaction

8th International Conference, IHCI 2016, Pilani, India,

December 12-13, 2016, Proceedings

Basu, A.; Das, S.; Horain, P.; Samit, B. (Eds.)

2017, XIX, 290 p. 139 illus., Softcover

ISBN: 978-3-319-52502-0