Proceedings of the 7th RSI
International Conference on Robotics and Mechatronics (ICRoM 2019)
November 20-21, 2019, Tehran, Iran

Position Estimation for Drones based on Visual SLAM and IMU in GPS-denied Environment

Hamid Didari Khamseh Motlagh, Faraz Lotfi, Hamid D.Taghirad
Advanced Robotics and Automated Systems (ARAS),
Faculty of Electrical Engineering,
K. N. Toosi University of Technology,
(Email: hamididari and f.lotfi@email.kntu.ac.ir, Tghirad@kntu.ac.ir)

Saeed Bakhshi Germi
Machine Learning Research Group
Tampere University
Tampere, Finland
(Email : Saeed.BakhshiGermi@tuni.fi)

***Abstract* – Due to the increased rate of drone usage in various commercial and industrial fields, the need for their autonomous operation is rapidly increasing. One major aspect of autonomous movement is the ability to operate safely in an unknown environment. The majority of current works are persistently using a global positioning system (GPS) to directly find the absolute position of the drone. However, GPS accuracy might be not suitable in some applications and this solution is not applicable to all situations. In this paper, a positioning system based on monocular SLAM and inertial measurement unit (IMU) is presented. The position is calculated through the semi-direct visual odometry (SVO) method alongside IMU data, and is integrated with an extended Kalman filter (EKF) to enhance the efficiency of the algorithm. The data is then employed to control the drone without any requirement to any source of external input. The experiment results for long-distance flying paths is very promising.**

Keywords – Position estimation, Kalman filtering, SLAM, monocular camera, UAV.

I. INTRODUCTION

With the increasing rate of drone technology development, the idea of drones playing a major role in transportation, rescue, and other commercial or safety purposes is very likely. For these applications, a drone must be able to operate in an unknown outdoor environment, while its absolute position is usually determined by using an external positioning system like GPS. However, since this system relies on an external source of input, it is quite plausible to hijack the drone. This is not only financially very costly, but may also cause serious implications due to the drone malfunctions. To tackle this problem, a positioning system independent of any external signals are advised. Since the drones used for rescue or transportation missions usually fly in an altitude of over 50 meters for a relatively long time, a vision-based SLAM method is chosen for estimation of the local position of the drone using images taken from a vertical monocular camera. To further enhance the accuracy of the embedded system, the estimated position may be fused to onboard IMU measurements.

In recent years, many researches are conducted research related to vision-based position estimation. These works are usually based on three types of camera, namely monocular, stereo, or RGBD. In [1], it is proposed to employ an RGBD SLAM method for positioning. Although the results are promising, the method cannot be used for drones due to the low range of depth camera (usually under 10 meters). Reference [2] proposes using a stereo camera for visual SLAM. While the depth measurement of a stereo camera has a higher range compared with an RGBD camera, for high altitude flying drones is flying at a high altitude the depth measurement via a stereo camera is not an accurate and reliable source of input.

As such, a suitable solution could be to employ a monocular-camera-based SLAM. PTAM [3], is a feature-based SLAM algorithm which achieves robustness through tracking and mapping of numerous features. The method runs in real-time by parallelizing the motion estimation and mapping tasks based on keyframes bundle adjustment (BA) [4]. However, PTAM is only suited for small environments. ORB-SLAM is another feature-based monocular SLAM system [5], Although this algorithm works well in a large-scale environment, the requirement for a powerful processor restricts its usage on commercial embedded systems. Another critical problem in using monocular SLAM is its scale difference to the reality. In other words, the position estimated by SLAM is not metric, and therefore, before utilizing the SLAM measurements for controlling the drone, a scale factor $\lambda \epsilon R^{+}$ must be estimated first. Due to the nature of the problem, this factor cannot be calculated directly and requires a separate estimator. The main challenge in scale factor estimation is its time variance and requirement to be updated regularly.

A method is proposed in [7] to calculate the scale factor for a monocular-camera-based SLAM. In this work, the scale factor is calculated by linear regression of height measurements of a sonar sensor and SLAM. This method has two major drawbacks. Firstly, the low range of sonar sensor may cause an issue when the drone is flying at a high altitude. Secondly, sonar sensors measure distance, and since the drone might fly over obstacles, this measurement is not always give the right altitude. Utilizing IMU data alongside with an observer is very promising to estimate the scale factor. This method does not rely on any external sensors and is independent to the environment topology. Moreover, due to the dynamic acceleration of the drone, the scale factor may be updated regularly. In [8], a solution for scale factor estimation is suggested based on fusing the SLAM output with the measurements obtained from an IMU. In this method, offline spline fitting is used, while its main drawback is that the scale factor cannot be updated quite frequently, This is due to the fact that this method requires a set of data for at least a period of 10s, to achieve a decent result. Furthermore, in [9], EKF is employed with two measurements for estimating the scale factor. Although the method is working properly at the beginning, when the scale factor drifts over time, the EKF fails to correctly estimate the scale factor.

Taking into account the limited processing power and the requirement for a fast and accurate method, in this paper it is proposed to use the SVO [6] empowered by an EKF scaling factor estimation. In order to verify the effectiveness of the proposed method, the results are compared to that of reference [9]. Furthermore, a transformation matrix between the camera and IMU is required for data fusion. Reference [10] proposes using a turntable to calculate the transformation matrix, while [11-12] propose employing an EKF to calculate the transformation matrix. These algorithms are elaborated in Section II.

The rest of the paper is organized as follows: Part II covers methods for IMU and camera calibration, IMU model used in this paper, proposed EKF filter for scale estimation, altitude estimation methods and finally control algorithms. Part III deals with the practical experiments and their results. Finally, part IV concludes the paper.

II. Control And State Estimation

This section is devoted to describe the proposed control and state estimation algorithms used in this research. To ensure the safety of the drone, controller and attitude-altitude estimator are deployed separately in the autopilot of the experimental setup. The section also covers the monocular SLAM, scale factor estimation, and data fusion for accuracy enhancement. Fig. 1 illustrates the diagram of the developed algorithm. The controller is required to be implemented in a real-time embedded system. Due to this fact, the controller and Attitude and Heading Reference System(AHRS) are deployed on the autopilot that utilizes a low level processing scheme. Furthermore, due to the possibility of system failure caused either by the malfunction of the main processing board or the communication link, the altitude estimator is also developed on the autopilot to further enhance the safety.

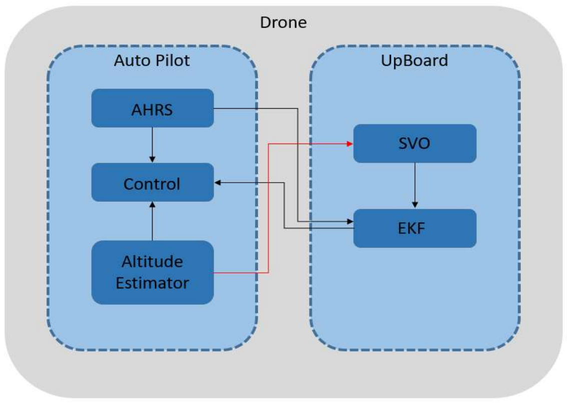

Fig. 1: Different parts of the developed algorithm. The red line uses for only initializing and the Black line is always used.

A. Camera and IMU Coordinate Systems

Since the frame coordinates of the camera and the IMU are not aligned, for the purpose of fusion the relating transformation matrix is required. To find such transformation several methods are proposed in the literature. A few representatives of such schemes are reviewed in here. In [10], a method is proposed to estimate the rotation matrix of camera with respect to IMU in two steps. First, the rotation is estimated by comparing two vertical references while both sensors observe a vertical object. For the camera, a vertical visual target such as chessboard may be used, while for the IMU, the accelerometer data may be used in a non-accelerated drone motion. Then, a translation matrix is estimated by using a turntable. In [11], the IMU and camera coordination is formed as a grey box identification problem. In this regard, an EKF model is developed to fuse IMU and camera data, while trying to minimize the error between prediction and measurement by optimizing the initial orientation and translation between the IMU and the camera. Furthermore, another EKF solution is presented in [12]. The algorithm estimates an approximate value for orientation and translation between the IMU and the camera and then optimizes the value by EKF.

The main benefit of using EKF to estimate the transformation matrix is that it does not require any external hardware such as a turntable. On the other hand, due to the nature of EKF, if the initial approximations for orientation and translation are not close, the results may converge to a local minimum or even diverge, since the number of optimization states is more than that of the measurements.

In this paper, it is proposed to use KALIBR toolbox [14 - 18], in order to coordinate the camera and IMU. This toolbox is based on [13], which has formulated the calibration problem as a maximum a posteriori problem in continuous time, while the optimization is solved by Levenberg-Marquardt (LM) algorithm. The estimation of rotation and translation parts is accomplished using records of images and IMU data. The details of this method are elaborated in [13].

B. Accelerometer

A MEMS accelerometer is employed in this research. Low price MEMS accelerometers suffer from various error sources that will cause uncertainty in the model which is not desired. In this paper, the measurement noise and bias for the accelerometer are taken into account based on the model proposed in [19]. The accelerometer model is presented as follows:

$$y_{I,k} = a + \delta^{a}_{I,t} + v^{a}_{I,t} \qquad (1)$$

where $y_{I,k}$ is the measured data, a is the actual acceleration, $\delta^{a}_{I,t}$ is the bias modeled as a random walk, and $v^{a}_{I,t}$ is the white Gaussian measurement noise. Furthermore, the random walk model for bias is considered as:

$$\delta^{a}_{I,t+1} = \delta^{a}_{I,t} + v^{\delta_a}_{I,t} \qquad (2)$$

where $v^{\delta_a}_{I,t}$ is a white Gaussian noise.

C. State Estimation

IMU data and monocular SLAM output are also in different frame coordinates. Before fusion, the IMU data must be transformed into the camera frame. By using the matrix obtained in subsection A, the acceleration in the camera frame may be derived as:

$$\vec{a}_c = R_{ci}(\vec{a}_i - R_{iw} * \vec{g}_n) \qquad (3)$$

where $\vec{a}_c$ is the acceleration vector in the camera frame, R_{ci} is the rotation matrix between camera and IMU frames, $\vec{a}_i$ is the acceleration vector in the IMU frame, R_{iw} is the rotation matrix between IMU and world frames, and $\vec{g}_n$ is the gravity vector in world frame.

To estimate the state of the system, two different methods based on EKF are presented in this paper and compared. Let us define the notations required for these two algorithms as follows:

- $\vec{x} \epsilon R^{3*1}$ is the position obtained from SLAM.
- $\vec{v}, \vec{a} \epsilon R^{3*1}$ are the velocity and the acceleration of the drone, respectively.
- $b_a \epsilon R^{3*1}$ is accelerometer bias.

- $\lambda \epsilon R^+$ is the scale factor between the world frame and SLAM output.
- v, ω are the Gaussian process noises.

The first method is using an EKF with two measurements. This method uses the formulation in [9], in which the non-linear prediction model is described as follows:

$$\vec{z}_{k+1} = \vec{f}_k(\vec{z}_k) + v_k$$

$$\begin{bmatrix} \vec{x}_{k+1} \\ \vec{v}_{k+1} \\ \vec{a}_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} I_3 & \frac{dt}{\lambda} I_3 & \frac{dt^2}{2\lambda} I_3 & 0 \\ 0 & I_3 & dt I_3 & 0 \\ 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vec{v}_k \\ \vec{a}_k \\ \lambda_k \end{bmatrix} + v_k \tag{4}$$

The linearized matrix model is as following:

$$F_k = \begin{bmatrix} I_3 & \frac{dt}{\lambda} I_3 & \frac{dt^2}{2\lambda} I_3 & -\frac{dt}{\lambda^2} I_3 - \frac{dt^2}{2\lambda^2} I_3 \\ 0 & I_3 & T I_3 & 0 \\ 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

Due to the difference in update rate of IMU and SLAM, a multi-rate EKF is used in this research. The measurement update yields:

$$\begin{aligned} \vec{y}_{V,k} &= H_{V,k}\vec{z}_k = [1\ 0\ 0\ 0]\vec{z}_k \\ \vec{y}_{I,k} &= H_{I,k}\vec{z}_k = [0\ 0\ 1\ 0]\vec{z}_k \end{aligned} \tag{6}$$

Where, index *I* is used for IMU and index *V* is used for vision.

The proposed method in this research is an EKF with only one measurement. In this approach, the IMU data is considered as the control signal to the notion model and the position of SLAM is used as the measurement. The non-linear prediction model is then formulated as:

$$\begin{bmatrix} \vec{x}_{k+1} \\ \vec{v}_{k+1} \\ \overrightarrow{b_a}_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} I_3 & dt I_3 & -0.5 dt^2 I_3 & 0 \\ 0 & I_3 & -dt I_3 & 0 \\ 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vec{v}_k \\ \overrightarrow{b_a}_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} 0.5dt^2 \\ dt \\ 0 \\ 0 \end{bmatrix} a_c \tag{7}$$

and the measurement model is:

$$z = h(x) + \omega = \lambda x + \omega \tag{8}$$

The linearized model is:

$$H_x = [\lambda \ \ 0 \ \ 0 \ \ x] \tag{9}$$

By using the presented motion model, the discrete-time covariance matrix Q_d is derived as:

$$Q_d = A G_c Q_c {G_c}^T A^T \tag{10}$$

where $Q_c = diag\left(\sigma_{n_a}^2, \sigma_{n_{b_a}}^2\right)$ is the continuous-time system noise covariance matrix, A is the state-transition model, and G_c is defined as:

$$G_c = \begin{bmatrix} 0 & 0 \\ -dt & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \tag{11}$$

To compare these methods, a simulation study is forwarded for one dimensional data and the results are reported in Fig. 2. The real value of λ is assumed to be 1; At t=48, this value starts to increase over the time. The initial value of λ for both methods is set to be 0.8 for the sake of comparison.

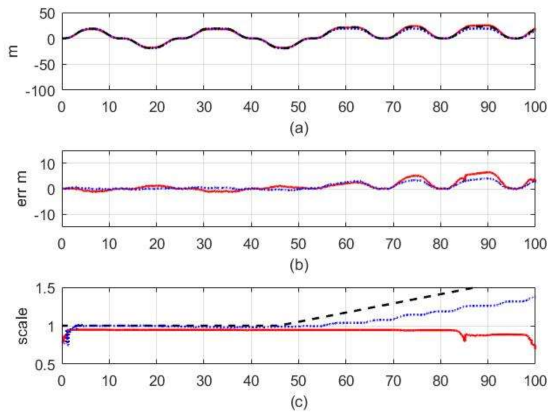

Fig. 2. The output of the first method is shown in solid red line, the second method in dotted blue line, and the ground truth in dashed black line. (a) Position vs. Time. (b) Estimated position error vs. Time. (c) Scale factor vs. Time.

As it is seen in this figure, as long as the scale factor is constant, both methods are similarly able to accurately estimate the scale factor and the position. However, when the scale factor becomes time-variant, the first method fails to estimate it accurately, while the second approach does not suffers as much.

D. Altitude

Since the vertical position estimation of monocular SLAM has a large drift, it is better to have a separate altitude control system. This will also prevent crashing if the vision system failed during the flight due to lack of features in the environment. For this purpose, an extra sensor is needed to estimate the altitude. Among all the different types of options, a pressure sensor has been chosen to fulfill this task due to the robustness to environment topography. The altitude of the drone may be estimated by calculating the difference between the current pressure and the initial pressure of the air. To increase the accuracy of the measurement, a Kalman filter is used to fuse the data from this sensor to that of the IMU. Thus, the prediction model is formed as:

$$\begin{bmatrix} \dot{x_z} \\ \dot{v_z} \\ \dot{b}_{a_z} \end{bmatrix} = \begin{bmatrix} 1 & dt & -0.5dt^2 \\ 0 & 1 & -dt \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_z \\ v_z \\ b_{a_z} \end{bmatrix} + \begin{bmatrix} 0.5dt^2 \\ dt \\ 0 \end{bmatrix} a_z \tag{12}$$

the vertical acceleration may be obtained as:

$$a_z = R_{wi} * \vec{a}_i * [0 \quad 0 \quad 1] - g \tag{13}$$

and finally, the measurement model yields:

$$z = [1 \quad 0 \quad 0] \begin{bmatrix} x \\ v \\ b_a \end{bmatrix} + w \tag{14}$$

E. GPS

To verify the algorithm, the ground truth for drone position is needed. Since this paper is focused on a drone, flying in an outdoor environment with altitude in range of 20m to 150m, an appropriate choice for validation data would be GPS. To compensate the measurement noise, a Kalman filter is used with IMU data. For this case, the motion model is formed as:

$$\begin{bmatrix}\dot{x}\\ \dot{v}\\ \dot{b}_a\end{bmatrix}=\begin{bmatrix}1 & dt & -0.5dt^2\\ 0 & 1 & -dt\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\ v\\ b_a\end{bmatrix}+\begin{bmatrix}0.5dt^2\\ dt\\ 0\end{bmatrix}a_w \qquad (15)$$

the acceleration in world frame may be obtained as:

$$\vec{a}_w = R_{wi} * \vec{a} - g \qquad (16)$$

and the observation model is formed as:

$$z = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}x\\ v\\ b_a\end{bmatrix}+w \qquad (17)$$

F. *Control*

Since monocular SLAM performs poorly in high velocity movements, it is important to limit the linear and angular velocity of the drone. This limitation cannot be forced with a solely position control system, thus a cascade controller is employed to control the altitude and horizontal position. With this system, the controller first regulates the angular and linear velocity and then it deals with the position. A separate controller is used for vertical position which controls the acceleration such that the drone moves smoothly in vertical direction. Fig. 3 shows the block diagram of the proposed method for controller in each channel:

- Attitude: First a P controller is used to convert attitude error to a desired angular rate. Next, a PID controller is used in the inner loop to convert the angular rate error to motor command.
- Horizontal: First a P controller is used to convert position error to a desired velocity. Then, a PID controller is used to convert the velocity error to angular error which is then used for the inner attitude control loop.
- Vertical: First a P controller is used to convert altitude error to a desired velocity. Next a PID controller is used to convert the velocity error to a desired acceleration. Then another PID controller is used to convert the acceleration error to motor command.

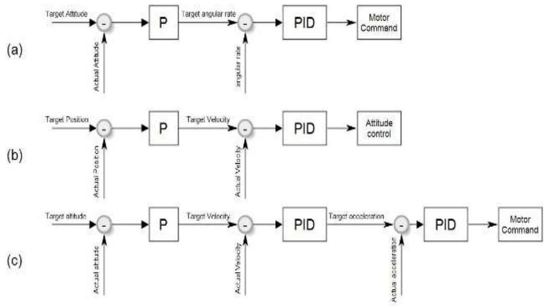

Fig. 3: (a) Attitude control diagram (b) Horizontal control diagram (c) Vertical control diagram

To test the performance of the controller, a flight path of 1.5 kilometers is considered for the drone in about 18 minutes. The experimental result is illustrated in Fig. 4. while the drone position is displayed in solid red line and the desired flight path is displayed in dashed-dot blue line. As it is seen in this figure, the velocity of the drone depends extensively on the accuracy of the roll and pitch angles. Since for the SLAM algorithm to perform well, it is required that the camera be hold in a relatively small angle with respect to the world frame, the maximum speed of the drone is limited to 2 m/s. As seen in the figure, the drone has suitably tracked the desired trajectory.

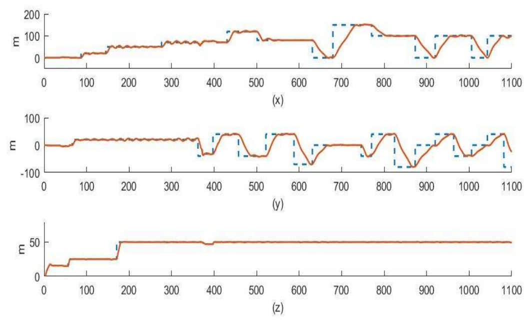

Fig. 4: Position control signal vs. Time for three dimensions.

G. *Error calculation*

To validate the estimated position from proposed algorithm, GPS data is fused with IMU to create the ground truth. The error is then calculated based on Root Mean Square Error (RMSE) which is formed as:

$$RMSE = \frac{\sum_{i=1}\sqrt{\left(x_i^{gps}-x_i^{EKF}\right)^2+\left(y_i^{gps}-y_i^{EKF}\right)^2}}{N} \qquad (18)$$

Where N is the total amount of data.

III. Experimental Results

To validate the proposed algorithm, various tests are examined in an outdoor environment. In these tests, the maximum wind speed is about 3 m/s and the maximum speed of drone is 2 m/s. To ensure the safety of the drone in case any fault happens, a backup controller with GPS is employed to return the drone to the starting point. The experimental drone is a Hexa-rotor equipped with a down facing camera with fish eye lens. The control algorithm is implemented in a non-commercial autopilot board and the image processing is executed on an upboard processor, while the results are fed to the autopilot board.

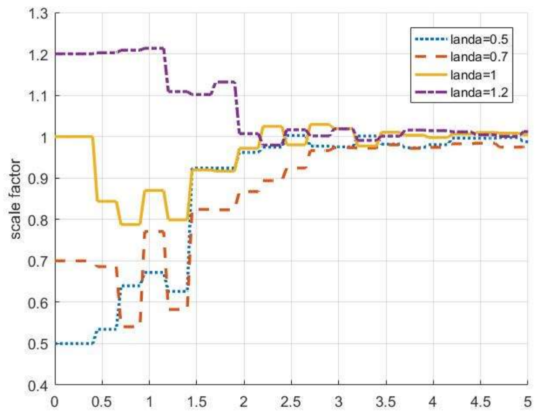

Fig. 5: Scale factor estimation. The true value is equal to one.

Since the exact value of the scale factor is not accessible in all situations, the proposed algorithm should work properly even when the initialized value has some errors. Thus, some tests are advised to determine the performance of the presented EKF. Fig. 5 illustrates the estimated scale factor with different initial values. Considering the resulting performance, the EKF is able to estimate the scale factor correctly as long as the initial error is under %50.

The first test was accomplished on the recorded data of the drone flying in an altitude of 150 meters for a total length of 1.5 kilometers. The recorded data contains IMU, camera, and GPS outputs. After applying the proposed method on the dataset, the estimated 2D position (solid red line) and the ground truth (dashed blue line) are compared in Fig. 6 . As seen in this figure, the proposed algorithm managed to estimate the position of the drone with decent accuracy. The RMSE of error is 9.43 meters which is 0.62% of the traveled distance.

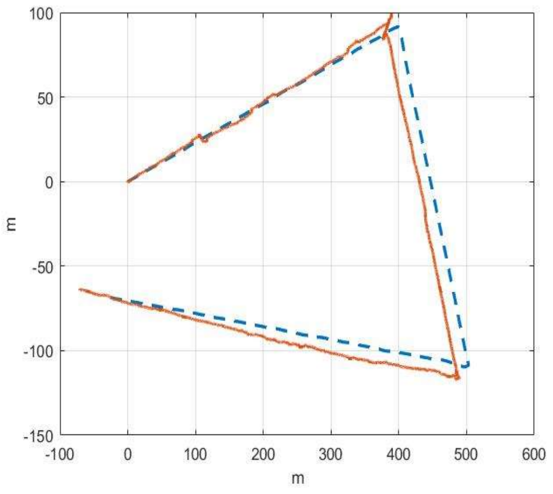

Fig. 6: 2D position representation of drone in first test.

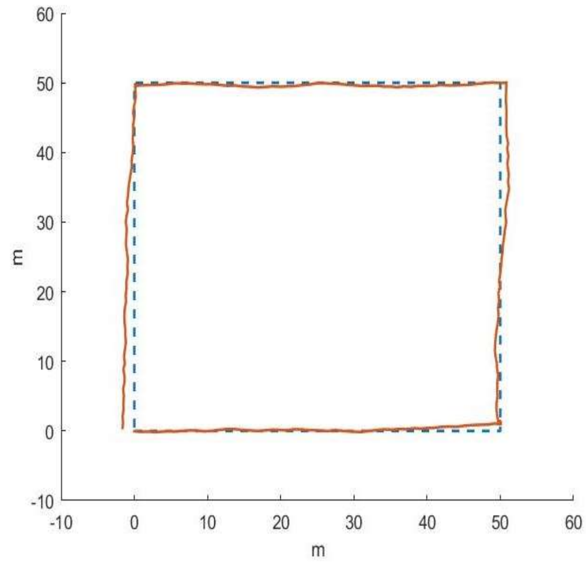

Fig. 7: 2D position representation of drone in second test.

The second involves traversing a rectangular path automatically using the proposed algorithm. During the flight the position of the drone is recorded from the GPS (solid red line) to verify the desired path (dashed blue line) and is compared in Fig. 7. As shown in the figure, the drone managed to accurately track the desired path while the RMSE of error is 0.7 meters in this closed path. which is 0.35% of the traveled distance. Since traveled distance in second test is far less than the first one, the result in second test is better.

IV. Conclusion

This paper presents an algorithm for the 3d positioning of a flying robot in an outdoor environment. The proposed method uses a monocular-camera-based SLAM and an IMU. This method allows the drone to operate safely even in GPS-denied environments. After conducting several tests, the results verify the effectiveness of the algorithm. The accuracy of the method is observed to be within % 0.5 of the traversed path in various experiments, which is very promising.

References

[1] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation", *Journal of Field Robotics*, vol. 36, no. 2, pp. 416-446, 2018.

[2] J. Engel, J. Stuckler and D. Cremers, "Large-scale direct SLAM with stereo cameras", *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[3] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces", *IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.

[4] H. Strasdat, J. Montiel and A. Davison, "Real-time monocular SLAM: Why filter?", *IEEE International Conference on Robotics and Automation*, 2010.

[5] R. Mur-Artal, J. Montiel and J. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System", *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147-1163, 2015.

[6] C. Forster, M. Pizzoli and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry", *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[7] O. Esrafilian and H. Taghirad, "Autonomous flight and obstacle avoidance of a quadrotor by monocular SLAM", *International Conference on Robotics and Mechatronics (ICROM)*, 2016.

[8] S. Jung and C. Taylor, "Camera trajectory estimation using inertial sensor measurements and structure from motion results", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, *2001*.

[9] G. Nützi, S. Weiss, D. Scaramuzza and R. Siegwart, "Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM", *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 287-299, 2010.

[10] J. Lobo and J. Dias, "Relative Pose Calibration Between Visual and Inertial Sensors", *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 561-575, 2007.

[11] J. Hol, T. Schön and F. Gustafsson, "Modeling and Calibration of Inertial and Vision Sensors", *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 231-244, 2010.

[12] F. Mirzaei and S. Roumeliotis, "A Kalman Filter-Based Algorithm for IMU-Camera Calibration: Observability Analysis and Performance Evaluation", *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1143-1156, 2008.

[13] P. Furgale, J. Rehder and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[14] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes", *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[15] P. Furgale, J. Rehder and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems", *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013.

[16] P. Furgale, T. Barfoot and G. Sibley, "Continuous-time batch estimation using temporal basis functions", *IEEE International Conference on Robotics and Automation*, 2012.

[17] J. Maye, P. Furgale and R. Siegwart, "Self-supervised calibration for robotic systems", *IEEE Intelligent Vehicles Symposium (IV)*, 2013.

[18] L. Oth, P. Furgale, L. Kneip and R. Siegwart, "Rolling Shutter Camera Calibration", *IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[19] J. D. Hol, "Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS", PhD dissertation, Linköping University Electronic Press, Linköping, 2011.