

Simultaneous State Initialization and Gyroscope Bias Calibration in Visual Inertial aided Navigation

Jacques Kaiser¹, Agostino Martinelli¹, Flavio Fontana² and Davide Scaramuzza²

Abstract—State of the art approaches for visual-inertial sensor fusion use filter-based or optimization-based algorithms. Due to the nonlinearity of the system, a poor initialization can have a dramatic impact on the performance of these estimation methods. Recently, a closed-form solution providing such an initialization was derived in [1]. That solution determines the velocity (angular and linear) of a monocular camera in metric units by only using inertial measurements and image features acquired in a short time interval. In this paper, we study the impact of noisy sensors on the performance of this closed-form solution. We show that the gyroscope bias, not accounted for in [1], significantly affects the performance of the method. Therefore, we introduce a new method to automatically estimate this bias. Compared to the original method, the new approach now models the gyroscope bias and is robust to it. The performance of the proposed approach is successfully demonstrated on real data from a quadrotor MAV.

I. INTRODUCTION

Autonomous mobile robots navigating in unknown environments have an intrinsic need to perform localization and mapping using only on-board sensors. Concerning Micro Aerial Vehicles (MAV), a critical issue is to limit the number of on-board sensors to reduce weight and power consumption. Therefore, a common setup is to combine a monocular camera with an inertial measurements unit (IMU). On top of being cheap, these sensors have very interesting complementarities. Additionally, they can operate in indoor environments, where Global Positioning System (GPS) signals are shadowed. An open question is how to optimally fuse the information provided by these sensors.

Currently, most sensor-fusion algorithms are either filter-based or iterative. That is, given a current state and measurements, they return an updated state. While working well in practice, these algorithms need to be provided with an initial state. The initialization of these

methods is critical. Due to nonlinearities of the system, a poor initialization can result into converging towards local minima and providing faulty states with high confidence.

In this paper, we demonstrate the efficiency of a recent closed-form solution introduced in [2][1], which fuses visual and inertial data to obtain the structure of the environment at the global scale along with the attitude and the speed of the robot. By nature, a closed-form solution is deterministic and, thus, does not require any initialization.

The method introduced in [2][1] was only described in theory and demonstrated in simulations. In this paper, we implement this method in order to test it with real world data. This allow us to identify its limitations and bring modifications to overcome them. Specifically, we investigate the impact of biased inertial measurements. Although the case of biased accelerometer was originally studied in [1], here we show that a large bias on the accelerometer does not significantly worsen the performance. One major limitation of [1] is the impact of biased gyroscope measurements. In other words, the performance becomes very poor in presence of a bias on the gyroscope and, in practice, the overall method can only be successfully used with a very precise - and expensive - gyroscope. Here, we introduce a simple method that automatically estimates this bias. By adding this new method for the bias estimation to the original method [1], we obtain results that are equivalent to the ones in absence of bias. Compared to [1], the new method is now robust to the gyroscope bias and automatically calibrates the gyroscope.

II. RELATED WORK

The problem of fusing visual and inertial data has been extensively investigated in the past. However, most of the proposed methods require a state initialization. Because of the system nonlinearities, lack of precise initialization can irreparably damage the entire estimation process. In literature, this initialization is often guessed or assumed to be known [3][4][5][6][7]. Recently, this sensor fusion problem has been addressed by using

*This work was supported by the French National Research Agency ANR through the project VIMAD

¹ INRIA Rhone Alpes, Grenoble, France. Email: jacko.kaiser@gmail.com, agostino.martinelli@ieee.org

² Robotics and Perception Group, University of Zurich, Switzerland. Email: fforster,sdavid@ifi.uzh.ch

optimization-based approaches [8][9][10][11][12][13]. These methods outperform filter-based algorithms in terms of accuracy due to their capability of relinearizing past states. On the other hand, the optimization process can be affected by the presence of local minima. We are therefore interested in a deterministic solution that analytically expresses the state in terms of the measurements provided by the sensors during a short time-interval.

In computer vision, several deterministic solutions have been introduced. These techniques, known as *Structure from Motion*, can recover the relative rotation and translation up to an unknown scale factor between two camera poses [14]. Such methods are currently used in state-of-the-art visual navigation methods for MAVs to initialize maps [15][7][16]. However, the knowledge of the absolute scale, and, at least, of the absolute roll and pitch angles, is essential for many applications ranging from autonomous navigation in GPS-denied environments to 3D reconstruction and augmented reality. For these applications, it is crucial to take the inertial measurements into consideration to compute these values deterministically.

A procedure to quickly re-initialize a MAV after a failure was presented in [17]. However, this method requires an altimeter to initialize the scale.

Recently, a closed-form solution has been introduced in [2]. From integrating inertial and visual measurements over a short time-interval, this solution provides the absolute scale, roll and pitch angles, initial velocity, and distance to 3D features. Specifically, all the physical quantities are obtained by simply inverting a linear system. The solution of the linear system can be refined with a quadratic equation assuming the knowledge of the gravity magnitude. This closed-form was improved in [4] to work with unknown camera-IMU calibration; however, since in this case the problem cannot be solved by simply inverting a linear system, a method to determine the six parameters that characterize the camera-IMU transformation was proposed. As a result, this method is independent of external camera-IMU calibration, hence, suitable for power-on-and-go systems.

A more intuitive expression of this closed-form solution was derived in [1]. While being mathematically sound, this closed-form solution is not robust to noisy sensor data. For this reason, to the best of our knowledge, it has never been used in an actual application. In this paper, we perform an analysis to find out its limitations. We start by reminding the reader the basic equations that characterize this solution (section III). In section IV, we show that this solution is resilient

to the accelerometer bias but strongly affected by the gyroscope bias. We then introduce a simple method that automatically estimates the gyroscope bias (section V). By adding this new method for the bias estimation to the original method, we obtain results that are equivalent to the ones obtained in absence of bias. Compared to the original method, the new method is now robust to the gyroscope bias and also calibrates the gyroscope. In section VI, we validate our new method against real world data from a flying quadrotor MAV to prove its robustness against noisy sensors during actual navigation. Finally, we provide the conclusions in section VII.

III. CLOSED-FORM SOLUTION

In this section, we provide the basic equations that characterize the closed-form solution proposed in [1]. We also provide the main features of this solution¹.

Let us refer to a short interval of time (e.g., of the order of 3 seconds). We assume that during this interval of time the camera observes simultaneously N point-features and we denote by t_1, t_2, \dots, t_{n_i} the times of this interval at which the camera provides an image of these points. Without loss of generality, we can assume that $t_1 = 0$. The following equation holds (see [1] for its derivation):

$$S_j = \lambda_1^i \mu_1^i - V t_j - G \frac{t_j^2}{2} - \lambda_j^i \mu_j^i \quad (1)$$

with:

- μ_j^i the normalized bearing of point feature i at time t_j in the local frame at time t_1 ;
- λ_j^i the distance to the point feature i at time t_j ;
- V the velocity in the local frame at time t_1 ;
- G the gravity in the local frame at time t_1 ;
- S_j the integration in the interval $[t_1, t_j]$ of the rotated linear acceleration data (i.e., the integration of the inertial measurements).

The local frame refers to a frame of reference common to the IMU and the camera. In a real application, we would work in the IMU frame and have some additional constant terms accounting for the camera-IMU transformation. We do not express these constant calibration terms explicitly here for clarity reasons.

The unknowns of Equation 1 are the scalars λ_j^i and the vectors V and G . Note that the knowledge of G is equivalent to the knowledge of the roll and pitch angles. The vectors μ_j^i are fully determined by visual and gyroscope measurements², and the vectors S_j are determined

¹Note that in this paper we do not provide a new derivation of this solution for which the reader is addressed to [1], section 3.

²The gyroscope measurements in the interval $[t_1, t_j]$ are needed to express the bearing at time t_j in the frame at time t_1



Fig. 1: Visual representation of Equation 1. The unknowns of the equation are colored in purple.

by accelerometer and gyroscope measurements.

Equation (1) provides three scalar equations for each point feature $i = 1, \dots, N$ and each frame starting from the second one $j = 2, \dots, n_i$. We therefore have a linear system consisting of $3(n_i - 1)N$ equations in $6 + Nn_i$ unknowns. Indeed, note that, when the first frame is taken at $t_1 = 0$, Equation (1) is always satisfied; thus does not provide information. We can write our system using matrix formulation. Solving the system is equivalent to inverting a matrix of $3(n_i - 1)N$ rows and $6 + Nn_i$ columns.

In [1], the author proceeded to one more step before expressing the underlying linear system. For a given frame j , the equation of the first point feature $i = 1$ is subtracted from all other point feature equations $1 < i \leq N$ (Equation (7)). This additional step, very useful to detect system singularities, has the effect to corrupt all measurements with the first measurement, hence worsening the performance of the closed-form solution. Therefore, in this paper we discard this additional step.

The linear system in Equation (1) can be written in the following compact form:

$$\Xi X = S. \quad (2)$$

Matrix Ξ and vector S are fully determined by the measurements, while X is the unknown vector. We have:

$$\begin{aligned} S &\equiv [S_2^T, \dots, S_2^T, S_3^T, \dots, S_3^T, \dots, S_{n_i}^T, \dots, S_{n_i}^T]^T \\ X &\equiv [G^T, V^T, \lambda_1^1, \dots, \lambda_1^N, \dots, \lambda_{n_i}^1, \dots, \lambda_{n_i}^N]^T \\ \Xi &\equiv \begin{bmatrix} T_2 & S_2 & \mu_1^1 & 0_3 & 0_3 & -\mu_2^1 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 \\ T_2 & S_2 & 0_3 & \mu_1^2 & 0_3 & 0_3 & -\mu_2^2 & 0_3 & 0_3 & 0_3 & 0_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ T_2 & S_2 & 0_3 & 0_3 & \mu_1^N & 0_3 & 0_3 & -\mu_2^N & 0_3 & 0_3 & 0_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ T_{n_i} & S_{n_i} & \mu_1^1 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & -\mu_{n_i}^1 & 0_3 & 0_3 \\ T_{n_i} & S_{n_i} & 0_3 & \mu_1^2 & 0_3 & 0_3 & 0_3 & 0_3 & 0_3 & -\mu_{n_i}^2 & 0_3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ T_{n_i} & S_{n_i} & 0_3 & 0_3 & \mu_1^N & 0_3 & 0_3 & 0_3 & 0_3 & -\mu_{n_i}^N & 0_3 \end{bmatrix}, \end{aligned}$$

where $T_j \equiv -\frac{t_j^2}{2}I_3$, $S_j \equiv -t_j I_3$ and I_3 is the identity

3×3 matrix, 0_3 is the 3×1 zero matrix. Note that matrix Ξ and vector S are slightly different from the ones proposed in [1]. This is due to the additional step that, as we explained in the previous paragraph, we discarded for numerical stability reasons (see [1] section 3 for further details).

The sensor information is completely contained in the above linear system. Additionally, in [1], the author added a quadratic equation assuming the gravitational acceleration is a priori known. Let us denote the gravitational magnitude by g . We have the extra constraint $|G| = g$. We can express this constraint in matrix formulation:

$$|\Pi X|^2 = g^2, \quad (3)$$

with $\Pi \equiv [I_3, 0_3, \dots, 0_3]$. We can therefore recover the initial velocity, the roll and pitch angles, and the distances to the point features by finding the vector X that satisfies (2) and (3).

In the next sections, we will evaluate the performance of this method on real-world data. This will allow us to identify its weaknesses and bring modifications to overcome them.

IV. LIMITATIONS OF [1]

The goal of this section is to find out the limitations of the solution proposed in [1]. This is obtained by running the closed-form solution on a combination of both real and synthetic data. In particular, we will add an artificial bias to the real measurements delivered by the inertial sensors (both the accelerometers and the gyroscopes). This will allow us to evaluate the impact of the bias on the performance.

A. Considered data

The setup is similar to the one we use in the experiments on real data described in Section VI-A. Specifically, the ground truth pose of the MAV is obtained with a motion-capture system. However, unlike Section VI, we simulate the camera measurements by creating artificial 3D-landmarks, and re-projecting these landmarks into the camera views (μ_j^i in Equation (1)). Simulating the camera measurements allows us to compare our estimations of the distances to the features (λ_j^i in Equation (1)) with the ground truth. We measure our error on the absolute scale by computing the mean error over all estimated distances to point features λ_j^i .

Inertial data were recorded from a quadrotor MAV while executing a circular trajectory of about 1m radius (see Fig. 2).

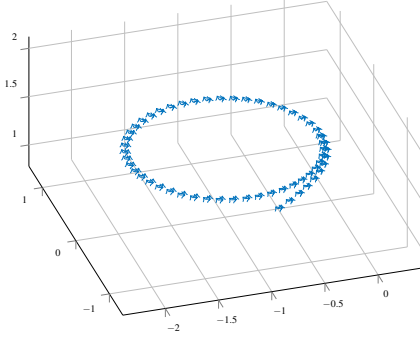


Fig. 2: Motion performed by the drone during 5 seconds.

We define the relative error as the euclidean distance between the estimation and the ground truth, normalized by the ground truth.

We set the simulated 3D-landmarks about 3m away from the MAV and the drone flies at a speed of around 1m/s. We found that setting the frame rate of the simulated camera at 10Hz provides a sufficient pixel disparity with the following setup. In practice, increasing the frame rate above 30Hz decreases the pixel disparity and introduces numerical instability for this setup. The theoretical cases in which our system admits singularities are provided in [2][1]. Reducing the number of considered frames also reduces the size of the matrices and, thus, speeds up the computations. As an example, over a time interval of 3 seconds, we obtain 31 distinct frames. When observing 7 features, solving the closed-form solution is equivalent to inverting a linear system of $3 \times 30 \times 7 = 630$ equations and $6 + 7 \times 31 = 223$ unknowns (see section III).

The method we use to solve the overconstrained linear system $\Xi X = S$ is a Singular Value Decomposition (SVD) since it yields numerically robust solutions.

In section IV, we will present the results obtained with the original closed-form solution on real IMU data with simulated camera measurements. Our goal is to identify its performance limitations and introduce modifications to overcome them.

B. Original closed-form solution performance

The original closed-form solution described in Equation (2) will be used as a basis for our work. Moreover, we can also use the knowledge of the gravity magnitude to refine our results (Equation (3)). In this case, we are minimizing a linear objective function with a quadratic constraint. In Fig. 3, we display the performance of the original Closed-Form (CF) solution in estimating speed, gravity in the local frame, and distances to the features with and without this additional constraint.

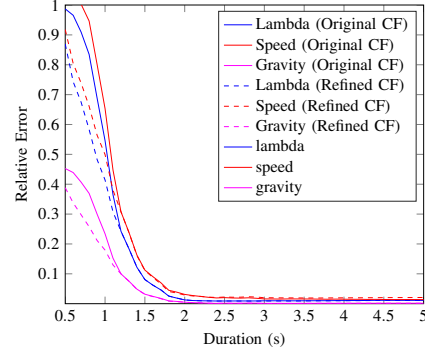


Fig. 3: Original closed-form solution estimations with and without using the knowledge of the gravity (3). We are observing 12 features over a variable duration of integration.

Note how the evaluations get better as we increase the integration time. Indeed, our equations come from an extended triangulation [2]. Therefore, it requires a significant difference in the measurements over time to robustly estimate the state. Also, note that the gravity is robustly estimated (around 5% error), whereas the speed and the distance to the features is more erroneous (above 10% error).

Since the gravity is well estimated with the original closed-form, it comes without surprise that constraining its magnitude does not improve the performance much. That is why the curves representing the performance with and without using the gravity refinement (3) are so close. We therefore ease this constraint for the following sections.

C. Impact of accelerometer bias on the performance

In order to visualize the impact of the accelerometer bias on the performance, we corrupt the accelerometer measurements provided by the IMU by an artificial bias (Fig. 4). Despite a high accelerometer bias, the closed-form solution still provides robust results. As seen in Fig. 4, neither the estimation of the gravity, the velocity or the lambdas is impacted by the accelerometer bias. In [1], the author provides an alternative formulation of the closed-form solution including the accelerometer bias as an observable unknown of the system. However, the estimation of the accelerometer bias with that method is not robust since our system is only slightly affected by it. This is a counterintuitive result. Since our equations contain an integration of the acceleration, we also perform an integration of the accelerometer bias. We would have expected the accelerometer bias to have a greater impact on the solutions yielded by the system.

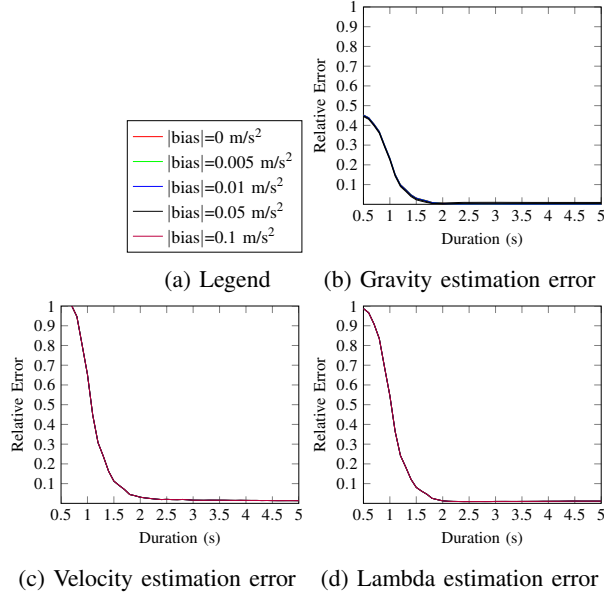


Fig. 4: Impact of the accelerometer bias on the performance of the closed-form solution. We are observing 7 features over a variable duration of integration.

D. Impact of gyroscope bias on the performance

To visualize the impact of the gyroscope bias on the performance, we corrupt the gyroscope measurements provided by the IMU by an artificial bias (Fig. 5). As seen in Fig. 5, the performance becomes very poor in presence of a bias on the gyroscope and, in practice, the overall method could only be successfully used with a very precise—and expensive—gyroscope.

V. ESTIMATING THE GYROSCOPE BIAS

Previous work has shown that the gyroscope bias is an observable mode when using an IMU and a camera, which means that it can be estimated [2].

Optimally, we would add the gyroscope bias in our unknown vector X and determine X by simply inverting the system $\Xi X = S$ as in the standard closed-form solution. However, we cannot express the gyroscope bias linearly with this system.

In this section, we propose a different approach to estimate the gyroscope bias using the closed-form solution.

A. Nonlinear minimization of the residual

Since our system of equations (1) is overconstrained, inverting it is equivalent to finding the vector X that minimizes the residual $\|\Xi X - S\|^2$.

Since we cannot express the gyroscope bias linearly, we define the following cost function:

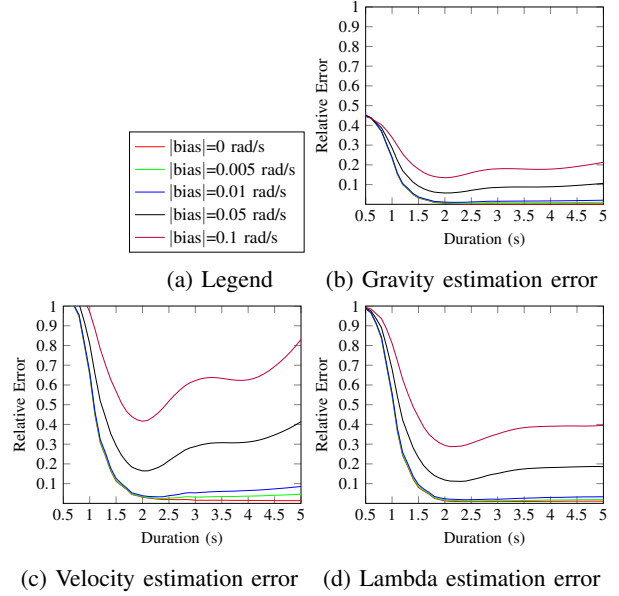


Fig. 5: Impact of the gyroscope bias on the performance of the closed-form solution. We are observing 7 features over a variable duration of integration.

$$\text{cost}(B) = \|\Xi X - S\|^2, \quad (4)$$

with:

- B the gyroscope bias;
- Ξ and S computed by replacing the angular velocity provided by the gyroscope ω by $\omega - B$.

By minimizing this cost function, we recover the gyroscope bias B and the unknown vector X . Since our cost function requires an initialization and is non-convex (see Fig.), the optimization process can be stuck in local minima. However, We can initialize the optimization process with $B = 0_3$ since the bias is usually rather small. Fig. 6 displays the three components of the bias estimated by this minimization.

The optimized closed-form solution provides better results than the standard closed-form solution. Fig. 7 shows an improvement in precision of around 13% for the distance to the features, and around 30% for the speed after 4 seconds of integration.

Moreover, this method is robust even for high values of the gyroscope bias. Fig. 8 displays the performance of the proposed method in estimating speed, gravity in the local frame, and distances to the features in presence of the same artificial gyroscope bias from Fig. 5.

As seen in Fig. 8, after a certain integration duration, the estimations agree no matter how high the bias is. In other words, given that the integration duration is long enough, this method is unaffected by the gyroscope bias.

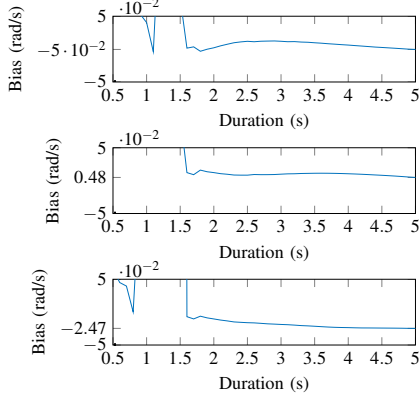


Fig. 6: Gyroscope bias estimation from nonlinear minimization of the residual. We are observing 12 features over a variable duration of integration.

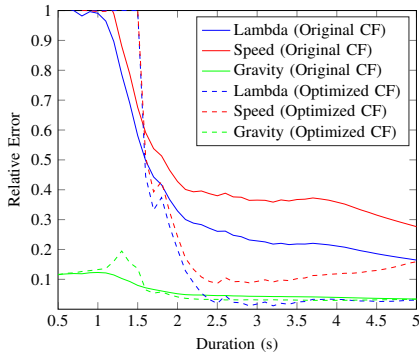
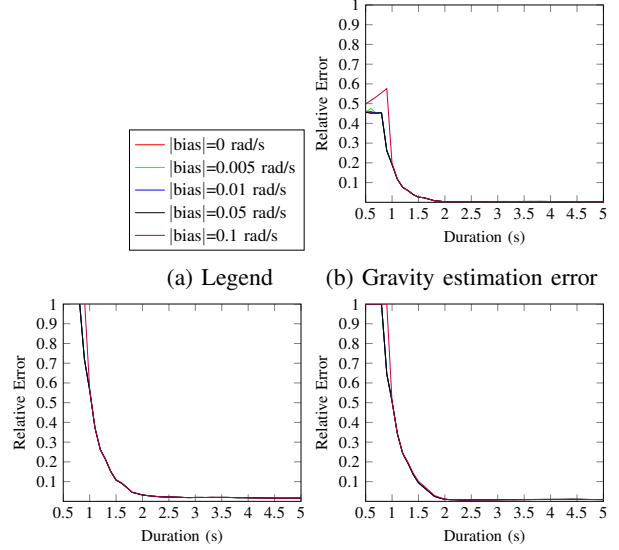


Fig. 7: Estimation error of the original closed-form solution against the optimized closed-form solution. We are observing 12 features over a variable duration of integration.

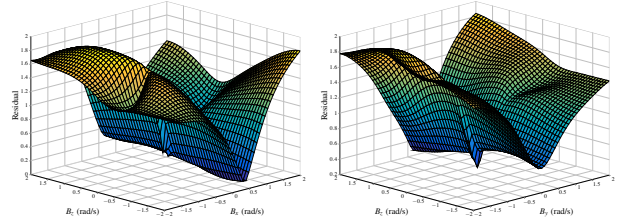
However, for very short time of integration (< 2.25 seconds), the proposed method can fail by providing a gyroscope bias much larger than the correct one. To understand this misestimation, in Fig. 9 we plot the residual with respect to the bias, which is the cost function we are minimizing. We highlight a misestimation of the gyroscope bias by setting the duration of integration to 2 seconds while observing 7 features. We refer to the components of the gyroscope bias by $B = [B_x, B_y, B_z]$.

As we can see in Fig. 9, the cost function admits a symmetry with respect to B_z . This symmetry replicates the minima of the true gyroscope bias along B_z . The optimization process can therefore diverge from the true gyroscope bias. In the next section, we present a method to use a priori knowledge to guide the optimization process.



(c) Velocity estimation error (d) Lambda estimation error

Fig. 8: Impact of the gyroscope bias on the performance of the optimized closed-form solution. We are observing 7 features over a variable duration of integration.



(a) Residual with respect to B_x (b) Residual with respect to B_y and B_z

Fig. 9: Cost function (residual) with respect to the gyroscope bias for a small amount of available measurements (integration of 2 seconds while observing 7 features)

B. Removing the symmetry in the cost function

The symmetry in the cost function is induced by the strong weight of the gravity in the Equation (1). In general, the residual is almost constant with respect to the component of the gyroscope bias along the direction \vec{u} when \vec{u} is collinear with the gravity throughout the motion. Since a MAV normally operates in near-hover configurations, \vec{u} is approximated to the vector pointing upward in the gyroscope frame when the MAV is hovering.

If the MAV rotates such that \vec{u} becomes noncollinear with the gravity, the cost function does not exhibit this symmetry anymore. In this case, the gyroscope bias is well estimated.

A simple solution to avoid having that symmetry in our system would be to enforce that there is no such

\vec{u} by forcing our MAV to perform rotations while it is operating. Another way to artificially get rid of this symmetry is to tweak the cost function. Specifically, we can add a regularization term that penalizes high estimations of the component of the gyroscope bias along \vec{u} :

$$\text{cost}(B) = \|\Xi X - S\|^2 + \lambda (\vec{u} \cdot B)^2, \quad (5)$$

with:

- \vec{u} the direction collinear with the gravity throughout the motion;
- λ the coefficient given to how much we want to penalize this bias component.

For small values of λ , our cost function is similar to the previous one and the bias can grow arbitrarily high. Note that, instead of forcing this gyroscope bias component to be close to 0, we can easily force it to be close to any value. Therefore, we can use the a priori knowledge of a gyroscope bias approximation:

$$\text{cost}(B) = \|\Xi X - S\|^2 + \lambda (\vec{u} \cdot (B - B^{\text{approx}}))^2,$$

with B^{approx} being the known approximate gyroscope bias. This method allows us to reuse previously-computed gyroscope bias since it is known to slowly vary over time.

Selecting a reliable and safe value for the regularization parameter λ is complicated. In this paper, we picked the value of λ empirically.

VI. EXPERIMENTS ON REAL DATA

We validate our method on a real dataset containing IMU and camera measurements from a flying quadrotor along with ground truth.

A. Experimental setup

For our evaluation, we consider a MAV flying in a room equipped with a motion-capture system. This allows us to compare the estimations of the velocity along with the roll and pitch angles against ground truth.

We use the same MAV used in [16], Section 3.4. Specifically, our quadrotor relies on the frame of the Parrot AR.Drone 2.01 including their motors, motor controllers, gears, and propellers. It is equipped with a PX4FMU autopilot and a PX4IOAR adapter board. The PX4FMU includes a 200Hz IMU. The MAV is also equipped with a downward-looking MatrixVision mvBlueFOX-MLC200w (752×480 -pixel) monochrome camera with a 130-degree field-of-view lens (Fig. 10a). The data are recorded using an Odroid-U3 single-board computer. The MAV flies indoors at low altitude (1.5m)



(a) A closeup of our quadrotor: 1) down-looking camera, with an OptiTrack motion-capture system (for ground-truth recording), 2) Odroid U3 quad-core computer, 3) PIXHAWK autopilot.

Fig. 10: Experimental setup for identifying the limitations of the performance. The drone is equipped with an IMU and a down-looking camera.

(Fig. 10b). The feature extraction and matching is done via the FAST corners [18].

B. Results

We compare the performance on the estimations of the gravity and the initial velocity obtained with three different methods:

- The original closed-form solution [1] (Equation 2);
- Our modified closed-form solution (this paper) (Equation 4);
- The loosely-coupled visual-inertial algorithm (MSF) in [19] using pose estimates from the Semi-direct Visual Odometry (SVO) package [7] (how to combine MSF [19] with SVO can be found in [16]).

The reason we included SVO+MSF in the validation is to have a reference state-of-the-art pose estimation method. However, MSF requires to be initialized with a rough absolute scale, whereas our method works without initialization.

We set the integration duration for the closed-form solution to 2.8 seconds, since it is sufficient for the closed-form to provide robust results (see Fig. 7). The camera provides 60fps, but we discard most of the frames and consider only 10Hz (this is discussed in section IV-A).

As seen in Fig. 11a, the performance obtained by our method is similar than the performance obtained by a well-initialized MSF. We remind the reader that unlike MSF, the closed-form solution does not require the knowledge of the absolute scale to be provided.

Moreover, the original closed-form solution and the optimized closed-form solution have similar performance. Indeed, for this dataset the gyroscope bias was estimated to $B = [0.0003, 0.009, 0.001]$, which is very small ($\|B\| = 0.0091$). To prove the robustness of our

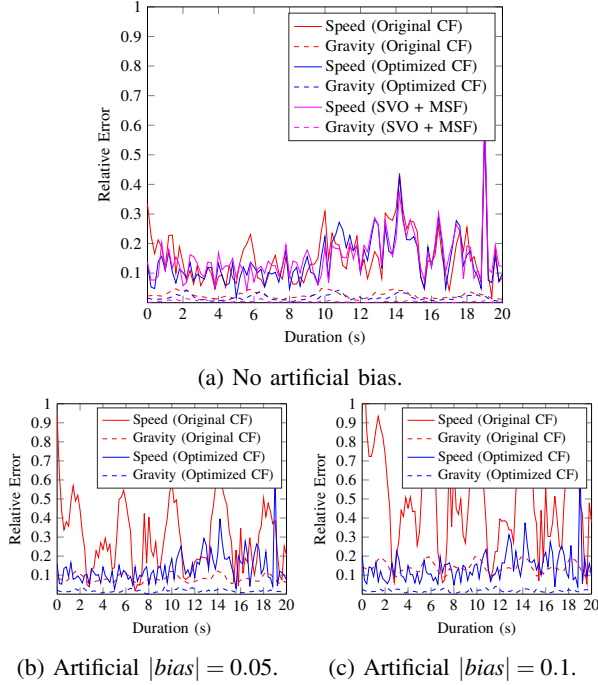


Fig. 11: Estimation error of the optimized closed-form solution against the original closed-form solution [1] and SVO [16]. The duration of integration is set to 2.8 seconds, and 10 point features are observed throughout the whole operation.

method compared to the original closed-form, we corrupt the gyroscope measurements with an artificial bias in Fig. 11b and Fig. 11c. As seen in these figures, our method is robust against gyroscope bias whereas the original closed-form is not.

VII. CONCLUSION

In this paper, we studied the recent closed-form solution proposed by [1] which performs visual-inertial sensor fusion without requiring an initialization. We implemented this method in order to test it with real-world data. This allowed us to identify its performance limitations and bring modifications to overcome them. We investigated the impact of biased inertial measurements. Although the case of biased accelerometer was originally studied in [1], we showed that the accelerometer bias does not significantly worsen the performance. One major performance limitation of this method was due to the impact of biased gyroscope measurements. In other words, the performance becomes very poor in presence of a bias on the gyroscope and, in practice, the overall method could only be successfully used with a very precise (and expensive) gyroscope. We then introduced a simple method that automatically estimates this bias.

We validated this method by comparing its performance against state-of-the-art pose estimation approach for MAV. For future work, we see this optimized closed-form solution being used on a MAV to provide accurate state initialization. This would allow aggressive take-off maneuvers, such as hand throwing the MAV in the air, as already demonstrated in [17] with a range sensor. With our technique, we could get rid of the range sensor.

REFERENCES

- [1] A. Martinelli, "Closed-form solution of visual-inertial structure from motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014.
- [2] —, "Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.
- [3] L. Armesto, J. Tornero, and M. Vincze, "Fast Ego-motion Estimation with Multi-rate Fusion of Inertial and Vision," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 577–589, 2007.
- [4] M. Li and a. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *The International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013. [Online]. Available: <http://ijr.sagepub.com/cgi/doi/10.1177/0278364913481251>
- [5] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "On the complexity and consistency of UKF-based SLAM," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4401–4408, 2009.
- [6] M. Bibuli, M. Caccia, and L. Lapierre, "Sliding Window Filter with Application to Planetary Landing," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 7, no. PART 1, pp. 81–86, 2007.
- [7] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [8] L. Stefan, L. Simon, B. Michael, S. Roland, and F. Paul, "Keyframe-based visual-inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2014.
- [9] F. C., C. L., D. F., and S. D., "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation," in *Robotics: Science and Systems (RSS)*, 2015.
- [10] A. Mourikis, S. Roumeliotis, et al., "A dual-layer estimator architecture for long-term localization," in *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. IEEE, 2008, pp. 1–8.
- [11] T. Lupton and S. Sukkarieh, "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *Robotics, IEEE Transactions on*, vol. 28, no. 1, pp. 61–76, 2012.
- [12] G. P. Huang, A. Mourikis, S. Roumeliotis, et al., "An observability-constrained sliding window filter for slam," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 65–72.
- [13] A. Mourikis, S. Roumeliotis, et al., "A multi-state constraint kalman filter for vision-aided inertial navigation," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 3565–3572.
- [14] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [15] S. M. Weiss, "Vision Based Navigation for Micro Helicopters (PhD Thesis - Weiss 2012)," no. 20305, 2012.

- [16] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, "Autonomous, vision-based flight and live dense 3d mapping with a quadrotor micro aerial vehicle," *Journal of Field Robotics*, 2015.
- [17] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor," in *International Conference on Robotics & Automation*, 2015.
- [18] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951 LNCS, pp. 430–443, 2006.
- [19] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 3923–3929, 2013.