

# IMU-based pose-estimation for spherical robots with limited resources

Jasper Zevering\*, Anton Bredenbeck, Fabian Arzberger, Dorit Borrmann, Andreas Nuechter<sup>1</sup>

**Abstract**—Spherical robots are a robot format that is not yet thoroughly studied for the application of mobile mapping. However, in contrast to other forms, they provide some unique advantages. For one, the spherical shell provides protection against harsh environments, e.g., guarding the sensors and actuators against dust and solid rock. This is particularly useful in space applications. Furthermore, the inherent rotation the robot uses for locomotion can be exploited to measure in all directions without having the sensor itself actuated. A reasonable estimation of the robot pose is required to exploit this rotation in combination with sensor data to create a consistent environment map. This raises the need for interpolating instances for calculation-intensive slow algorithms such as optical localization algorithms or as an initial estimate for subsequent SLAM. In such cases, inertial measurements from sensors such as accelerometers and gyroscopes generate a pose estimation for these interpolation steps.

The paper presents a pose estimation procedure based on inertial measurements, that exploits the known dynamics of a spherical robot. It emphasizes a low jitter to maintain constant world measurements during standstill and avoids exponentially growing error in position estimates. Evaluating the position and orientation estimates with given ground truth frames shows that we reduce the jitter in orientation and handle slip and partly slide behavior as other commonly used filters such as the Madgwick filter.

## I. INTRODUCTION

Spherical robots are a relatively narrow field of robotics. Still, they could be useful in situations where the measurement equipment needs to be protected against harsh environments. As an example, the 2021 CDF study LunarCaves by ESA about the feasibility of a spherical robot, called DAEDALUS, for exploration of lunar-lava tubes [1] shows the need for orientation and position estimation of spherical robots with the limitations on resources common for space-applications. Although a purely Inertial Measurement Unit (IMU)-based estimation is not recommended for various reasons, the lack of an absolute reference among others, IMU-based pose estimation is used as part of an overall multi-sensor-fusion-based estimation. IMUs are sensors with higher refresh rates than Simultaneous Localization and Mapping (SLAM) based algorithms, such as optical or Light Detection and Ranging (LIDAR) SLAM. This makes them suitable for interpolating the more precise and absolute

position reference points of these slower algorithms. The limited computational power of space-qualified CPUs further imposes the need for good orientation estimation without computational-intensive algorithms like Kalman filters. Lastly, the geometry of a spherical robot opens the possibility for the not often used position estimation by IMUs.

This paper introduces an IMU-based orientation and position algorithm for spherical robots with limited computational power and the need for real-time estimations. Also, the algorithm does not rely on one specific locomotion approach, but it assumes a rotation of the IMU together with at least the shell of the robot, if not the whole robot. Such as the prototype developed at the computer science chair VII of the University of Wuerzburg, depicted in Figure 1, which is inspired by previous work such as Daedalus [1]. The evaluations made in this paper were done with the setup of the IMUs of this prototype. The Livox Mid-100 was used as laser scanner. Due to the lack of a locomotion system, the sphere needs to be rolled externally. The evaluation will also show and discuss the resulting point clouds of this prototype with the presented pose estimation.

## II. ASSUMPTIONS AND PRELIMITATIONS

The presented approach works for any spherical robot which moves based on rotation and possesses IMUs. Nevertheless, it requires multiple assumptions to be beneficial in comparison to existing approaches. Some of these arise from the fundamental behavior of spherical robots and some more specific from the DAEDALUS project. Therefore we state the following assumptions and requirements for the approach to become beneficial:

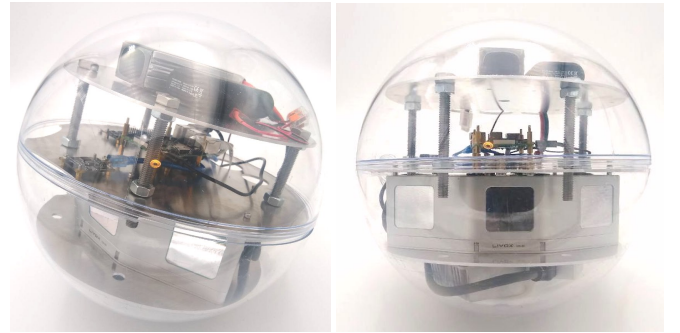


Fig. 1. Prototype to test the algorithm in an envisaged technical environment. The main payload is the Livox Mid-100 laser-scanner. For pose-estimation, three IMUs of the manufacturer Phidget are placed on the middle plate and a Raspberry Pi 4 for the calculations. On the top are two batteries and on the bottom one voltage stabilizer and breakout box of the laser-scanner.

\* Corresponding author:  
info@jasperzevering.de

<sup>1</sup> All authors are with Informatics VII : Robotics and Telematics - University of Wuerzburg

We acknowledge funding from the ESA Contract No. 4000130925/20/NL/GLC for the “DAEDALUS – Descent And Exploration in Deep Autonomy of Lava Underground Structures Open Space Innovation Platform (OSIP) lunar caves-system study and the Elite Network Bavaria (ENB) for providing funds for the academic program Satellite Technology”.

1. *Real Time Operations:* The operation of the robot does not allow post-calculation of the pose. This implies the usage of onboard computational power.

2. *Limitation of Computational Power:* The need for efficient code without massive bottleneck operations, as the inversion of a matrix with the extended Kalman Filter. The code must minimize the usage of non-basic operations.

3. *Low Costs of Hardware:* We assume biased and noisy data in a non-negligible amount from the sensors.

4. *Sphere Shape:* The code relies heavily on the assumption of a sphere as a shape with a constant robot radius. We also assume that rotation of the sphere on the ground leads to translation.

5. *Slip and Sliding:* We assume the sphere has both slip, i.e., the sphere rotates, but there is no translation, and sliding, i.e., the sphere has translation without rotation in a limited, non-permanent way. The limitation is the assumption that sliding and slipping reduce the amount of translation for a given rotation and vice versa, however, not for complete absence.

6. *Stability of Pose:* For simultaneous localization and mapping (SLAM) purposes using LIDARs, a stable but maybe minimal false pose is preferred over a more exact but noisy and jumping position.

7. *Further Processing of Pose:* The approach should avoid jumping values or abrupt changes of data if not clearly indicated by the sensors. Suppose there is an internal algorithmic change of behavior due to the change of state from standing to rolling. This change shall not be represented by an unnatural acceleration in the data.

8. *Uncertainty of Iterative Position Integration:* IMUs deliver no absolute reference for a position as they do for orientation in the form of the noisy but measurable gravity vector. So without external or secondary sensors relying on absolute waypoints, the position estimation will always underlie the integration of errors.

9. *Space Suitability:* As the code is considered to be applied to space missions, the magnetometer data is not considered in the algorithm, despite its positive impact on orientation estimation on earth.

10. *Short Term Mobility:* As the code still relies on integration parts, the sphere is not considered to roll continuously for a long period but rather makes multiple short paths. This will also be necessary, as, in further extension, the fusion of a pose estimation by the laser scanner will be merged, requiring standstill periods for scanning.

11. *Non-reliability of locomotion commands:* For DAEDALUS, the same input of the locomotion controller leads to heavily different behaviors [1]. Also, the DAEDALUS robot has two utterly different locomotion methods; therefore, we will not take locomotion itself, nor the controller input or output into account of pose estimation.

We conclude that the generated data is a compromise between on-board simple filtration of data, while still being usable for heavy calculation post-record, e.g. map generation. It is easy to calculate and adapt the specific needs. Still, as it makes assumptions about the dynamics of the system,

it delivers reliable data which takes physical behavior into account. It is real-time capable and does not rely on delayed steps. The use is restricted to spherical or cylindrical robots. It is optimized for gathering laser data where noise position while standing means a lot of calculation in the map creation, and therefore, even if slightly false, a stable position is preferred.

### III. STATE OF THE ART

For pose estimation on embedded hardware, there are three main algorithms.

#### A. Kalman Filter

The Kalman Filter or the more suitable Extended Kalman Filter (EKF) uses state prediction and merges with sensor data for orientation estimation [2]. For the defined limitations, the Kalman Filter is not suitable. First, the EKF requires a considerable amount of computational power, mainly because of its matrix inversion. For this, the needed calculations grow exponentially for input data. Further, the Kalman Filter relies on input to a system to calculate an estimation of the pose for the next iterative step. But, as we described in the preliminary Section II, we can not take the input to locomotion control as a reference to pose estimation as the behavior is not linear or even predictable without exact knowledge of the surface. Therefore it is not possible to implement a physical behavior of the system to the filter, as easily as it is for other robotic applications. There exists the Error State Extended Kalman Filter, which works only with IMUs [3]. Still, this requires substantial computational power. Furthermore, for the later introduced position estimation for spherical robots, some calculations are used for the orientation as well as position. After all, if computational power is not limited, Kalman Filters provide principally better results.

#### B. Madgwick Filter

The Madgwick filter is a widely used filter for orientation estimation with IMUs [4]. It is efficient and easy to calculate but does not consider the physics of the motion. It uses a gradient descent algorithm. It does nearly provide the sought-after classification for the orientation. The problem is the optimization for movement. The Madgwick filter has a jittering behavior at the standstill of the IMU. This does not meet the desired behavior as described in the prelimitations Section (II). The behavior is shown in comparison to the complementary filter in Figure 2. This jittering behavior is most prominent during standstill as the Madgwick filter was designed for movement. As we do not use a magnetometer, we can not use the gyro bias estimation of the Madgwick filter.

#### C. Complementary Filter

A complementary filter is a fundamental approach of combining precise but not exact with nonprecise but exact data. Therefore it is widely used for combining noisy accelerometer data with gravity as a reference with the precise

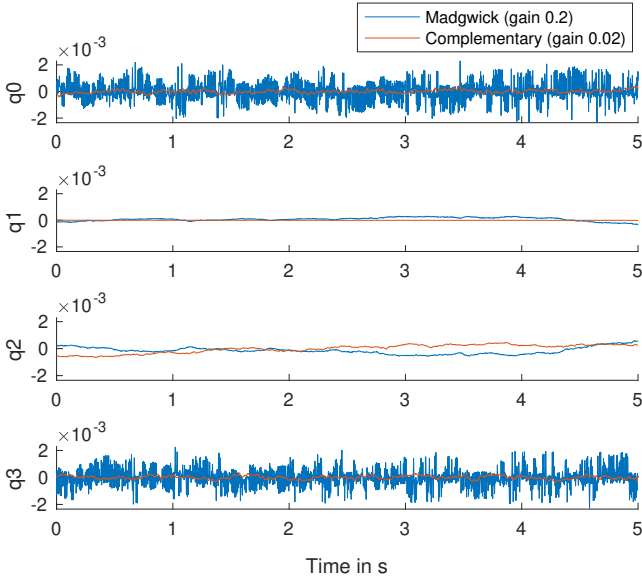


Fig. 2. Quaternion values of Madgwick filter and complementary filter with default gains when IMU is at a standstill. The Madgwick filter has clearly visible jitter, the complementary filter has not.

but drifting measurement of gyroscopes. The basic concept is to have the main part of a value dominated by a gyroscope but, over time, a slow drift towards the measurement of the accelerometer. E.g., for pitch, this results in

$$\theta_n = (1 - \alpha) \cdot (\theta_{n-1} + \Delta\theta_{gyro}) + \alpha \cdot \theta_{acc} \quad (1)$$

where  $\alpha$  is the complementary gain, it determines how much weight the accelerometer data gets and how fast the value drifts towards the measurement of the accelerometer. Therefore the gyroscope functions as high-pass and the accelerometer as low-pass part of the basic complementary filter [5]. A commonly used value for  $\alpha$  is 0.02, meaning every iteration, 2 percent of the new values come from the accelerometer. This corresponds, with the used 125[Hz] calculation, to frequency-border of the low and high-pass of 2.5[Hz] [6]. An  $\alpha$  of 1 means the resulting data is completely the one of the accelerometers, and the gyroscope has no influence.

The complementary filter is best for orientation calculation without position change as the accelerometer value relies on calculating the gravity value. With an acceleration in one direction, this leads to a miscalculation of the gravitational vector and, therefore, a false orientation estimation. Simple orientation estimations of quadcopters often just change the  $\alpha$  gain to 0 while moving to avoid the miscalculation of the gravity vector [7].

#### D. Position Estimation

Position estimation purely by IMU is not recommended as the double integration of the accelerometer measurements leads to exponential error summation [8]. Therefore there are very few approaches targeting this way of position calculation [9]. The very impressive work in [10] also uses IMUs for 6D pose estimation. They follow a pre-specified

trajectory with a manipulator and manage to estimate its pose within 10% of the ground truth. Given their application is rather broad, they do not exploit known dynamics of the system, which could be helpful for spherical robots.

### IV. FILTER-CONCEPT

The pose orientation for the spherical robot is a concept with interconnected components. Both orientation and position estimates are computed based on the gyroscope and accelerometer measurements, taking into account the limitations of the system described in Section ???. The generalized overview of orientation and position estimation is shown in Figure 3.

#### A. Orientation

The orientation estimation is the combination of the Madgwick and a complementary filter. The weaknesses and strengths of both described in Section III are combined. The general idea is to distinguish between motion and standstill. In motion, the Madgwick filter is used, and without motion, the complementary filter. This approach of distinguishing the evaluation algorithm depending on the state of motion of a robot has been done before, like in [11]. The transition needs to be in a differentiable way, as this is one of our limitations. As both filters are applied with weights, this is done by an allocation of a given gain towards one or the other filter depending on the likelihood the sphere is moving or standing still. Therefore, this requires some adapting mechanism and an estimation of how likely the sphere is to rotate and hence translate. Both filters share the same meaning of the gain, describing the influence of the accelerometer on the orientation. The Madgwick filter weight determines how much influence the accelerometer measurement has on the overall orientation, just in a different way than the complementary filter. Thus the implementation of this approach does not endanger the stability of the estimation due to the change between two completely independent approaches. The algorithm can shift the gain from one filter to the other in a smooth way without jumping values or abruptly changing behavior.

#### B. Position

As there is no absolute reference of position neither with calculation by gyroscope nor by the accelerometer, the filter can only try to improve the calculation of the velocity leading to the change of position. The straightforward approach is to double integrate the accelerometer, which leads to the earlier described unwanted behavior of exponential error integration and makes the position estimation completely unusable. The second straightforward approach multiplies the rotation speed with the known radius of the sphere and takes this as velocity, integrating once. The more precise and less noisy data of the gyroscope leads to a usable estimation of position. There is no such extensive error integration, and more importantly: no exponentially growing error with time. However, this approach neglects slip and sliding, as it is one of our limitations and vertical movement. If the

sphere slips, this approach will predict the full translation. Also, the other way around, if there is translation but just slight rotation because of sliding, it will predict too little translation. Furthermore, integrating the rotation does not let us estimate vertical movement. Every rotation will be predicted as translation in a given plane, which is most likely parallel to the ground. For instance, if the sphere rolls up an obstacle, the length of the path is estimated to be the ground-plane.

Our algorithm uses both approaches of position estimation and combines them, and limits all components according to the other components. This implies a connection between translation and rotation. Overall the extreme situation (full translation, no rotation, or the other way round) will automatically lead to wrong predictions, as there is no way to tell which of the two approaches has at that moment more reliable data. Therefore, the combination of both rather than only one is chosen.

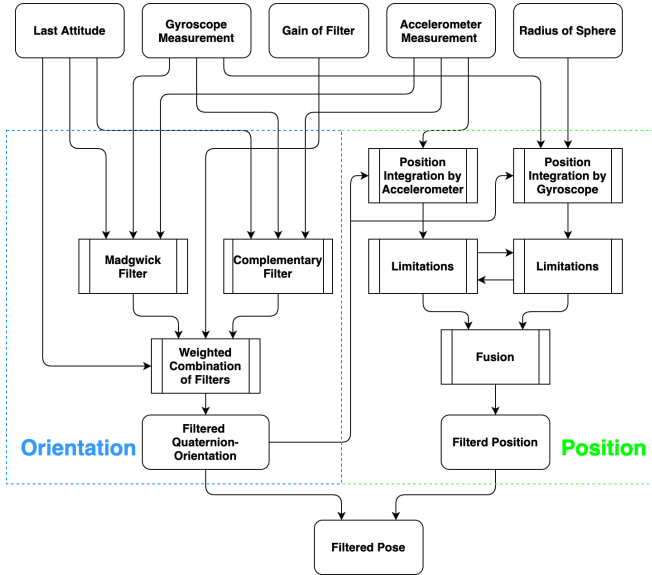


Fig. 3. Diagram of the estimation.

## V. FILTER-IMPLEMENTATION

### A. Orientation

The orientation is a symbiosis of the Madgwick filter and complementary filter. For no- or slow-moving, the complementary filter is chosen; for fast-moving, the Madgwick filter. The Madgwick filter itself is untouched. The standard complimentary filter is adapted to work in quaternion representation, as the direct integration of the gyroscope give angles in the Euler domain. Quaternions were designed under the premise of being free from singularities, i.e., for a continuous change in orientation, there exists a continuous change in quaternion representation [12]. This is not the case for the Euler angle, as a transition of a 359-degree roll to a 0-degree roll has no continuous value representation if limited to 360 degrees. But combining both leads to contradicting updates from each filter with respect to the other. Therefore

the normal way of the complementary filter of the new orientation is not used, as it implies a transition of each value independently. Hence, the RPY-orientation is transformed to a quaternion, and then a quaternion slerp is performed [13]. The quaternion slerp works over the representation of two quaternions on a sphere. The transition from one quaternion to the other is now not done directly but always over the surface of the sphere. This leads to the same handling of quaternions as the Madgwick-filter without endangering robustness due to using two different quaternions with very different values representing nearly the same position.

The transition between the two filters is done by shifting a fixed factor among the  $\alpha$ -gain of the complementary filter and the gain factor of the Madgwick filter. As the common values widely used for both values differ by one order of magnitude, the gain shifted to the complementary filter is divided by ten and the other way around. The values of the gyroscope axes determine in which manner to shift the overall gain to both Filters. The defining values are the threshold until which value the algorithm should use a purely complementary filter and at which threshold a full Madgwick filter should determine the orientation. These values have been determined empirically. Future work will investigate improving those values by developing a suitable heuristic for the thresholds. The transition between the full complementary and the full Madgwick estimation, as is required by the prelimitations, can be done by a function suitable for the specific scenario. This avoids large accelerations of the orientation due to the rapid change from one algorithm to another. In our experiments, when starting from stillstand, the incipient Madgwick filter has a more rapid impact on the orientation values than the incipient complementary filter when stopping rotation. Therefore the transition between  $\alpha$ -gain of the complementary filter and the Madgwick-gain was chosen to have quadratic behavior. The value referred to as autogain  $\Theta$  indicates how much impact both employed filters have on the orientation estimation. Therefore we calculate factors from the gyroscope measurements and scale the autogain with the maximum  $\beta$  (cf. equation 2 and 3).  $g_k$  represents the gyro measurement in the corresponding axis  $k$ , in rad/s. Later on, these factors will again be used for position estimation, namely to determine if the sphere rotates or not. These factors are heuristically defined as:

$$f_k = \begin{cases} 0 & g_k \leq 0.1 \\ 0.25 \cdot (g_k - 0.1)^2 & 0.1 < g_k < 2.1 \\ 1 & g_k \geq 2.1 \end{cases} \quad (2)$$

$$\beta = \max(f_x, f_y, f_z) \quad (3)$$

Then the Madgwick gain  $\gamma$  for a given autogain  $\Theta$  is calculated by

$$\gamma = \Theta \cdot \beta \quad (4)$$

The complementary filter gain  $\alpha$  is calculated by

$$\alpha = \Theta \cdot \theta \cdot (1 - \beta). \quad (5)$$

Thus, the Madgwick gain has a maximum of  $\Theta$  and  $\alpha$  a maximum of  $\theta \cdot \Theta$ . This ratio  $\theta$  can be adapted to specific needs. We used  $\theta = 0.1$  because of the two often used standard values  $\gamma = 0.2$  and  $\alpha = 0.02$ , which have the ration 0.1. Hence for adapting the position estimation to a given robot, the function for calculating the factors needs to be adapted in such a way that it is a good indication for movement or standstill. The ratio  $\theta$  needs to be chosen in a way that the desired ratio between Madgwick and complementary gain is reached. And lastly, the autogain needs to be chosen to fit the specific needs in terms of how strong both filters should be able to influence the orientation. We used  $\Theta = 0.2$  to get the described common values.

### B. Position

The calculated orientation is directly used for position calculation. With the orientation, a gravity vector is calculated, which, once normalized, represents the measurement of the accelerometer in abstinence of movement. Let  $\mathbf{g} = [0 \ 0 \ -1]^T$  represent the gravity vector in the world frame and the matrix  $\mathbf{R}$  represent the current orientation of the IMU, then

$$\mathbf{g}' = \begin{bmatrix} g'_x \\ g'_y \\ g'_z \end{bmatrix} = \mathbf{R} \cdot \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad (6)$$

describes the effect of gravity in the coordinate system of the IMU. This gravity vector is then subtracted from the accelerometer measurement. If there is no translation, the subtraction should lead to  $[0 \ 0 \ 0]^T$  meaning there is no acceleration other than gravity. If there are non-zero components, there is an acceleration in that direction. Given the accelerometer measurement  $\mathbf{a}$  this leads to the

$$\begin{bmatrix} v_{xAcc} \\ v_{yAcc} \\ v_{zAcc} \end{bmatrix} = \int_0^T \left( \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - \begin{bmatrix} g'_x \\ g'_y \\ g'_z \end{bmatrix} \right) dt \quad (7)$$

These values are now coupled to the rotation of both axes other than their own. Therefore, the factors from the orientation step, representing the likelihood of rotation, are used as limiting factors. If one of the three factors is 1, this means there is a translation in this direction. When there is no rotation, it will set the velocity to 0. During fast movement, this allows exponential error integration. This exponential error is taken care of later on.

$$\begin{bmatrix} v_{xAcc} \\ v_{yAcc} \\ v_{zAcc} \end{bmatrix} = \begin{bmatrix} v_{xAcc} \\ v_{yAcc} \\ v_{zAcc} \end{bmatrix} \cdot \begin{bmatrix} \max(f_y, f_z) \\ \max(f_x, f_z) \\ \max(f_x, f_y) \end{bmatrix} \quad (8)$$

The rotation into the world frame is computed as:

$$\begin{bmatrix} v_{xAccWorld} \\ v_{yAccWorld} \\ v_{zAccWorld} \end{bmatrix} = \mathbf{R}^T \cdot \begin{bmatrix} v_{xAcc} \\ v_{yAcc} \\ v_{zAcc} \end{bmatrix}. \quad (9)$$

The next step requires the velocity calculated by the rotation. Therefore the rotations (direct values from the gyro) are rotated into the world frame, i.e.,

$$\begin{bmatrix} \omega_{xWorld} \\ \omega_{yWorld} \\ \omega_{zWorld} \end{bmatrix} = \mathbf{R}^T \cdot \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}. \quad (10)$$

The rotation around the world  $z$ -axis is not used to calculate the velocity. Its effect on the position is considered when determining the orientation in the world coordinate system.  $\omega_{zWorld}$  does not need to be computed as it has no influence on the position.

The resulting velocities are calculated by multiplying the circumference of the sphere:

$$\begin{bmatrix} v_{xzGyroWorld} \\ v_{yzGyroWorld} \end{bmatrix} = \begin{bmatrix} \omega_{yWorld}/(2\pi) \\ \omega_{xWorld}/(2\pi) \end{bmatrix} \cdot 2\pi r = \begin{bmatrix} \omega_{yWorld} \\ \omega_{xWorld} \end{bmatrix} \cdot r. \quad (11)$$

As the gyro measures rad/s the value is divided by  $2\pi$ . Note that the two-dimensional velocities  $v_{xzGyroWorld}$  and  $v_{yzGyroWorld}$  do not correspond to the velocity on the ground plane but may consist of motion in  $z$  direction.

Next, the  $v_{xAccWorld}$  and  $v_{yAccWorld}$  are limited depending on  $v_{xzGyroWorld}$  and  $v_{yzGyroWorld}$ , in our example, this was 120 percent of the gyro calculated velocity. This ensures no exponential error integration when in motion. This is coupled to the physical act of sliding. The higher the likelihood of sliding, the higher the velocity should be compared to the velocity based on rotation. This does not count for slipping, as this implies the adaption of limitation of the velocity by rotation by the velocity measured by the accelerometer. But there is no need for a limitation as there is no double integration with the rotation.

The last step for calculating the velocity is taking the velocity in  $z$ -direction into account for the velocity by rotation. This velocity in  $z$  is from the noisy double integration, as this is the only indication for change of the  $z$ -axis. To avoid large errors and unlimited double error integration,  $v_{zAccWorld}$  is limited to the mean of  $v_{xzGyroWorld}$  and  $v_{yzGyroWorld}$ . With the Pythagorean theorem the velocity in  $x$  and  $y$  is calculated accepting the velocity in  $z$ -direction, which can not be derived by the velocity from gyro:

$$\begin{bmatrix} v_{xGyroWorld} \\ v_{yGyroWorld} \end{bmatrix} = \begin{bmatrix} \sqrt{v_{xzGyroWorld}^2 - v_{zAccWorld}^2} \\ \sqrt{v_{yzGyroWorld}^2 - v_{zAccWorld}^2} \end{bmatrix} \quad (12)$$

. The last step to get the position is the integration of the velocity:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \int_0^T \begin{bmatrix} (1 - \beta)v_{xAccWorld} + \beta \cdot v_{xGyroWorld} \\ (1 - \beta)v_{yAccWorld} + \beta \cdot v_{yGyroWorld} \\ v_{zAccWorld} \end{bmatrix} dt \quad (13)$$

. Depending on the practical circumstances and quality of measurement, the factor  $\beta$  is chosen, that the position relies more on the accelerometer calculated velocity or the gyro. There should always be a mixture of both if there is slipping and sliding.  $\beta = 0.5$  results in just the mean of both approaches.



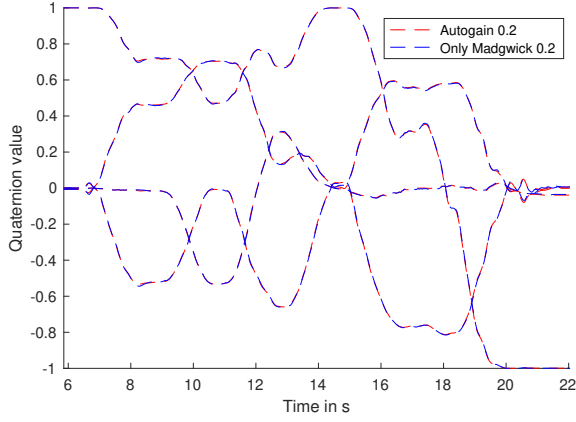


Fig. 4. Quaternion-values of the presented fusion of complementary and Madgwick filter and only the Madgwick filter.

### C. Time Complexity

Overall a pose estimation step consists only of operations that are constant in time. Given that we use the fast inverse square root algorithm [14] for normalization, the trigonometric functions are considered the remaining bottleneck. Overall there are  $4 \times \arctan$ ,  $4 \times \sin$ ,  $4 \times \cos$ ,  $1 \times \arcsin$ ,  $1 \times \arccos$  and 5 divisions. The IMU measurements are collected at 125 Hz and are processed without problems on a raspberry pi. Hence no overwhelming computational load is to be expected.

## VI. EVALUATION

The proposed algorithm is evaluated by two sets of experiments performed with three Phidgets 3/3/3 IMUs mounted on a round acrylic glass plate. For position experiments, the plate is put inside an acrylic glass sphere with the same radius as the glass plate. First, the fusion of Madgwick and complementary filter, here denoted as Autogain, is compared to a Madgwick only estimate. Second, the trajectory computation using both gyroscope and accelerometer is evaluated.

### A. Orientation

A first experiment is performed to show the overall performance of the Autogain computation for the orientation. The IMU was manually rotated in arbitrary directions, and the orientation was plotted using the Madgwick filter and the Autogain filter. As expected, the overall behavior of the presented algorithm and the Madgwick filter is very similar in the dynamic case, as shown in Figure 4. Therefore the more expressive part is the change between motion and standstill, as in motion, both filters use the same algorithm but differ during standstill and slow motion. As described in Section III, the Madgwick filter has a rather strong jitter during slow movements and standstill. This comes due to the (too) large steps of gradient descent iterations. To compensate for this, one needs to lower the gain, leading to more imprecise results during movement, as large iteration steps are beneficial there. Alternatively, one needs to increase

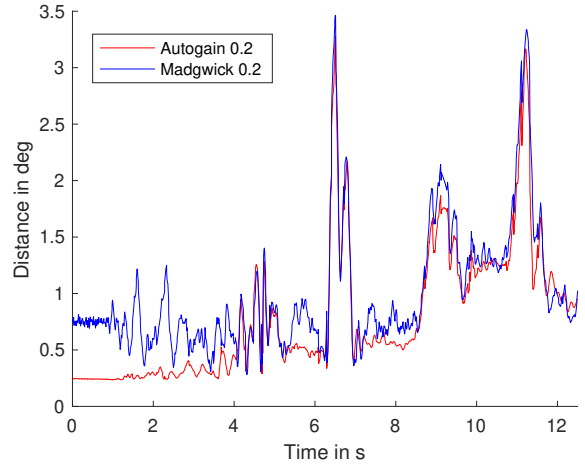


Fig. 5. Distance of the proposed and the Madgwick algorithm to the optical tracking system, using the distance definition of [15].

calculation frequency, which is in our scenarios limited by the hardware capabilities of the IMU.

To evaluate specifically the change between slow motion and standstill, a further experiment is performed. Figure 5 show the distance to a precise external reference over a short excerpt of the experiment with three phases of slow motion and rotation at second 7, 9 and 11. For external reference, we used a precise optical tracking system. The definition and proposed calculation of the distance of 2 quaternions of [15] was used. It describes the direct angle between two quaternions. The optical system matches the overall orientation of both IMU-based algorithms quite well and lacks the perturbations of the pure Madgwick filter. The transformation of the orientation by the optical tracking system and the IMU-based algorithms yield two unknown transformations, the relative orientation of the coordinate system of IMU to the optical coordinate system and how the axes are interpreted on the tracked structure itself. Therefore, the algorithm proposed in [16] was used to solve these two unknown transformations on basis of the resulting data, to match the orientations of optical and IMU-based data. As this matching of the optical system happens on basis of one of the two IMU orientation data-sets, the comparison of absolute differences may be misleading, as the transformations were matched on basis of the autogain filter. To stress the jitter behavior of the Madgwick without relying on absolute orientation, which will be part of further research, Figure 6 shows the derivative of the distance. Here the change of direction due to small oscillations and the smoother fusion algorithm in comparison to the tracking is even more apparent.

### B. Position

The primary evaluation for the position focuses on slip and slide-behavior. Here the plate with the IMU(s) is fixed inside a sphere with a diameter of 29 cm and rolled along an “L”-shaped track of one meter by one meter. Slip behavior was manually provoked. Figure 7 shows the resulting trajectories. Pure rotation integration leads to an enlarged “L” by

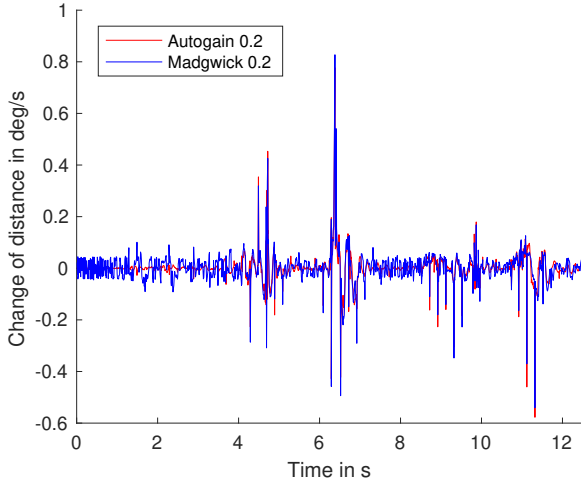


Fig. 6. Derivative of the distance of the proposed and the Madgwick algorithm to the optical tracking system, using the distance definition of [15].

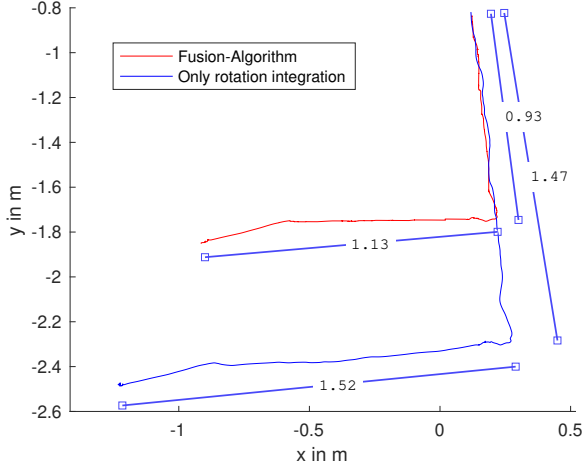


Fig. 7. Ground position (Z-axis, perpendicular to the ground) by integration of rotation and the fusion algorithm. An L-shape with 1 meter by 1 meter was performed with provoked slip(rotation without translation). The boxes are manually set points as a reference. The distance in meter between these points is written on the lines connecting them. The rolling started in the upper right corner and ended in the bottom left corner.

about 50 percent, as the manually provoked rotation is just integrated and therefore interpreted as linear motion. The fusion between the accelerometer and gyroscope interprets an “L” with dimensions much closer to the original. The estimation leaves room for improvement, as in one direction, it misses the actual length about -10 percent, and in the other direction overestimates it by 10 percent. But it shows the overall capability of the algorithm to compensate for slip.

The same experiment was performed with added slide, i.e., more translation than just the rotation produces. Here the results shown in Figure 8 were not as reliably good as with the slip experiment. This is due to the timing of the acceleration. In order to recognize the acceleration as translation, the rotation needs to be above a certain threshold, as

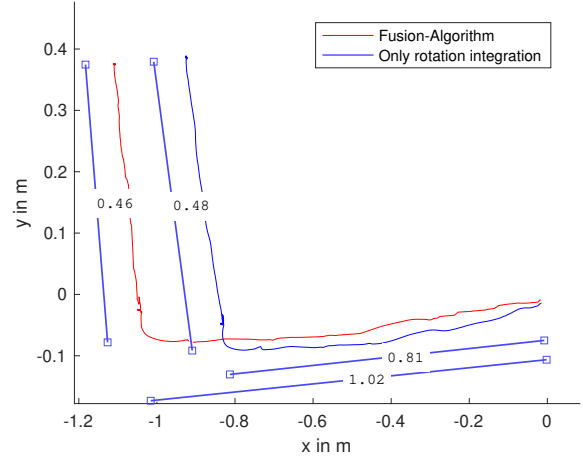


Fig. 8. Ground position (Z-axis, perpendicular to the ground) by integration of rotation and the fusion algorithm. An L-shape with 1 meter by 1 meter was performed with provoked slide(translation without rotation). The boxes are manually set points as a reference. The distance in meter between these points is written on the lines connecting them. The rolling started in the lower right corner and ended in the upper left corner.

the algorithm does not allow any translation without rotation at all. The translation is computed using the accelerometer via double integration. Therefore the acceleration of the slide movement needs to be at the same time as a minimal amount of rotation. If the rotation starts after the sliding, the double integration of the acceleration is suppressed. When the rotation starts and the translation could be double integrated, it is already zero as the velocity is constant. This can be seen in Figure 8 where the one side of the L was interpreted exactly as long as the pure integration, i.e., approx. 50 cm, and therefore the slide was not recognized, and the other part was correctly interpreted longer than just the integration and with 1.02 m as long as the actual trajectory. This is to be blamed on the execution of the experiment, as it was not perfectly ensured only to start sliding with ongoing rotation, which is hard to achieve when manually guiding a sphere. However, the sliding behavior in a real environment could be in both ways, too, so sliding while rolling and starting sliding before rolling. With slip, this is not a problem, as, by definition, the translation cannot begin before the rotation when slipping. If so, it would be sliding. The experiments for sliding did not reliably detect the slide event and, on some occasions, interpreted the distance as shorter than just the integration, as the acceleration of the linear motion was missed. However, the deceleration of the sphere was still recognized while rotation existed, and therefore, the sphere miscalculated a couple of centimeters.

### C. Impact on 3D mapping

One major design driver for the presented algorithm is the suitability for spherical SLAM purposes. Therefore, a pose estimate is preferred to be slightly wrong, if it is more continuous, i.e. less noisy and with less jumping values, than the exact pose has.

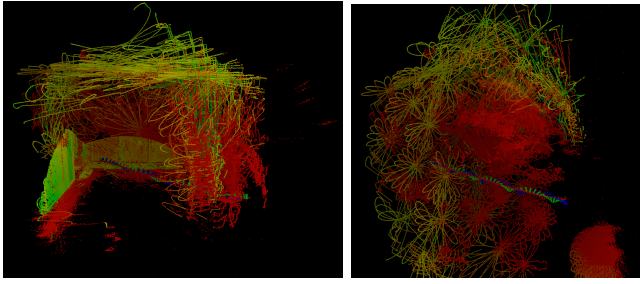


Fig. 9. Above: resulting 3D map using the presented algorithm for pose filtering. Below: resulting 3D map using Madgwick + double integrator for pose filtering. Both images were shot from the exact same point of view. Both plots use the same LIDAR data and apply one of the simultaneously generated pose estimations.

To test the impact of the filter on 3D mapping, we used the prototype in Figure 1. After 2 s of initial standstill, the sphere rolled approx. 4 m within 5 s in an indoor environment from one corner of the room to the opposing one. We extracted the laserscan data, as well as the filtered pose using the presented algorithm. Without further pre- or post-processing, the pose data is combined with the laserscan data to create a 3D map. Figure 9 compares the result with another 3D map, using the in the same run recorded pose-estimation by the Madgwick filter and simple rotation integration. Therefore the same laserscan-data is mapped with different pose estimations. The presented filter improves the resulting map compared to the Madgwick/integrator approach. In particular, the walls and the ceiling are more recognizable, which is favorable for many SLAM algorithms. Another improvement is that points were no longer sensed below the ground, i.e. the plane on which the sphere was moving on. The more detailed part of the map, as seen in the bottom left corner of the above image of Figure 9, corresponds to the initial standstill. It is more detailed than the rest of the map because of the non-repeating flower-shaped scanning pattern, where point density increases with time. Note that the same detailed part is also present in the image below. However, it is out of view (at the bottom right corner) as the Madgwick filter misses the initial pose estimation by a lot.

#### D. Conclusion

This paper introduced an IMU-based pose estimation filter based on the fusion of multiple already existing filters and approaches. It is optimized for specific needs and circumstances of spherical robots with laser scanning tasks in a space-suitable way. Needless to say, a lot of work remains to be done. The fusion of complementary and Madgwick orientation filter solves the shortcomings of the Madgwick when at a standstill, particularly the jitter. A first qualitative evaluation shows the functionality of the filter. In further experiments, a quantitative Orientation comparison with a synchronized optical system needs to be performed. Also, as further work, all puzzle pieces, this filter, a locomotion system, a scanning system need to be combined to an overall working prototype.

The presented position estimation shows the capability to

overcome the problem of the missing absolute reference of IMUs partly. It mainly improves the measurements in the ground plane with slip behavior present. Still, there is a need for parameter tuning. Sliding was not detected all the time reliably. Here a mechanism for a delayed rotation needs to be investigated, that acceleration shortly before a rotation is integrated and taken into account once rotation starts. The approach for, at least roughly, indicating the change in the  $z$ -direction, which is now done only by the accelerometer values, shows no promising results nor usable results in first qualitative experiments with rolling over obstacles and therefore needs to be re-investigated.

#### REFERENCES

- [1] Angelo Pio Rossi, Francesco Maurelli, Vikram Unnithan, Hendrik Dreger, Kedus Mathewos, Nayan Pradhan, Dan-Andrei Corbeanu, Riccardo Pozzobon, Matteo Massironi, Sabrina Ferrari, et al. Daedalus-descent and exploration in deep autonomy of lava underground structures: Open space innovation platform (osip) lunar caves-system study. 2021.
- [2] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [3] Mohammad Ahmadi, Alireza Khayatian, and Paknush Karimaghaee. Orientation estimation by error-state extended kalman filter in quaternion vector space. In *SICE Annual Conference 2007*, pages 60–67. IEEE, 2007.
- [4] Sebastian Madgwick. An efficient orientation filter for inertial and inertial/magnetic sensor arrays. *Report x-io and University of Bristol (UK)*, 25:113–118, 2010.
- [5] Hyung Gi Min and Eun Tae Jeung. Complementary filter design for angle estimation using mems accelerometer and gyroscope. *Department of Control and Instrumentation, Changwon National University, Changwon, Korea*, pages 641–773, 2015.
- [6] Szabolcs Hajdu, Sándor Tihamér Brassai, and Iuliu Szekely. Complementary filter based sensor fusion on fpga platforms. In *2017 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM) & 2017 Intl Aegean Conference on Electrical Machines and Power Electronics (ACEMP)*, pages 851–856. IEEE, 2017.
- [7] Robert Mahony, Tarek Hamel, and J-M Pflimlin. Complementary filter design on the special orthogonal group so (3). In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 1477–1484. IEEE, 2005.
- [8] YK Thong, MS Woolfson, JA Crowe, BR Hayes-Gill, and DA Jones. Numerical double integration of acceleration measurements in noise. *Measurement*, 36(1):73–92, 2004.
- [9] Manon Kok, Jeroen D Hol, and Thomas B Schön. Using inertial sensors for position and orientation estimation. *arXiv preprint arXiv:1704.06053*, 2017.
- [10] J-S Botero Valencia, M Rico Garcia, and J-P Villegas Ceballos. A simple method to estimate the trajectory of a low cost mobile robotic platform using an imu. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, 11(4):823–828, 2017.
- [11] Joachim Hertzberg, Kai Lingemann, and Andreas Nüchter. *Mobile Roboter: Eine Einführung aus Sicht der Informatik*, page 145. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
- [12] Richard Becker. Dealing with the quaternion antipodal problem for advecting fields. Technical report, US Army Research Laboratory US Army Research Laboratory United States, 2017.
- [13] Erik B Dam, Martin Koch, and Martin Lillholm. *Quaternions, interpolation and animation*, volume 2. Citeseer, 1998.
- [14] Origin of quake3’s fast invsqrt(). [beyond3d.com/content/articles/8/](http://beyond3d.com/content/articles/8/). Accessed: 2021-05-01.
- [15] James J Kuffner. Effective sampling and distance metrics for 3d rigid body path planning. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004*, volume 4, pages 3993–3998. IEEE, 2004.
- [16] Fadi Dornaika and Radu Horaud. Simultaneous robot-world and hand-eye calibration. *IEEE transactions on Robotics and Automation*, 14(4):617–622, 1998.