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

Jasper Zevering*, Anton Bredenbeck, Fabian Arzberger, Dorit Borrmann, Andreas Nuechter[1]

***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 a protection against harsh environments, e.g. guarding the sensors and actuators from 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. To use this rotation in combination with sensor data to create a consistent environment map a reasonable estimation of the robot pose is required. In such cases the pose estimation can be done by inertial measurements from sensors such as accelerometers and gyroscopes, as interpolating instance for calculation-intensive slow algorithms as optical localization algorithms or as rough estimation. We propose 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. The evaluation of the position and orientation estimates with given ground truth frames shows that we can reduce the jitter in orientation and can 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 the 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 can be used as part of an overall multi-sensor-fusion-based estimation. IMUs are sensors with higher refresh rates than other Simultaneous Localization and Mapping (SLAM) based algorithms, such as optical or Light Detection and Ranging (LIDAR) SLAM, making 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 complex 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.

II. ASSUMPTIONS AND PRELIMITATIONS

The presented approach requires multiple assumptions, to be beneficial-Therefore, we classify the calculated pose concerning its usage and define the usage as follows:

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

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.

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

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.

Slip and Sliding: We assume the sphere has both slip (the sphere rotates, but there is no translation) and sliding (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.

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.

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.

Uncertainty of Iterative Position Integration: IMUs deliever 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

* Corresponding author: jasper.zevering@stud-mail.uni-wuerzburg.de

[1] 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".

on absolute waypoints, the position estimation will always underlie the integration of errors.

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.

Short Term Mobility: As the code still relies on integration parts, the sphere is not considered to have a straight motion 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.

Non-reliablity of locomotion: The sphere from the DAEDALUS example has two completely different locomotion methods; therefore, we will not take locomotion itself into account of pose estimation. Also, the same locomotion-approach can lead to heavily different behavior with the same input to the controller. [1]

Classification of Data: The generated data is a compromise between simple filtration of data and heavy calculation post-record. It is easy to calculate and adapt but due to the number of pre-assumptions, as it makes assumptions about the dynamics of the system, it still delivers reliable data which takes physical behavior into account. It is real-time capable and does not rely on delay steps.

Field of Use: 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. 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, 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.

B. Madgwick Filter

The Madgwick filter is a widely used filter for orientation estimation with IMUs [2]. It is efficient and easy to calculate but does not consider the physics of the motion. 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 1. This behavior is most prominent during standstill as the Madgwick filter was designed for movement.

C. Complementary Filter

A complimentary 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 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 would result in

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

where α 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. A commonly used value for α is 0.02, meaning every iteration, 2 percent of the new values come from the accelerometer. An α 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 quadrocopters often just change the α gain to 0 while moving to avoid the miscalculation of the gravity vector. [3]

D. Position Estimation

Position estimation purely by IMU is not very recommended as the double integration of the accelerometer measurements leads to exponential error summation. Therefore there are very few approaches targeting this way of position calculating [4]. The very impressive [5] 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 useful 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 limitations of the system described in Section II. The generalized overview of orientation and position estimation is shown in Figure 2.

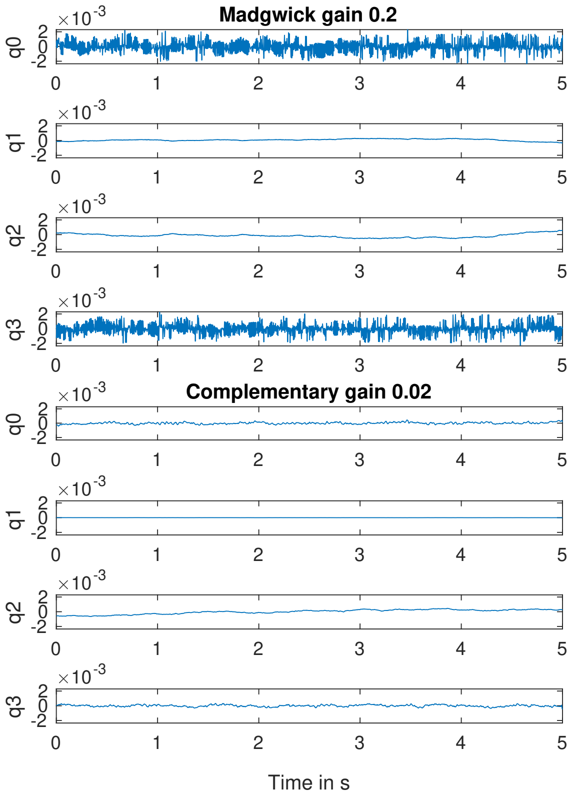

Fig. 1. Quaternion values of Madgwick filter and complementary filter with default gains when IMU is at a stillstand. The Madgwick filter has clearly visible jitter, the complementary filter not

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 can be combined. Therefore the general idea is to distinguish between motion and standstill. In motion, the Madgwick filter is used, and without motion, the complementary filter. 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. If for instance the sphere rolls up an obstacle, the length of the path is estimated to 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 the 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 and therefore the combination of both rather then only one is chosen.

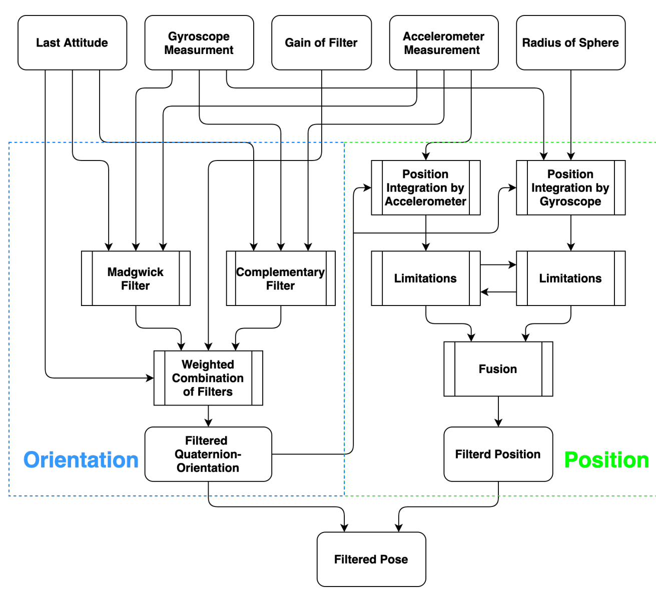

Fig. 2. Generalized diagram of the estimation

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 integration only works in the Euler angles domain, the jumps of values at the border of the domain like 0 and 360 degrees do not match the mostly continuous change of quaternions. As the representation is still valid (0 degree and 360 degree roll being the same position), this is not a problem when using at one point any of the filters exclusively. But combining both, this 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 [6]. The quaternion slerp works over the representation of two quaternions on a sphere. The transisiton 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 transistion between the two filters is done by shifting a fixed factor among the α-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 on 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 then the incipient complementary filter, when stopping rotation. Therefore the transition between α-gain of the complementary filter and the Madgwick-gain was chosen to have quadratic behavior. The value referred to as autogain Θ 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 β (cf. equation 2 and 3). The g_k represent the gyro measurement in the corresponding an 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:

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

$$\beta = \max(f_x, f_y, f_z) \tag{3}$$

Then the Madgwick gain γ for a given autogain Θ is calculated by

$$\gamma = \Theta \cdot \beta \tag{4}$$

The complementary filter gain α is calculated by

$$\alpha = \Theta \cdot \theta \cdot (1 - \beta) \tag{5}$$

So the Madgwick gain has a maximum of Θ and α a maximum of $\theta \cdot \Theta$. This ratio θ can be adapted to the specific needs. We used $\theta = 0.1$ because of the 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 θ 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 standard 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 $\mathbf{g} = \begin{bmatrix} 0 & 0 & -1 \end{bmatrix}^T$ represent the gravity vector in the world frame and the matrix $\mathbf{R}$ represent the current orientation of the IMU, then

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

describes the effect of gravity in the coordinate system of the IMU. This gravity vector is then subtracted from the accelerometer measurement. If the there is no translation, the subtraction should lead to $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ 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

$$\begin{bmatrix} v_{x\text{Acc}} \\ v_{y\text{Acc}} \\ v_{z\text{Acc}} \end{bmatrix} = \int_0^T \left(\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} - \begin{bmatrix} g'_x \\ g'_y \\ g'_z \end{bmatrix} dt \right) \tag{7}$$

These Values are now coupled to the rotation of both axes other than their own. Meaning there can be a translation in x if there is rotation around y or z. 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 we let translation in this direction happen. When there is no rotation, it will hold the velocity to 0. During fast movement, this would allow exponential error integration. This exponential error is taken care of later on.

$$\begin{bmatrix} v_{x\text{Acc}} \\ v_{y\text{Acc}} \\ v_{z\text{Acc}} \end{bmatrix} = \begin{bmatrix} v_{x\text{Acc}} \\ v_{y\text{Acc}} \\ v_{z\text{Acc}} \end{bmatrix} \cdot \begin{bmatrix} \max(f_y, f_z) \\ \max(f_x, f_z) \\ \max(f_x, f_y) \end{bmatrix} \tag{8}$$

The rotation into the world frame is computed as:

$$\begin{bmatrix} v_{x\text{AccWorld}} \\ v_{y\text{AccWorld}} \\ v_{z\text{AccWorld}} \end{bmatrix} = \mathbf{R}^T \cdot \begin{bmatrix} v_{x\text{Acc}} \\ v_{y\text{Acc}} \\ v_{z\text{Acc}} \end{bmatrix} \tag{9}$$

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

$$\begin{bmatrix} \omega_{x\text{World}} \\ \omega_{y\text{World}} \\ (\omega_{z\text{World}}) \end{bmatrix} = \mathbf{R}^T \cdot \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} \tag{10}$$

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. $\omega_{z\text{world}}$ needs not to be computed as it has no influence on the position.

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

$$\begin{bmatrix} v_{xz\text{GyroWorld}} \\ v_{yz\text{GyroWorld}} \end{bmatrix} = \begin{bmatrix} \omega_{y\text{World}}/(2\pi) \\ \omega_{x\text{World}}/(2\pi) \end{bmatrix} \cdot 2\pi r = \begin{bmatrix} \omega_{y\text{World}} \\ \omega_{x\text{World}} \end{bmatrix} \cdot r \tag{11}$$

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

Next the $v_{x\text{AccWorld}}$and $v_{y\text{AccWorld}}$ are limited depending on $v_{xz\text{GyroWorld}}$ and $v_{yz\text{GyroWorld}}$, 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 would imply 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, $v_{z\text{AccWorld}}$ is limited to the mean of $v_{xz\text{GyroWorld}}$ and $v_{yz\text{GyroWorld}}$. 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:

$$\begin{bmatrix} v_{x\text{GyroWorld}} \\ v_{y\text{GyroWorld}} \end{bmatrix} = \begin{bmatrix} \sqrt{v_{xz\text{GyroWorld}}^2 - v_{z\text{AccWorld}}^2} \\ \sqrt{v_{yz\text{GyroWorld}}^2 - v_{z\text{AccWorld}}^2} \end{bmatrix} \tag{12}$$

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

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \int_0^T \begin{bmatrix} (1-\beta)v_{x\text{AccWorld}} + \beta \cdot v_{x\text{GyroWorld}} \\ (1-\beta)v_{y\text{AccWorld}} + \beta \cdot v_{y\text{GyroWorld}} \\ v_{z\text{AccWorld}} \end{bmatrix} dt \tag{13}$$

. Depending on the practical circumstances and quality of measurement, the factor β can be 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. $\beta = 0.5$ would result in just the mean of both approaches.

C. *Time Complexity*

Overall an pose estimation step consist only of operations that are constant in time. Given that we use the fast inverse square root algorithm [7] 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 an round acrylic glass plate. For position experiments the plate is put inside a acrylic glass sphere where the plate exactly fits in the middle, as the radii are the same. 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 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 3. Therefore the more interesting 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 string jitter during slow movements and standstill. To evaluate specifically the change between slow motion and standstill, a further experiment was performed. The results in Figure 4 show a short excerpt of the experiment with four phases of slow motion. For external reference, we used a precise optical tracking system. Manually transforming the measured IMU values to the optical reference frame then yields the above plot. The optical system has the same shape as the presented algorithm and lacks the perturbations of the pure Madgwick filter. To stress the same behavior without relying on absolute orientation, which will be part of further research, Figure 5

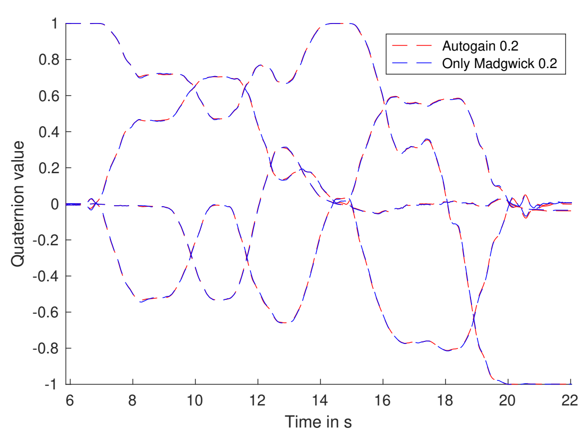

Fig. 3. Quaternion-values of the presented fusion of complementary and Madgwick filter and only the Madgwick filter

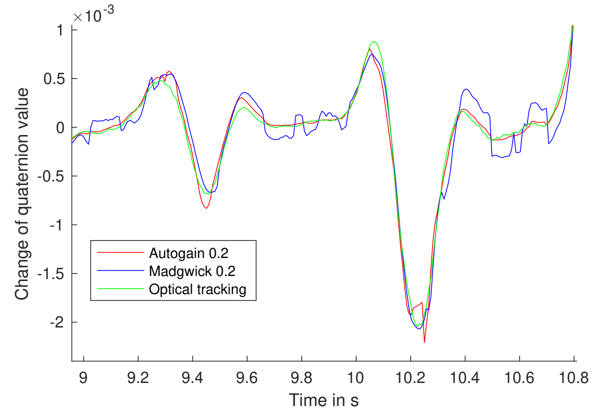

Fig. 5. Derivative of q0-quaternion-value of the presented fusion of complementary and Madgwick filter, only the Madgwick filter and an evaluation via optical tracking

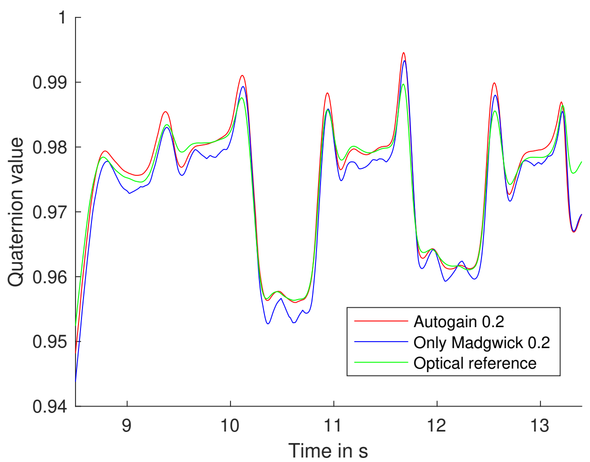

Fig. 4. q0-quaternion-value of the presented fusion of complementary and Madgwick filter, only the Madgwick filter and an evaluation via optical tracking

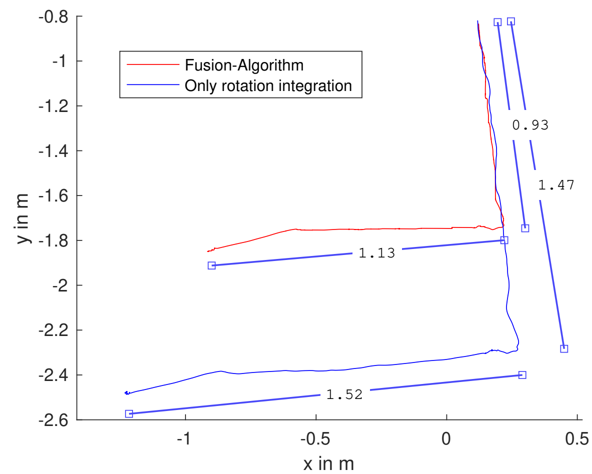

Fig. 6. Ground position by integration of rotation and the fusion algorithm. An L-shape with 1 meter by 1 meter was performed with provoked slip(rotation without translation)

shoes the derivative of all three approaches. 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 position focuses on the 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 6 shows the resulting trajectories. Pure rotation integration leads to an enlarged L by 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. This should be improvable by parameter tuning. 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 would produce. Here the results shown in Figure 7 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, there needs to be a little bit of rotation, as the algorithm does not allow any translation without rotation at all. As the translation computed using the accelerometer is by double integration, 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 7 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

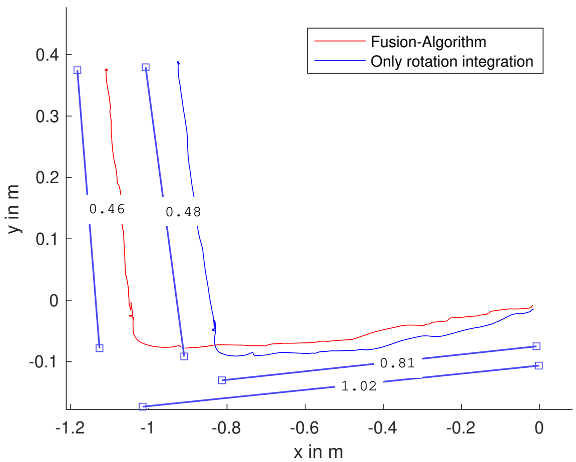

Fig. 7. X-Y-Position by integration of rotation and the fusion algorithm. An L-shape with 1 meter by 1 meter was performed with provoked slide(translateion without rotation)

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. But 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 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. 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, particular 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. The position estimation as well as the combination of the integration by rotation of the gyroscope and the linear acceleration of the accelerometer shows the capability to overcome the problem of the missing absolute reference of IMUs partly, which mainly improves the measurements in the ground plane with slip behavior present. Still, there is a need for parameter tuning. Sliding was not detected reliably all the time. 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] Rossi A.P. et al.. "DAEDALUS - Descent And Exploration in Deep Autonomy of Lava Underground Structures". Wuerzburger Forschungsberichte in Robotik und Telematik (21). 978-3-945459-33-1. (2021).

[2] Madgwick, Sebastian O. H.. "An efficient orientation filter for inertial and inertial / magnetic sensor arrays". (2010).

[3] Mahony, R. ; Hamel, T.; Pflimlin, J.-M. ."Complementary filter design on the special orthogonal group SO(3)", Proceedings of the 44th IEEE Conference on Decision and Control. (2005).

[4] Kok, Manon; Hol, Jeroen D.; Schoen,Thomas B. . "Using Inertial Sensors for Position and Orientation Estimation", Foundations and Trends in Signal Processing: Vol. 11.(2017).

[5] Botero Valencia, J.-S. ; García, M. R. ; Ceballos, J. V. . "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). (2017).

[6] Dam, Erik B.; Koch, Martin; Lillholm, Martin. "Quaternions, Interpolation and Animation". (1998).

[7] "Origin of Quake3's Fast InvSqrt()". beyond3d.com/content/articles/8/. Accessed: 2021-05-01.