2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

On the 3D trochoidal motion model of LiDAR sensors placed off-centered inside spherical mobile mapping systems

Fabian Arzberger[1] and Andreas Nüchter[1]

***Abstract*— We study the motion model of a sensor rigidly mounted inside a ball. Due to the rigid placement inside the ball, the geometry of the sensor trajectory resembles a 3D curate trochoid. A new calibration method for spherical systems estimates the extrinsic parameters of the sensor with respect to the balls center of rotation. We deploy the calibration and motion model on our spherical mobile mapping platform to estimate the trajectory of a LiDAR sensor and compare it to trajectories of state-of-the-art LiDAR-Inertial odometry (LIO) methods. The motion model, which is solely based on IMU measurements, produces comparable results to the LIO methods, sometimes even outperforming them in positional accuracy. Although the LIO methods provide better rotational accuracy due to the utilization of LiDAR data, they struggle to reproduce the trochoidal nature of the trajectory and only provide pose estimations at the LiDAR frequency, whereas the motion model produces a more consistent trochoidal trajectory at the much higher IMU frequency. The results demonstrate the difficulty that current LIO methods have on spherical systems and indicate that our motion model is suitable for overcoming these issues.**

I. Introduction

Spherical systems are still a niche format in robotics for the use of mobile mapping, compared to more prominent systems like unmanned aerial vehicles (UAV) or rovers. The majority of research in that area focuses on the locomotion mechanism to move the ball around [1]–[5]. Often, these approaches use actuators inside the ball, e.g., moving masses or conservation of angular momentum via flywheels, and study the mobility, controllability, and corresponding path planning methods. Others focus more on the mathematical description of the motion of the ball itself [6]–[10]. However, there is a lack of research that focuses on the use of spherical systems for mobile mapping using its internal sensors. Some authors [11]–[13], including the European Space Agency (ESA) [14], have suggested the suitability of the ball-shaped design for exoplanetary exploration and exploration of other inaccessible, dangerous, or harsh terrestrial environments like mine shafts or narrow funnels. The advantages of spherical systems in that context include protection of the internal sensors, an advantage in maneuverability and mass efficiency, and a locomotion principle that leads to sensor coverage without needing additional actuators for the sensor. However, one major disadvantage considering mobile mapping is the large angular velocities and aggressive system dynamics, which introduce degrading effects on inertial measurement units (IMUs), LiDAR sensors, and cameras. In our previous work, we have built spherical mobile mapping prototypes including LiDAR sensors and IMUs, assuming that each sensor would be mounted in the balls center [15]–[17]. In reality, though, the trajectory of a sensor mounted rigidly inside the ball resembles a curate trochoid, which is illustrated in Figure 1. We approach this problem in our paper, having the following contributions:

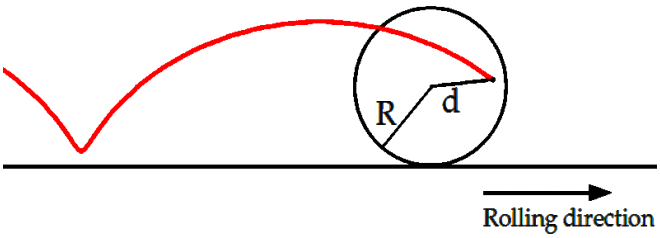


Fig. 1: Illustration of a curate trochoid (red line), which is the trace of a point rigidly mounted inside a rolling sphere with radius R and distance d to the balls center point of rotation, when the ball is rolling without slipping.

- A motion model that estimates the trochoidal 6-DoF trajectory of a sensor mounted rigidly inside the spherical system, based on IMU measurements and an extrinsic calibration of the sensor with respect to the center of the ball.
- A calibration procedure for LiDAR sensors that estimates the extrinsic translation parameters of the sensor with respect to the center of the ball.
- A comparison of the motion model with two state-of-the-art Lidar-Inertial odometry (LIO) methods, illustrating their difficulty to produce trochoidal trajectories.

The rest of the paper is structured as follows: In the next section, we introduce other work related to extrinsic LiDAR calibration and LiDAR-inertial odometry. Then, we introduce our own calibration procedure for spherical mobile mapping systems, followed by the introduction of our 3D trochoidal motion model. Afterwards, we perform experiments with our spherical prototype and discuss the quality of the resulting trochoidal trajectories, followed by conclusions and possible future work.

II. Related work

In the previous section, we have already introduced some spherical systems and their possible applications. Our paper

*We acknowledge funding from the Elite Network Bavaria (ENB) for providing funds for the academic program "Satellite Technology"

[1]The authors are with Computer Science XVII – Robotics, Julius-Maximilians-Universität Würzburg, 97074 Am Hubland, Germany. Contact: fabian.arzberger@uni-wuerzburg.de

focuses on two major topics: 1) Extrinsic LiDAR calibration and 2) LiDAR-inertial odometry (LIO). Note that the motion model presented in our paper only falls into the category of *Inertial* odometry, yet we compare it with other LIO methods.

A. Extrinsic LiDAR calibration

Finding the extrinsic parameters between the coordinate systems of a LiDAR and some other sensor or reference frame is a broad, well studied field. State-of-the-art approaches do not require the user to place artificial external markers in the environment but utilize the features or geometries of the environment directly. The following methods are only a few examples that address the calibration of LiDAR-IMU systems. In [18] the authors introduced an on-site calibration method for LiDAR-IMU systems that combines point, sphere, line, cylinder, and plane features. Their approach employs a full information maximum likelihood estimation to obtain both the LiDAR-to-IMU extrinsics, but also the IMU intrinsic parameters. Another similar approach that estimates both LiDAR-IMU extrinsic and IMU intrinsic parameters is [19], where the authors also utilize point, plane, cone, and cylinder features to construct a geometrically constrained optimization problem, followed by a restricted maximum likelihood estimation. Li et al. present a method that employs "continuous-time trajectory estimation wherein the IMU trajectory is modeled by Gaussian process regression with respect to the independent sampling timestamps" [20]. Furthermore, they account for motion distortion effects of the LiDAR and match the corrected point clouds to a structured plane representation of the environment in an on-manifold batch-optimization framework. In [21] the authors also utilize a continuous-time batch-optimization framework. This approach has been designed for usage in degenerate cases by leveraging observability-aware modules, including an information-theoretic data selection policy and a state update mechanism that updates only the identifiable directions in the state space. Note that in this work, it is specifically required to find the extrinsic translation parameters between the LiDAR and not the IMU, but rather the center point of rotation of the ball, which none of the abovementioned methods provide. These methods are still useful to find the extrinsic rotations. However, as for translation, we have to introduce a new procedure specifically tailored for our use case. Our approach is also marker-less and uses only the geometry of the environment via globally consistent scan-matching and utilizes the different radii that the sensor has when rotating around the center point.

B. LiDAR-inertial odometry

Many state-of-the-art approaches exist to solve the LIO problem, which is especially important for autonomous systems in GPS-denied environments. Often, these methods have been developed and evaluated on cars [22], UAV [23], or other rotationally more restricted systems when compared to a rolling ball. Currently, the most prominent approach to solving the LIO problem is to construct a tightly coupled optimization problem. Some examples include [24], where a factor graph is utilized to solve the optimization problem, or [25], where the authors use an iterated error-state Kalman Filter. Usually, state-of-the-art LIO methods also provide a motion distortion correction algorithm via IMU preintegration. The most popular methods are open-source implementations like [26] for systems experiencing racing velocities, LIO-SAM [27] which utilizes local keyframes and also estimates the IMU bias errors, FAST-LIO [28] which keeps a representation of the global map in an iteratively growing KD-tree, or DLIO [29] which provides a computationally efficient continuous-time coarse-to-fine approach. In this work, we deploy two of the abovementioned state-of-the-art approaches on our spherical mobile mapping system, namely FAST-LIO in its most current version [28], and DLIO [29]. In doing so we test if state-of-the-art methods are able to keep up with the faster than usual rotations and if these methods can reconstruct the trochoidal trajectory of the LiDAR. We expect these methods to perform sub-optimally because the motion state propagation includes the accelerometer readings which, on rolling balls, are mostly governed by centripetal forces degrading their quality.

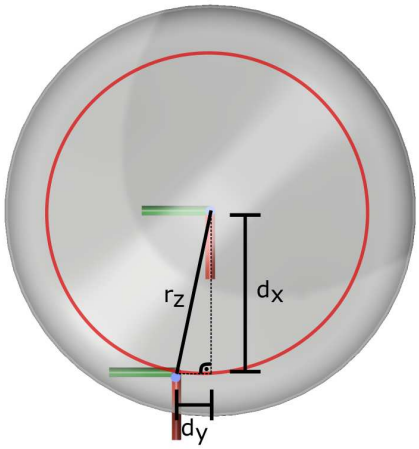


Fig. 2: Calibration principle of a sensor inside a ball, illustrated in one axis. The sensor will make a circular trajectory (red) with radius r_z when rotating without translation around the z-axis. The key to extrinsic parameter calibration is finding the offsets d_y and d_x that define the radius r_z.

III. Extrinsic Calibration

In this section, we introduce a method of finding the extrinsic 3D translation parameters of the LiDAR sensor with respect to the center of the ball. Knowing these offsets in all principal axes is of utmost importance for the trochoidal motion model to work. If the ball rotates precisely around its center point, the sensor mounted rigidly inside the ball follows a circular trajectory. The key idea behind the procedure is to measure the radii of the circular trajectory that the sensor follows when rotating around all three principal axes, and then use these radii to calculate the offsets. This is illustrated for one principal axis in Figure 2, where a rotation

Fig. 3: Left: The spherical mobile mapping system housing a laser-scanner, IMUs, and an onboard computer inside the spherical protective shell. Right: 3D printed calibration apparatus needed to perform rotations of the spherical system without translation around the center point of the ball.

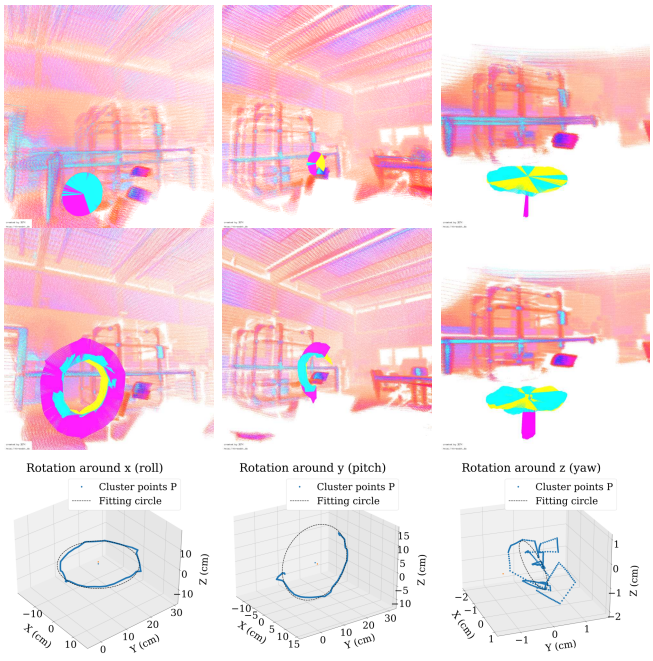


Fig. 4: Calibration results. Each column corresponds to one principal axis. The first row shows the input data, i.e., the unaligned point clouds using the orientation provided by the IMUs. The second row shows the aligned point clouds and refined poses. The third row shows the fitted circles.

around the z-axis results in a radius r_z which depends on the extrinsic offsets in the x and y direction, d_x and d_y respectively. The same is true for rotations around the other principal axes, leading to

$$\begin{aligned} r_x^2 &= d_y^2 + d_z^2 \\ r_y^2 &= d_x^2 + d_z^2 \\ r_z^2 &= d_x^2 + d_y^2 \,, \end{aligned} \tag{1}$$

which we solve directly by expressing the equation system in matrix-form

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} d_x^2 \\ d_y^2 \\ d_z^2 \end{pmatrix} = \begin{pmatrix} r_x^2 \\ r_y^2 \\ r_z^2 \end{pmatrix}$$
$$\Rightarrow \begin{pmatrix} d_x^2 \\ d_y^2 \\ d_z^2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix} \cdot \begin{pmatrix} r_x^2 \\ r_y^2 \\ r_z^2 \end{pmatrix} . \tag{2}$$

We use Equation (2) to calculate the extrinsic offsets from the measured radii, as described in the next subsections.

A. Estimating the radii

In order to measure the radii as precisely as possible, it is necessary to rotate the ball around its center point. We achieve this by using a 3D-printed design that houses three ball bearings, as shown in Figure 3. The ball bearings touch the spherical shell at three positions, such that any rotation of the system will happen around its center point. When rotating the system in that way, we first assume that the LiDAR sensor is placed in the center point and use the data of the IMUs to estimate the orientation. The first row of Figure 4 shows the resulting poses and misaligned point clouds, which is the input to the globally consistent, time-continuous, offline LiDAR-SLAM algorithm "Semi-Rigid Registration" (SRR) [30]. The second row of Figure 4 shows the corresponding output, i.e., the aligned point clouds and circular trajectories. In both rows, each column corresponds to one principal axis of the system. For the purpose of obtaining the radii, we fit circles through the SRR-estimated sensor positions for each axis using the following steps: First, we use singular value decomposition (SVD) to fit a plane through the sensor positions, then project all sensor positions to that plane. Second, now that the sensor positions are projected onto a 2D plane, we fit a circle by the method of least-squares to obtain the radius. Both of these steps are common, well known techniques [31]. The last row of Figure 4 shows the fitted circles.

B. Obtaining the offsets

Note that often it is not possible to direclty use the radii from the previous subsection in Equation (2). Insisting that the offsets must not be imaginary yields the constraints

$$\begin{aligned} 0 < -r_x^2 + r_y^2 + r_z^2 \\ 0 < \;\; r_x^2 - r_y^2 + r_z^2 \\ 0 < \;\; r_x^2 + r_y^2 - r_z^2 \,, \end{aligned} \tag{3}$$

which are prone to be violated in the presence of measurement noise from the LiDAR or residual registration errors. Thus, we use the sums of the squared residuals S_i from the least-squares circle fitting method to calculate a 95% confidence interval for the radii:

$$r_i \pm z_{0.975} \frac{S_i}{n-1} \frac{1}{\sqrt{n}} \,, \tag{4}$$

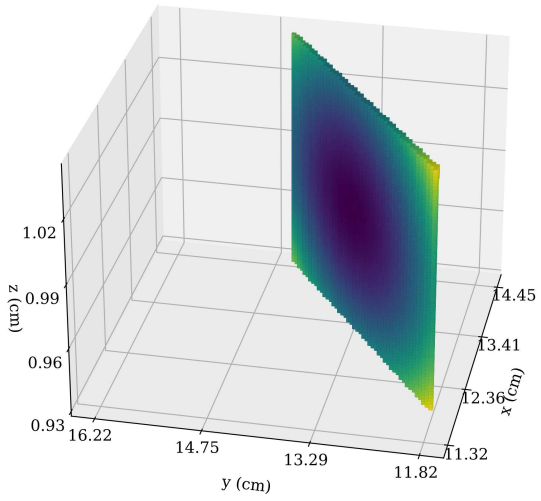


Fig. 5: The calibration space corresponding to the estimated radii and their confidence intervals. Each voxel represents a solution that contains only non-imaginary offsets.

where n is the number of measurments taken with the LiDAR and $z_{0.975}$ is the value of the 97.5% quantile of the unit normal distribution needed for a double-sided 95% confidence test. We check the whole calibration space spanned by the confidence intervals in a greedy fashion for non-imaginary solutions, picking the one that is closest to the estimated radii r_i. Figure 5 shows the calibration space where a voxel is plotted if a solution only has non-imaginary extrinsic offsets. As illustrated by the color of the voxels, which represent the distance to the originally estimated radii, we pick the closest one corresponding to the extrinsic offsets

$$\mathbf{d} = \begin{pmatrix} \pm 0.972401 \\ \pm 0.0639203 \\ \pm 13.2604 \end{pmatrix} \text{cm} \, .$$

Note that both negative and positive directions are possible due to the squares in Equation (2), thus it is up to the user to inspect the axes definitions of their system and choose the correct sign.

IV. Trochoidal Motion Model

In this section we introduce the 3D trochoidal motion model for the LiDAR, using the previously calculated extrinsic offsets to the center of the ball. Figure 6 introduces the notation used in this section. The superscript $\cdot^r$ denotes a vector with respect to the local coordinate system of the ball with its origin at the center point, e.g., $\boldsymbol{\omega}^r$ is the angular velocity vector as given by the gyroscope. We deduce the extrinsic parameters of the LiDAR with respect to the local frame $\mathbf{s}^r = (0.972401, 0.0639203, -13.2604)^\tau$, using $\mathbf{d}$ obtained from the previous section. Furthermore, the normal vector of the floor in the global frame, $\mathbf{n}$, must be available, for example via a flat-floor assumption or measurement via the LiDAR sensor. In this work, we use a flat-floor assumption but plan on measuring the normal vector using the LiDAR sensor in future work. We denote the current orientation of the ball with $\mathbf{R}$ which is available from the onboard IMUs. Using this orientation we express the angular velocity in the global frame:

$$\boldsymbol{\omega} = \mathbf{R}^{-1} \cdot \boldsymbol{\omega}^r \, . \tag{5}$$

The linear velocity of the balls center over ground must be

$$\mathbf{v} = r_s \boldsymbol{\omega} \times \mathbf{n} \, , \tag{6}$$

where r_s is the radius of the ball. The position of the balls center $\mathbf{p}$ is the integral over the linear velocity

$$\mathbf{p} = \int \mathbf{v} \, , \tag{7}$$

whereas the position of the LiDAR $\mathbf{s}$ is the sum of $\mathbf{p}$ and the extrinsic parameters expressed in the global frame

$$\mathbf{s} = \mathbf{R}^{-1} \cdot \mathbf{s}^r + \mathbf{p} \, . \tag{8}$$

Thus, the combined motion model using Equations (5)-(8) is

$$\mathbf{s} = \mathbf{R}^{-1} \cdot \mathbf{s}^r + \int \left(\left[\mathbf{R}^{-1} \cdot r_s \cdot \boldsymbol{\omega}^r \right] \times \mathbf{n} \right) \, , \tag{9}$$

which expands, not ommiting the time dependence, to

$$\mathbf{s}(t) = \mathbf{R}(t)^{-1} \mathbf{s}^r + \int_0^t \left(\left[\mathbf{R}(\tau)^{-1} r_s \boldsymbol{\omega}^r(\tau) \right] \times \mathbf{n}(\tau) \right) d\tau \, . \tag{10}$$

V. Experiments and Evaluation

In this section, we deploy our motion model from Equation (10) on our spherical mobile mapping system, shown in the left image of Figure 3. It is equipped with a "Hesai PandarXT32" LiDAR sensor and three "Phidgets Spatial 3/3/3 1044b" IMUs which are mounted rigidly inside the spherical shell of the system. A "BMAX B2ro" Mini PC running Ubuntu Linux on an "Intel Celeron N4120" 4-core CPU (2.6 GHz) serves as the onboard processing unit. We currently roll the spherical system manually to record our datasets, yet we plan on including a locomotion mechanism, e.g., the ones mentioned in Section I, with future prototypes. In this work, we compare the trochoidal motion model with two LIO methods: FAST-LIO [28] and DLIO [29], by utilizing an offline-batch continuous-time globally consistent SLAM algorithm, SRR [30], to provide reference trajectories. Figure 7 shows that we use the output of the motion model directly as an initial guess for SRR, which outputs a globally consistent map and trajectory. The latter serves as a reference to compare against the trajectories of the other three estimators. Figure 9 shows the position components for all 4 trajectories. Qualitatively speaking, the motion model has notable drift in the yaw axis, which is due to the inability of the IMUs to provide reliable yaw estimations without the use of their magnetometer. We forbid the use of the magnetometer by design because the prototype has been designed in an exoplanetary exploration context [32]. Notably, the motion model provides the most consistent estimate in the z-axis, where most of the trochoidal

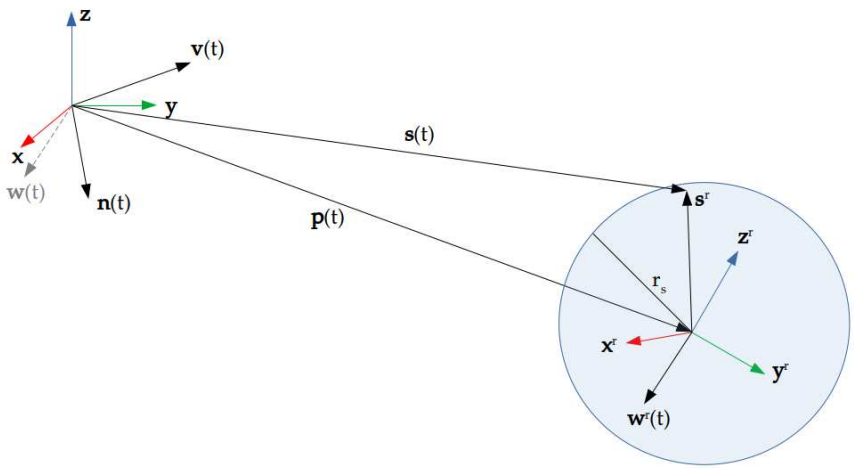


Fig. 6: Schematics and notation of the motion model. The x^r, y^r, z^r frame describes the local coordinate system of the center point of the ball. The x, y, z frame is the fixed global coordinate system in which the trochoidal motion of the sensor is described by the vector $s(t)$.

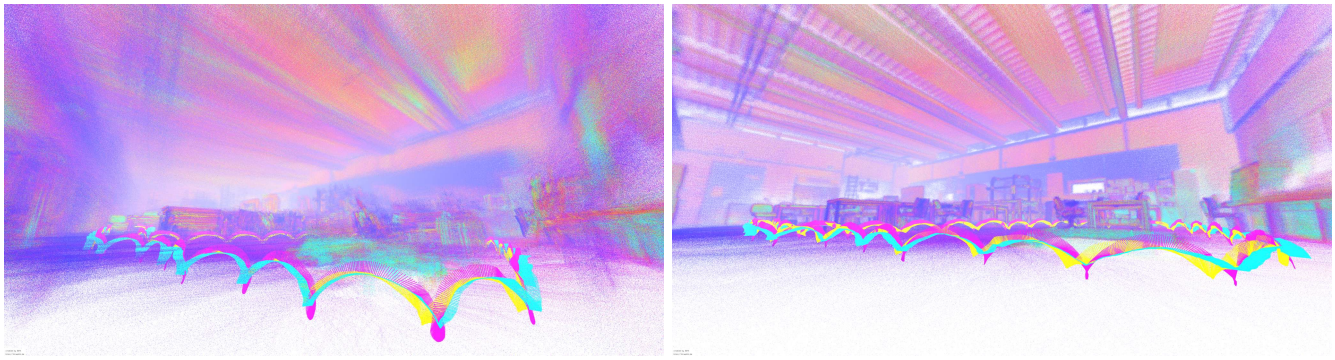


Fig. 7: Left: The trajectory is the output of the trochoidal motion model, which is applied to the LiDAR data. Integration errors are present especially in the yaw axis leading to drift. Right: The LiDAR and trajectory data from the left image is used as input for SRR [30], which results in a globally consistent point cloud and trajectory. The color of the points denote intensity of the laser return.

motion is happening. Figure 10 shows also the resulting point clouds when applying the trajectories to the LiDAR data in a zoomed sliced birds-eye view. Again, the drift of the motion model, especially in the yaw axis, is apparent. FAST-LIO has issues with scan matching leading to ghosting in the point cloud. DLIO provides the best point cloud, although some residual error is still present, considering the misalignment of objects like chairs.

A. Error metrics

We have considered recording ground truth trajectory data with an external tracking system based on infrared markers, as we did in previous work [17]. However, we have deliberately chosen not to do that in this paper due to tracking problems caused by marker reflections off the spherical shell and loss of vision on them when rolling. To evaluate the accuracy of the estimated trajectories quantitatively, we compare them against the reference trajectory obtained by SRR. We use the "evo" odometry evaluation tool from Grupp [33] to calculate the full absolute pose error (APE) of the trajectories, which splits into two parts: position and rotation error.

- **Position error:**
  The absolute pose error with respect to position in meters ($\mathrm{APE}\ [\mathrm{m}]$) represents the unsigned error for each pose between the estimated position $\mathbf{p}_{\mathrm{est},i}$ and refernce position $\mathbf{p}_{\mathrm{ref},i}$

$$\left|\mathbf{p}_{\mathrm{est},i} - \mathbf{p}_{\mathrm{ref},i}\right| . \tag{11}$$

- **Rotation error:**
  The abolute pose error with respect to rotation in degrees ($\mathrm{APE}\ [^\circ]$) represents the error for each pose between the orientation estimation $\mathbf{R}_{\mathrm{est},i}$ and the orientation reference $\mathbf{R}_{\mathrm{est},i}$

$$\left|\angle\left(\mathbf{R}_{\mathrm{est},i}^{-1}\mathbf{R}_{\mathrm{ref},i}\right)\right| . \tag{12}$$

B. Results

Metrics for both individual components of the APE are listed in Table I, whereas the full APE for each trajectory is plotted in Figure 8. The results indicate that the motion model is outperformed by DLIO and FAST-LIO, especially for the rotation part, which is unsurprising since these methods use additional LiDAR data instead of only magnetometer-denied IMU data. However, the metrics are in the same order of magnitude and the motion model seems to perform better than FAST-LIO regarding the positional error for some metrics. We make that statement cautiously, though, since the motion model alone has no way of recovering from the accumulated drift in longer trajectories. It is for this reason, that we plan to implement the motion model into a LIO method specifically tailored towards spherical systems. Our results show that both FAST-LIO and DLIO do perform sub-optimally in this context, and also that our motion model has the potential to aid such methods to reach their full capabilities which have been demonstrated in their respective papers [28], [29]. Furthermore, keep in mind that the LIO methods only provide their pose estimates at the LiDAR frequency, 20 Hz in our case, whereas the motion model provides them at the higher IMU frequency which is 125 Hz.

VI. Conclusions

In this work, we have addressed the prediction of the trochoidal motion that a LiDAR sensor experiences when mounted rigidly inside a spherical mobile mapping system. Our new calibration procedure, which is specifically designed for this use case, estimates the extrinsic LiDAR-to-center-point parameters needed to construct the trochoidal trajectory. We evaluated the predicted motion model trajectory and compared it to the trajectories that state-of-the-art LIO methods produce. The results show that the motion model trajectory better resembles the trochoidal geometry than the state-of-the-art approaches, which have no information on the spherical nature of the system. Nevertheless, a lot of work remains to be done. We need to measure the ground normal vector using the LiDAR data to account for rolling on slopes. Additionally, since our motion model only utilizes IMU data it is still prone to drift, thus we do not plan on using it as is. Furthermore, the state-of-the-art LIO approaches perform sub-optimally due to the motion profile of the rolling ball, which they were not designed for. Thus, in future work, we plan on utilizing our motion model to implement a LIO method that is suited for this context. Another solution is to modify existing LIO methods, e.g., bootstrap them with our motion model or modify their state propagation mechanisms. Our results indicate that DLIO [29] is a promising approach for such modifications due to the overall better performance.

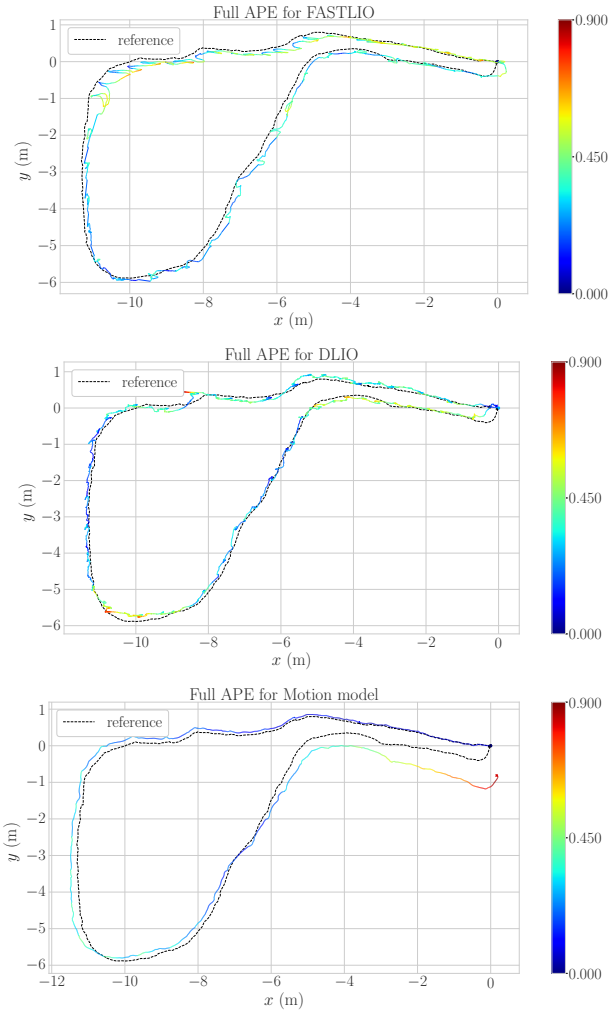


Fig. 8: Comparison of the estimated trajectories. The color denotes the combined absolute pose error (unit-less). Our motion model experiences drift, the LIO methods inconsistently jump.

References

[1] W.-H. Chen, C.-P. Chen, J.-S. Tsai, J. Yang, and P.-C. Lin, "Design and implementation of a ball-driven omnidirectional spherical robot," *Mechanism and Machine Theory*, vol. 68, pp. 35–48, 2013.

[2] T. Ylikorpi and J. Suomela, *Ball-Shaped Robots*. 2007.

[3] R. Chase and A. Pandya, "A review of active mechanical driving principles of spherical robots," *Robotics*, vol. 1, pp. 3–23, 2012.

[4] V. Joshi, R. Banavar, and R. Hippalgaonkar, "Design and analysis of a spherical mobile robot," *Mechanism and Machine Theory*, vol. 45, pp. 130–136, 2010.

[5] M. Anwar, L. H. Omar, Y. M. Ekhsan, R. Ramli, N. Nadzri, F. Ahmad, and M. H. Shafii, "Conceptual of spherical robot," in *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pp. 547–550, 2014.

[6] F. Hogan and J. Forbes, "Modeling of spherical robots rolling on generic surfaces," *Multibody System Dynamics*, vol. 35, pp. 91–109, 2015.

[7] I. Mamaev and E. V. Vetchanin, "Dynamics of a spherical robot with periodically changing moments of inertia," *2020 International Conference Nonlinearity, Information and Robotics (NIR)*, pp. 1–5, 2020.

[8] M. R. Burkhardt and J. Burdick, "Reduced dynamical equations for barycentric spherical robots," *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2725–2732, 2016.

[9] O. K. Vanjpe, M. Narasimhappa, and A. D. Mahindrakar, "Global attitude estimation and dead reckoning of a mobile spherical robot using extended kalman filter," in *Proceedings of the 2019 4th International Conference on Advances in Robotics*, AIR '19, (New York, NY, USA), Association for Computing Machinery, 2020.

[10] S. Fiori, "Lie-group modeling and simulation of a spherical robot,

TABLE I: Comparison of statistical metrics for the positional and rotational components of the absolute pose error (APE) for all compared estimators. The best value is printed in **bold**, the second best in *italics* numbers where lower is better.

Estimator	Median		Mean		Std.		RMSE		Max.	
	APE [m]	APE [°]	APE [m]	APE [°]	APE [m]	APE [°]	APE [m]	APE [°]	APE [m]	APE [°]
DLIO	**0.230**	**4.345**	**0.230**	*4.456*	**0.085**	*1.791*	**0.245**	*4.637*	**0.579**	10.675
FASTLIO	0.413	*4.356*	0.428	**4.436**	*0.116*	**1.247**	0.444	**4.487**	*0.758*	**8.922**
Motion model	*0.246*	4.748	*0.272*	4.457	0.179	1.940	*0.326*	4.861	0.877	*8.968*

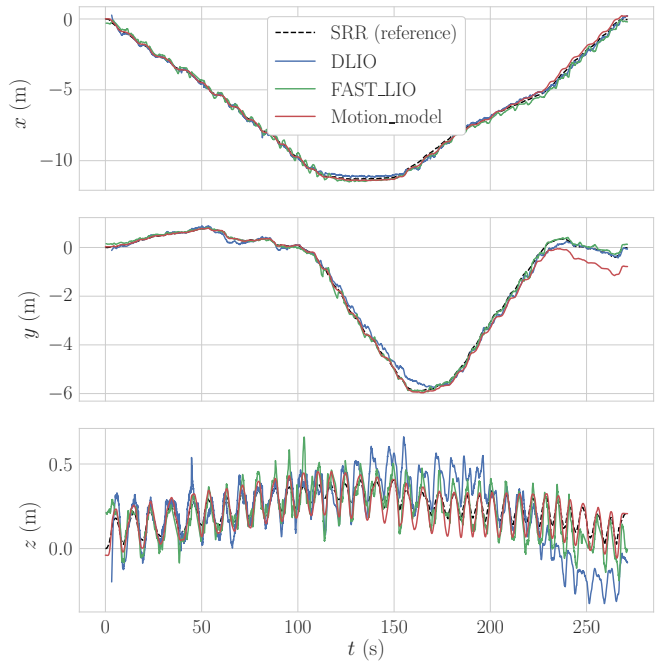


Fig. 9: Component-wise comparison of the estimated trajectories. The trochoidal nature of the LiDAR trajectory is best represented by the motion model.

actuated by a yoke-pendulum system, rolling over a flat surface without slipping," *Robotics and Autonomous Systems*, p. 104660, 2024.

[11] M. Bujňák, R. Pirník, K. Rástočný, A. Janota, D. Nemec, P. Kuchár, T. Tichý, and Z. Łukasik, "Spherical robots for special purposes: A review on current possibilities," *Sensors*, vol. 22, no. 4, 2022.

[12] M. Li, H. Sun, L. Ma, P. Gao, D. Huo, Z. Wang, and P. Sun, "Special spherical mobile robot for planetary surface exploration: A review," *International Journal of Advanced Robotic Systems*, vol. 20, 2023.

[13] F. Bruhn, H. Kratz, J. Warell, C. Lagerkvist, V. Kaznov, J. Jones, and L. Stenmark, "A preliminary design for a spherical inflatable microrover for planetary exploration," *Acta Astronautica*, vol. 63, pp. 618–631, 2008.

[14] European Space Agency, "Daedalus – descent and exploration in deep autonomy of lava underground structures." `https://nebula.esa.int/content/daedalus-%E2%80%93-descent-and-exploration-deep-autonomy-lava-underground-structures`. Accessed: 2024-03-14.

[15] J. Zevering, A. Bredenbeck, F. Arzberger, D. Borrmann, and A. Nuechter, "Imu-based pose-estimation for spherical robots with limited resources," in *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 1–8, 2021.

[16] F. Arzberger, A. Bredenbeck, J. Zevering, D. Borrmann, and

[17] F. Arzberger, F. Wiecha, J. Zevering, J. Rothe, D. Borrmann, S. Montenegro, and A. Nüchter, "Delta filter - robust visual-inertial pose estimation in real-time: A multi-trajectory filter on a spherical mobile

A. Nüchter, "Towards spherical robots for mobile mapping in human made environments," *ISPRS Open Journal of Photogrammetry and Remote Sensing*, vol. 1, p. 100004, 2021.
mapping system," in *2023 European Conference on Mobile Robots (ECMR)*, pp. 1–8, 2023.

[18] W. Liu, Z. Li, R. Malekian, M. Sotelo, Z. Ma, and W. Li, "A novel multifeature based on-site calibration method for lidar-imu system," *IEEE Transactions on Industrial Electronics*, vol. 67, pp. 9851–9861, 2020.

[19] W. Liu and Y. Li, "Error modeling and extrinsic-intrinsic calibration for lidar-imu system based on cone-cylinder features," *Robotics Auton. Syst.*, vol. 114, pp. 124–133, 2019.

[20] S. Li, L. Wang, J. Li, B. Wang, L. Chen, and G. Li, "3d lidar/imu calibration based on continuous-time trajectory estimation in structured environments," *IEEE Access*, vol. 9, pp. 138803–138816, 2021.

[21] J. Lv, X. Zuo, K. Hu, J. Xu, G. Huang, and Y. Liu, "Observability-aware intrinsic and extrinsic calibration of lidar-imu systems," *IEEE Transactions on Robotics*, vol. 38, pp. 3734–3753, 2022.

[22] S. Zhao, Z. Fang, H. Li, and S. Scherer, "A robust laser-inertial odometry and mapping method for large-scale highway environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1285–1292, 2019.

[23] J.-C. Yang, C.-J. Lin, B.-Y. You, Y.-L. Yan, and T.-H. Cheng, "Rtlio: Real-time lidar-inertial odometry and mapping for uavs," *Sensors*, vol. 21, no. 12, 2021.

[24] Z. Wang, L. Zhang, Y. Shen, and Y. Zhou, "D-liom: Tightly-coupled direct lidar-inertial odometry and mapping," *IEEE Transactions on Multimedia*, vol. 25, pp. 3905–3920, 2023.

[25] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "Lins: A lidar-inertial state estimator for robust and efficient navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8899–8906, 2020.

[26] A. Huguet Segarra *et al.*, "Limo-velo: A real-time, robust, centimeter-accurate estimator for vehicle localization and mapping under racing velocities," B.S. thesis, Universitat Politècnica de Catalunya, 2022.

[27] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5135–5142, 2020.

[28] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

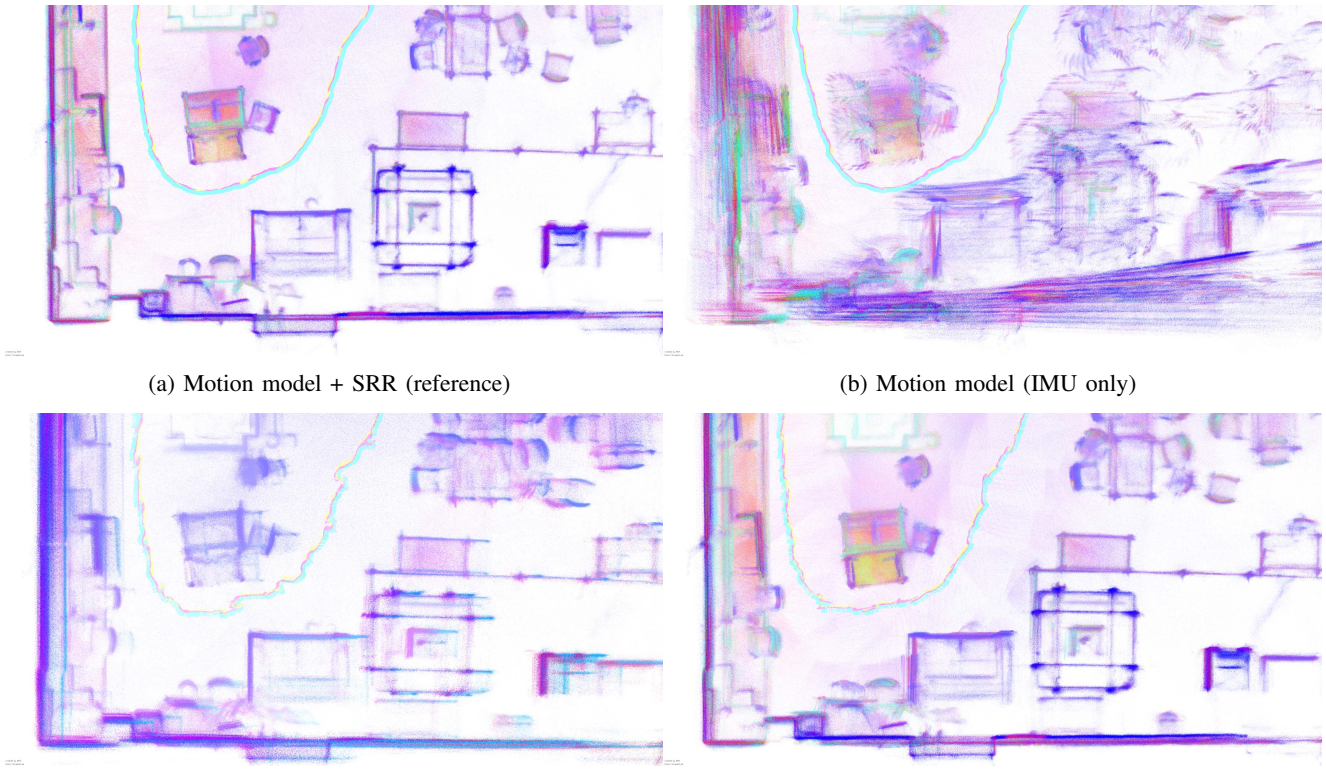
[29] K. Chen, R. Nemiroff, and B. T. Lopez, "Direct lidar-inertial odometry: Lightweight lio with continuous-time motion correction," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3983–3989, 2023.

[30] J. Elseberg, D. Borrmann, and A. Nüchter, "6dof semi-rigid slam for mobile scanning," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1865–1870, 2012.

[31] S. J. Ahn, W. Rauh, and M. Recknagel, "Geometric fitting of line, plane, circle, sphere, and ellipse," 01 1999.

[32] A. P. Rossi, F. Maurelli, V. Unnithan, H. Dreger, K. Mathewos, N. Pradhan, D.-A. Corbeanu, R. Pozzobon, M. Massironi, S. Ferrari, C. Pernechele, L. Paoletti, E. Simioni, P. Maurizio, T. Santagata, D. Borrmann, A. Nüchter, A. Bredenbeck, J. Zevering, F. Arzberger, and C. A. R. Mantilla, "Daedalus - descent and exploration in deep autonomy of lava underground structures," Tech. Rep. 21, Institut für Informatik, 2021.

[33] M. Grupp, "evo: Python package for the evaluation of odometry and slam.." `https://github.com/MichaelGrupp/evo`, 2017.



(a) Motion model + SRR (reference)

(b) Motion model (IMU only)

(c) FAST_LIO

(d) DLIO

Fig. 10: Comparison of the estimated trajectories and resulting point clouds in a zoomed, sliced birds-eye view. The color of the points denote the intensity of the LiDAR return signal. Only in the reference point cloud every object is aligned. The motion model drifts, FAST-LIO has ghosting, and DLIO has misaligned objects.