2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Mobile Robot Localization: a Modular, Odometry-Improving Approach

Luca Mozzarelli[1], Luca Cattaneo[1], Matteo Corno[1] and Sergio Matteo Savaresi[1]

Abstract— **Despite the number of works published in recent years, vehicle localization remains an open, challenging problem. While map-based localization and SLAM algorithms are getting better and better, they remain a single point of failure in typical localization pipelines. This paper proposes a modular localization architecture that fuses sensor measurements with the outputs of off-the-shelf localization algorithms. The fusion filter estimates model uncertainties to improve odometry in case absolute pose measurements are lost entirely. The architecture is validated experimentally on a real robot navigating autonomously proving a reduction of the position error of more than 90% with respect to the odometrical estimate without uncertainty estimation in a two-minute navigation period without position measurements.**

I. INTRODUCTION

As more and more autonomous mobile robots (AMRs) are getting deployed “into the wild”, the need for robust, reliable, and always-available localization is increasing. This is especially true when focusing on the urban environments, where last-mile delivery robots are becoming an interesting option for many businesses, while the localization problem is exacerbated by the challenging context. GNSS-based localization with Real Time Kinematics (RTK) corrections has been successfully employed in some application scenarios like agricultural fields, but its quality in urban areas is uneven, with vast areas where the signal has low quality due to multi-path reflections or is completely unavailable (indoor or covered areas and underpasses). For this reason, a lot of effort is directed towards the development of map-based localization algorithms [1], [2], that compare live sensor data (typically LiDARs or cameras) to an a-priori known map of the environment to estimate the vehicle location. While these provide invaluable data in some settings, they are not a silver bullet and previous works have shown how they may struggle in some scenarios like vast open areas and repetitive environments [3]. To mitigate these issues sensor fusion has been explored since the early days of robotics. Although some modern approaches integrate more sensors, like GNSS receivers, in a graph Simultaneous Localization and Mapping (SLAM) framework [4], these tend to be computationally complex and rarely consider the peculiarities of the vehicle on which the algorithm is deployed, resulting in high sensitivity to sensor calibrations and completely unrealistic motion estimates during failures (sudden jumps, lateral translations for holonomic vehicles or vertical movement for robots working in a planar environment, for example). Indeed, most of these SLAM systems aim to be off-the-shelf algorithms compatible with a vast variety of vehicles and setups, assuming unconstrained rigid body dynamics. Adapting them to exploit the motion constraints imposed by a specific vehicle can be a resource-intensive hurdle, even though the benefits could be noteworthy. More classical approaches employ Extended Kalman Filters (EKFs), as it is a lightweight and versatile sensor fusion algorithm: common examples fuse proprioceptive sensors (like IMUs and wheel encoders) with GNSS measurements [5], [6]. While approaches fusing a vast variety of sensors have been developed over the years [7], [8], GNSS receivers have long remained the only source of absolute localization data, with few application and environment-specific exceptions [9].

This paper proposes a modular, two-layered localization architecture that can incorporate raw sensor measurements, as well as the output of off-the-shelf localization algorithms. This enables the exploitation of complex, vehicle-agnostic algorithms in a localization system that is aware of the vehicle’s peculiarities, with relevant benefits in the reliability of odometrical estimates. Indeed, while in [9] little thought is given to the availability of the position measurements, in an urban area position sources might unexpectedly become unavailable. Such losses would force to robot to halt, even if continuing along the trajectory would allow it to re-localize (think of an underpass that blocks GNSS reception: reaching the end would allow the reception of new measures). While fusing two position sources definitely increases the robustness of the localization solution, the loss of both is not uncommon. In such conditions, having short or medium-term resilience to complete measurement loss can significantly increase the robot’s autonomous capabilities and reduce the need for human intervention. Since proprioceptive sensors are highly reliable, continuing to move trusting an odometrical estimate is an option but, without the necessary precautions, it is often too risky: the drift rates of odometry can be unpredictable and highly affected by model parameters like wheel radii. Our approach improves the odometrical estimation by estimating model uncertainties online, resulting in relevant improvements in positional errors.

To recap, the main contribution of this paper is the development of a localization architecture that merges the benefits of modern localization algorithms and more classical EKF-based state estimation approaches. The proposed architecture is:

- modular and capable of incorporating the output of complex off-the-shelf algorithms, guaranteeing competitive

[1] The authors are with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milano, Italy. Email: {luca.mozzarelli, luca1.cattaneo, matteo.corno, sergio.savaresi}@polimi.it

The experimental data presented in this paper was collected during the project “BUDD-e: Blind-assistive aUtonomous Droid Device”, which is financially supported by Politecnico di Milano, Polisocial Award 2021.

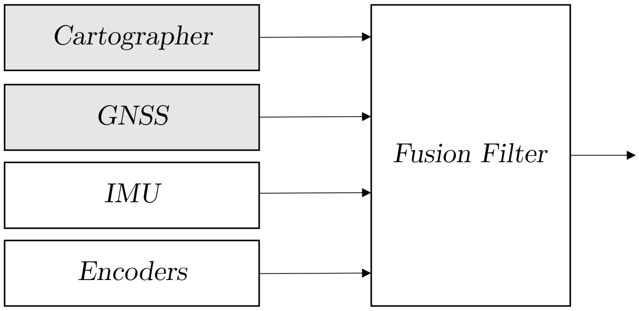


Fig. 1: Block scheme of the proposed localization architecture.

performance in GNSS-denied environments;
- aware of vehicle-specific motion models, as the EKF can be tailored to a specific vehicle;
- capable of improving the odometrical data by estimating model uncertainties;

The architecture is also capable of handling asynchronous, multi-rate, and unavailable measurements by adapting the estimated state based on time-varying observability conditions and can be easily ported to different vehicles and setups. The architecture is developed an deployed experimentally on an autonomous mobile robot.

The remainder of this paper is structured as follows. Section II discusses the core principles of the proposed localization architecture. Next, Section III will present the robotic platform used to develop and validate the proposed algorithm. The implementation of the architecture for the specific experimental platform is detailed in IV and the experimental results are reported in Section V.

II. LOCALIZATION ARCHITECTURE

The proposed localization architecture comprises two layers, exemplified in Figure 1. The first layer, on the left, is constituted by the localization sources, while the second layer performs the sensor fusion operations. Localization sources could be physical sensors, like GNSS receivers, encoders, IMUs and magnetometers or other algorithms that provide useful localization data, like map-based localization algorithms or LiDAR odometries. In general, three categories can be identified: absolute source, displacement sources and proprioceptive sensors. Absolute sources provide the robot's position with respect to a fixed reference frame. At least one absolute source is required to obtain an absoulute position estimate. Displacement sources include odometries based on exteroceptive sensors, like LiDAR or visual odometries. Proprioceptive sensors measure the internal state of the vehicle, like IMUs and encoders. It is worth noting that, when multiple absolute position sources are considered, each might provide data in a different reference frame. For the fusion to work correctly, the roto-translation between such references is estimated by aligning the raw sensor data from the mapping session using the closed form minimization of the point-to-point mean squared error as in [10].

The second layer of the proposed localization architecture consists of a filter based on the Extended Kalman Filter framework. Under this extremely popular framework, a model is employed to predict the evolution of the state between iterations and the prediction is subsequently corrected by integrating measures. The applied correction is proportional to the error between measured values and measurement predictions, obtained through an observation model from the predicted state. In the proposed approach, and contrary to [9], we advocate for vehicle-specific models. This enables the use of formulations that are better tailored: the more accurate the state transition model is the more robust the localization will be to failures in the absolute localization error sources.

Note that the filter must be capable of handling a varying number of measurements, which is not a given in the EKF paradigm. A common method to bypass this issue to handle multi-rate measurements is to design the filter to run at the frequency of the slowest sensor. This solution is clearly suboptimal and might not meet the output frequency requirements. Furthermore, some of the inputs might become unavailable for prolonged periods. For these reasons, we approach this issue differently: selecting the filter execution frequency based on the fastest sensor. At each iteration, we perform the prediction step and the correction step, but the latter considers only the available measurements. This is achieved by dynamically adjusting the dimensions of the involved matrices.

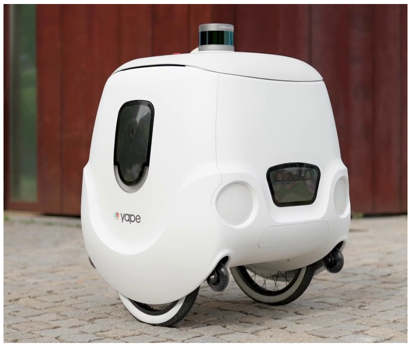


Fig. 2: Yape, the deployment target of the developed localization algorithm.

III. EXPERIMENTAL SETUP

The localization architecture presented in this paper was developed and tested on Yape, an autonomous ground drone designed for last-mile delivery ([11], [12], [13]). Yape (Figure 2) is a Two-Wheeled Inverted Pendulum (TWIP) robot, which independently controls the two wheels to achieve motion while keeping the chassis balanced. The robot is equipped with a sensor package comprising

- an Ardusimple simpleRTK2B Lite RTK GNSS receiver;
- a Robosense R16 16-layer mechanical LiDAR;
- a chassis-mounted e-shock DC_Perception-Lean: a 6 degrees of freedom Inertial Measurement Unit (IMU);
- two incremental wheel encoders with a resolution of 0.06°

The IMU measures linear accelerations and rotational velocities along the three axes, while wheel encoders provide wheel positions and rotational speeds. The presented algorithm was implemented running online on the onboard processing unit, an Nvidia Xavier AGX development board, by employing the ROS framework [14].

IV. ARCHITECTURE SPECIALIZATION FOR YAPE

This Section details how the architecture principles outlined in Section II translate in a practical implementation on the Yape AMR.

A. Localization sources

In this work, we fuse data from two types of sources: absolute pose sources and proprioceptive sources, leaving the integration of displacement sources to future develoments. To achieve absolute localization, the GNSS receiver provides position measures at 10 Hz. GNSS data achieves centimeter-level accuracy when a clear view of the sky and an internet connection to receive RTK corrections are available. Furthermore, a LiDAR, map-based localization algorithm was deployed to provide absolute pose data indoors and under poor GNSS coverage. Cartographer [2] was selected to cover this role after a detailed analysis of its performance, presented in [3]. The Cartographer algorithm is employed in pure localization mode: the localization map is considered available after a mapping experiment. In said experiment, the robot is teleoperated on the area that will be autonomously traversed while acquiring raw sensor data. The dataset is then fed to the Cartographer algorithm in SLAM mode, which builds and optimizes the map. Note that GNSS and Cartographer data are referred to different reference frames: to estimate the roto-translation between the two, the mapping phase is exploited, applying the alignment method described in Section II to the optimized SLAM trajectory and the GNSS one. Cartographer pose data is provided at a frequency of 20 Hz. Regarding proprioceptive sources, we fuse the wheel speeds measured by the encoders, as well as the yaw rate measurement from the IMU. Both sensors sample data at 100 Hz.

B. State transition model

To show the potential of implementing vehicle-specific models we developed an uncertain kinematic differential model. The estimated state x is

$$x_k = \begin{bmatrix} X_k & Y_k & \psi_k & \omega_k^R & \omega_k^L & R_k^R & R_k^L & b_k \end{bmatrix}^T \quad (1)$$

where X, Y and ψ represent the pose of the robot, ω^R and ω^L the wheels rotational speeds while R^R, R^L and b the wheel radii and IMU yaw rate bias. Subscript k indicates that the variables are referred to time instant k. The last three states represent uncertain model parameters, which strongly affect the odometrical performance of the localization solution. A sensitivity analysis of this phenomenon is reported in Figure 3. Notice how even small inaccuracies can have huge effects on the estimated trajectory.

We make the assumption of low wheel speeds, radii and IMU bias dynamics, which allows us to write the state transition function as

$$x_{k+1} = f(x_k, u_k) = \begin{bmatrix} X_k + \frac{\omega_k^R R_k^R + \omega_k^L R_k^L}{2} \cos(\psi_k) T_s \\ Y_k + \frac{\omega_k^R R_k^R + \omega_k^L R_k^L}{2} \sin(\psi_k) T_s \\ \psi_k + \frac{\omega_k^R R_k^R - \omega_k^L R_k^L}{T} T_s \\ \omega_k^R \\ \omega_k^L \\ R_k^R \\ R_k^L \\ b_k \end{bmatrix} \quad (2)$$

where T_s is the sampling time and T the vehicle's track width. Note that in our formulation $u_k = \emptyset$: we include all sensor data as outputs.

C. Measurement model

Since Cartographer and encoder measurements map directly to state variables the predicted measures $\hat{y}$ will be

$$\hat{y}_k^{enc} = \begin{bmatrix} \omega_k^R \\ \omega_k^L \end{bmatrix} \quad (3)$$

and

$$\hat{y}_k^{carto} = \begin{bmatrix} X_k \\ Y_k \\ \psi_k \end{bmatrix}. \quad (4)$$

The measurement model of the GNSS receiver is similar to (4) but lacks a heading measurement and needs to take into account the displacement of the antenna mounting position with respect to the vehicle's center of mass, expressed here in polar coordinates (d, α):

$$\hat{y}_k^{gnss} = \begin{bmatrix} X_k + d\cos(\psi_k + \alpha) \\ Y_k + d\sin(\psi_k + \alpha) \end{bmatrix}. \quad (5)$$

Finally, the IMU yaw rate measurement prediction needs to be computed from both wheel speeds, both radii as well as the bias:

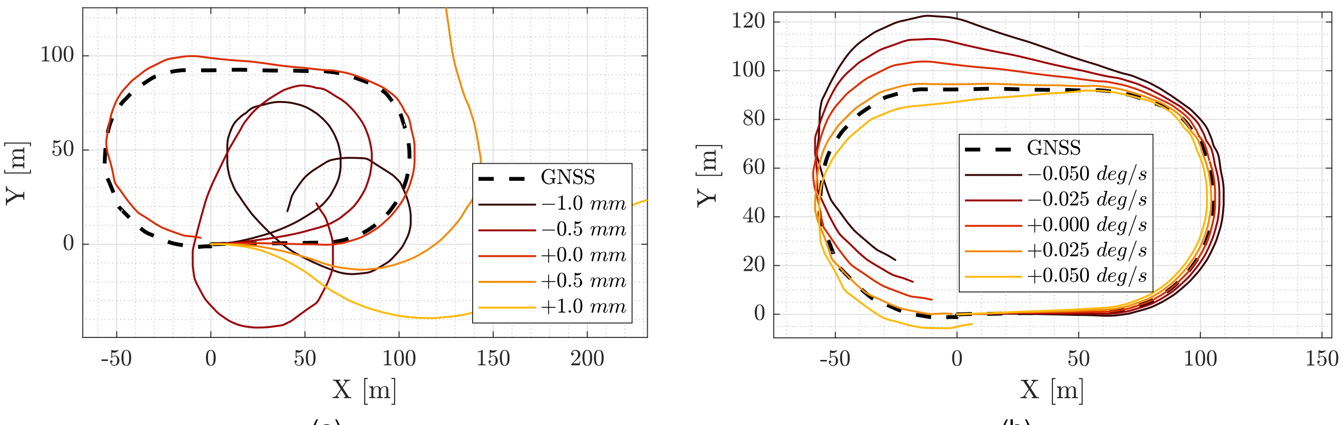
$$\hat{y}_k^{imu} = \frac{\omega_k^R R_k^R - \omega_k^L R_k^L}{T} + b_k. \quad (6)$$

D. Dealing with multi-rate and unavailable measures

As introduced above, we aim to achieve a high-frequency output, independently from the acquisition frequency of the slowest sensor. In our case, both the encoders and the IMU provide data at 100 Hz: we select the execution frequency $1/T_s = 60$ Hz to maximize the probability of receiving new samples from the proprioceptive sensors before each iteration, even under severe congestion of the ROS network. In practice, this means that the predicted measurements vector will have a minimum of 3 elements and a maximum of 8, with GNSS measures being integrated roughly every 6 iterations and Cartographer ones every 3.

An important side-effect of this design choice is that the observability properties of the system change between one iteration and the following one. Indeed, in all the iterations where no position measurements are available, the linearized measurement matrix would result in being

$$C = \begin{bmatrix} 0 & 0 & 0 & \frac{R_R}{T} & -\frac{R_L}{T} & \frac{\omega_R}{T} & \frac{\omega_L}{T} & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}. \quad (7)$$



(a)

(b)

Fig. 3: Sensitivity analysis of model uncertainties on the estimated (odometrical) trajectory: left wheel radius (a) and IMU yaw rate bias (b).

The 3×3 null block will cause the first three columns of the observability matrix $\mathcal{O} = \begin{bmatrix} C & CA & \dots & CA^7 \end{bmatrix}^T$ to be zero-valued as well. As a matter of fact, the number of linearly independent rows would be $\operatorname{rank} \mathcal{O} = 3$: the observable state is reduced to the two wheel speeds and one between the radii and the yaw rate bias.

Some works in the literature solve this issue by predicting the measurement as usual and inserting it as a virtual measurement [15] or by dividing the estimation problem into multiple sub-filters [16]. The former approach, however, induces an artificial convergence of the unobservable states covariances. The latter, while being theoretically sounder, cannot consider cross-correlations between the states of different filters and its implementation is inevitably complex. We opt for a different approach, which applies the prediction steps of the EKF to the full state, while the correction step is applied only to the observable states. Although one of the model uncertainty states would theoretically be observable even without position measures, we choose to correct just the wheel speeds. Indeed if, as an example, we opted to estimate the IMU bias as well, the measurement model in (6) would become

$$\hat{y}_k^{imu} = \frac{\omega_k^R \bar{R}^R - \omega_k^L \bar{R}^L}{T} + b_k \tag{8}$$

where $\bar{R}^R$ and $\bar{R}^L$ are the latest available estimates of the wheel radii. However, it is not possible to assert that such estimates have reached convergence, and employing erroneous values would result in an incorrect estimate of the bias b as well.

In practice, this approach results in the model uncertainty states remaining constant, while the position states get updated in an odometrical fashion. Notice that the covariance of the unobservable states grows during these "open-loop" periods, since the state covariance prediction step is applied to the full state.

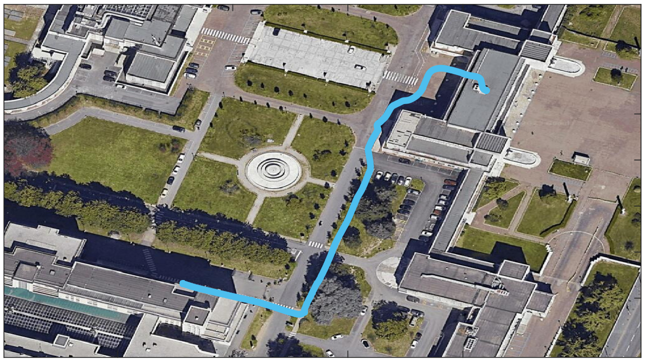


Fig. 4: Aerial view of the trajectory traveled by Yape in the courtyard of the Niguarda Hospital in Milan.

V. VALIDATION RESULTS

The proposed algorithm has been extensively tested in field trials in which the robot autonomously navigated public sidewalks. The experiment reported in this paper was conducted in the spacious courtyard of the Niguarda Hospital (*ASST Grande Ospedale Metropolitano Niguarda*) in Milan during test runs of the Budd-e project [17]. The traversed courtyard is a vast area with heavy pedestrian and vehicular traffic, presenting numerous buildings and green areas. Figure 4 shows the aerial view of the robot trajectory.

The presented analysis focuses on the robustness of the localization solution to the loss of absolute position measurements and the effect of model uncertainty estimation. The first set of experiments proves the robustness of the algorithm to the loss of one of the two absolute pose measurements, either the GNSS or Cartographer. The second analysis presents the effect of model uncertainty estimation with position measurement available. The last set of experiments combines the previous two, by assessing the performance when all position measures are lost and evaluating the impact of estimating model uncertainties under such circumstances.

A. Loss of one absolute pose source

In the presented experiment, both GNSS and Cartographer provide high-quality measurements during the entirety of the test, exception made for the final seconds in which GNSS quality degrades as the robot enters a indoor area (top right in Figure 4). This allowed us to simulate the loss of one sensor by disabling the fusion in the filter while using it for ground truth purposes. Figure 5 presents the results of these experiments: in blue, the error between the output of the filter and Cartographer measures remains close to the raw sensor data, even when fed GNSS data only (right plots). Note that in spite of not having direct yaw measurements, the error maintains a median of $1\,\mathrm{deg}$. Similar conclusions can be drawn by inspecting the red plots, which report errors with respect to the GNSS position.

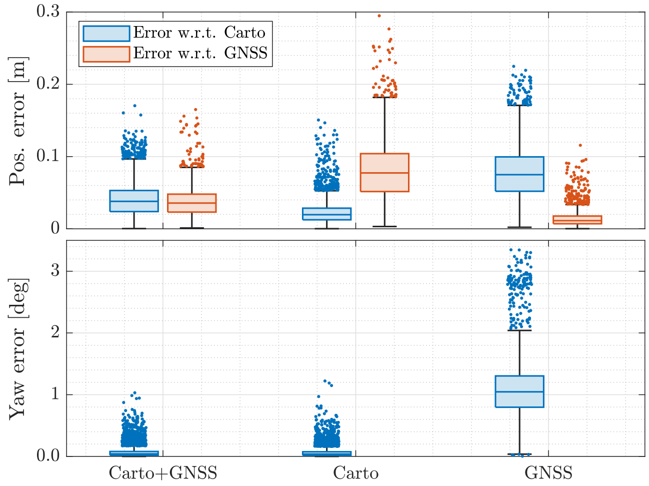


Fig. 5: Position (top) and yaw (bottom) error distribution of filter estimate with respect to Cartographer and GNSS measures. The X-axis refers to the fused position inputs (GNSS, Cartographer, or both).

B. Model uncertainties estimation

The next set of experiments shows the effect of the model uncertainty estimation on the velocity states. The goal of this analysis is to show how estimating the wheel radii and IMU yaw rate bias is necessary to achieve coherency between the position and velocity states of the filter. Two experiments were performed, with and without model uncertainty estimation, and for each one the position outputs were used to compute the cumulative traveled distance s. Similarly, the output velocities were integrated to obtain both the traveled distance and cumulated yaw angle. The top plot in Figure 6 displays the error between the cumulated distance computed from the poses and the one computed from the velocities, while the bottom one shows the error between the yaw angle estimated from the EKF and the one obtained by integrating the estimated yaw rate. With the model uncertainty estimation disabled (dark red dashed lines) both errors diverge: this means that the velocities are not getting estimated correctly, and their integration drifts. With the estimation enabled (light green lines), after a convergence phase, such drift is significantly reduced from the longitudinal distance and even eliminated entirely from the yaw angle.

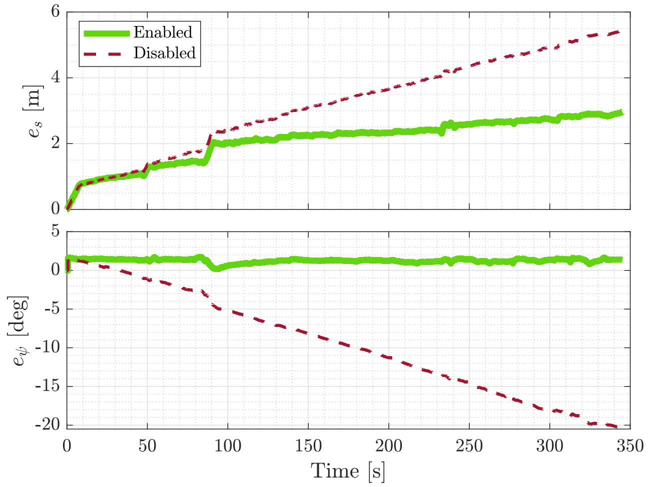


Fig. 6: Effect of model uncertainties estimation on the curvilinear abscissa (top) and yaw (bottom) error between the filter output pose and integrated output velocities. The experiment was repeated with (green lines) and without (red dashed lines) model uncertainties estimation.

C. Loss of all absolute pose sources

The final validation experiments aim to assess the effect of losing both position measures, a severe but not uncommon circumstance in real-world deployments of autonomous vehicles. To simulate this scenario, both the GNSS and Cartographer fusion are disabled from $t = 220\ s$, leaving the algorithm running with proprioceptive inputs only for $130\ s$. The trajectory estimated by the filter fed with both inputs was employed as ground truth. The effect of the estimation is clearly visible in Figure 7a, which depicts the trajectories after the measurement loss occurs (black dot). With the model uncertainties estimated until such moment, the resulting model is much more accurate and enables to maintain significantly lower errors and minimize odometrical drift, as visible in Figure 7b, which reports the time series of the filter output with respect to Cartographer's pose. Notice how, after more than two minutes of navigation and $\approx 70\ m$ traveled the cumulated drift reduces from $5.3\ m$ to $0.35\ m$ in terms of position error and from $7\ \mathrm{deg}$ to less than $2\ \mathrm{deg}$ in terms of yaw error.

VI. CONCLUSIONS

This paper presented a localization architecture for autonomous mobile robots. Structured on two layers, the first one encompasses localization sources including both physical sensors (IMU, encoders and GNSS) and fully-featured off-the-shelf localization algorithms (Cartographer). The second layer is a fusion filter based on the Extended Kalman Filter framework and tailored to the vehicle motion model, allowing the estimation of model uncertainties and

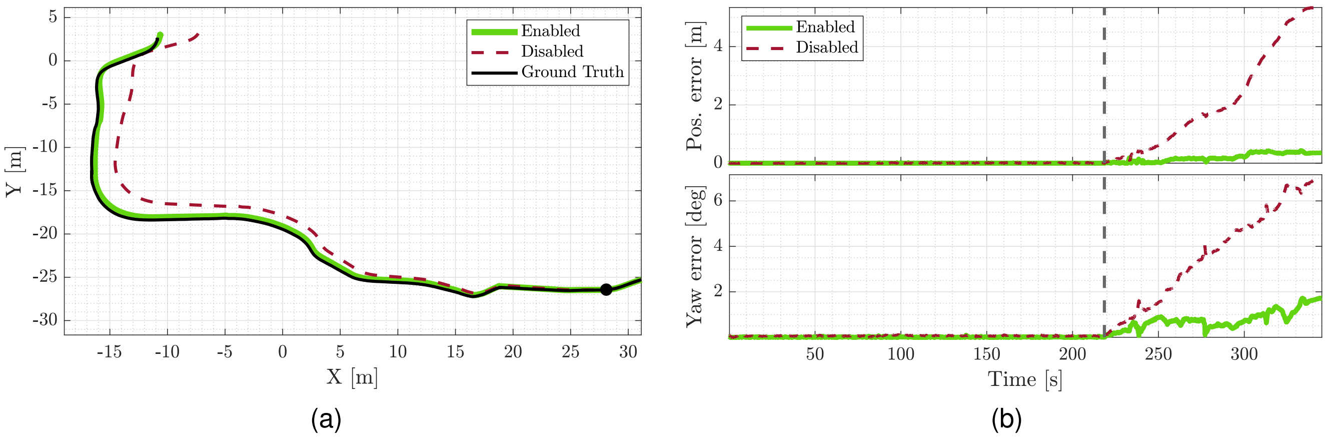


Fig. 7: Effect of model uncertainties estimation on the estimated (odometrical) trajectory (a)when position measures are lost: position ((b), top) and yaw ((b), bottom) errors. The black dot in (a) and dashed lines in (b) indicate the last position measurement received. The experiment was repeated with (green lines) and without (red dashed lines) model uncertainties estimation.

the subsequent improvement of odometry estimates. This represents a relevant improvement with respect of vehicle-agnostic localization algorithms, which despite their undoubtable usefulness, can fail in unrealistic ways. The filter is capable of handling multi-rate and unavailable measurements by adapting the estimated state online based on to the time-varying observability properties. Robustness to complete loss of position measurements is demonstrated by disabling Cartographer and GNSS corrections for more than two minutes, resulting in a cumulated error of 0.35 m, with respect to the 5.3 m achieved without uncertainties estimation.

REFERENCES

[1] D. Fox, "Kld-sampling: Adaptive particle filters and mobile robot localization," *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[2] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. Stockholm, Sweden: IEEE, May 2016, pp. 1271–1278.

[3] L. Mozzarelli, M. Corno, and S. M. Savaresi, "A comparative experimental study of LiDAR, camera, and LiDAR-camera localization algorithms for autonomous mobile robots on urban sidewalks," in *2023 IEEE Conference on Control Technology and Applications (CCTA)*. © 2023 IEEE, Aug. 2023, pp. 53–58.

[4] 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)*, Oct. 2020, pp. 5135–5142.

[5] B. Boberg and S.-L. Wirkander, "Integrating GPS and INS: comparing the Kalman estimator and particle estimator," in *7th International Conference on Control, Automation, Robotics and Vision, 2002. ICARCV 2002.*, vol. 1, Dec. 2002, pp. 484–490 vol.1.

[6] S. Rezaei and R. Sengupta, "Kalman Filter-Based Integration of DGPS and Vehicle Sensors for Localization," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 6, pp. 1080–1088, Nov. 2007.

[7] A. M. Sabatini and V. Genovese, "A Sensor Fusion Method for Tracking Vertical Velocity and Height Based on Inertial and Barometric Altimeter Measurements," *Sensors*, vol. 14, no. 8, pp. 13 324–13 347, Aug. 2014.

[8] S. Pizzocaro, M. Corno, M. Pantano, and S. Savaresi, "Magnetometer Aided GPS-free Localization of an Autonomous Vineyard Drone," in *2021 European Control Conference (ECC)*, Jun. 2021, pp. 1132–1137.

[9] A. Wischnewski, T. Stahl, J. Betz, and B. Lohmann, "Vehicle Dynamics State Estimation and Localization for High Performance Race Cars," *IFAC-PapersOnLine*, vol. 52, no. 8, pp. 154–161, 2019.

[10] B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *Journal of the Optical Society of America A*, vol. 5, no. 7, p. 1127, Jul. 1988.

[11] F. Parravicini, M. Corno, and S. Savaresi, "Robust State Observers for Two Wheeled Inverted Pendulum under wheel-slip," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, Oct. 2019, pp. 1525–1530.

[12] ——, "Extended target tracking for autonomous street crossing," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 440–15 445, 2020.

[13] M. Corno and S. Savaresi, "Measuring Urban Sidewalk Practicability: a Sidewalk Robot Feasibility Index," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15 053–15 058, Jan. 2020.

[14] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009, p. 6.

[15] A. Katriniok and D. Abel, "Adaptive EKF-Based Vehicle State Estimation With Online Assessment of Local Observability," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1368–1381, Jul. 2016.

[16] P.-J. Bristeau, N. Petit, and L. Praly, "Design of a navigation filter by analysis of local observability," in *49th IEEE Conference on Decision and Control (CDC)*. Atlanta, GA, USA: IEEE, Dec. 2010, pp. 1298–1305.

[17] A. Rebecchi, M. Farina, G. Andreoni, S. Capolongo, M. Corno, P. Perego, and E. Lettieri, "Shaping accessible public spaces for visually impaired people. The BUDD-e research experience," *TECHNE - Journal of Technology for Architecture and Environment*, no. 25, pp. 192–203, May 2023.