

# LiDAR-Inertial SLAM-Based Navigation and Safety-Oriented AI-Driven Control System for Skid-Steer Robots

Mehdi Heydari Shahna, Eemil Haaparanta, Pauli Mustalahti, and Jouni Mattila

**Abstract**—Integrating artificial intelligence (AI) and stochastic technologies into the mobile robot navigation and control (MRNC) framework while adhering to rigorous safety standards presents significant challenges. To address these challenges, this paper proposes a comprehensively integrated MRNC framework for skid-steer wheeled mobile robots (SSWMRs), in which all components are actively engaged in real-time execution. The framework comprises: 1) a LiDAR-inertial simultaneous localization and mapping (SLAM) algorithm for estimating the current pose of the robot within the built map; 2) an effective path-following control system for generating desired linear and angular velocity commands based on the current pose and the desired pose; 3) inverse kinematics for transferring linear and angular velocity commands into left and right side velocity commands; and 4) a robust AI-driven (RAID) control system incorporating a radial basis function network (RBFN) with a new adaptive algorithm to enforce in-wheel actuation systems to track each side motion commands. To further meet safety requirements, the proposed RAID control within the MRNC framework of the SSWMR constrains AI-generated tracking performance within predefined overshoot and steady-state error limits, while ensuring robustness and system stability by compensating for modeling errors, unknown RBF weights, and external forces. Experimental results verify the proposed MRNC framework performance for a 4, 836 kg SSWMR operating on soft terrain.

**Index Terms**—Robust control, mobile robot, sensor fusion.

## I. INTRODUCTION

Skid-steer wheeled robots (SSWMRs) are a common choice for tasks performed in off-road settings due to their straightforward design, ability to turn in place, and high maneuverability. However, they exhibit complex wheel-ground interactions and require an advanced integration of technologies within the mobile robot navigation and control (MRNC) framework to operate autonomously under uncertainties and disturbances. Hence, ensuring compatibility in integration and control while adhering to high safety standards is a significant challenge [1]. MRNC involves executing a series of actions that move it from its current position to the intended destination. This requires the robot to accurately determine its current location. Simultaneous localization and mapping (SLAM) enables a robot to concurrently build a map of an unknown environment while estimating its pose. Most SLAM methods are based on either visual or LiDAR technology. [2]. Although visual SLAM is a more affordable option that relies on camera images, performing well in feature-rich environments with good lighting, it struggles in low-light or feature-poor conditions. In addition, it is prone to drift, particularly over extended

operation [3], [4]. On the other hand, LiDAR-based SLAM uses laser scanning data to provide highly accurate depth information, making it more robust in diverse conditions, including low-light and feature-poor areas. A state-of-the-art approach to improve pose estimation in SLAM is sensor fusion, where data from multiple sensors are integrated to enhance robustness, especially beneficial in environments with sensory degradation caused by factors such as geometric ambiguities, illumination variations, or environmental obscurants (e.g., dust, smoke, fog, rain, or snowfall) [5], [6]. To obtain more accurate positioning and create better maps, LiDAR data can be combined with information from other sensors like GPS and IMUs. Real-time SSWMR trajectory estimation is accomplished in this paper through the use of the LIO-SAM algorithm, a technique derived from the findings of [6] that performs LiDAR inertial odometry via smoothing and mapping.

With this essential information about the SSWMR pose in a built map, the control system must then develop a robust tracking strategy to reach the desired motion by continuously generating appropriate control signals [7], monitoring motion feedback, and compensating for external disturbances such as uneven terrain, load variations, or environmental forces [8]. The use of artificial intelligence (AI) in state-of-the-art control system technologies has grown rapidly due to benefits such as adaptability to different environments and autonomy and learning decision-making for reducing the need for human oversight, thereby lowering operational costs while increasing productivity [9], [10]. However, using AI-driven control technologies in the MRNC framework while adhering to rigorous safety standards presents significant challenges [11], [12]. According to ISO/IEC TR 5469 [12], a supervisory control module using methods like barrier function approaches can ensure the system remains within a safe operating region. Hence, to avoid violations of safety-related system constraints, [13] employed a logarithmic barrier Lyapunov function (BLF) through an adaptive neural network (NN) optimized control for marine vessels suffering from ocean disturbances. However, it assumed a constant state constraint that does not directly guarantee restrictions on control performance metrics such as overshooting or steady-state error. Although [14] proposed a fractional BLF through sliding-mode adaptive radial basis function networks (RBFNs) with predefined control performance, the study overlooked the safety-defined constraints on input control signals, which are limited by mechanical restrictions in the actuator's working space. Building on [15], [16] and taking a step further in meeting safety requirements, this paper proposes a comprehensive control system within the MRNC framework of SSWMR to constrain AI-generated tracking performance within predefined overshoot and steady-state error boundaries as well as safety-defined control inputs.

Funding for this research was provided by the Business Finland partnership project Future All-Electric Rough Terrain Autonomous Mobile Manipulators (Grant No. 2334/31/2022).

M. H. Shahna, E. Haaparanta, P. Mustalahti, and J. Mattila are with the Faculty of Engineering and Natural Sciences, Tampere University, Finland. (e-mail: firstname.lastname@tuni.fi).

The control system design includes: 1) A pure pursuit algorithm for path-following control for generating desired linear and angular velocity commands of the robot based on the current pose (generated by LIO-SLAM) and the desired pose; 2) inverse kinematics for transferring linear and angular velocity commands into left- and right-side linear velocity commands; and 3) a robust AI-driven (RAID) control system incorporating RBFNs with a new adaptive algorithm to enforce in-wheel actuation systems to track each side motion commands within the predefined performance. The contributions of the study include: 1) this is the first paper to present a comprehensive MRNC framework for SSWMRs, integrating a state-of-the-art LiDAR-inertial SLAM-based navigation with a novel model-based, safety-oriented, AI-driven control system; 2) unlike [13], [14], [17], safety-defined input-output constraints are addressed in the design of the RAID control by using enhanced logarithmic BLF with user-defined transient and steady-state performance; 3) employing the RAID control guarantees that the actual motion of SSWMRs tracks the reference motion with an exponential convergence rate, even under modeling errors and external disturbances; and 4) the proposed MRNC performance is validated by implementing it on a 4,836 kg SSWMR operating on a slippery and wet terrain during low-light conditions. The entire proposed MRNC framework for SSWMRs is shown in Fig. 1, where the integrated framework and the displayed parameters will be elaborated on later.

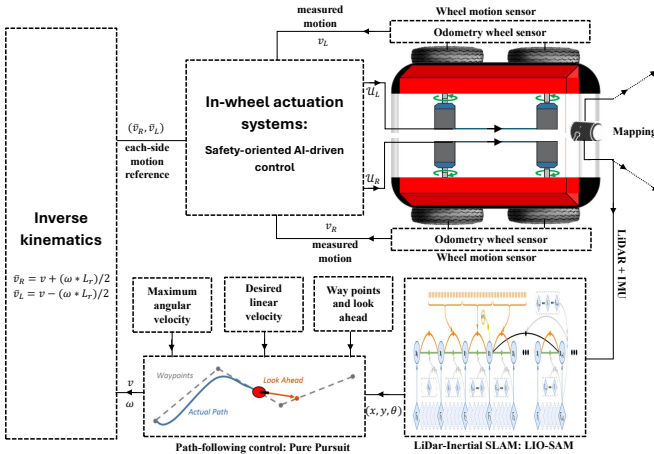


Fig. 1: The whole proposed MRNC framework for SSWMRs.

The remainder of this paper is organized as follows: Section II introduces the LiDAR-inertial SLAM algorithm used for SSWMRs, based on the LIO-SAM algorithm, which lays the foundation for the control system. Section III designs the proposed control system, including path-following control, inverse kinematics, and the safety-oriented AI-driven in-wheel actuation system control. In addition, this section provides a rigorous stability proof. In Section IV, the proposed MRNC framework is experimentally applied to a heavy-weight SSWMR to demonstrate the MRNC framework's effectiveness.

## II. LIDAR-INERTIAL SLAM-BASED NAVIGATION

Building upon the work of [6], this paper employs the LIO-SAM algorithm. This system uses a factor graph to

combine data from LiDAR and inertial sensors for accurate motion tracking. It integrates various types of measurements, including loop closures identified through radius-search, into this graph. The IMU data is processed to correct LiDAR point cloud distortion and provide an initial motion estimate. This estimate then helps refine the IMU's bias. To achieve real-time speed, the system optimizes poses using only recent LiDAR scans, rather than a full global map. This local scan-matching approach, along with strategic keyframe selection and a sliding window technique, significantly improves efficiency. Specifically, new keyframes are compared to a limited set of recent, representative poses (sub-keyframes). Fig. 2 illustrates the factor graph architecture that forms the SLAM system basis. It integrates data from an IMU, a LiDAR, and, when available, a GPS. To build this graph, four distinct factor types are incorporated: (a) factors derived from IMU preintegration, (b) LiDAR odometry factors, (c) GPS measurement factors, and (d) loop closure factors, as detailed in [6].

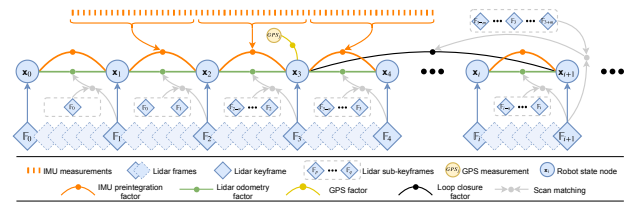


Fig. 2: The system structure of LIO-SAM [6].

### A. Lidar Odometry

The LiDAR odometry pipeline of LIO-SAM can be separated into distinct parts: feature extraction, voxel map creation, correspondence matching, and transform optimization. These modules follow the LiDAR odometry algorithm presented in [18] and [6]. Features are extracted from each incoming LiDAR scan. The features are divided into edge and planar features, depending on the smoothness value  $c$  of a local surface defined as:

$$c = \frac{1}{|\mathcal{S}| \cdot \|p_{i+1,k}\|} \left\| \sum_{u \in \mathcal{S}, u \neq k} (p_{i+1,k} - p_{i+1,u}) \right\| \quad (1)$$

where  $p_{i+1}$  is a point represented in the LiDAR frame at index  $k$  from the current LiDAR scan at time  $t = i+1$ . The set  $\mathcal{S}$  contains consecutive points distributed equally around point  $k$ . The normalized sum of differences between the point  $k$  and points  $u \in \mathcal{S}$  classifies if a point is considered as a feature. Points with the largest value of  $c$  are selected as edge points, as this would indicate a greater deviation between local points. Accordingly, points attaining the smallest smoothness values are considered as planar points [18]. Every LiDAR frame  $\mathbb{F}_i$  is composed of the extracted planar features  $F_i^p$  and edge features  $F_i^e$  represented in a body-fixed frame. For efficient processing, LiDAR frame  $\mathbb{F}_{i+1}$  becomes a keyframe only if the robot's movement significantly differs from its prior position, as determined by a set threshold. This frame is linked to the new node in the factor graph [19]. A point cloud map is created using the  $n$  most recent keyframes, called sub-keyframes. Employing the associated transformations  $\{\mathbf{T}_{i-n}, \dots, \mathbf{T}_i\}$ ,

the sub-keyframes  $\{\mathbb{F}_{i-n}, \dots, \mathbb{F}_i\}$  become transformed to a fixed world frame. The point cloud map is represented as a voxel map  $\mathbf{M}_i$  comprising sub-voxel maps  $\mathbf{M}_i^p$  and  $\mathbf{M}_i^e$  for transformed planar and edge features, respectively. The sub-voxel maps are defined as the intersection of the transformed feature points to avoid duplicate features inside single voxels:

$$\begin{aligned} \mathbf{M}_i^p &= 'F_i^p \cap 'F_{i-1}^p \cap \dots \cap 'F_{i-n}^p \\ \mathbf{M}_i^e &= 'F_i^e \cap 'F_{i-1}^e \cap \dots \cap 'F_{i-n}^e, \end{aligned} \quad (2)$$

where  $'F_i^p$  and  $'F_i^e$  denote the point sets transformed from the body-fixed frame to world-fixed one. The initial transform is given by an IMU measurement [6]. A kD-tree is built for both local planar and surface maps. This allows the use of a nearest neighbor search to find the closest points for the extracted feature points  $'F_{i+1}^p$  and  $'F_{i+1}^e$  from the current scan. The neighboring points are used for distance calculation between a feature point and its correspondent edge line in  $\mathbf{M}_i^e$  and between a feature point and its correspondent planar patch in  $\mathbf{M}_i^p$ . The edge-to-line distance is defined as

$$d_{ek} = \frac{\left| \left( p_{i+1,k}^e - p_{i,u}^e \right) \times \left( p_{i+1,k}^e - p_{i,v}^e \right) \right|}{\left| p_{i,u}^e - p_{i,v}^e \right|} \quad (3)$$

The points  $p_{i,u}^e$  and  $p_{i,v}^e$  form an edge line in  $\mathbf{M}_i^e$ . The distance metric for point-to-plane is:

$$d_{pk} = \frac{\left| \begin{array}{c} \left( p_{i+1,k}^p - p_{i,u}^p \right) \\ \left( p_{i+1,k}^p - p_{i,u}^p \right) \times \left( p_{i,u}^p - p_{i,w}^p \right) \end{array} \right|}{\left| \left( p_{i,u}^p - p_{i,v}^p \right) \times \left( p_{i,u}^p - p_{i,w}^p \right) \right|} \quad (4)$$

The points  $p_{i,u}^p$ ,  $p_{i,v}^p$  and  $p_{i,w}^p$  form the correspondent planar patch in  $\mathbf{M}_i^p$  [18]. Using the functions above, the optimal transform can be solved by minimizing point-to-plane and point-to-edge distances using the iterative Gauss-Newton optimization:

$$\min_{\mathbf{T}_{i+1}} \left\{ \sum_{p_{i+1,k}^e \in 'F_{i+1}^e} d_{ek} + \sum_{p_{i+1,k}^p \in 'F_{i+1}^p} d_{pk} \right\} \quad (5)$$

The LiDAR odometry factor, which is the relative transformation between poses at  $i$  and  $i+1$ , is then obtained by [6]  $\Delta \mathbf{T}_{i,i+1} = \mathbf{T}_i^T \mathbf{T}_{i+1}$ .

### B. Loop closure: radius search

To address drifts, the default loop closure presented in [6] is a Euclidean distance-based solution referred to as radius search. To detect loop closures, the system checks prior states located within a radius around each newly added state in the factor graph. The transformation  $\Delta T_{n,i+1}$  between the current frame  $\mathbb{F}_{i+1}$  and the prior sub-keyframes  $\{\mathbb{F}_{n-m}, \dots, \mathbb{F}_n, \dots, \mathbb{F}_{n+m}\}$  is obtained through the scan matching procedure described in the LiDAR odometry section. The transformation obtained through scan matching is added as a loop closure factor to the factor graph.  $\mathbb{F}_n$  is the loop closure candidate frame and  $m$  is a sliding window parameter that determines the number of sub-keyframes chosen for scan matching. In summary, LiDAR provides spatial measurements

for mapping, while the IMU refines motion estimation, especially in featureless areas. The system outputs a consistent map of the environment and the robot's pose estimation  $(x, y, \theta)$ , representing the position and orientation in a 2D plane.

## III. DESIGN OF CONTROL SYSTEM

### A. Path-Following Control

The pure pursuit algorithm is a straightforward method for implementation on SSWMRs [20]. It designates a target point ahead of the robot based on the current pose [20], [21] and conveys it to the in-wheel actuation system for control purposes. The next target point (waypoint)  $(x_t, y_t)$  is selected at the look-ahead distance  $L$  from the path such that the distance from the robot's current position  $(x, y)$  is at least  $D_t \geq L > 0$ . Then, the angle between the robot's current position and the target point is obtained as  $\alpha_t = \text{atan2}(y_t - y, x_t - x)$ . The steering angle can be calculated as  $\delta_t = \text{atan2}\left(\frac{2L \sin(\alpha_t)}{D_t}\right)$ . Finally, the commanded angular velocity based on the desired linear velocity  $v$  for the next step time can be obtained as:

$$\omega = \frac{v}{\frac{D_t}{2 \tan(\delta_t)}}.$$

*Remark 1:*  $\text{atan2}(\cdot)$  function is inherently designed to handle singularities gracefully and returns the four-quadrant inverse tangent ( $\tan^{-1}$ ), which must be real in the closed interval  $(-\pi, \pi)$ . In addition, if  $\delta_t = 0$ , then  $\omega$  becomes infinite. To avoid singularity, we set  $\omega = 0$  if  $\delta = 0$ . Finally, after reaching the last waypoint, both the angular velocity  $\omega$  and linear velocity  $v$  are set to zero to ensure the robot stops at the last waypoint without any further velocity commands.

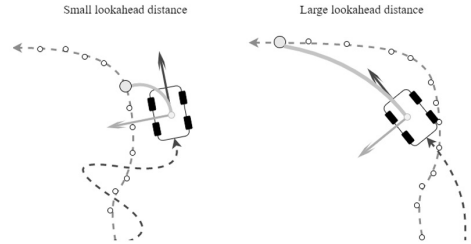


Fig. 3: Impacts of lookahead distance [20].

The algorithm's successful application depends on the proper size of the lookahead distance [20]. As shown in Fig. 3, if the lookahead distance is too small, oscillation might occur. In contrast, in case of an inconveniently large lookahead distance, the problem of truncating might be risen.

### B. Inverse Kinematics of SSWMRs

The inverse kinematics of SSWMRs from the output of Section III-A can be provided as:

$$\bar{v}_R = v + (\omega L_r)/2, \quad \bar{v}_L = v - (\omega L_r)/2 \quad (6)$$

$L_r$  represents the wheelbase width of the robot. Note, based on Section III-A and Eq. (6), the desired linear velocity robot must be set as:  $v \leq \frac{v_{max}}{2(1+\tan(\delta_t)\frac{L_r}{D_t})}$ , where  $v_{max}$  is the predefined maximum velocity of the SSWMR. These outputs  $(\bar{v}_R, \bar{v}_L)$  are then sent to both sides of in-wheel actuation controllers as references, ensuring proper execution of the motion.

### C. Safety-Oriented RAID Control for In-wheel Actuators

We define the linear velocity for each side ( $i : R, L$ ) as  $v_i \in \mathbb{R}$ . The motion dynamics of the in-wheel actuation system is

$$\dot{v}_i = g_i(v_i, t) \mathcal{U}_i(t) + d_i(v_i, t) + \Delta_i(t) \quad (7)$$

where  $d_i : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$  represents the bounded modeling uncertainties,  $\mathcal{U}_i : \mathbb{R}_+ \rightarrow \mathbb{R}$  denotes the model-based control generated by the actuator mechanism, and  $\Delta_i : \mathbb{R}_+ \rightarrow \mathbb{R}$  accounts for the bounded external disturbance. Note that  $g_i : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$  is the control coefficient.

*Assumption 1:*  $g_i$  is allowed to be non-constant and unknown, but it must be positive, ensuring that the control actuator signal has the same sign as the robot motion.

*Assumption 2* [15], [22]: Assume positive constants  $g_i^* \in \mathbb{R}_+$  and  $\Delta_i^* \in \mathbb{R}_+$  for bounds of functional control gain and external disturbances as  $|g_i(v_i, t)| \leq g_i^*$  and  $|\Delta_i(t)| \leq \Delta_i^*$ .

To define safety-oriented control inputs for preventing harm to mechanical actuators, let us define the following constraints:

$$\lambda_i = \begin{cases} \frac{1}{|\mathcal{U}_i(t)|+1}, & \mathcal{U}_i(t) \geq \bar{\mathcal{U}}_i \text{ or } \mathcal{U}_i(t) \leq \underline{\mathcal{U}}_i \\ 1 & \underline{\mathcal{U}}_i \leq \mathcal{U}_i(t) \leq \bar{\mathcal{U}}_i \end{cases} \quad (8)$$

and

$$\bar{\lambda}_i = \begin{cases} \bar{\mathcal{U}}_i - \frac{\mathcal{U}_i(t)}{|\mathcal{U}_i(t)+1}, & \mathcal{U}_i(t) \geq \bar{\mathcal{U}}_i \\ 0 & \underline{\mathcal{U}}_i \leq \mathcal{U}_i(t) \leq \bar{\mathcal{U}}_i \\ \underline{\mathcal{U}}_i - \frac{\mathcal{U}_i(t)}{|\mathcal{U}_i(t)+1} & \mathcal{U}_i(t) \leq \underline{\mathcal{U}}_i \end{cases} \quad (9)$$

$\underline{\mathcal{U}}_i$  and  $\bar{\mathcal{U}}_i$  are the acceptable boundaries following mechanical actuator specifications, respectively. Then, if we define safety-constrained control inputs as

$$\mathcal{U}_i^* = \lambda_i \mathcal{U}_i + \bar{\lambda}_i \quad (10)$$

we will always have  $\mathcal{U}_i^*$ , as

$$\mathcal{U}_i^* = \begin{cases} \bar{\mathcal{U}}_i, & \mathcal{U}_i(t) \geq \bar{\mathcal{U}}_i \\ \mathcal{U}_i & \underline{\mathcal{U}}_i \leq \mathcal{U}_i(t) \leq \bar{\mathcal{U}}_i \\ \underline{\mathcal{U}}_i & \mathcal{U}_i(t) \leq \underline{\mathcal{U}}_i \end{cases} \quad (11)$$

We define the tracking error as  $e_i = v_i - \bar{v}_i$ , where  $\bar{v}_i$  is the reference velocity of  $i$  side received from Section III-B. By altering control inputs  $\mathcal{U}_i$  with  $\mathcal{U}_i^*$  in (7), we have

$$\dot{e}_i = g_i \mathcal{U}_i^* + d_i + \Delta_i - \dot{\bar{v}}_i \quad (12)$$

By employing RBFNs, the uncertainty term can be defined as

$$d_i(v_i) = (a_i)^\top \Phi_i(v_i) + \delta_i(v_i) \quad (13)$$

where  $v_i$  serves as the input to the RBFNs. The unknown output layer's weight vector  $a_i \in \mathbb{R}^{N_i}$  corresponds to  $N_i$  hidden-layer neurons.  $\delta_i(\cdot) : \mathbb{R} \rightarrow \mathbb{R}$  represents the unknown approximation error.  $\Phi(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{N_i}$  is the function defined as

$$\Phi_i(v_i) = [\Phi_{i1}(v_i) \quad \Phi_{i2}(v_i) \cdots \Phi_{iN_i}(v_i)]^\top \quad (14)$$

$\Phi_{ik}(\cdot)$  can be selected a common Gaussian functions, as

$$\Phi_{ik}(v_i) = \exp\left(-\frac{\|v_i - \alpha_{ik}\|^2}{\gamma_{ik}^2}\right), \quad k = 1, 2, \dots, N_i \quad (15)$$

where  $\alpha_{ik} \in \mathbb{R}$  can be selected stochastically and defines the center of the receptive field, while  $\gamma_{ik} \in \mathbb{R}$  is a given constant

determining the width of the Gaussian functions. The symbol  $\|\cdot\|$  denotes the Euclidean norm.

*Assumption 3:* Suppose the unknown approximation error satisfies  $|\delta_i(v_i)| \leq \delta_i^* < \infty$ . However, the weights  $a_i$ , the approximation error  $\delta_i$ , and its upper bound  $\delta_i^*$  are unknown.

From (12) and (13), we can have:

$$\dot{e}_i = g_i \mathcal{U}_i^* + (a_i)^\top \Phi_i(v_i) + \delta_i(v_i) + \Delta_i - \dot{\bar{v}}_i \quad (16)$$

According to ISO/IEC TR 5469 [12], safety-defined constraints can be formulated as barrier functions to define an invariant set. This method ensures that the system remains within a safe operating region. Let us define the bound of tracking control performance in terms of overshooting and steady-state error as follows  $|e_i| < o_i(t)$ , where:

$$o_i = (o_i^{ov} - o_i^b) e^{-o_i^* t} + o_i^b \quad (17)$$

where  $o_i^{ov}, o_i^b$ , and  $o_i^*$  are positive constants. The transient response of  $e_i$  must be restricted to  $o_i^{ov}$ . The exponential convergence rate and steady state response of  $e_i$  are  $o_i^*$  and  $o_i^b$ . The model-based constrained control input in Eq. (10) is defined as

$$\mathcal{U}_i^* = \lambda_i \mathcal{U}_i + \bar{\lambda}_i = \lambda_i [u_i + f_i(v_i)] + \bar{\lambda}_i \quad (18)$$

where  $f_i : \mathbb{R} \times \mathbb{R}_+ \rightarrow \mathbb{R}$  is the known modeling term of the in-wheel actuation system. To constrain system states to be within the safety-defined tracking performance, we can use singularities for designing the control term  $u_i$  in which by choosing  $|e_i(t_0)| < o_i(t_0)$ , the control input must tend toward infinity as  $e_i$  approaches  $o_i$ . In this case, an error such as ‘‘Infinity or NaN value encountered’’ will occur in programming environments, causing the execution to halt and triggering emergency safety stops. Let us propose a novel RAID control  $u_i$ , as

$$u_i = -f_i - \frac{1}{2} \beta_i e_i - \frac{\dot{o}_i^2 e_i^3}{o_i^2 (o_i^2 - e_i^2)} - \frac{e_i}{o_i^2 - e_i^2} (\hat{\phi}_i^4 \|\Phi_i\| + \|\Phi_i\|^2 + \hat{v}_i^2 + 1) \quad (19)$$

where  $\beta_i > 0$  is the control design parameter.  $\hat{\phi}_i$  is the adaptive parameter designed to adjust the weights of the RBFNs in Eq. (13) and compensate for external disturbances in Eq. (16). The adaptive law is defined as

$$\dot{\hat{\phi}}_i = -\frac{1}{2} \kappa_i \hat{\phi}_i + \left(\frac{e_i}{o_i^2 - e_i^2}\right)^2 \hat{\phi}_i^3 \|\Phi_i\| \quad (20)$$

where  $\kappa_i > 0$ . Eqs. (19) and (20) contain the fractional barrier function with safety-defined singularities.

*Definition 1* [23]–[25] Consider a Lyapunov function  $V$  with initial value  $V(t_0)$  related to any system states. For  $t \geq t_0$ , the system is uniformly exponentially stable if

$$V \leq \bar{c} V(t_0) e^{-\rho(t-t_0)} + \ell \rho^{-1} \quad (21)$$

where  $\bar{c}$ ,  $\rho$ , and  $\ell$  are positive constants.

*Theorem 1:* Based on *Definition 1*, employing Eqs. (19) and (20) ensures uniform exponential stability of the system provided in Eq. (7) in the presence of uncertainties, external disturbance, and predefined safety-related constraints.

*Proof.* See Section III-D.

*Lemma 1.* For any positive value of  $o_i$ , the next inequality is valid for every  $e_i$  that meets the condition  $|e_i| < o_i$  [15], [26]:

$$\log\left(\frac{o_i^2}{o_i^2 - e_i^2}\right) < \frac{e_i^2}{o_i^2 - e_i^2} \quad (22)$$

*Remark 2:* Based on the Cauchy-Schwarz inequality for any  $a$  and  $b$  we can say  $ab \leq ca^2 + \frac{1}{4c}b^2$ , where  $c$  is positive.

#### D. Stability Analysis

Let us introduce a BLF, as

$$V_i = \frac{1}{2} \log\left(\frac{o_i^2}{o_i^2 - e_i^2}\right) + \frac{1}{2} \lambda_{\min_i} \hat{\phi}_i^2 \quad (23)$$

where  $\lambda_{\min_i}$  is an unknown constant. After differentiating Eq. (23), we have

$$\dot{V}_i = \frac{-\dot{o}_i e_i^2 + e_i o_i \dot{e}_i}{o_i(o_i^2 - e_i^2)} + \lambda_{\min_i} \hat{\phi}_i \dot{\phi}_i \quad (24)$$

By inserting Eq. (16), we have

$$\begin{aligned} \dot{V}_i &= \frac{-\dot{o}_i e_i^2}{o_i(o_i^2 - e_i^2)} + \frac{e_i}{o_i^2 - e_i^2} (g_i \mathcal{U}_i^* + a_i^\top \Phi_i + \delta_i \\ &\quad + \Delta_i - \dot{v}_i) + \lambda_{\min_i} \hat{\phi}_i \dot{\phi}_i \end{aligned} \quad (25)$$

From (18),

$$\begin{aligned} \dot{V}_i &= \frac{-\dot{o}_i e_i^2}{o_i(o_i^2 - e_i^2)} + \frac{e_i}{o_i^2 - e_i^2} (g_i \lambda_i u_i + g_i \lambda_i f_i \\ &\quad + a_i^\top \Phi_i + \delta_i + g_i \bar{\lambda}_i + \Delta_i - \dot{v}_i) + \lambda_{\min_i} \hat{\phi}_i \dot{\phi}_i \end{aligned} \quad (26)$$

Based on *Remark 2*,

$$\begin{aligned} \dot{V}_i &\leq \lambda_{\min_i} \frac{\dot{o}_i^2 e_i^4}{o_i^2(o_i^2 - e_i^2)^2} + \frac{1}{4\lambda_{\min_i}} + \frac{e_i}{o_i^2 - e_i^2} (g_i \lambda_i u_i \\ &\quad + g_i \lambda_i f_i) + \lambda_{\min_i} \frac{e_i^2}{(o_i^2 - e_i^2)^2} \dot{v}_i^2 + \frac{1}{4\lambda_{\min_i}} \\ &\quad + \lambda_{\min_i} \frac{e_i^2}{(o_i^2 - e_i^2)^2} \|\Phi_i\|^2 + \frac{1}{4\lambda_{\min_i}} \|a_i\|^2 \\ &\quad + \lambda_{\min_i} \frac{e_i^2}{(o_i^2 - e_i^2)^2} + \frac{1}{4\lambda_{\min_i}} (g_i \bar{\lambda}_i + \Delta_i + \delta_i)^2 \\ &\quad + \lambda_{\min_i} \hat{\phi}_i \dot{\phi}_i \end{aligned} \quad (27)$$

After mathematical simplifications,

$$\begin{aligned} \dot{V}_i &\leq \lambda_{\min_i} \frac{\dot{o}_i^2 e_i^4}{o_i^2(o_i^2 - e_i^2)^2} + \frac{e_i}{o_i^2 - e_i^2} (g_i \lambda_i u_i + g_i \lambda_i f_i) \\ &\quad + \lambda_{\min_i} \frac{e_i^2}{(o_i^2 - e_i^2)^2} \dot{v}_i^2 + \lambda_{\min_i} \frac{e_i^2}{(o_i^2 - e_i^2)^2} \|\Phi_i\|^2 \\ &\quad + \lambda_{\min_i} \frac{e_i^2}{(o_i^2 - e_i^2)^2} + \lambda_{\min_i} \hat{\phi}_i \dot{\phi}_i \\ &\quad + \frac{(g_i \bar{\lambda}_i + \Delta_i + \delta_i)^2 + \|a_i\|^2 + 2}{4\lambda_{\min_i}} \end{aligned} \quad (28)$$

Inserting the control input  $u_i$  and adaptive law from Eqs. (19) and (20), we obtain

$$\dot{V}_i \leq \lambda_{\min_i} \frac{\dot{o}_i^2 e_i^4}{o_i^2(o_i^2 - e_i^2)^2}$$

$$\begin{aligned} & - \frac{1}{2} g_i \lambda_i \beta_i \frac{e_i^2}{o_i^2 - e_i^2} - g_i \lambda_i \frac{\dot{o}_i^2 e_i^4}{o_i^2(o_i^2 - e_i^2)^2} \\ & - g_i \lambda_i \frac{e_i^2}{(o_i^2 - e_i^2)^2} (\hat{\phi}_i^4 \|\Phi_i\| + \|\Phi_i\|^2 + \dot{v}_i^2 + 1) \\ & + \lambda_{\min_i} \frac{e_i^2}{(o_i^2 - e_i^2)^2} (\dot{v}_i^2 + \|\Phi_i\|^2 + 1 + \hat{\phi}_i^4 \|\Phi_i\|) \\ & - \frac{1}{2} \lambda_{\min_i} \kappa_i \hat{\phi}_i^2 + \frac{(g_i \bar{\lambda}_i + \Delta_i + \delta_i)^2 + \|a_i\|^2 + 2}{4\lambda_{\min_i}} \end{aligned} \quad (29)$$

As  $0 < \lambda_{\min_i} \leq g_i \lambda_i$ , based on Assumptions 1,2, and 3,

$$\begin{aligned} \dot{V}_i &\leq - \frac{1}{2} \lambda_{\min_i} \beta_i \frac{e_i^2}{o_i^2 - e_i^2} - \frac{1}{2} \lambda_{\min_i} \kappa_i \hat{\phi}_i^2 \\ & + \frac{(g_i^* \bar{\lambda}_i + \Delta_i^* + \delta_i^*)^2 + \|a_i\|^2 + 2}{4\lambda_{\min_i}} \end{aligned} \quad (30)$$

Thus, from *Lemma 1*, Eq. (30) will be

$$\begin{aligned} \dot{V}_i &\leq - \frac{1}{2} \lambda_{\min_i} \beta_i \log\left(\frac{o_i^2}{o_i^2 - e_i^2}\right) - \frac{1}{2} \lambda_{\min_i} \kappa_i \hat{\phi}_i^2 \\ & + \ell_i \end{aligned} \quad (31)$$

where  $\ell_i = \frac{(g_i^* \bar{\lambda}_i + \Delta_i^* + \delta_i^*)^2 + \|a_i\|^2 + 2}{4\lambda_{\min_i}}$  is a positive constant depending to the intensity of the disturbances, over-saturation control, approximation error, and fitting weights of RBFNs. From [27], we can say

$$\dot{V}_i \leq -\rho_i V_i + \ell_i, \quad \rho_i = \min[\lambda_{\min_i} \beta_i, \kappa_i] \quad (32)$$

Now, we define an additional quadratic function based on (23) for both sides of the SWMR, as

$$V = V_R + V_L = \frac{1}{2} \sum_{i=R,L} \log\left(\frac{o_i^2}{o_i^2 - e_i^2}\right) + \hat{\phi}_i^2 \quad (33)$$

By differentiating Eq. (33), and from (32), we obtain

$$\begin{aligned} \dot{V} &= \dot{V}_R + \dot{V}_L \\ &\leq -\rho_R V_R + \ell_R - \rho_L V_L + \ell_L \leq -\rho V + \ell \end{aligned} \quad (34)$$

where

$$\rho = \min[\rho_R, \rho_L], \quad \ell = \ell_R + \ell_L \quad (35)$$

We can solve it based on [28], as follows

$$V \leq V(t_0) e^{-\rho(t-t_0)} + \ell \int_{t_0}^t e^{-\rho(t-T)} dT \quad (36)$$

Because  $e^{-\rho(t-T)}$  is always decreasing,

$$V \leq V(t_0) e^{-\{\rho(t-t_0)\}} + \ell \rho^{-1} \quad (37)$$

From (33), we can say

$$\sum_{i=R,L} \log\left(\frac{o_i^2}{o_i^2 - e_i^2}\right) \leq 2(V(t_0) e^{-\{\rho(t-t_0)\}} + \ell \rho^{-1}) \quad (38)$$

Thus, based on *Definition 1* and Eq. (38), *Theorem 1* is proved. Note that  $\sum_{i=R,L} \log\left(\frac{o_i^2}{o_i^2 - e_i^2}\right)$  reaches a defined region  $G(\tau)$  in uniformly exponential convergence, such that  $G(\tau) := \left\{ \sum_{i=R,L} \log\left(\frac{o_i^2}{o_i^2 - e_i^2}\right) \leq \bar{\tau}_i := 2 \ell \rho^{-1} \right\}$ .

TABLE I: Absolute pose error (APE) in meters for different SLAM methods.

Sequence	LIO-SAM	FAST-LIO2	ORB-SLAM3	ORB-SLAM3 inertial	VINS-Fusion	VINS-Fusion inertial
1	0.217127	0.409905	<b>0.164774</b>	0.485586	0.474715	0.440113
2	<b>0.86914</b>	1.754343	Failed	1.715077	2.216691	1.925925
3	0.81162	1.555683	<b>0.542835</b>	1.252041	1.387264	1.23698
4	0.417588	0.70375	<b>0.280474</b>	0.516507	0.529805	0.454944
5	0.681154	1.664873	<b>0.440216</b>	1.520726	1.743379	1.691131
6	0.869949	1.886515	<b>0.735056</b>	1.968528	3.277011	2.850892
7	<b>0.088294</b>	0.189954	0.103472	0.198334	1.130662	0.889479
8	0.941384	0.993397	<b>0.879032</b>	1.466838	2.90835	2.532334
9	<b>0.768462</b>	0.82185	0.851645	1.379856	1.742125	1.593643
10	0.903748	1.524194	<b>0.860641</b>	Failed	3.012128	2.583019
Mean	<b>0.6568466</b>	1.1504464	<b>0.53979389</b>	1.16705478	1.842213	1.619846

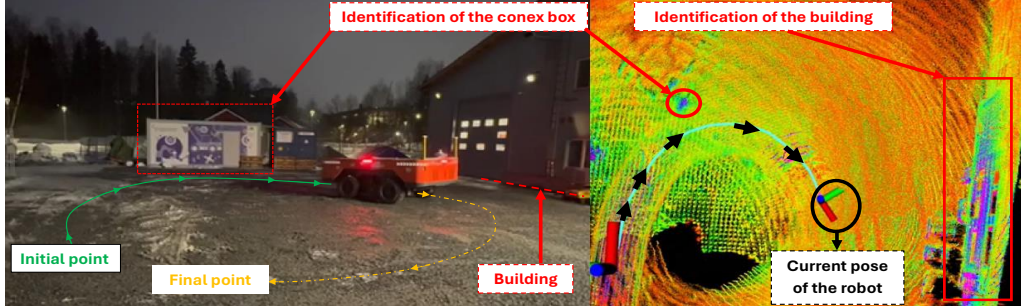


Fig. 4: The operation under low-light conditions and on soft terrain using the proposed MRNC framework (see Fig. 1). The relevant video is available at: <https://youtube.com/shorts/0bQO6-Syo08?feature=share>.

#### IV. EXPERIMENTAL RESULT ANALYSIS

The studied SSWMR, called the multi-purpose deployer (MPD), is a heavy-duty vehicle with a weight of 4,836 kg and a maximum speed of 0.97 m/s. The wheels are connected to bogies on each side, which ensures contact with the ground when traversing uneven terrain. The wheels are powered by hydraulic motors. The mechanical structure and sensor setup of the MPD are presented in Figs. 5 and 6, respectively. The pump flow model without uncertainties due to leakage, compressibility, or other inefficiencies is  $Q_p = V_p n_p$ , where  $V_p = 1.8 \times 10^{-5} \text{m}^3/\text{rev}$  is the displacement volume. This represents the fluid volume the pump moves per revolution of its shaft.  $n_p$  is the control input and is the rotational speed of the pump shaft, measured in revolutions per minute (RPM). The required volumetric flow rate for a fixed-displacement motor is specified as follows:

$$V_p n_p = Q_p = Q_m = \omega_m \frac{V_m}{2\pi} = \bar{v}_i \frac{\bar{G}}{r} \frac{V_m}{2\pi} \quad (39)$$

where  $\bar{G}$  is the gear ratio for conversion between the wheel velocity and the motor velocity.  $V_m$  is the motor displacement and  $r$  is the radius of the wheel. Thus, from (18), the modeling term of the proposed control is  $n_p = f_i = \bar{v}_i \frac{\bar{G}}{r} \frac{V_m}{2\pi} \frac{1}{V_p}$ , and the control signal is  $n_p^* = \mathcal{U}_i = u_i + n_p = u_i + f_i$ . Now, from Eq. (18), we design the model-based control with safety constraints, based on machine modeling specifications, as

$$\mathcal{U}_i^* = \lambda_i [u_i + f_i] + \bar{\lambda}_i = \lambda_i [u_i + 3000 \bar{v}_i] + \bar{\lambda}_i \quad (40)$$

The hardware of the perception system is divided into two separate host PCs. The embedded Beckhoff PC reads the IMU data from bogies and the chassis IMUs through the

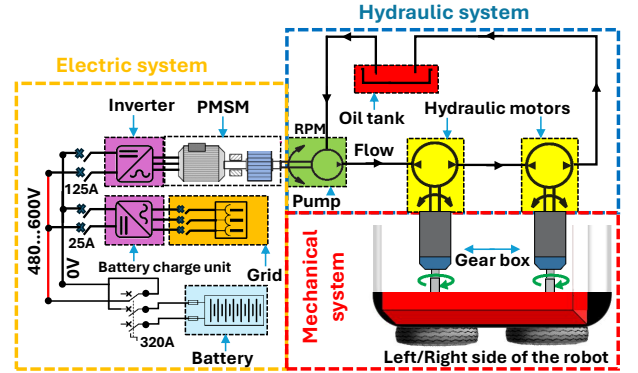


Fig. 5: Mechanical structure of each side of studied MPD.

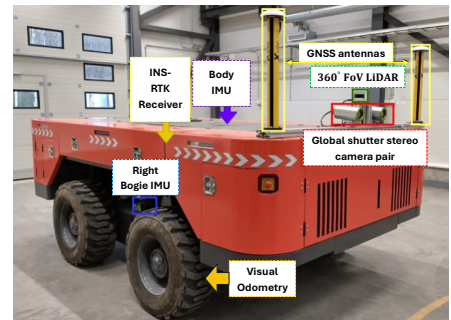


Fig. 6: The dimensions of the MPD are  $2\text{m} \times 4\text{m} \times 1.5\text{m}$ .

EtherCAT protocol. The measurements are then sent using User Datagram Protocol (UDP) to the Linux side of the system, which is powered by an industrial PC (IPC). The

Ouster OSO-128 LiDAR is connected to the Linux PC through a multi-gigabit switch to ensure enough bandwidth for point clouds. The data from the LiDAR are read utilizing the ROS framework in UDP to transfer the data from sensors to the host machine. The internal IMU of the Ouster LiDAR is also included. The external IMUs are read at 1000 Hz due to their usage in control applications. The LiDAR IMU is read at 100 Hz by default. The IMU for the chassis is ADIS16475-2, configured in its default settings of providing  $1024 \times 128$  points at 10 Hz. Various SLAM methods within the MRNC of the studied SSWMR are evaluated across 10 sequences in daylight conditions. The comparative APE is presented in Table I, where the last four algorithms use stereo cameras instead of LiDar. Although the lowest mean APE is attained by ORB-SLAM3 in stereo mode and Lidar-inertial LIO-SAM, using ORB-SLAM3 in low-light conditions is challenging. Eight waypoints, depending on the initial pose, and a 1 m lookahead distance are chosen for the pure pursuit control to achieve a circular trajectory with about 5 m radius. In addition, for safety reasons, the maximum linear velocity and the desired velocity are set to 0.4 m/s and 0.38 m/s (about 50% of the robot's capability), respectively. The safely predefined control performance in terms of over-shooting and steady-state error is:  $o_i = 0.19e^{-0.00009t} + 0.11$ . The actuator control effort constraint is  $|\mathcal{U}_i^*| < 1250$  RPM. The control gains are  $\beta_i = 1.2$  and  $\kappa_i = 5.2$ ,  $N_i = 9$ ,  $\alpha_i = [2 \text{ rand} - 1]$ , and  $\gamma_i = 0.13$ , where *rand* outputs a variable between 0 and 1.

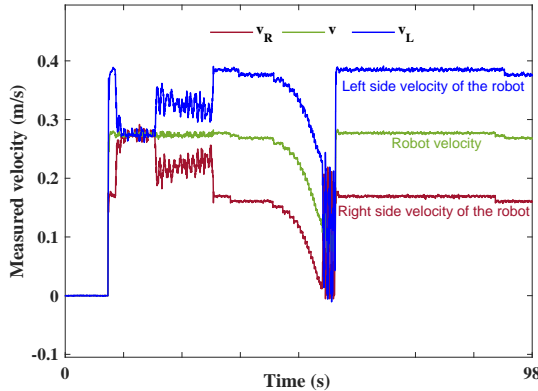


Fig. 7: Measured velocities through wheel odometry sensors.

Fig. 4 shows the entire operation as well as the recorded data from the employed LiDar-inertial SLAM algorithm. Note that it successfully identifies obstacles that can be used for designing collision-avoidance algorithms in future work. The conditions involves low light, and the terrain is wet and soft. The recorded velocities are indicated in Fig. 7 in the autonomous operation. As defined in the pure pursuit algorithm, the robot has the right side heading on the circle trajectory. Thus, the linear velocity of the left side is greater than that of the right side. The actual velocities at approximately 10-20s and again at 50s show wheel slippage due to the wet and soft ground; however, the RIAD control maintains stable tracking despite these disturbances. Fig. 8 indicates the control

tracking error between the measured wheel velocity (obtained from the wheel odometry sensor) and the desired velocity for each side (generated by the pure pursuit and inverse kinematics algorithms). The results show that both sides adhere to the safely predefined control performance.

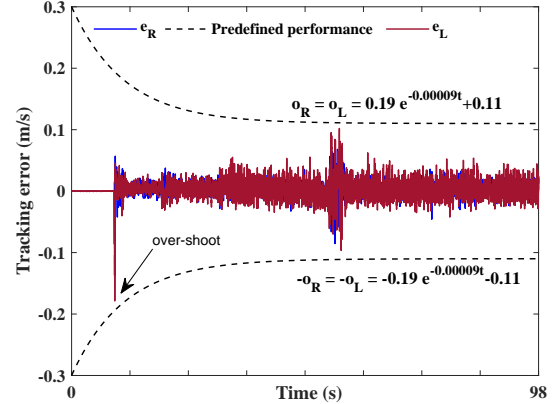


Fig. 8: RAID tracking control performance under predefined performance constraints.

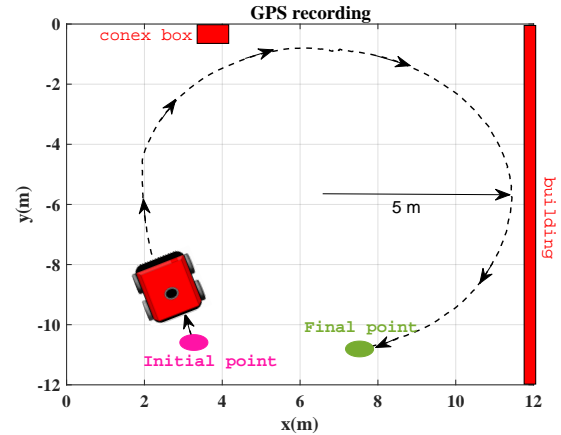


Fig. 9: The passed trajectory in 2D plane recorded by GPS.

Fig. 9 shows the trajectory of the robot in a 2D plane, recorded by GPS, forming a roughly circular shape with a radius of approximately 5 m. Wheel slippage from the location (2, -8) to (2, -4) and at (7, -1) are evident, while the RIAD maintains tracking stability under wheel slippage. The generated output efforts of the RAID control for each side of the pumps based on RPM are shown in Fig. 10, maintaining the signals within the safety actuator constraint of 1250 RPM. The performance of the RAID control in the same condition is compared with three control algorithms: an adaptive predictive control (APC) provided in [29], PID control, and a fractional-order prescribed-time controller (FOPTC) provided in [30]. The comparative results are summarized in Table II, validating the proposed RAID control efficacy.

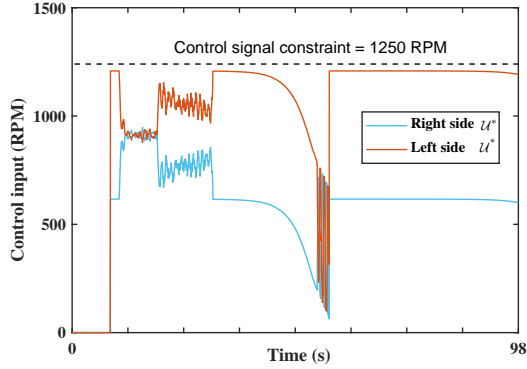


Fig. 10: RAID control signals for each side of the SSWMR.

TABLE II: Control performance metrics in the experiment.

Stability Criteria	RAID	APC	PID	FOPTC
Settling time	3.2 s	3.6 s	5.8 s	5.3 s
Overshoot	1.7%	2.1%	2.3%	1.8%
Steady-state error	0.043	0.069	0.12	0.056

## V. CONCLUSION

To meet rigorous safety standards and the complexity of SSWMRs, this paper proposes a novel integrated MRNC framework, in which all components are actively engaged in real-time execution, including: 1) a LiDAR-inertial SLAM algorithm, 2) an effective path-following control system, 3) inverse kinematics for transferring linear and angular velocity commands, and 4) a novel robust RAID control system to enforce in-wheel actuation systems. The proposed RAID control within the MRNC framework constrained the AI-generated tracking performance within predefined overshoot and steady-state error boundaries while ensuring robustness and system stability by compensating for uncertainties and external forces.

## REFERENCES

- [1] R. Khan, F. M. Malik, A. Raza, and N. Mazhar, "Comprehensive study of skid-steer wheeled mobile robots: development and challenges," *Industrial Robot: The International Journal of Robotics Research and Application*, vol. 48, no. 1, pp. 142–156, 2021.
- [2] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, 2019.
- [3] S. A. Mohamed, M.-H. Haghbayan, T. Westerlund, J. Heikkinen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems," *IEEE Access*, vol. 7, pp. 97466–97486, 2019.
- [4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [5] S. Khattak, H. Nguyen, F. Mascari, T. Dang, and K. Alexis, "Complementary multi-modal sensor fusion for resilient robot pose estimation in subterranean environments," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 1024–1029.
- [6] 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)*. IEEE, 2020, pp. 5135–5142.
- [7] G. Huskić, S. Buck, and A. Zell, "Gerona: generic robot navigation: a modular framework for robot navigation and control," *Journal of Intelligent & Robotic Systems*, vol. 95, no. 2, pp. 419–442, 2019.

- [8] M. H. Shahna, P. Mustalahti, and J. Mattila, "Fault-tolerant control for system availability and continuous operation in heavy-duty wheeled mobile robots," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2025.
- [9] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki, and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Science Robotics*, vol. 9, no. 89, p. eadi9641, 2024.
- [10] M. H. Shahna, S. A. A. Kolagar, and J. Mattila, "Integrating deepRL with robust low-level control in robotic manipulators for non-repetitive reaching tasks," in *2024 IEEE International Conference on Mechatronics and Automation (ICMA)*. IEEE, 2024, pp. 329–336.
- [11] M. Carabantes, "Black-box artificial intelligence: an epistemological and critical analysis," *AI & Society*, vol. 35, no. 2, pp. 309–317, 2020.
- [12] International Organization for Standardization, "ISO/IEC TR 5469 - Artificial intelligence — Functional safety and AI systems," 2024.
- [13] Q. Shi, R. Wang, X. Li, and J. Yang, "Barrier lyapunov function-based adaptive optimized control for full-state and input-constrained dynamic positioning of marine vessels with simulation and model-scale tests," *Ocean Engineering*, vol. 301, p. 117534, 2024.
- [14] S. I. Han, J. Y. Cheong, and J. M. Lee, "Barrier lyapunov function-based sliding mode control for guaranteed tracking performance of robot manipulator," *Mathematical Problems in Engineering*, vol. 2013, no. 1, p. 978241, 2013.
- [15] M. H. Shahna, J.-P. Humaloja, and J. Mattila, "Model reference-based control with guaranteed predefined performance for uncertain strict-feedback systems," *Control Engineering Practice*, vol. 164, p. 106516, 2025.
- [16] M. H. Shahna, P. Mustalahti, and J. Mattila, "Robust torque-observed control with safe input–output constraints for hydraulic in-wheel drive systems in mobile robots," *Control Engineering Practice*, vol. 164, p. 106459, 2025.
- [17] J.-X. Zhang and G.-H. Yang, "Low-complexity tracking control of strict-feedback systems with unknown control directions," *IEEE Transactions on Automatic Control*, vol. 64, no. 12, pp. 5175–5182, 2019.
- [18] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, pp. 401–416, 2017.
- [19] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.
- [20] R. Krecht, C. Hajdu, and Á. Ballagi, "Possible control methods for skid-steer mobile robot platforms," in *2020 2nd IEEE International Conference on Gridding and Polytope Based Modelling and Control (GPMC)*. IEEE, 2020, pp. 31–34.
- [21] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," 1992.
- [22] Y. Huang and Y. Liu, "Practical tracking via adaptive event-triggered feedback for uncertain nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 64, no. 9, pp. 3920–3927, January 2019.
- [23] M. Corless and G. Leitmann, "Bounded controllers for robust exponential convergence," *Journal of Optimization Theory and Applications*, vol. 76, no. 1, pp. 1–12, 1993.
- [24] M. H. Shahna, M. Bahari, and J. Mattila, "Robustness-guaranteed observer-based control strategy with modularity for clean-tech emla-driven heavy-duty robotic manipulator," *IEEE Transactions on Automation Science and Engineering*, 2024.
- [25] —, "Robust decomposed system control for an electro-mechanical linear actuator mechanism under input constraints," *International Journal of Robust Nonlinear Control*, 2024.
- [26] B. Ren, S. S. Ge, K. P. Tee, and T. H. Lee, "Adaptive neural control for output feedback nonlinear systems using a barrier lyapunov function," *IEEE Transactions on Neural Networks*, vol. 21, no. 8, pp. 1339–1345, July 2010.
- [27] M. H. Shahna, P. Mustalahti, and J. Mattila, "Robust model-free control framework with safety constraints for a fully electric linear actuator system," in *2024 IEEE 21st International Power Electronics and Motion Control Conference (PEMC)*. IEEE, 2024, pp. 1–10.
- [28] M. H. Shahna and J. Mattila, "Exponential auto-tuning fault-tolerant control of n degrees-of-freedom manipulators subject to torque constraints," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024, pp. 7263–7270.
- [29] Z. Zhu and Z. Zhu, "Backstepping model-free adaptive control for a class of second-order nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 34, no. 5, pp. 3057–3073, 2024.
- [30] M. Ge, H. Xu, and Q. Song, "Prescribed-time control of four-wheeled independently driven skid-steering mobile robots with prescribed performance," *Nonlinear Dynamics*, vol. 111, no. 22, pp. 20991–21005, 2023.