sensors

MDPI

Article

Optimal Trajectory Planning for Wheeled Mobile Robots under Localization Uncertainty and Energy Efficiency Constraints

Xiaolong Zhang [1], Yu Huang [1], Youmin Rong [1], Gen Li [2], Hui Wang [1] and Chao Liu [1,*]

[1] School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; xiaolong172@hust.edu.cn (X.Z.); D201677192@hust.edu.cn (Y.H.); D201980282@hust.edu.cn (Y.R.); hustMrwang@hust.edu.cn (H.W.)

[2] Guangzhou Institute of Advanced Technology, Chinese Academy of Sciences, Guangzhou 511400, China; gen.li@giat.ac.cn

* Correspondence: yuanlcay@hust.edu.cn

Abstract: With the rapid development of robotics, wheeled mobile robots are widely used in smart factories to perform navigation tasks. In this paper, an optimal trajectory planning method based on an improved dolphin swarm algorithm is proposed to balance localization uncertainty and energy efficiency, such that a minimum total cost trajectory is obtained for wheeled mobile robots. Since environmental information has different effects on the robot localization process at different positions, a novel localizability measure method based on the likelihood function is presented to explicitly quantify the localization ability of the robot over a prior map. To generate the robot trajectory, we incorporate localizability and energy efficiency criteria into the parameterized trajectory as the cost function. In terms of trajectory optimization issues, an improved dolphin swarm algorithm is then proposed to generate better localization performance and more energy efficiency trajectories. It utilizes the proposed adaptive step strategy and learning strategy to minimize the cost function during the robot motions. Simulations are carried out in various autonomous navigation scenarios to validate the efficiency of the proposed trajectory planning method. Experiments are performed on the prototype "Forbot" four-wheel independently driven-steered mobile robot; the results demonstrate that the proposed method effectively improves energy efficiency while reducing localization errors along the generated trajectory.

Keywords: trajectory planning; localization uncertainty; energy efficiency; dolphin swarm optimization; swarm intelligence; wheeled mobile robot

check for updates

Citation: Zhang, X.; Huang, Y.; Rong, Y.; Li, G.; Wang, H.; Liu, C. Optimal Trajectory Planning for Wheeled Mobile Robots under Localization Uncertainty and Energy Efficiency Constraints. *Sensors* **2021**, *21*, 335. https://doi.org/10.3390/s21020335

Received: 23 November 2020
Accepted: 1 January 2021
Published: 6 January 2021

1. Introduction

In the intelligent manufacturing field, wheeled mobile robots are one of the most widely used groups of robots in factories and warehouses to perform material transportation tasks that benefit from their automation and efficiency [1]. As one of the key technologies of mobile robots, trajectory planning has recently attracted plenty of research. A series of trajectory planning schemes have been reported until now, such as the graph search-based method [2,3], interpolating curve planning method [4,5], sampling-based planning method [6], and numerical optimization method [7]. Among these methods, the interpolating curve planning method is a widely examined planning strategy due to its optimized performance and strong ability to handle external constraints. In [8], the authors propose a spline-based trajectory planning method to guarantee constraint satisfaction for autonomous guided vehicles. In [9], a minimum time trajectory planning method is proposed for a two-wheeled mobile robot by combining a straight line and a Bézier curve under the torque constraints. Although the interpolating curve planning method offers potential ways for trajectory generation, the trajectory optimization brings a heavy calculation burden owing to the nonlinearity, non-differentiability, and multi-objective requirements of the wheeled mobile robot system [10]. Swarm intelligence algorithms are proved to be a solution for this challenge [11]. In [12], the authors propose a novel chaotic grouping particle swarm optimization algorithm with a dynamic regrouping strategy to solve complex numerical optimization problems. In [13], a new evolutionary computation approach based on an artificial bee colony algorithm for solving the multi-objective orienteering problem is presented. In [14], the authors first propose a dolphin swarm algorithm (DSA) based on the biological characteristics and living habits of dolphins to solve the optimization problem with first-slow-then-fast convergence. The DSA simulates the predatory process of dolphins for multi-dimensional numerical optimization problems with four phases, i.e., search, call, reception, and predation. Many practical engineering optimization problems can be solved by DSA due to its high efficiency, low computational burden and excellent optimization performance [15,16]. In [16], the authors optimize the localization process by using DSA to process the sensed data in Wireless Sensor Networks. However, in the process of dolphin updating, the positions of the dolphins are updated based on a random method. Thus, the efficiency and convergence of the DSA decrease and the obtained solution may lead to low optimization accuracy or even failure [17]. Moreover, dolphins are explored completely based on fixed step size during the searching stage; this results in an inability to perform the searching efficiently.

Reliable localization is a fundamental requirement for mobile robots during autonomous navigation. Large localization errors or localization failure will lead to excessive tracking errors of mobile robots and even safety accidents. Nevertheless, most existing localization methods are flawed in terms of applicability to ambiguous environments, such as long corridors and empty areas [18]. This may result in localization uncertainty for mobile robots. The localization ability of the mobile robot, i.e., the localizability of the robot, is used to measure localization uncertainty in the environment. It varies over a given map since environmental information has different effects on robot localization at different positions [18,19]. In ambiguous environments, it is of importance to take localizability into consideration to avoid large localization errors on the planned trajectory. The early effort of this is presented in [20], where the authors first provide theoretical analysis for localization accuracy based on Cramér–Rao Bound. An ambiguity model of the indistinguishability is presented to formalize and quantify the perceptual ambiguity in the static environment [21]. In [22], this paper proposes a novel method for real-time localizability estimation by analyzing the constraints in each direction to predict localization performance. In [23], a minimum uncertainty planning method is proposed based on a partially observable Markov decision process (POMDP) with continuous observation spaces. In [24], the authors present an uncertainty-augmented Markov Decision Process to approximate POMDP to estimate the localization of the robot. However, POMDPs may not be computationally achievable owing to the expansion of the state space, although they are theoretically satisfied.

Energy efficiency is a crucial technology for wheeled mobile robots to run automatically for a long period. It is significant to optimize energy consumption due to the limited capacity of the equipped batteries. To extend the running time of the robot system, researchers have made many contributions to the energy efficiency of wheeled mobile robots. An energy optimal trajectory generation method is presented for a two-wheel differential mobile robot by constructing a cost function related to the integral of the Lagrangian [25]. As reported in [26], the authors propose a minimum-energy translational and rotational velocity trajectory planning for three-wheeled omnidirectional mobile robots by using Pontryagin's minimum principle. In [27], a closed-form trajectory planning method based on Dubins' path is presented to obtain the minimum energy trajectory for car-like robots. However, in the research studies mentioned above, very little attention has been devoted to building a generic energy model of different styles of wheeled mobile robots to provide a solid foundation for energy efficiency efforts.

Localizability and energy efficiency may be conflicting trajectories for optimization goals. Energy efficiency is mainly affected by speed, ground friction conditions, etc [28]. However, the localizability of the robot is chiefly influenced by the environmental map information. As a matter of fact, meeting energy efficiency may reduce the reliability of localization along the trajectory and vice versa. Therefore, how to balance energy efficiency and localizability poses a crucial challenge to trajectory planning, which motivates us to make a contribution in this paper.

The aim of this paper is to explore an effective trajectory planning method, considering energy efficiency and localizability simultaneously to bring the wheeled mobile robot from the start position to the goal position. The main contributions of this paper can be reflected in: (1) A novel localizability measure method is proposed based on the likelihood function to explicitly quantify the influence resulting from environmental information on the localization of mobile robots. (2) By utilizing the localizability measure method, the localizability aware map (LAM) is achieved such that the localization error at a given position can be effectively estimated. (3) An optimal trajectory planning method is proposed under localization uncertainty and energy efficiency constraints by incorporating an improved dolphin swarm algorithm (DSA) into the trajectory optimization process. The DSA utilizes the proposed adaptive step strategy and learning strategy to minimize the cost function during the robot motions. (4) Implemented on a developed prototype "Forbot" mobile robot, comprehensive experiments demonstrate that the proposed trajectory planning method guarantees the minimum total cost to balance localization uncertainty and energy efficiency while the robot navigates along the trajectory.

The rest of this paper is organized as follows. Section 2 explains how to establish the energy consumption model and describes the localizability measure method. In Section 3, the path planning with the energy and localizability criteria is developed. The proposed trajectory optimization based on the improved DSA is explained in detail in Section 4. Experimental results of the proposed trajectory planning method are shown and performances of the proposed method are analyzed in Section 5. Concluding remarks and future work are given in Section 6.

2. Energy Model and Localizability Measure

2.1. Energy Model

As shown in Figure 1, the common hardware architecture for mobile robots mainly consists of a perception module, decision module, control module, actuator module, and energy module. The energy module, equipped with batteries, and the battery management system provide electrical energy to other modules to drive the robot. The total energy consumption $E_t(t)$ of the wheeled mobile robot is derived as [28]:

$$E_t(t) = E_k(t) + E_a(t) + E_e(t), \tag{1}$$

where E_k, $E_a(t)$ and $E_e(t)$ are kinematic energy consumption, actuators' energy consumption and nonmechanical devices' energy consumption, respectively, which are defined as follows:

$$\left\{ \begin{array}{c} E_k(t) = \int (m\max(v(t)a(t),0) + I\max(\gamma(t)\delta(t),0))dt \\ E_a(t) = \int \left(\sum\limits_{i=0}^{N} \mu \frac{mg}{N} v_{wi} \right) dt \\ E_e(t) = P_e t \end{array} \right. , \tag{2}$$

where v and γ denote the x-direction and yaw velocities, respectively. m and I represent the mass and the rotational inertia of the mobile robot, separately. a and δ are the x-direction and yaw accelerations, respectively. μ denotes the coefficient of the rolling friction influenced by the ground types. v_{wi} denotes the linear velocity of the ith wheel. N is the number of wheels. g is the gravitational acceleration. P_e defines the total power of nonmechanical devices.

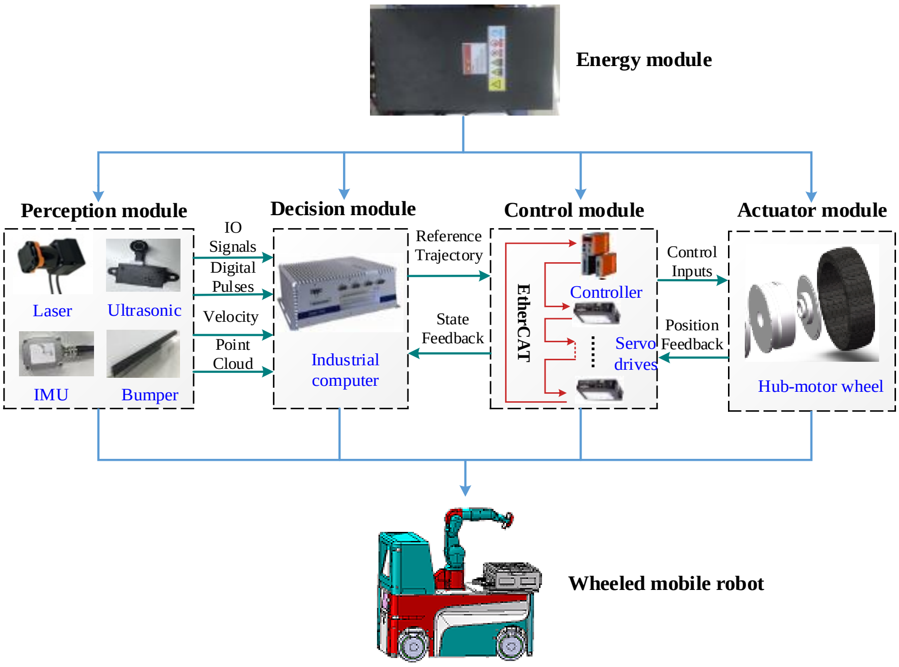

Figure 1. Hardware architecture.

2.2. Localizability Measure

In the field of robot navigation, mobile robots may inevitably run in ambiguous environments that include symmetrical or featureless map regions, resulting in the perceptual aliasing of external sensors. Therefore, the localization errors may accumulate in such regions, which brings a high risk of choosing a path without considering localization uncertainty [18]. Given that the perceptual aliasing region causes relatively strong localization confusion, it is necessary to avoid such regions as much as possible to ensure safety when the robot runs automatically on the planned path. Then, the probability of localization confusion (LCP) is proposed to estimate the probability of the specific event A, i.e., a robot's true pose is confused with some other poses over a grid map based on a given observation model. To be more mathematically precise, we provide the following definitions.

Definition 1. *Let $z_{\boldsymbol{x}_a}$ be the observation data captured by the robot-mounted sensors at the pose $\boldsymbol{x}_a$. Considering an input pose $\boldsymbol{x}_m$ as the independent variable, the likelihood function for sensor-based localization can be defined as [21]*

$$L(\boldsymbol{x}_m) \triangleq P(z_{\boldsymbol{x}_a}|\boldsymbol{x}_m). \tag{3}$$

According to (3), we can infer that if the arbitrary value of $L(\boldsymbol{x}_m)$ for a given $\boldsymbol{x}_a$ is smaller than $L(\boldsymbol{x}_a)$, the pose of the robot can be accurately obtained via the commonly used maximum likelihood estimation method. Unfortunately, $L(\boldsymbol{x}_m) \geq L(\boldsymbol{x}_a)$ may occur as a result of the mentioned ambiguous environments such that the maximum likelihood estimation method may confuse the pose $\boldsymbol{x}_m$ with the actual pose $\boldsymbol{x}_a$.

Definition 2. *Assume that ε is a small scalar used to compensate for the unmodeled factors; for the event $A: \ L(\boldsymbol{x}_m) + \varepsilon \geq L(\boldsymbol{x}_a)$, we define the LCP as follows*

$$P^{A}_{\boldsymbol{x}_m|\boldsymbol{x}_a} \triangleq P(L(\boldsymbol{x}_m) + \varepsilon \geq L(\boldsymbol{x}_a)), \tag{4}$$

where $\boldsymbol{x}_a$ and $\boldsymbol{x}_m$ are deterministic values, and the only random variable is $z_{\boldsymbol{x}_a}$ because of sensor measurement uncertainty. If the observation data $z_{\boldsymbol{x}_a}$ is known, the likelihood value of $L(\boldsymbol{x}_m)$ can be uniquely calculated. So, we can conclude that the LCP is dependent on the distribution of $z_{\boldsymbol{x}_a}$.

Then, the LCP defined by (4) can be rewritten based on Monte Carlo integration as follows:

$$P^A_{\boldsymbol{x}_m|\boldsymbol{x}_a} \approx \frac{1}{N_p}\sum_{i=1}^{N_p} I_A(z^i_{\boldsymbol{x}_a}), \tag{5}$$

where $z^i_{\boldsymbol{x}_a}$ represents the ith sample of the observation data. N_p is the number of all sampling.$I_A(z_{\boldsymbol{x}_a})$ denotes an indicator function, as depicted below [24]:

$$I_A(z_{\boldsymbol{x}_a}) = \begin{cases} 0 & L(\boldsymbol{x}_m) + \varepsilon < L(\boldsymbol{x}_a)|z_{\boldsymbol{x}_a} \\ 1 & L(\boldsymbol{x}_m) + \varepsilon \geq L(\boldsymbol{x}_a)|z_{\boldsymbol{x}_a} \end{cases}. \tag{6}$$

To quantitatively estimate the localization quality of poses over the map, we adopt LCP as the weight to compute the localizability evaluation value (LEV) as follows:

$$\mathrm{LEV}(\boldsymbol{x}) = \frac{\sum\limits_{\Delta\boldsymbol{x}_i\in\Omega} \Gamma(\boldsymbol{x}+\Delta\boldsymbol{x}_i)P^A_{\boldsymbol{x}+\Delta\boldsymbol{x}_i|\boldsymbol{x}}}{\sum\limits_{\Delta\boldsymbol{x}_i\in\Omega} P^A_{\boldsymbol{x}+\Delta\boldsymbol{x}_i|\boldsymbol{x}}}, \tag{7}$$

where $\Delta\boldsymbol{x}_i \triangleq (\Delta x_i, \Delta y_i, \Delta\theta_i)$ is the ith incremental pose relative to $\boldsymbol{x}$. The difference between $\boldsymbol{x}$ and $\boldsymbol{x}+\Delta\boldsymbol{x}_i$ determined by $\Gamma(\boldsymbol{x}+\Delta\boldsymbol{x}_i) = (\Delta x_i^2 + \Delta y_i^2)^{\frac{1}{2}} + \delta|\Delta\theta_i|$ with a coefficient δ used to transform the orientation difference into the distance difference. As shown in Figure 2, the center of the diagram represents the actual pose and we can find that the closer the color is to red, the smaller the Γ value at that pose. Ω denotes a set including all $\Delta\boldsymbol{x}_i$ as follows:

$$\Omega = \left\{\Delta\boldsymbol{x}_i|((\Delta x_i^2+\Delta y_i^2)^{\frac{1}{2}} \leq \Phi_d(\Omega)\&\Delta\theta = 0) \cap (|\Delta\theta_i| \leq \Phi_\theta(\Omega)\&(\Delta x_i^2+\Delta y_i^2)^{\frac{1}{2}} = 0)\right\} \tag{8}$$

where $\Phi_d(\Omega)$ and $\Phi_\theta(\Omega)$ are the distance range and the orientation range of the set Ω, respectively.

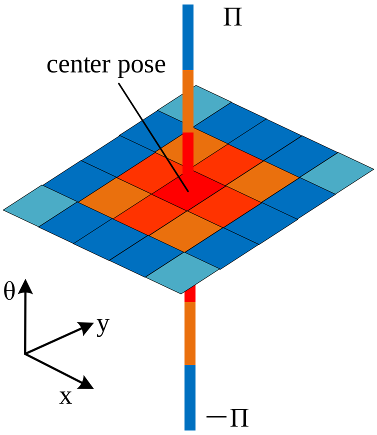

Figure 2. Schematic diagram of the set Ω.

$\mathrm{LEV}(\boldsymbol{x})$ is a criterion that reflects the expected localization performance based on the given observation model at the pose $\boldsymbol{x}$. According to (7), we can conclude that the $\mathrm{LEV}(\boldsymbol{x})$ represents the weighted average localization error resulting from the ambiguity between the actual pose $\boldsymbol{x}$ and the other poses from Ω. A larger $\mathrm{LEV}(\boldsymbol{x})$ implies a larger localization error caused by perceptual aliasing when a robot passes the pose $\boldsymbol{x}$.

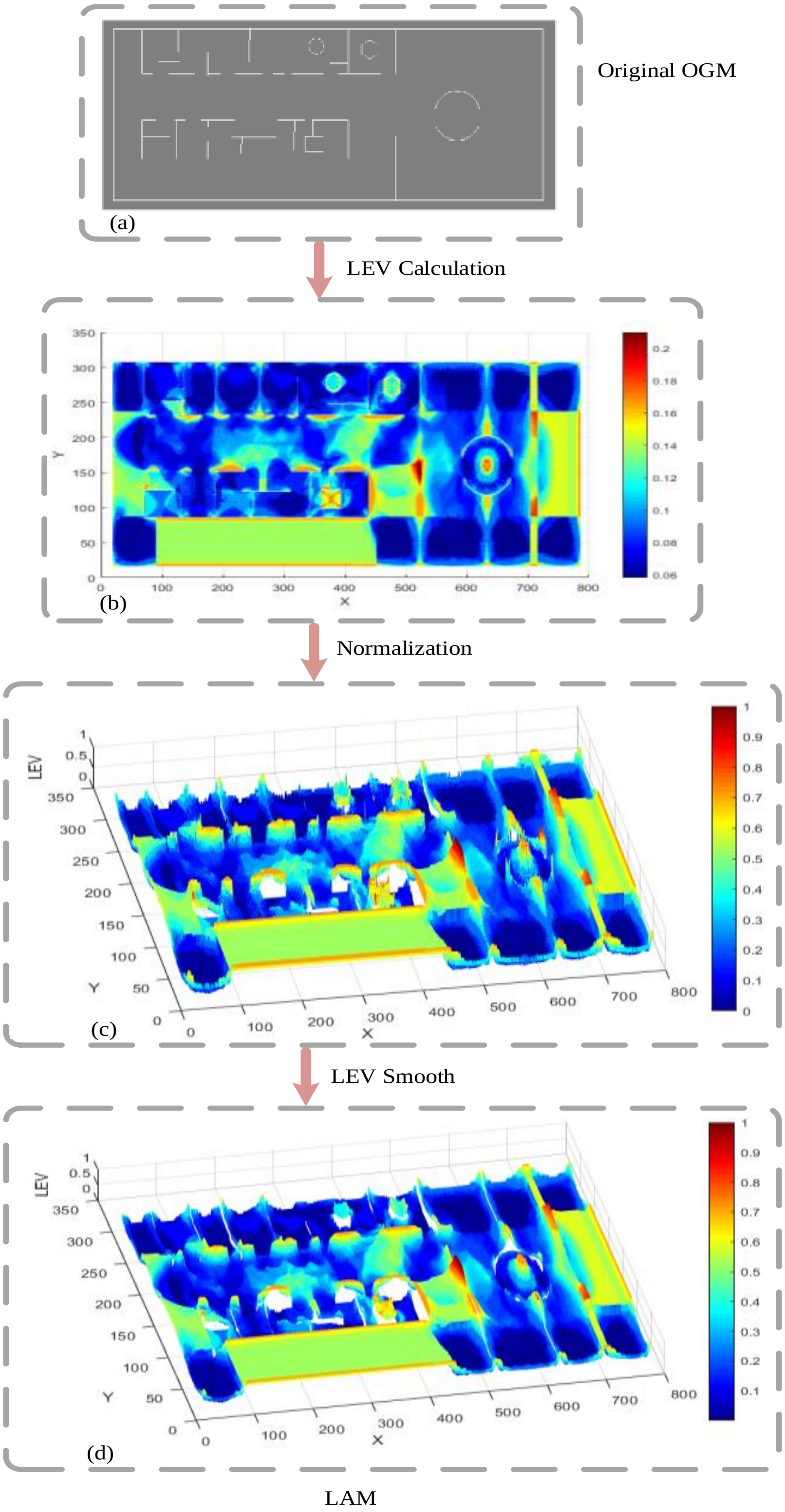

Figure 3. The construction process of the LAM. (**a**) Original OGM; (**b**) LEV calculation map; (**c**) LEV smooth map; (**d**) LAM.

2.3. *The Localizability-Aware Map Construction*

The LEV is achieved by the aforementioned localizability evaluation method so that the localization reliability can be evaluated, and the unambiguous areas over the occupancy grid map (OGM) can be identified. However, the calculated values of the LEV cannot be used directly for path planning, due to the large fluctuations of the LEV. To derive LAM, the process is explained in detail as listed below:

(1) *OGM Building:* The generation of the OGM is an increasingly mature technology in the field of robotics. In this paper, the simultaneous localization and mapping method presented in [29] is adopted to build the OGM of the environment.

(2) *LEV Calculation:* To calculate the LEV over the generated OGM by Equation (7), the specific calculation method of $P(z_{\boldsymbol{x}_a})$ needs to be determined.

As investigated in [30], sampling large amounts of data is time-consuming for a 2-D LIDAR in a large-scale environment and it is noted that the beam model complies with the physical measurement model of LIDAR. In this research, the beam model is adopted as the observation model to perform the calculation of $P(z_{\boldsymbol{x}_a})$. Then, we consider the parameters ε, $\Phi_d(\Omega)$ and $\Phi_\theta(\Omega)$. The δ specifies as $\delta = \tau_d/\tau_o$ where τ_d and τ_o are the maximum tolerable distance and orientation errors, respectively. Since $\mathrm{LEV}(\boldsymbol{x})$ is the weighted average localization error, it can be inferred that inequality $\mathrm{LEV}(\boldsymbol{x}) \leq \Phi_d(\Omega) + \delta\Phi_\theta(\Omega)$. By combining the inequality with (7), we have $\tau_d \leq \Phi_d(\Omega)$ and $\tau_o \leq \Phi_\theta(\Omega)$. To improve computing efficiency, we select $\Phi_d(\Omega) = \tau_d$ and $\Phi_\theta(\Omega) = \tau_o$. In this paper, the related parameters are set as $N_p = 20$, $\varepsilon = 0.5$, $\Phi_d(\Omega) = 0.25$ m, $\Phi_\theta(\Omega) = 5^\circ$. It is noted that we calculate the LEV by integrating over angles, i.e., $\mathrm{LEV}(\boldsymbol{x}) = \int \mathrm{LEV}(x, y, \theta)d_\theta$, since this paper does not use angle information.

(3) *Normalization:* To adjust values measured on different scales to a notionally common scale, the LEV is normalized between 0 and 1, representing lower and higher localizability, respectively.

(4) *LEV Smoothing:* If a robot enters the connected regions with fluctuating LEVs, the robot may fall into local oscillations during motion planning. In this step, the Gaussian Filter with a mask is used to smooth the values of LEV to suppress oscillations. After this step, the LAM graph is obtained to provide localizability constraints for path planning.

Figure 3 illustrates the construction process of the LAM graph. Figure 3a is the original OGM with 308 × 786 pixels used to represent the environment in the form of block grids; each grid is either occupied or unoccupied. By calculating the LEV, we have Figure 3b that shows the LEV of each grid (ranging from 0.06 to 0.21). Normalization and LEV smoothing are then performed. As shown in Figure 3d, the LEV is getting larger as the color changes from blue to red, that is, the localizability is getting worse.

3. Optimal Path with Localization Uncertainty and Energy Efficiency

In what follows, a heuristic search algorithm is examined with the novel localizability and energy related criterion to search for an optimal motion path by employing the aforementioned localizability and energy model constraint.

3.1. Heuristic Search Algorithm

A search algorithm that exploits a heuristic function at each node is often called a heuristic search algorithm; it has better computational complexity than brute-force algorithms. The A* algorithm, known to belong to this category of algorithms, has become a ubiquitous searching technology in path planning in the mobile robots field [31]. By searching all possible solutions, the A^* algorithm determines a least-cost path based on the specific cost index from the start node to the goal node in a grid map. Based on this property, a novel cost function considering important factors, such as energy consumption and localizability, is designed to obtain an optimal path for mobile robots.

As a heuristic algorithm, the cost function $f(\cdot)$, which is an estimate for the importance of the candidate node in the path, is one of the key elements of the A^* algorithm. In this report, we employ a grid map to describe the distribution of the obstacles, in which each grid that is either occupied by obstacles or is free is taken into account as a node n. By offering a search order, the function $f(\cdot)$ represents a cost of the path starting from the start node n_s to the goal node n_g, which includes the actual cost $g(n)$ and the heuristic cost $h(n)$, i.e.:

$$f(n) = g(n) + h(n) \tag{9}$$

worked out as follows:

$$g(n_i) = g(n_i - 1) + s_{n_i} \tag{10}$$

where s_{n_i} is the travel distance between the node $n_i - 1$ and the node n_i.

However, the shortest path may not mean minimum energy consumption or a low risk of failure to perform localization in some scenarios. To avoid large localization errors and large energy consumption, it is essential to consider the factors that have a great influence on the localization process and energy consumption when planning a path. Therefore, a novel cost function considering both localizability and energy efficiency will be presented to plan a path for mobile robots.

3.2. Path Planning

To satisfy the demands of considering localizability and energy efficiency simultaneously for the robot path planning, the actual cost model (10) is redesigned as:

$$g(n_i) = g(n_i - 1) + \alpha_1 C_l + \alpha_2 C_e, \tag{11}$$

where C_l and C_e denote the localization-related term and energy-related term, respectively. α_1 and α_2 are the coefficients to balance the weight between C_l and C_e, which are defined by:

$$C_l = \mathrm{LEV}(n_i), \tag{12}$$

$$C_e(n_i) = 2\eta(n_i)\mu_{n_i-1,n_i} mg \int |v_r(t)| dt = 2\eta(n_i)\mu_{n_i-1,n_i} mg s_{n_i-1,n_i}\ , \tag{13}$$

where μ_{n_i-1,n_i} denotes the friction parameter, s_{n_i-1,n_i} is the length of the path segment between the nodes $n_i - 1$ and n_i. $\eta(n_i)$ is a penalty factor to maintain a safe distance from surrounding obstacles and is expressed as:

$$\eta(n_i) = \begin{cases} 1, & d_o > D_s \\ \frac{D_s - D_m}{d_o - D_m}, & D_m < d_o \leq D_s \\ 1000, & d_o \leq D_m \end{cases}\ , \tag{14}$$

where d_o denotes the distance between the current node n_i and the nearest obstacle. D_s and D_m are the maximum and minimum safety distance, respectively. $d_o > D_s$ indicates that the current node is far away from obstacles. If $D_m < d_o \leq D_s$, the path cost increases with the increase in d_o. The current node is too close to the obstacle when $d_o \leq D_m$.

Next, we focus on the heuristic function $h(n_i)$. As investigated in [32], the heuristic function is admissible if it does not overestimate the cost of reaching the goal node from a particular node. The ideal situation is the heuristic function evaluating the accurate cost. However, in most of the robot planning problems, it is impractical to find such heuristic functions. Thus, the heuristic effectiveness and computational efficiency of heuristic search algorithms depend on the selection of the heuristics.

A new localizability-energy-related heuristic cost function is proposed as:

$$h(n_i) = \beta_1 \mathrm{LEV}(n_i) + \beta_2 \mu_{n_i-1,n_i} mg s_{n_i,n_g}, \tag{15}$$

where β_1 and β_2 are the weighted coefficients. s_{n_i,n_g} denotes the distance between node n_i and the goal node n_g.

In summary, we present the novel cost function $f(n_i)$ by combining (11) and (15) as follows:

$$f(n_i) = g(n_i - 1) + \alpha_1 C_l + \alpha_2 C_e + \beta_1 \mathrm{LEV}(n_i) + \beta_2 \mu_{n_i-1,n_i} mg s_{n_i,n_g}. \tag{16}$$

By using the cost function (16), an optimal collision-free path with localizability awareness and energy efficiency can be achieved.

4. Trajectory Optimization with Improved DSA

4.1. Parameterized Trajectory

The path generated by the heuristic search method is often a piecewise linear path or even a sharp path. The mobile robot needs to start, stop, and rotate frequently due to the discontinuity of the path, resulting in time delay, energy consumption, and unnecessary wear on the robot parts. To handle these problems, we employ a parameterized cubic Bézier curve to smooth the path, which benefits from the continuity and local controllability of the Bézier curve. Furthermore, the trajectory is further optimized by explicitly taking into account localizability and energy efficiency.

A series of Bézier curves are inserted between each segment defined by two consecutive waypoints and connected to obtain a complete smooth trajectory. As shown in Figure 4, the knee points along the generated path are chosen as waypoints. If two near waypoints are extremely close, they are merged into one waypoint. On the other hand, if a path segment is too long, new waypoints need to be inserted [33]. Thus, we can achieve a series of generated waypoints expressed by $P_{w0}, \cdots, P_{wi}, \cdots, P_{wN}$. Further, we define the angular bisector of the two near segments as the orientation of the corresponding waypoints.

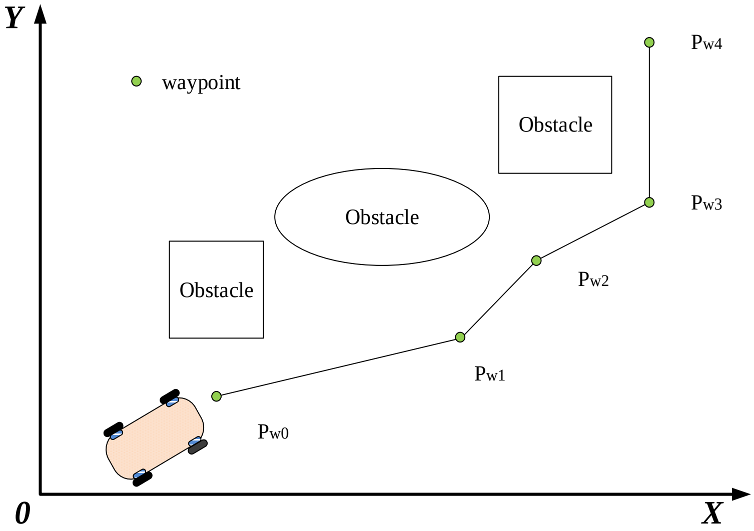

Figure 4. Selection of waypoints.

The cubic Bézier curve passes through the initial waypoint $P_{i-1}^{w} = [x_{i-1}^{w}, y_{i-1}^{w}, \theta_{i-1}^{w}]^{T}$ and finial waypoint $P_{i}^{w} = [x_{i}^{w}, y_{i}^{w}, \theta_{i}^{w}]^{T}$ of a path segment. Its sharpness can be reshaped by adjusting the two control points $P_{i1}^{c} = [x_{i1}^{c}, y_{i1}^{c}]^{T}$ and $P_{i2}^{c} = [x_{i2}^{c}, y_{i2}^{c}]^{T}$. $[x, y]$ and θ denote the position and orientation of the corresponding waypoint, respectively. Then, the parameterized trajectory of the cubic Bézier curve is designed as follows:

$$\begin{cases} x(\tau_i) = (1-\tau_i)^3 x_{i-1}^{w} + 3\tau_i(1-\tau_i)^2 x_{i1}^{c} + 3\tau_i^2(1-\tau_i) x_{i2}^{c} + \tau_i^3 x_i^{w} \\ y(\tau_i) = (1-\tau_i)^3 y_{i-1}^{w} + 3\tau_i(1-\tau_i)^2 y_{i1}^{c} + 3\tau_i^2(1-\tau_i) y_{i2}^{c} + \tau_i^3 y_i^{w} \end{cases}. \tag{17}$$

Subject to:

$$\begin{aligned} \frac{dx(\tau_i)}{dt}\Big|_{t=T_{i-1}} &= \frac{3(x_{i1}^{c} - x_{i-1}^{w})}{T_i - T_{i-1}} = v_{i-1}\cos\theta_{i-1}^{w} \\ \frac{dy(\tau_i)}{dt}\Big|_{t=T_{i-1}} &= \frac{3(y_{i1}^{c} - y_{i-1}^{w})}{T_i - T_{i-1}} = v_{i-1}\sin\theta_{i-1}^{w} \\ \frac{dx(\tau_i)}{dt}\Big|_{t=T_{i}} &= \frac{3(x_{i}^{w} - x_{i2}^{c})}{T_i - T_{i-1}} = v_{i}\cos\theta_{i}^{w} \\ \frac{dy(\tau_i)}{dt}\Big|_{t=T_{i}} &= \frac{3(y_{i}^{w} - y_{i2}^{c})}{T_i - T_{i-1}} = v_{i}\sin\theta_{i}^{w} \end{aligned} \tag{18}$$

where $x(\tau_i)$ and $y(\tau_i)$ are the trajectory with parameter $\tau_i = \frac{t-T_{i-1}}{T_i-T_{i-1}} \in [0,1], t \in [T_{i-1}, T_i]$ in the x and y direction, respectively. T_{i-1} and T_i are the arrival time at each waypoint for the segment $P^w_{i-1}P^w_i$. Then, when t varies from T_{i-1} to T_i, the trajectory varies from P^w_{i-1} to P^w_i. v_{i-1} and v_i are the velocities at waypoints P^w_{i-1} and P^w_i, respectively.

It is noted that the control points P^c_{i1} and P^c_{i2} are dependent on the arrival time T_i and the velocity v_i. According to (18), we can obtain the control points as follows:

$$\begin{aligned} x^c_{i1} &= x^w_{i-1} + \tfrac{1}{3}(T_i - T_{i-1})v_{i-1}\cos\theta^w_{i-1} \\ y^c_{i1} &= y^w_{i-1} + \tfrac{1}{3}(T_i - T_{i-1})v_{i-1}\sin\theta^w_{i-1} \\ x^c_{i2} &= x^w_i - \tfrac{1}{3}(T_i - T_{i-1})v_i\cos\theta^w_i \\ y^c_{i2} &= y^w_i - \tfrac{1}{3}(T_i - T_{i-1})v_i\sin\theta^w_i \end{aligned} \tag{19}$$

Consequently, by choosing suitable T_i and v_i, we will achieve an optimized trajectory such that the mobile robot can approach the final goal P_{wN} along all these waypoints.

4.2. Improved DSA

In this section, we present a trajectory smoothing method based on improved DSA by selecting parameters T_i and v_i while minimizing the localization error and minimizing energy consumption. The energy model (1) is calculated by combining the designed Bézier curve (17) as follows:

$$\begin{aligned} &E(T_i, v_i) = E_k + E_a + E_e \\ &= \int_0^{T_i}(m\max(v_i(\tau_i)a(\tau_i), 0) + I\max(\gamma(\tau_i)\delta(\tau_i), 0))dt + \int_0^{T_i}\sum_{j=0}^{N}\mu\frac{mg}{N}v_i(\tau_i)dt + P_eT_i \end{aligned} \tag{20}$$

Then, we aim to solve the following optimization problem:

$$\min J(T_i, v_i) = \min(\rho_1\int \mathrm{LEV}(p(\tau_i))d\tau_i + \rho_2E(T_i, v_i)) \tag{21}$$

where J is objective function. $p(\tau_i) = (x(\tau_i), y(\tau_i))$ denotes the point on the parameterized trajectory. ρ_1 and ρ_2 are the coefficients to balance the weight between and localizability and energy consumption.

To find the most appropriate T_i and v_i, we present an improved DSA to enhance optimization efficiency. The DSA simulates the predatory process of dolphins for multi-dimensional numerical optimization problems with four phases, i.e., search, call, reception, and predation. This algorithm has numerous advantages of few parameters and strong search capability, which makes it have better global search ability and better stability compared with the conventional evolutionary algorithms 14.

For the DSA, for each dolphin $\boldsymbol{Dol}_i(i = 1, 2\cdots, N_{dol})$, we define $\boldsymbol{L}_i$ as the individual optimal solution that $\boldsymbol{Dol}_i$ finds in a single time and define $\boldsymbol{K}_i$ as the neighborhood optimal solution. Then, we have three types of distances, i.e., $DD_{i,j} = \|\boldsymbol{Dol}_i - \boldsymbol{Dol}_j\|$, $DK_i = \|\boldsymbol{Dol}_i - \boldsymbol{K}_i\|$ and $DKL_i = \|\boldsymbol{L}_i - \boldsymbol{K}_i\|$.

In the search stage, a sound wave mechanism is employed between the individual dolphin $\boldsymbol{Dol}_i$ and a new child dolphin $\boldsymbol{X}_{ijt}$. More specifically, the sound is defined as Vd_j $(j = 1, \cdots, M)$, where M is the number of sounds. A maximum search time T_{sm} is set to prevent the DSA from falling into the search stage. Within the maximum search time T_{sm}, the dolphin $\boldsymbol{Dol}_i$ will search a new child solution $\boldsymbol{X}_{ijt}$ based on the sound wave during the propagation time t, namely:

$$\boldsymbol{X}_{ijt} = \boldsymbol{Dol}_i + Vd_jt \tag{22}$$

After calculating the fitness $Fit(\boldsymbol{X}_{ijt})$ of the new solution $\boldsymbol{X}_{ijt}$, we can obtain the optimal fitness as:

$$Fit_{iab} = \min_{j=1,2,\cdots,M;t=1,2,\cdots,T_{sm}} Fit(\boldsymbol{X}_{ijt}) \tag{23}$$

Then, the individual optimal solution $\boldsymbol{L}_i$ is specified as $\boldsymbol{L}_i = \boldsymbol{X}_{iab}$. If $Fit(\boldsymbol{L}_i) < Fit(\boldsymbol{K}_i)$, the neighborhood optimal solution $\boldsymbol{K}_i$ is replaced by $\boldsymbol{L}_i$.

In the call stage and reception stage, each dolphin informs other dolphins, through sound wave, as to whether a better solution is found and where it is located. The detailed process can be found in the literature [14].

In the predation stage, each dolphin updates its own location to hunt for preys within a certain surrounding radius R_2. Besides, the maximum range R_1 is set as $R_1 = T_{sm} \times \left\| Vd_j \right\|$. The update process is divided into three cases based the distance DK_i.

(1) If $DK_i \leq R_1$, a new dolphin $\boldsymbol{newDol}_i$ is derived as follows:

$$\boldsymbol{newDol}_i = \boldsymbol{K}_i + \frac{\boldsymbol{Dol}_i - \boldsymbol{K}_i}{DK_i} R_2 \tag{24}$$

where the surrounding radius $R_2 = \left(\frac{e_d - 2}{e_d} \right) DK_i, e_d > 2$.

(2) If $DK_i > R_1 \& DK_i \geq DKL_i$, $\boldsymbol{Dol}_i$ moves in a random way to obtain a new dolphin, as below:

$$\boldsymbol{newDol}_i = \boldsymbol{K}_i + \frac{\boldsymbol{Random}}{\|\boldsymbol{Random}\|} R_2 \tag{25}$$

where the surrounding radius $R_2 = \left(1 - \frac{DK_i Fit(\boldsymbol{L}_i) + (DK_i - DKL_i) Fit(\boldsymbol{K}_i)}{e_d DK_i Fit(\boldsymbol{L}_i)} \right) DK_i$.

(3) If $DK_i > R_1 \& DK_i < DKL_i$, we obtain a new dolphin $\boldsymbol{newDol}_i$ as follows:

$$\boldsymbol{newDol}_i = \boldsymbol{K}_i + \frac{\boldsymbol{Random}}{\|\boldsymbol{Random}\|} R_2 \tag{26}$$

where the surrounding radius $R_2 = \left(1 - \frac{DK_i Fit(\boldsymbol{L}_i) - (DKL_i - DK_i) Fit(\boldsymbol{K}_i)}{e_d DK_i Fit(\boldsymbol{L}_i)} \right) DK_i$.

If the iterative termination condition is fulfilled, the DSA ends. Otherwise, the DSA gets into the search stage again.

The standard DSA updates the position of each dolphin and expands child dolphins to search for the optimal solution. However, it should be pointed out that a new child dolphin is fully expanded with fixed step size during the searching stage, which leads to a fall in the local optimum for one clan of a dolphin group. Moreover, the positions of the dolphins are updated based on a random way in the process of dolphin updating. Therefore, the efficiency and convergence of the DSA decrease, and the obtained solution may result in low optimization accuracy or even failure.

To address these problems, we propose an adaptive step strategy and a novel learning strategy in the optimization process to balance the convergence speed and precision of the algorithm and better exchange information between dolphins.

(1) ***Adaptive step strategy***: we offer the following adaptive step parameter λ_d.

$$\lambda_d = \lambda_0 + \frac{w_1 Fit_{opt}}{w_2 \ln N_d + w_3 Fit_0} \tag{27}$$

where λ_0 is the minimum step. Fit_{opt} and Fit_0 are the current optimal fitness and the initial fitness, respectively. N_d denotes the current number of iterations. The factors $w_i, i = 1,2,3$ are the regulation parameters.

Then, the formula for searching the new child dolphin of the improved algorithm is as follows:

$$\boldsymbol{X}_{ijt} = \boldsymbol{Dol}_i + \lambda_d V d_j t \tag{28}$$

(2) ***Learning strategy:*** The searching efficiency may be enhanced if the child dolphin $\boldsymbol{X}_{ijt}$ learns from the information of the dolphin $\boldsymbol{L}_i$ since $\boldsymbol{L}_i$ is the individual optimal solution that $\boldsymbol{Dol}_i$ finds in a single time. Hence, the position of $\boldsymbol{X}_{ijt}$ can be obtained as:

$$Vd'_j =_{x1} \times Vd_j +_{x2} \times (\boldsymbol{L}_i - \boldsymbol{Dol}_i)/t \tag{29}$$

$$\boldsymbol{X}_{ijt} = \boldsymbol{Dol}_i + \lambda_d Vd'_j t \tag{30}$$

where $_1$ and $_2$ are the impact factors.

Next, the dolphin $\boldsymbol{Dol}_i$ will learn from the individual optimal solution $\boldsymbol{L}_i$. The dolphin $\boldsymbol{Dol}_i$ updates itself according to information of the dolphin $\boldsymbol{L}_i$ as follows:

$$\boldsymbol{newDol}_i = \boldsymbol{K}_i +_{d1} \frac{\boldsymbol{Dol}_i - \boldsymbol{K}_i}{DK_i} R_2 +_{d2} \frac{\boldsymbol{L}_i - \boldsymbol{K}_i}{DK_i} R_2, \text{ if } DK_i \leq R_1 \tag{31}$$

$$\boldsymbol{newDol}_i = \boldsymbol{K}_i +_{d1} \frac{\boldsymbol{Random}}{\|\boldsymbol{Random}\|} R_2 +_{d2} \frac{\boldsymbol{L}_i - \boldsymbol{K}_i}{DK_i} R_2, \text{ if } DK_i > R_1 \tag{32}$$

The improved DSA related to our optimization problem (21) solution is depicted in Algorithm 1. In this way, the optimal parameters can be obtained such that the mobile robot can reach the goal with energy efficiency and localizability simultaneously.

In the end, the optimal trajectory can thus be obtained through the use of this algorithm by iteration for every segment trajectory.

Algorithm 1: Improved DSA

```
Input: the objective function $J$
Output: the parameters $T_i$ and $v_i$
  // Initialization
  Initialize randomly a poption of $N_{dol}$ dolphin swarm
  $\boldsymbol{Dol}$= $\{\boldsymbol{Dol}_1, \boldsymbol{Dol}_2, \cdots, \boldsymbol{Dol}_{N_{dol}}\}$
  Define $DD_{i,j}, DK_i, DKL_i$ as the distance bewteen two corresponding dolphins
  and $TS_{i,j}$ is the rest time for the sound of moving from $\boldsymbol{Dol}_i$ to $\boldsymbol{Dol}_j$.
  Specify the related parameters : $\rho_1, \rho_2, \lambda_0, w_1, w_2, w_3, _{x1}, _{x2}, _{d2}, _{d2}$
  while the termination condition is not satisfied do
  // the search stage
    Search new child dolphin $\boldsymbol{X}_{ijt} = \boldsymbol{Dol}_i + \lambda_d Vd_j t$
    Calculate optimal fitness, $Fit_{iab} = \min_{j=1,2,\cdots,M;t=1,2,\cdots,T_{sm}} Fit(\boldsymbol{X}_{ijt})$
    Individual optimal solution $\boldsymbol{L}_i = \boldsymbol{X}_{iab}$
    if $Fit(\boldsymbol{L}_i) < Fit(\boldsymbol{K}_i)$ then
    Neighborhood optimal solution $\boldsymbol{K}_i = \boldsymbol{L}_i$
    end if
  // the call stage
    Update the rest time $TS_{i,j}$
  // the reception stage
    Exchange the information between dolphins
  // the predation stage
    Calculate $DD_{i,j}$, $DK_i$ and $DKL_i$
    Determine search Radius $R_2$
    if $DK_i \leq R_1$ then
   $\boldsymbol{newDol}_i = \boldsymbol{K}_i + o_{d1} \frac{\boldsymbol{Dol} - \boldsymbol{K}_i}{DK_i} R_2 + o_{d2} \frac{\boldsymbol{L}_i - \boldsymbol{K}_i}{DK_i} R_2$
    else if $DK_i > R_1 \& DK_i \geq DKL_i$ then
   $\boldsymbol{newDol}_i = \boldsymbol{K}_i + o_{d1} \frac{\boldsymbol{Random}}{\|\boldsymbol{Random}\|} R_2 + o_{d2} \frac{\boldsymbol{L}_i - \boldsymbol{K}_i}{DK_i} R_2$
    else if $DK_i > R_1 \& DK_i < DKL_i$ then
    $\boldsymbol{newDol}_i = \boldsymbol{K}_i +_{d1} \frac{\boldsymbol{Random}}{\|\boldsymbol{Random}\|} R_2 +_{d2} \frac{\boldsymbol{L}_i - \boldsymbol{K}_i}{DK_i} R_2$
   end if
   Calculates $newDol_i$ fitness
   Update optimal solution $\boldsymbol{K}_i$
  end while
```

5. Simulation and Experiment Results

5.1. Experimental Setup and Experimental Identification of the Energy Model

To test the effectiveness of the proposed optimal trajectory planning method, a four-wheel independently driven-steered mobile robot is employed in the experiments. The developed prototype robot, defined as Forbot, is shown in Figure 5. The Forbot installs two LIDARs diagonally with a 360-degree viewing angle. The Forbot has some prominent features, such as automatic charging, anti-crash measures, trackless autonomous navigation, and vision-based operating of the work-piece. The wheels with hub motors of the Forbot realize continuous wheel-ground contact so that the robot can move on an uneven floor with impurities. As each wheel of the robot has two degrees of freedom to roll and turn actively, the Forbot is able to achieve the diagonal move steer mode with a specific kinematic model [34]. The mass and inertia of the robot are 1000 kg and 60 kg·m^2, respectively, and its battery capacity is 120 Ah. The maximum speed, maximum acceleration and minimum acceleration are 1 m/s, 0.1 m/s^2 and -0.1 m/s^2, respectively. More specifications are shown in Table 1.

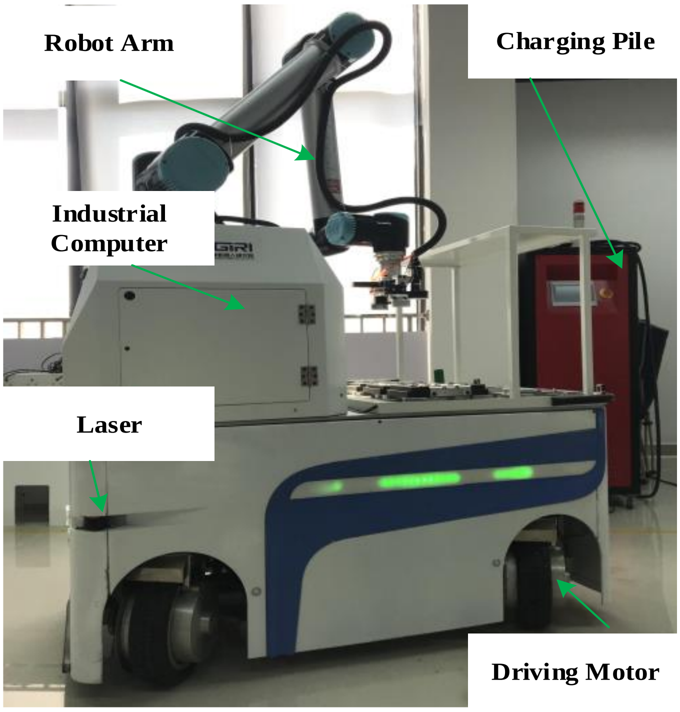

Figure 5. The prototype of the developed Forbot.

Table 1. Parameters of the Forbot.

Description	Quantity	Description	Quantity
Robot Length	0.89 m	Incremental Encoder	2500 pulses/turn
Robot Width	0.53 m	Main Frequency of PC	2.2 GHz
Wheel Diameter	0.28 m	RAM of PC	8 G

The power parameter P_e in the energy model (1) is determined when the robot is still. In this situation, the current is stable, and we have $P_e = 336$ w. By performing friction coefficient experiments [35], we get $\mu = 0.048$ for the tile surface, and $\mu = 0.086$ for the carpet surface.

To examine the validity of the energy model (1), the Forbot moves according to the designed velocity profile presented in Figure 6. We compare the experimental results and modeling results that are given by Figure 7. It can be concluded that the experimental results verify the correctness of the presented energy model.

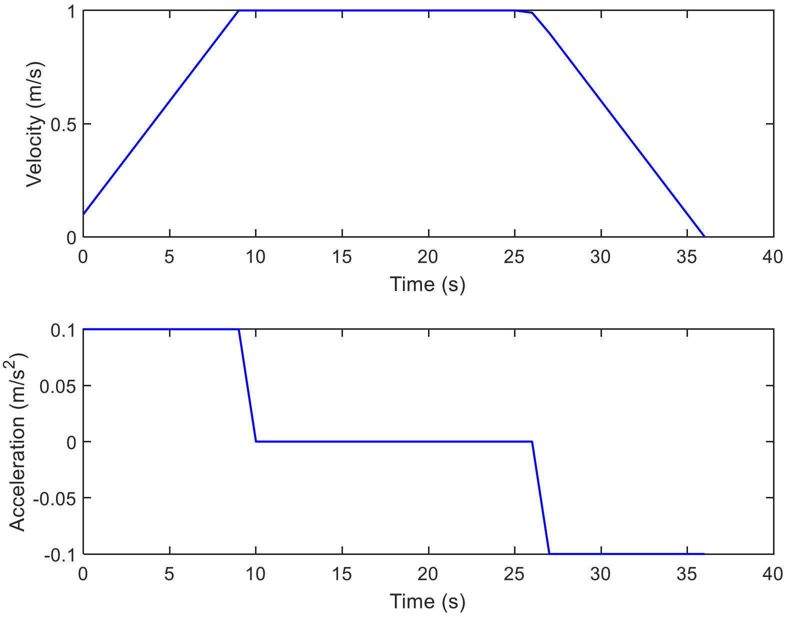

Figure 6. Motor's input current with respect to the velocity.

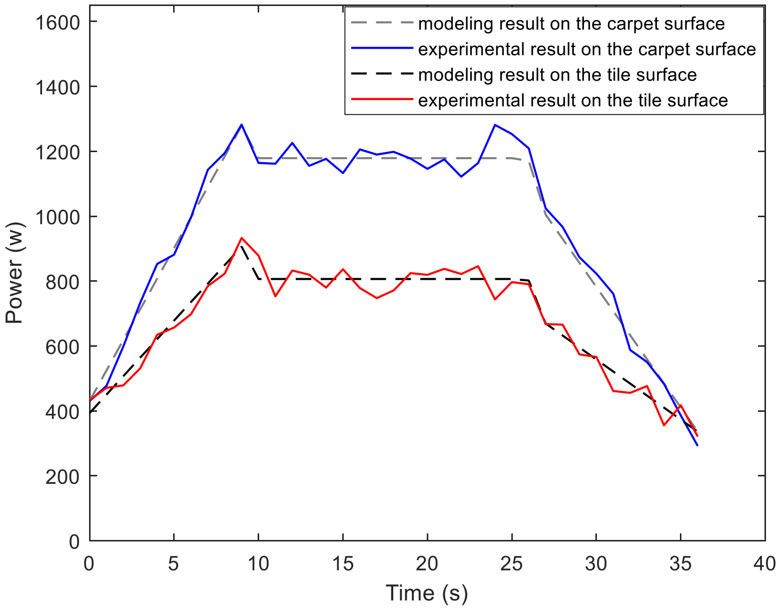

Figure 7. Energy model verification experiment.

5.2. Simulation Results

In this section, we first illustrate the advantages of the improved DSA for trajectory optimization. After that, the trajectory planning results in the long corridor region and circle region, with two ground surface types, are demonstrated to verify the trajectory planning performance.

(Case 1) *Improved DSA:* In this case, we will validate the advantages of the improved DSA through two experiments based on different individuals, i.e., 10 individuals and 100 individuals. Regarding the computational efficiency, we compare the classic swarm intelligence algorithms, such as the standard PSO algorithm [12], standard ABC algorithm [13], and the traditional DSA. The comparison experiments are carried out on the computer with 2.2 GHz CPU and 8 GB RAM. The initializations of dolphins are distributed randomly and evenly. The related parameters are set as $\lambda_0 = 1$, $\rho_{i=1,2} = [0.2, 0.8]$, $w_{i=1,2,3} = [0.27, 0.28, 0.45]$, $_{di=1,2} = [0.75, 0.25]$, $_{xi=1,2} = [0.75, 0.25]$, which are decided through experiments to find optimal values. To ensure fairness and show consistency, the experimental results are achieved by performing each experiment 20 times independently, and the algorithm returns when the convergence condition meets the convergence accuracy or the maximum iterations.

Figures 8 and 9 show the running iterations based on the compared algorithms for 10 individuals and 100 individuals, respectively. As shown in Figure 8, the iteration number for trajectory optimization is prominently reduced under the improved DSA algorithm. The improved algorithm can obtain the optimal results within 29 repeated trails while the traditional one requires more than 51 iterations searching for the optimal solutions. The reason is that we present an adaptive step strategy and learning strategy in the optimization process to drive the current individual to approach the best solution. In terms of Figure 9, the ABC algorithm obtains the fastest convergence rate in the initial loops while DSA is the slowest one. Gradually, the improved DSA becomes the fastest one when the iteration reaches 13. The is because the information exchanges between dolphins result in time-delay, which is proportional to the number of dolphins and the distance between dolphins.

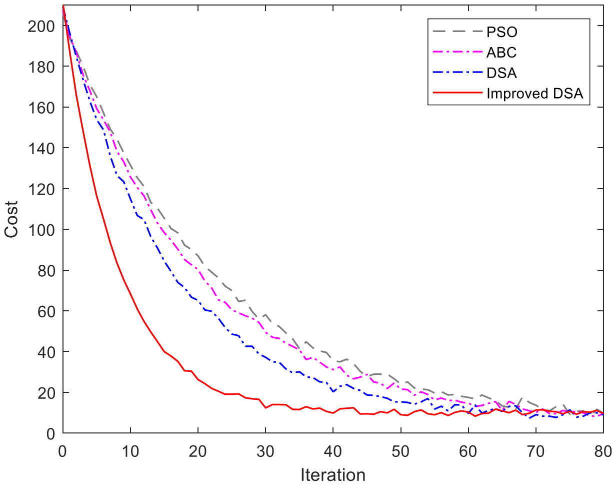

Figure 8. Evaluation of improved DSA algorithm under 10 individuals.

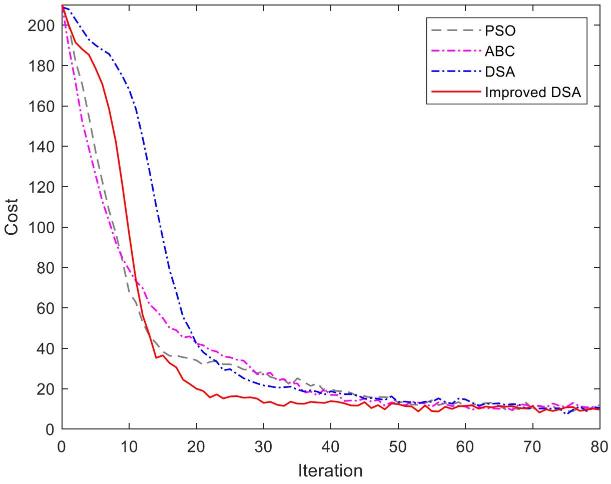

Figure 9. Evaluation of improved DSA algorithm under 100 individuals.

The statistics index of the computational performance is given in Tables 2 and 3. It can be seen that our improved DSA algorithm performs better in our trajectory planning problem and has significant advantages in terms of iteration times over the other three algorithms. Moreover, the Wilcoxon's rank-sum tests at the 5% significance level are carried out to verify the significance of the improved dolphin swarm algorithm. The Wilcoxon test results are shown in Table 4. In Table 4, "–", "0" and "1" represent the best result, significantly different and not significantly different from the best one, respectively. The results show that our improved DSA algorithm outperform other three algorithms for the trajectory planning problem.

Table 2. Simulation results in case (1): 10 individuals.

Criteria	PSO	ABC	Standard DSA	Improved DSA
Iteration times	80	69	51	29
Mean	2.3776	3.998	0.8659	0.5231
SD	0.985	1.709	0.4022	0.1988

Table 3. Simulation results in case (1): 100 individuals.

Criteria	PSO	ABC	Standard DSA	Improved DSA
Iteration times	63	52	46	22
Mean	0.9524	1.2462	0.4982	0.3013
SD	0.4309	0.5758	0.2097	0.1258

Table 4. Results of Wilcoxon's tests of the four algorithms.

Simulation	PSO	ABC	Standard DSA	Improved DSA
10 individuals	1	1	1	–
100 individuals	1	1	0	–

(Case 2) *Trajectory planning in a long corridor region:* This case considers the trajectory planning performance in a long corridor region with the two different ground surfaces shown in Figures 10–12. Simulations on a four-wheel independently driven-steered mobile robot are carried out. For this simulation, the corresponding parameters are chosen as $D_s = 1.2$ m, $D_m = 0.25$ m, $\beta_1 = 0.5$, $\beta_2 = 1.2$. To balance the weighting coefficients, α_1 α_2 β_1 and β_2 are determined by $[0.2, 0.8, 0.2, 0.8]$. The start point and the goal point are located at [48, 104] and [500, 75], respectively.

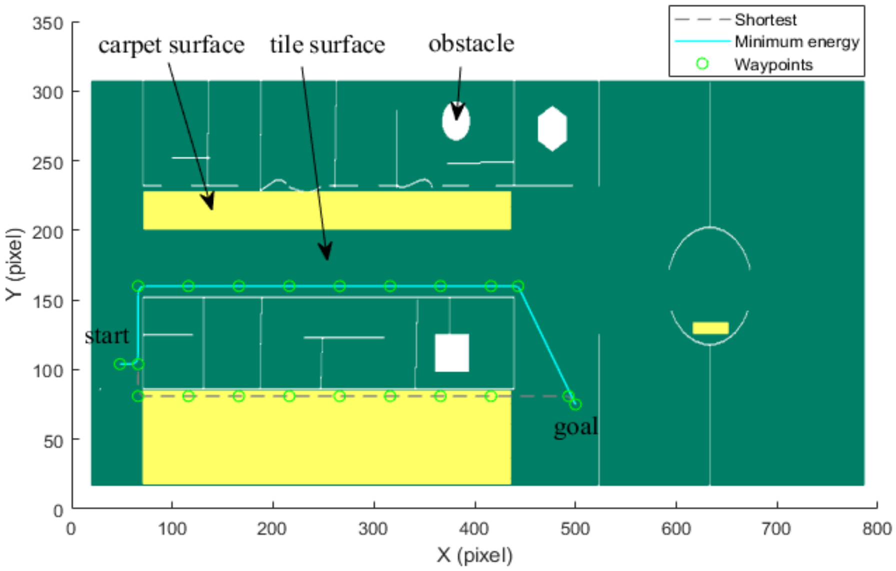

Figure 10. Optimal trajectory based on minimum energy in the long corridor region.

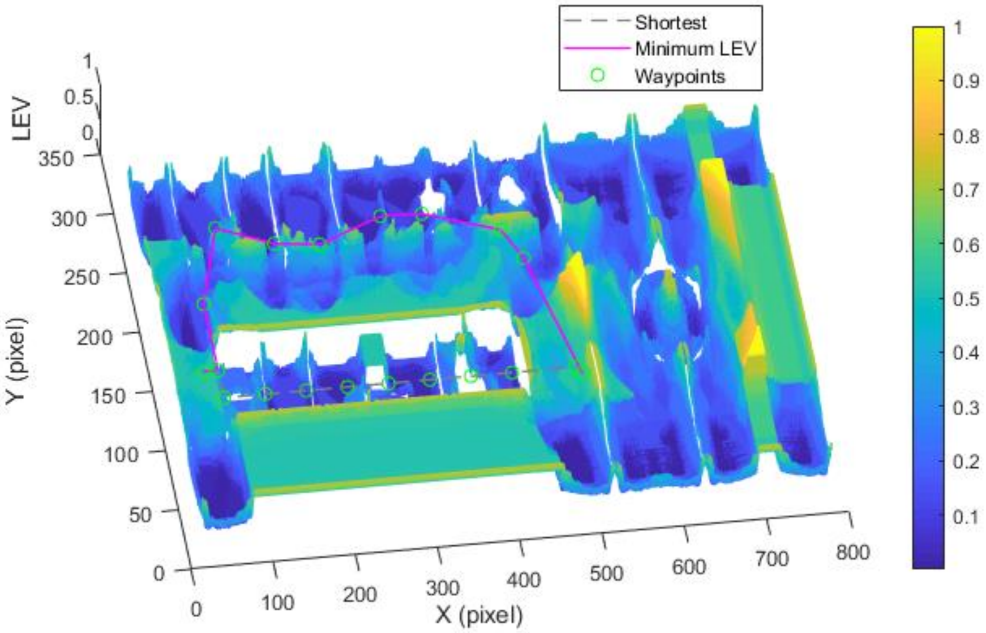

Figure 11. Optimal trajectory based on minimum localization error in the long corridor region.

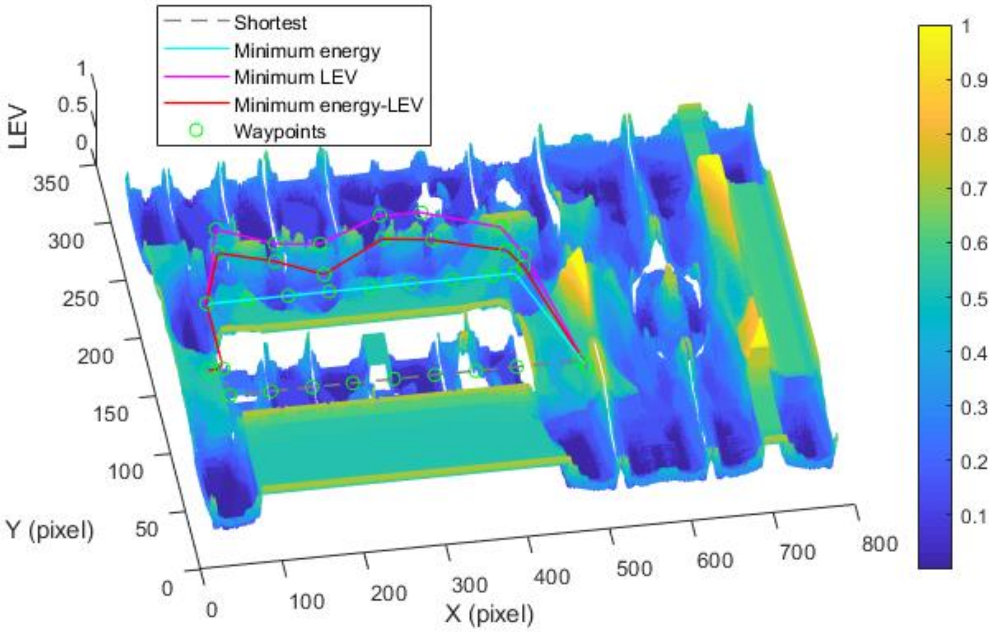

Figure 12. Comparison of trajectories of the case (2).

By utilizing our trajectory planning method, the trajectory is generated considering localizability and energy efficiency. Figure 10 illustrates the generated trajectory only based on minimum energy consumption. As can be seen from the result, the mobile robot travels on the smooth tile surface to avoid consuming too much energy to overcome friction. Although the resulting trajectory is longer, it consumes less energy throughout the trajectory. Figure 11 shows the generated trajectory in the resulting LAM graph only based on minimum localization error. In this case, rather than moving the shortest trajectory that has the regions with high LEV, the generated trajectory is selected to be closer to featured structure regions with low LEV. As shown in Figure 12, an optimal trajectory is then generated considering localizability and energy efficiency simultaneously. Figures 13 and 14 present the power and the LEV of each trajectory, respectively. Compared with other trajectories, the trajectory generated by the proposed trajectory planning method chooses the minimum total cost to balance the localization error and energy efficiency.

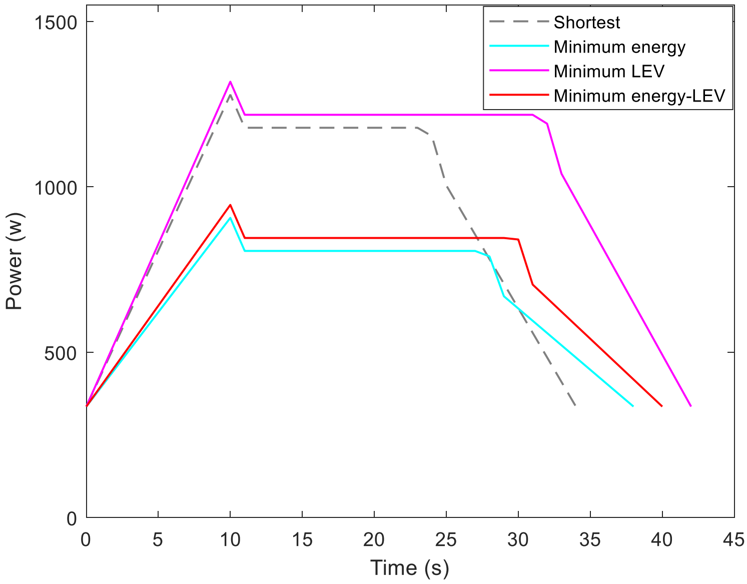

Figure 13. Motional power in the experiments.

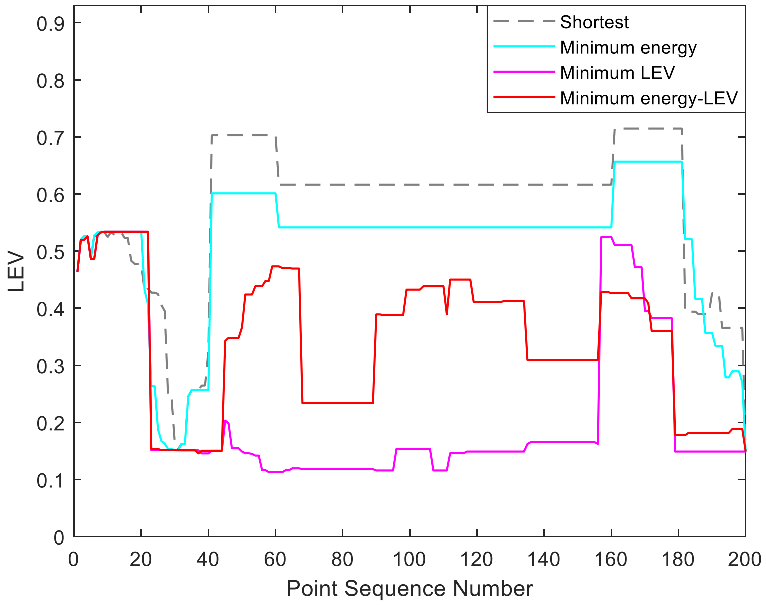

Figure 14. The LEV in the experiments.

Table 5. Simulation results in case (2).

Trajectory	**Energy (kJ)**	**LEV**	**Travel Distance (m)**	**Total Cost**
Shortest	32.06	114.03	23.58	48.45
Minimum energy	26.36	101.83	27.37	41.45
Minimum LEV	42.75	43.91	31.33	42.98
Minimum energy-LEV	29.16	68.34	29.82	36.99

For quantitative analysis, energy consumption, localization error, travel distance, and total cost achieved by using the presented methods are given in Table 5. As we can see from the table, it is concluded that the proposed trajectory planning method guarantees optimal performance in a comprehensive way. To be more specific, the minimal energy-LEV trajectory has the lowest total cost of 36.99, which is enhanced by 23.65% in comparison with the shortest trajectory.

(Case 3) *Trajectory planning in a circle region:* In this case, we consider the trajectory planning performance in a circle region. As depicted in Figure 15, the experimental environment and the corresponding parameters are the same as the case (2). The difference is that the start point (550, 158) and goal point (750, 158) are located in the circle region.

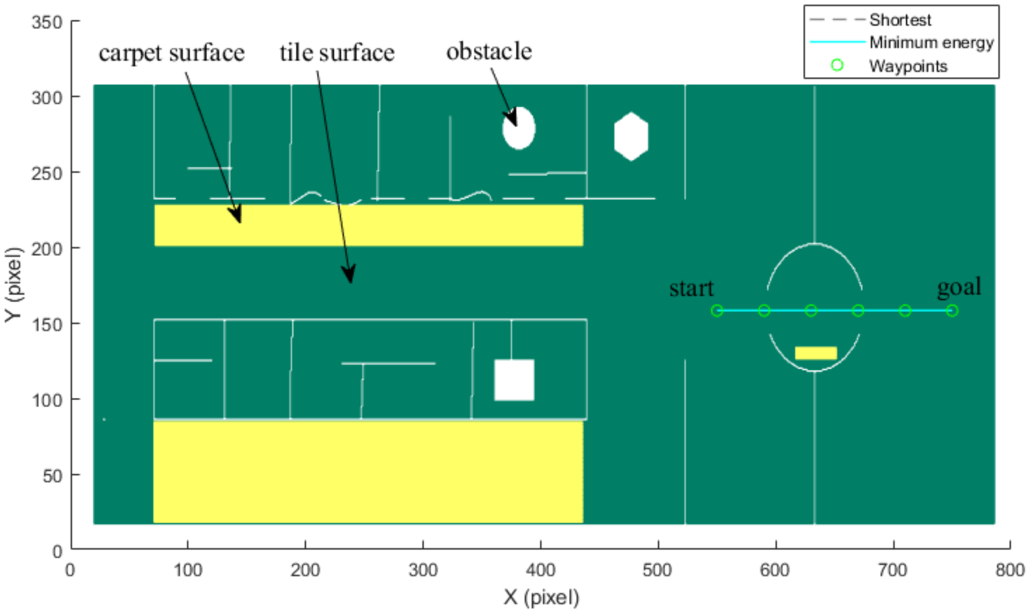

Figure 15. Optimal trajectory based on minimum energy in the circle region.

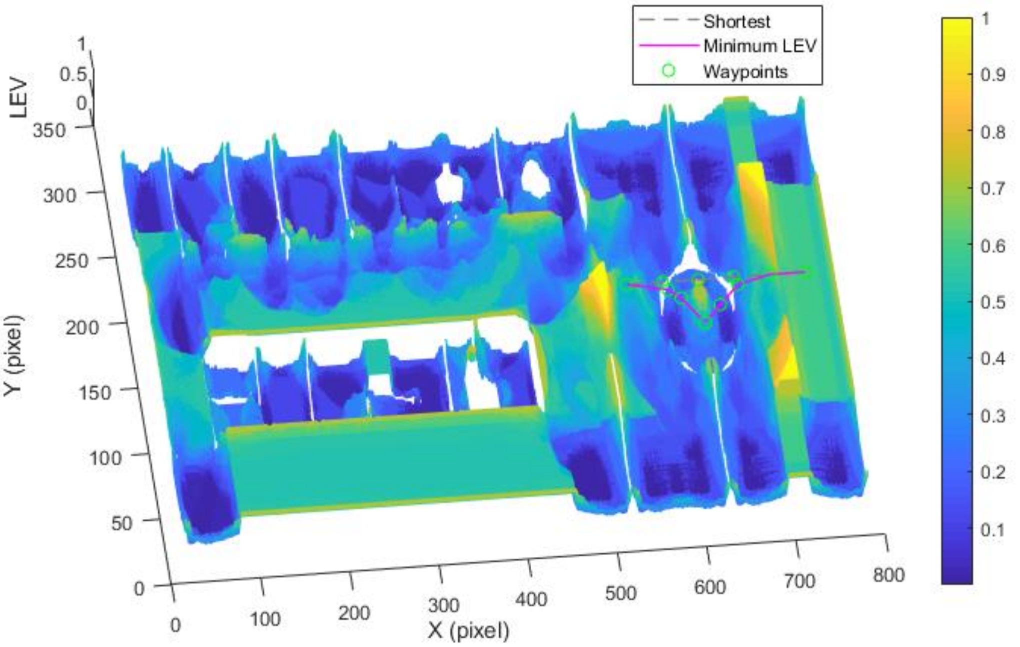

Figure 16. Optimal trajectory based on minimum localization error in the circle region.

Figure 15 shows the generated trajectory only considering the minimum energy consumption. This trajectory is the same as the shortest trajectory since both trajectories are on the tile surface. Figure 16 presents the generated trajectory in the resulting LAM graph only based on the minimum localization error. As we can see from the result, at the center of the circle, the LEV is very high relative to other regions nearby due to the same sensor reading regardless of the orientation. To stay away from the high LEV region, the generated trajectory is around the circumference to avoid large localization errors. In Figure 17, we can see all the resulting trajectories of this case, one of which is the optimal trajectory that is simultaneously based on energy efficiency and localizability. Figures 18 and 19 show the power and the LEV of the resulting trajectories, respectively.

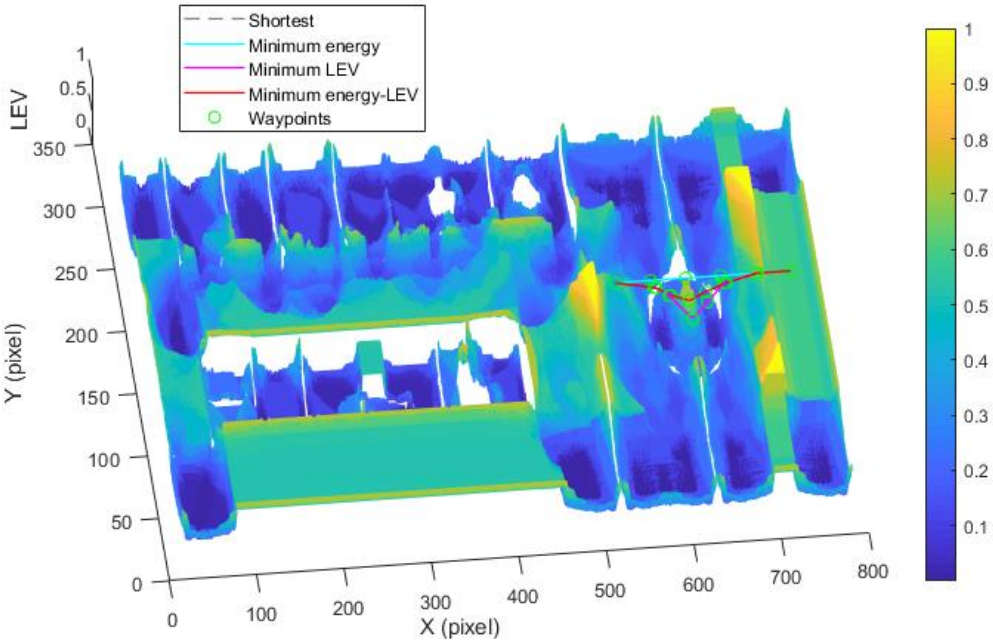

Figure 17. Comparison of trajectories of the case (3).

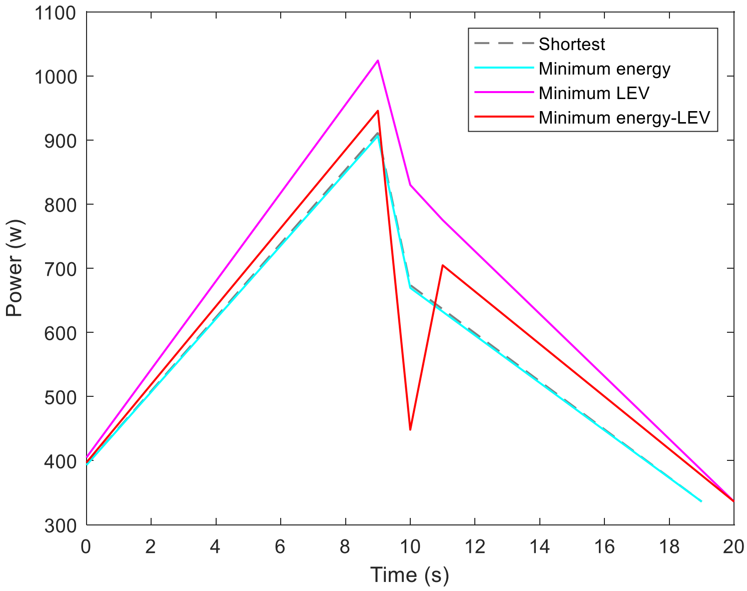

Figure 18. Motional power in the experiments.

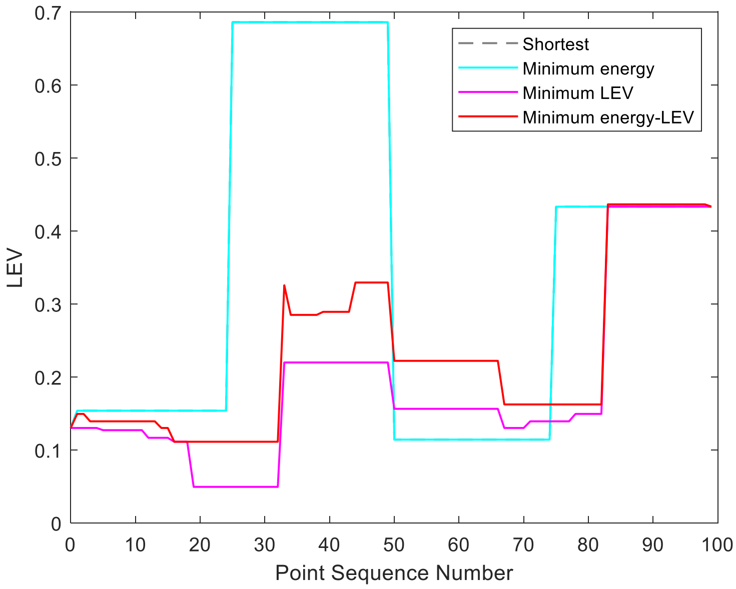

Figure 19. The LEV in the experiments.

By examining Table 6, which presents the energy consumption, localization error, travel distance, and total cost, it can be seen that the proposed trajectory planning method achieves an optimal trajectory to maintain an acceptable level of energy consumption as well as to decrease the localization errors throughout the trajectory. Particularly, the minimal energy-LEV trajectory saves 14.93% energy consumption compared to the minimal LEV trajectory, and is 33.41% lower than the minimum energy trajectory in terms of the localization error.

Table 6. Simulation results in case (3).

Trajectory	Energy (kJ)	LEV	Travel Distance (m)	Total Cost
Shortest	11.57	34.66	10.00	16.19
Minimum energy	11.53	34.66	10.00	16.16
Minimum LEV	14.53	19.04	10.84	15.43
Minimum energy-LEV	12.36	23.08	10.22	14.50

5.3. Experiment Results

In this experiment, we verify the trajectory planning performance in an experimental environment with 799 × 985 grids, and each grid is a square with 50 mm edges. Boxes are placed into the environment, forming rich structural features, and ground surface types in the environment contain tile and carpet. The localization method proposed in [36] is applied to the Forbot robot. The sampling frequency of the control loop is specified as 10 Hz.

Figure 20 shows that the Forbot approaches the goal point along the minimum energy trajectory. The dotted line represents the shortest trajectory that is mostly on the carpet surface. When the Forbot moves along the shortest trajectory, higher frictional resistance leads to more energy consumption. In contrast, although the minimum energy trajectory is longer in this case, it is found to consume less energy.

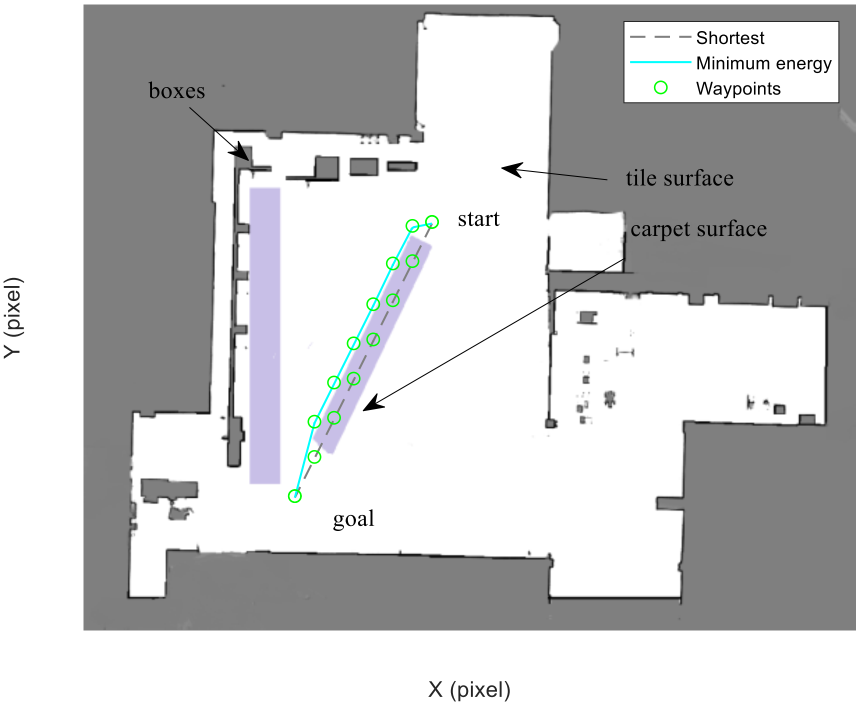

Figure 20. Optimal trajectory based on minimum energy.

As shown in Figure 21, four resulting trajectories, i.e., shortest, minimum energy, minimum LEV, and minimum energy-LEV, are generated in the LAM graph. The waypoints (green) are drawn along the trajectory in Figure 21. Figures 22 and 23 present the power and the LEV profile of the experiment, respectively. As can be seen intuitively, instead of taking the shortest trajectory, the minimum LEV trajectory is much closer to featured structure regions where the localization error is relatively low. The minimum energy-LEV trajectory has a tendency to balance the energy consumption and localization error as the robot moves toward the goal.

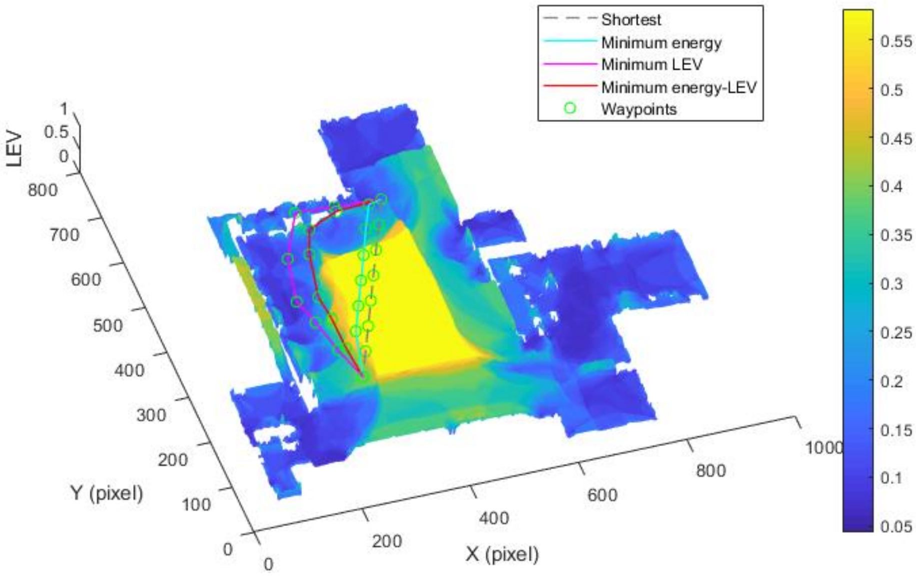

Figure 21. Comparison of trajectories of the case (2).

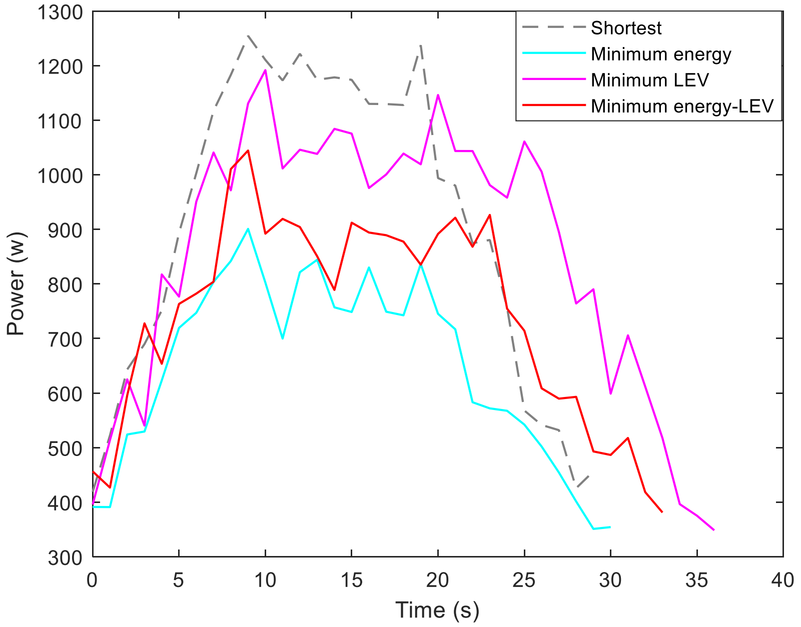

Figure 22. Motional power in the experiments.

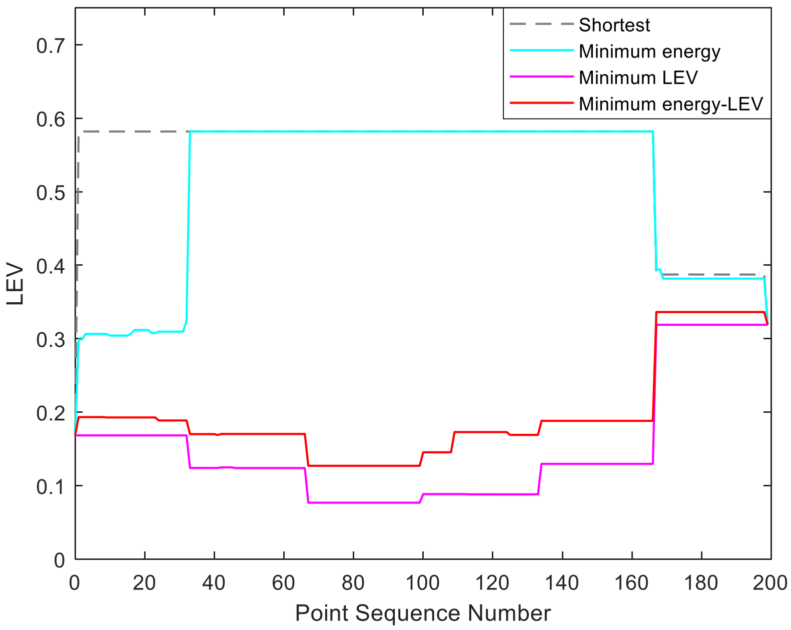

Figure 23. The LEV in the experiments.

To make the comparison more clearly and directly, we provide the energy consumption, localization error, travel distance, and total cost of the experiment in Table 7. As can be seen from the table, the proposed trajectory planning method obtains a comprehensively-optimized trajectory with guaranteed minimum total cost. More specifically, the minimal energy-LEV trajectory consumes 19.58% less energy than the minimal LEV trajectory, and reduces the localization error by 61.03% compared to the minimal energy trajectory, and ends up 36.78% more than the shortest trajectory in terms of the total cost.

Table 7. Experimental results of trajectory planning.

Trajectory	Energy (kJ)	LEV	Travel Distance (m)	Total Cost
Shortest	27.83	109.46	19.57	44.16
Minimum energy	20.30	100.54	20.15	36.35
Minimum LEV	31.21	30.13	26.83	30.99
Minimum energy-LEV	25.10	39.18	23.82	27.92

6. Conclusions

This paper proposed an optimal trajectory planning method to obtain a minimum total cost trajectory for wheeled mobile robots by balancing localization errors and energy efficiency. A novel localizability measure method based on the likelihood function was presented to explicitly quantify the localization ability of the robot over a prior map. Then, the localizability aware map was achieved such that the localization error at a given position can be effectively estimated. By incorporating an improved DSA into the trajectory optimization process, the optimal trajectory was generated during the robot motions. We carried out the comprehensive simulations and experiments. Specifically, in the real experiment, the minimal energy-LEV trajectory consumes 19.58% less energy than the minimal LEV trajectory, and reduces the localization error by 61.03% compared to the minimal energy trajectory. The results demonstrated the efficiency of the proposed trajectory planning method in localizability and energy efficiency.

The following directions will be considered in our future works: (1) integrate the neural network optimization to enhance the quality of the generated trajectory; (2) extend our trajectory planning method to multi-robot applications.

Author Contributions: Conceptualization, C.L. and X.Z.; methodology, X.Z.; software, G.L.; validation, G.L. and H.W.; formal analysis, Y.H.; investigation, Y.H.; resources, Y.R.; data curation, Y.R.; writing—original draft preparation, X.Z.; writing—review and editing, C.L. and G.L. visualization, G.L.; supervision, Y.H.; project administration, C.L.; funding acquisition, Y.R. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the Guangdong Major Science and Technology Project (Grant No. 2019B090919003), the Science and Technology Planning Project of Guangdong Province (Grant No. 2017B090913001), and the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2020A1515011393).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Hernández-Guzmán, V.M.; Silva-Ortigoza, R.; Tavera-Mosqueda, S.; Marcelino-Aranda, M.; Marciano-Melchor, M. Path-Tracking of a WMR Fed by Inverter-DC/DC Buck Power Electronic Converter Systems. *Sensors* **2020**, *20*, 6522. [CrossRef] [PubMed]
2. Fu, B.; Chen, L.; Zhou, Y.; Zheng, D.; Pan, H. An improved A* algorithm for the industrial robot path planning with high success rate and short length. *Robot. Auton. Syst.* **2018**, *106*, 26–37. [CrossRef]
3. Jing, Y.; Chen, Y.; Jiao, M.; Huang, J.; Zheng, W. An Improved A* Algorithm Based on Loop Iterative Optimization in Mobile Robot Path Planning. In Proceedings of the International Conference on Intelligent Robotics and Applications, Shenyang, China, 8–11 August 2019; pp. 118–130.
4. Gonzalez, D.; Perez, J.; Milanes, V. Parametric-based path generation for automated vehicles at roundabouts. *Expert Syst. Appl.* **2017**, *71*, 332–341. [CrossRef]

5. Liu, C.; Cao, G.H.; Qu, Y.-Y.; Cheng, Y. An improved PSO algorithm for time-optimal trajectory planning of Delta robot in intelligent packaging. *Int. J. Adv. Manuf. Technol.* **2019**, *107*, 1091–1099. [CrossRef]
6. Chi, W.; Wang, C.; Wang, J.; Meng, Q. Risk-DTRRT-Based Optimal Motion Planning Algorithm for Mobile Robots. *IEEE Trans. Autom. Sci. Eng.* **2018**, *16*, 1–18. [CrossRef]
7. Havlak, F.; Campbell, M. Campbell, Discrete and continuous, probabilistic anticipation for autonomous robots in urban environments. *IEEE Trans. Robot.* **2013**, *30*, 461–474. [CrossRef]
8. Mercy, T.; Van Parys, R.; Pipeleers, G. Spline-Based Motion Planning for Autonomous Guided Vehicles in a Dynamic Environment. *IEEE Trans. Control Syst. Technol.* **2017**, *26*, 1–8. [CrossRef]
9. Joonyoung, K. Trajectory Generation of a Two-Wheeled Mobile Robot in an Uncertain Environment. *IEEE Trans. Ind. Electron.* **2019**, *67*, 5586–5594.
10. González, D.; Pérez, J.; Milanés, V.; Nashashibi, F. A review of motion planning techniques for automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **2015**, *17*, 1135–1145. [CrossRef]
11. Peng, H.; Wang, H.; Chen, D. Optimization of remanufacturing process routes oriented toward eco-efficiency. *Front. Mech. Eng.* **2019**, *14*, 422–433. [CrossRef]
12. Chen, K.; Xue, B.; Zhang, M.; Zhou, F. Novel chaotic grouping particle swarm optimization with a dynamic regrouping strategy for solving numerical optimization tasks. *Knowl. Based Syst.* **2020**, *194*, 105568. [CrossRef]
13. Martin-Moreno, R.; Rodriguez, V.; Miguel, A. Multi-Objective Artificial Bee Colony algorithm applied to the bi-objective orienteering problem. *Knowl. Based Syst.* **2018**, *154*, 93–101. [CrossRef]
14. Wu, T.; Yao, M.; Yang, J. Dolphin swarm algorithm. *Front. Inf. Technol. Electron. Eng.* **2016**, *17*, 717–729. [CrossRef]
15. Amic, S.; Soyjaudah, K.S.; Ramsawock, G. Dolphin Swarm Algorithm for Cryptanalysis. In Proceedings of the Fifth International Conference, Kerala, India, 6 January 2019; pp. 149–163.
16. Kannadasan, K.; Edla, D.R.; Kongara, M.C.; Kuppili, V. M-curves path planning model for mobile anchor node and localization of sensor nodes using dolphin swarm algorithm. *Wirel. Netw.* **2019**, *30*, 1–15. [CrossRef]
17. Sharma, S.; Kaul, A. Hybrid fuzzy multi-criteria decision making based multi cluster head dolphin swarm optimized IDS for VANET. *Veh. Commun.* **2018**, *12*, 23–38. [CrossRef]
18. Irani, B.; Wang, J.; Chen, W. A localizability constraint-based path planning method for autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 2593–2604. [CrossRef]
19. Li, G.; Meng, J.; Xie, Y.; Zhang, X.; Liu, C. Reliable and fast localization in ambiguous environments using ambiguity grid map. *Sensors* **2019**, *19*, 3331. [CrossRef]
20. Censi, A. On Achievable Accuracy for Range-Finder Localization. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 4170–4175.
21. Ruiz-Mayor, A.; Crespo, J.C.; Trivino, G. Perceptual ambiguity maps for robot localizability with range perception. *Expert Syst. Appl.* **2017**, *85*, 33–45. [CrossRef]
22. Zhen, W.; Zeng, S.; Soberer, S. Robust Localization and Localizability Estimation with a Rotating Laser Scanner. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore; 2017; pp. 6240–6245.
23. Candido, S.; Hutchinson, S. Minimum Uncertainty Robot Path Planning using a POMDP Approach. In Proceedings of the International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 3 December 2010; pp. 1408–1413.
24. Nardi, L.; Stachniss, C. Uncertainty-Aware Path Planning for Navigation on Road Networks Using Augmented MDPs. In Proceedings of the International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019.
25. Kang, H.; Liu, C.; Jia, Y.B. Inverse dynamics and energy optimal trajectories for a wheeled mobile robot. *Int. J. Mech. Sci.* **2017**, *134*, 576–588. [CrossRef]
26. Kim, H.; Kim, B.K. Online minimum-energy trajectory planning and control on a straight-line path for three-wheeled omnidirectional mobile robots. *IEEE Trans. Ind. Electron.* **2013**, *61*, 4771–4779. [CrossRef]
27. Tokekar, P.; Karnad, N.; Isler, V. Energy-optimal trajectory planning for car-like robots. *Auton. Robot.* **2014**, *37*, 279–300. [CrossRef]
28. Serralheiro, W.; Maruyama, N.; Saggin, F. Self-tuning time-energy optimization for the trajectory planning of a wheeled mobile robot. *J. Intell. Robot. Syst.* **2019**, *95*, 987–997. [CrossRef]
29. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time Loop Closure in 2D LIDAR SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
30. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005; pp. 153–169.
31. Zuo, L.; Guo, Q.; Xu, X.; Fu, H. A hierarchical path planning approach based on A* and least-squares policy iteration for mobile robots. *Neurocomputing* **2015**, *170*, 257–266. [CrossRef]
32. Ganganath, N.; Cheng, C.T.; Tse, C.K. A constraint-aware heuristic path planner for finding energy-efficient paths on uneven terrains. *IEEE Trans. Ind. Inform.* **2015**, *11*, 601–611. [CrossRef]
33. Liu, S.; Sun, D.; Zhu, C.; Wen, S. A Dynamic Priority Strategy in Decentralized Motion Planning for Formation Forming of Multiple Mobile Robots. In Proceedings of the International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009; pp. 3774–3779.
34. Ni, J.; Hu, J.; Xiang, C. Robust control in diagonal move steer mode and experiment on an X-by-wire UGV. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 572–584. [CrossRef]

35. Feng, Y.; Chen, H.; Zhao, H.Y.; Zhou, H. Road tire friction coefficient estimation for four-wheel drive electric vehicle based on moving optimal estimation strategy. *Mech. Syst. Signal Process.* **2020**, *139*, 106–116. [CrossRef]
36. Yilmaz, A.; Temeltas, H. Self-adaptive Monte Carlo method for indoor localization of smart AGVs using LIDAR data. *Robot. Auton. Syst.* **2019**, *122*, 103–285. [CrossRef]