

Energy Consumption Minimization for Autonomous Mobile Robot: A Convex Approximation Approach

Nguyen Thi Thanh Van ¹, Ngo Manh Tien ^{2*}, Nguyen Cong Luong ³, Ha Thi Kim Duyen ⁴

¹ Faculty of Electrical and Electronic Engineering, Phenikaa University, Hanoi, Vietnam

² Institute of Physics, Vietnam Academy of Science and Technology, Hanoi, Vietnam

³ Faculty of Computer Science, Phenikaa University, Hanoi, Vietnam

⁴ Faculty of Electronic Engineering, Hanoi University of Industry, Hanoi, Vietnam

Email: ¹ van.nguyenthithanh@phenikaa-uni.edu.vn, ² nmtien@iop.vast.vn,

³ luong.nguyencong@phenikaa-uni.edu.vn, ⁴ ha.duyen@hau.edu.vn

*Corresponding Author

Abstract—In this paper, we consider a trajectory design problem of an autonomous mobile robot working in industrial environments. In particular, we formulate an optimization problem that jointly determines the trajectory of the robot and the time step duration to minimize the energy consumption without obstacle collisions. We consider both static and moving obstacles scenarios. The optimization problems are nonconvex, and the main contribution of this work proposing successive convex approximation (SCA) algorithms to solve the nonconvex problems with the presence of both static and moving obstacles. In particular, we first consider the optimization problem in the scenario with static obstacles and then consider the optimization problem in the scenario with static and moving obstacles. Then, we propose two SCA algorithms to solve the nonconvex optimization problems in both the scenarios. Simulation results clearly show that the proposed algorithms outperform the A* algorithm, in terms of energy consumption. This shows the effectiveness of the proposed algorithms.

Keywords—Autonomous mobile robot, energy consumption minimization, trajectory design, time step duration, nonconvex optimization problem.

I. INTRODUCTION

The robotic field has been developing rapidly, and autonomous mobile robots (AMRs) have recently been widely used in industrial environments as well as other applications due to their efficiency and productivity [1], [2] Recently, the AMR with differential drive wheels has been increasingly noticed and widely applied due to its advantages, such as flexible motion capabilities, a simple structure, and lower production costs. However, the mobile robots are typically equipped with a removable energy source that has limited capacity. Meanwhile, the mobile robots require a large amount of energy for sensor system, control system, and motion system. Therefore, the energy consumption minimization for the mobile robots need to be investigated. With the aim of reducing energy consumption and improving the energy efficiency of the mobile robots, both hardware design and algorithms have been proposed.

For the hardware-based solutions, the authors in [3], [4] designed the low-power hardware to reduce the overall electrical energy consumption of the robot. The authors in [5] designed a novel DC power supplier for industrial robots to reduce the total energy consumption for different electrical power profiles. In addition, the authors in [6]–[14] proposed power management, wireless power transfer systems and battery conservation methods to reduce the energy consumption for the robots. Besides, the energy harvesting method was proposed in [15]–[18] to have more power for the robot thus reduce the energy that need from the battery.

Authors in [19] introduced two energy-conservation techniques which are dynamic power management and real-time scheduling to save energy. The experiment is conducted on the Pioneer 3DX robot, and the results shown that the proposed techniques are promising to reduce the energy consumption and extend the lifetime of the robot. Authors in [20] considered the energy-efficient consumption problem by using illuminance instead of the laser-beacon in the navigation system. The experiments results show that the proposed system outperforms the system using laser beacons in terms of power consumption. Different from the aforementioned works, other works aim to save energy by optimizing motion planning of the mobile robot as proposed in [21].

The simulation results in [21] show that optimizing the motion planning can save battery energy up to 10% compared with the widely used trapezoidal velocity profile. Energy-efficient motion planning for the mobile robot exploration is proposed in [22], which can save up to 42% energy compared with the standard utility-based method. Differently, the authors in [23] proposed a motion planning approach, and the simulation results show that the energy can be saved up to 51%. With motion planning, the problems of path planning and path tracking are the most important. Otherwise, several works have investigated a range of control algorithms of path tracking



for the robots [24]–[35]. However, these researches have not considered the energy consumption yet.

Author in [36] considered the energy consumption in the trajectory tracking problems. This work shows some well-known motion controllers, such as feedback-based controller, the Lyapunov-based controller, the clever trigonometry-based controller, the advance controller, and the Dubins path-based controller. The simulation results show that the total energy loss by different controllers are different.

Authors in [37] proposed a optimized controller based on fuzzy logic systems to minimize the energy consumption of the robot. In particular, by using the genetic algorithm, the proposed controller minimizes acceleration, thus reducing the kinetic energy of the robot. The simulation results show that the energy consumption is reduced significantly, and thus demonstrating the efficiency of the proposed controller as well as the proposed algorithm. However, it is difficult to reduce more energy consumption with tracking task of the robot.

Related to path planning, many techniques based on Dijkstra, Genetic, Banker, Bellman-Ford, A* and D* algorithms have been developed over the years [38]–[53]. In general, the path planning optimization with the aim of minimizing the energy consumption is challenging to be solved due to the nonconvex objective function. For this, the authors in [54] incorporated the shortest path length into the objective function to minimize energy consumption using the Newton algorithm.

A different approach is based on A* algorithm [55], [56]. Authors in [56] proposed a energy-efficient A* (EA*) algorithm for optimizing the path for the robot. In this work, although authors only consider the energy consumption for stop and turns of the robot, the simulation results shows that the path optimization by EA* algorithm can save energy more than A* algorithm. Authors in [55] also designed the robot path based on the A* algorithm to generate an energy-efficient path where a new energy-related criterion is used in the cost function. Also, authors in [57] proposed trajectory planning algorithm based on A* algorithm to minimize energy loss. However, the aforementioned works only solve the problem with static obstacles. Authors in [58] proposed an algorithm base on a genetic algorithm (GA) to find an optimal path planning in a dynamic environment with many obstacles. A genetic algorithm was also implemented to optimize the path planning to decrease energy consumption [59].

Authors in [60] developed the optimal velocity profile to reduce energy consumption of the Differential Drive Mobile Robot (DDMR). In this work, authors consider the power model that combines the robot and motor dynamic models. The speed optimization was also used as a method for software-based power management in [61]. In [62], authors proposed a mobile robot motion framework based on the enhanced robust panel method (ERPM), which provides the robust and effective path

planning and motion control in partially known and unknown, static and dynamic environments. Differently, authors in [63]–[68] proposed deep reinforcement learning techniques for robot path planning. However, they did not consider the energy consumption.

From the existing works, we can observe that some of them are proposed to improve energy efficiency, but they do not consider the dynamic environment with moving obstacles. Moreover, they do not optimize the time step duration which has a great influence on the energy consumption of the robot. Meanwhile, some existing works are proposed for dynamic environment, but they do not investigate the energy efficiency.

In this work, we aim to jointly optimize the trajectory and time step duration, which is the interval time that the robot moves from the current position to the next position, so as to minimize the energy consumption of a mobile robot, i.e., an AMR, while avoiding the collisions in dynamic environments with both static and moving obstacles. Note that the introduction of the step time duration, along with the trajectory into the optimization problem makes the problem much more challenging. To solve the problem, we develop a successive convex approximation (SCA) algorithm. The simulation results clearly show the effectiveness of the proposed algorithm compared with the baseline schemes, i.e., the A* algorithm. Interestingly, the simulation results show that the proposed algorithm with the time step optimization further reduces the energy consumption of the mobile robot compared with the proposed algorithm without time step optimization. The main contributions of the paper are the followings:

- We formulate an optimization problem for the mobile robot in the industrial environment that jointly optimizes the path planning and time step duration to minimize the energy consumption, subject to the limited velocity, the limited time and collision avoidance.
- The optimization problem is nonconvex and computationally intractable. Thus, we develop an successive convex approximation (SCA) algorithm to solve the problem with efficient computation. This algorithm is proposed for the scenario in which there are only static obstacles in the working environment of the robot, and thus it is namely OSOW algorithm.
- We further propose an algorithm, namely MSOW, for the scenario including both static and moving obstacles. In this case, the robot first based on the global path that is created by OSOW algorithm to move. Then, the robot uses working environment information sensed by its sensors to find the local path to avoid the moving obstacles. We formulate the energy consumption optimization problem in which the moving collision-avoidance constraints are included. To solve this problem, we again use the successive convex approximation algorithm.
- We provide simulation results to show the improvement

of the proposed algorithms, i.e., the OSOW and MSOW algorithms, compared with the baseline algorithms in which the time step duration is random and with the A* based algorithm.

The rest of the paper is organized as follows. In Section II, we describe the AMR robot and formulate the optimization problem for the robot in environment with static obstacles. In Section III, we present the successive convex approximation algorithm proposed to solve the problem. In Section IV, we formulate the optimization problem in a dynamic environment with static and moving obstacles, followed by the the successive convex approximation algorithm. The simulation results and discussions are presented in Section V. The conclusions and future work are given in Section VI.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a system model as shown in Fig. 1. There is an AGV robot moving from a start point (SP) to a goal point (GP). The robot is in a fixed coordinate system Oxy . Also, we consider a moving coordinate system $O'x'y'$ associated with the robot. We denote $\mathbf{q}_s = (x_s, y_s)^T$, $\mathbf{q}_g = (x_g, y_g)^T$ as the coordinates of the SP and the GP, respectively. We further denote $\mathbf{q}(t) = (x(t), y(t))^T$ and v as the coordinate and velocity, respectively, of the robot at time t . The robot is assumed to perform a task, e.g., carrying boxes, in an industrial environment. We aim to determine the optimal trajectory and time step duration so as to minimize the total energy consumption over its trajectory while avoiding static and moving obstacles.

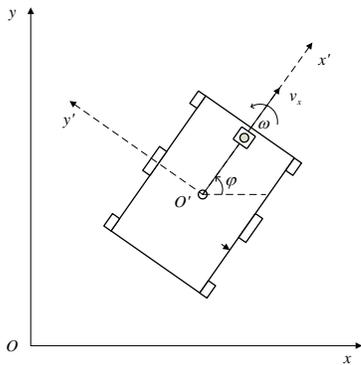


Fig. 1. AMR model.

A. Energy Model

The energy consumed for the robot motors include two main parts: the energy transformed into robot kinetic energy and the energy to overcome traction resistance, i.e., the friction [55]. In our work, we ignore transforming loss, the mechanical loss, and the energy loss as heat in the armatures of motors as in

[69]. The kinetic energy losses of the robot can be expressed as in [70] as follows:

$$E_k = \frac{1}{2}mv^2(t) = \int_{t=0}^T \left(d \left(\frac{1}{2}mv^2(t) \right) \right), \quad (1)$$

where m is the mass and the moment of inertia of the robot. The friction dissipation during the movement of the robot is [70]

$$E_{friction} = 2\mu mg \int_{t=0}^T |v(t)|dt, \quad (2)$$

where g is the gravitational acceleration, m is the weight of the robot, and μ is the rolling friction coefficient that depends on the surface type of the ground. Therefore, the energy consumption of the robot's motion system is determined by [55]

$$E_{motion} = \int_{t=0}^T \left(d \left(\frac{1}{2}mv^2(t) \right) \right) + 2\mu mg \int_{t=0}^T |v(t)|dt, \quad (3)$$

Now, we determine the energy consumed by the onboard computer, sensors, and electric circuits as follows: [70] [55]

$$E_s = P_s \int_{t=0}^T dt, \quad (4)$$

where P_s is the electrical power of the system.

The energy consumption formulation can be rewritten in time-discrete as follows:

$$\begin{aligned} E &= E_{motion} + E_s \\ &= E_k + E_{friction} + E_s \\ &= \frac{1}{2} \sum_{d=0}^{D-1} mv_d^2 + 2\mu mg \sum_{d=0}^{D-1} |\mathbf{q}_{d+1} - \mathbf{q}_d| + P_s \sum_{d=0}^{D-1} \tau, \\ &= \frac{1}{2} \sum_{d=0}^{D-1} m \left(\frac{|\mathbf{q}_{d+1} - \mathbf{q}_d|}{\tau} \right)^2 + 2\mu mg \sum_{d=0}^{D-1} |\mathbf{q}_{d+1} - \mathbf{q}_d| \\ &\quad + P_s(D-1)\tau, \end{aligned} \quad (5)$$

where D is the total requirement number of time steps, $d \in [0, D]$ as the time-varying path of the robot, $\mathbf{Q} = [\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_D]$, and $\mathbf{v} = [v_1, v_2, \dots, v_D]$ denote the set of possible coordinates and the set of velocity values of the robot, and τ is the time step duration.

B. Collision-avoidance Constraints

Given the factory environment, we assume that the obstacles are in rectangular shape. In particular, there is a set $\mathcal{K} = \{1, \dots, K\}$ of obstacles in the workspace of the AMR. Then, each obstacle is defined as the intersection of four half spaces [71]

$$\mathcal{P} = \{\mathbf{z} | \mathbf{A}\mathbf{z} < \mathbf{b}\}, \quad (6)$$

where $\mathbf{A} \in \mathbb{R}^{H \times 2}$ and $\mathbf{b} \in \mathbb{R}^H$. The point \mathbf{z} is defined to be outside of the shape \mathcal{P} if at least one of the four scalar inequalities $\mathbf{A}q \geq \mathbf{b}$ is satisfied, i.e.,

$$\mathbf{A}\mathbf{z} \geq \mathbf{b} + (\mathbf{e} - \mathbf{1})M, \quad (7)$$

$$\sum_{h=1}^H e_h = 1, \quad (8)$$

where $\mathbf{e} = (e_1, e_2, \dots, e_H)^T$ is a vector of binary variables, $e_h \in \{0, 1\}$, and $\mathbf{1} = \mathbf{1}^H$ is a vector of ones, M is a sufficiently large constant used in the Big-M method [72]. The constraints in (7) and (8) ensure that at least one element of \mathbf{e} is equal to 1 and the point \mathbf{z} is out of the obstacle. We denote d_r as the size of robot. To avoid the collision between the robot and obstacles, we define a safe distance of $\delta > 0$. Therefore, the location of the robot needs to satisfy the following constraint

$$\begin{aligned} \mathbf{A}_k \mathbf{q}_d &\geq \mathbf{b}_k + d_r + \delta + (\mathbf{e}_k - \mathbf{1})M, \\ \forall d &\in [0, D], \forall k \in \mathcal{K}, \end{aligned} \quad (9)$$

$$\mathbf{E} \in \{0, 1\}^{K \times H \times (D+1)}, \quad (10)$$

$$\sum_{h=1}^H e_{k,h,d} = 1, \forall k \in \mathcal{K}, d \in [0, D], \quad (11)$$

where $\mathbf{e}_{k,h,d} \in \{0, 1\}^H$ is a binary vector for obstacle k and $\mathbf{E} = (\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_K)$.

C. Optimization Formulation

In general, we can design the trajectory which makes to decrease the travel distance to reduce the kinetic energy consumption and the friction dissipation during the movement of the robot. However, it should be to satisfy the collision avoidance requirement. In addition, we can decrease the time step duration to reduce the energy consumed by the onboard computer, sensors, and electric circuits. However, this requires the mobile robot to move faster that can exceed its limited velocity. Therefore, in this work, we aim to minimize the energy consumption of the robot by optimizing its trajectory and time step duration while satisfying the collision avoidance requirement. The optimization problem is mathematically formulated as the mixed-binary optimization problem as follows:

$$\min_{\mathbf{Q}, \tau, \mathbf{e}} E \quad (12a)$$

s.t. (9) - (11),

$$\mathbf{q}_0 = \mathbf{q}_s, \quad (12b)$$

$$\mathbf{q}_D = \mathbf{q}_g, \quad (12c)$$

$$v_d \leq v_{\max}, \forall d \in \{1, 2, \dots, D\}, \quad (12d)$$

$$\tau_{\min} \leq \tau \leq \tau_{\max}, \quad (12e)$$

where v_{\max} is the maximum linear velocity of the robot, τ_{\min} and τ_{\max} are the minimum time step duration and the maximum time step duration, respectively.

III. SUCCESSIVE CONVEX APPROXIMATION ALGORITHM

It is observed that the optimization problem (12) is nonconvex because the objective in (12a) is nonconvex. In this section, we develop a successive convex approximation [73] algorithm to solve it.

Let $(\mathbf{Q}^{(\kappa)}, \tau^{(\kappa)}, \mathbf{e}^{(\kappa)})$ be a feasible point for (12) found from the $(\kappa - 1)$ -th iteration. In iteration κ , we determine the next feasible point $(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)}, \mathbf{e}^{(\kappa+1)})$. By applying the inequality (A.2) in Appendix A to E_k , we have

$$\begin{aligned} E_k &\leq \frac{1}{2}m \sum_{d=0}^{D-1} \left(\frac{|\mathbf{q}_{d+1}^{(\kappa)} - \mathbf{q}_d^{(\kappa)}|}{4\tau^{(\kappa)}} \left(\frac{|\mathbf{q}_{d+1} - \mathbf{q}_d|}{|\mathbf{q}_{d+1}^{(\kappa)} - \mathbf{q}_d^{(\kappa)}|} + \frac{\tau^{(\kappa)}}{\tau} \right)^2 \right)^2 \\ &= \frac{m}{32} \sum_{d=0}^{D-1} \frac{|\mathbf{q}_{d+1}^{(\kappa)} - \mathbf{q}_d^{(\kappa)}|^2}{(\tau^{(\kappa)})^2} \left(\frac{|\mathbf{q}_{d+1} - \mathbf{q}_d|}{|\mathbf{q}_{d+1}^{(\kappa)} - \mathbf{q}_d^{(\kappa)}|} + \frac{\tau^{(\kappa)}}{\tau} \right)^4 \\ &\triangleq E_k^{(\kappa)}. \end{aligned} \quad (13)$$

It is observed that the expression in the brackets of $E_k^{(\kappa)}$ is sum of the norm function, i.e., $\frac{|\mathbf{q}_{d+1} - \mathbf{q}_d|}{|\mathbf{q}_{d+1}^{(\kappa)} - \mathbf{q}_d^{(\kappa)}|}$ and the inverse of the positive portion, i.e., $\frac{\tau^{(\kappa)}}{\tau}$. These terms are convex functions, thus $E_k^{(\kappa)}$ is convex. In addition, $E_{friction}$ is already convex. Therefore, we can approximate the objective in (12a) by the following convex function

$$\begin{aligned} E &\leq E_k^{(\kappa)} + E_{friction} + E_s \\ &\triangleq E^{(\kappa)}. \end{aligned} \quad (14)$$

Based on (14), we solve the following convex optimization problem rather than (12)

$$\min_{\mathbf{Q}, \tau, \mathbf{e}} E^{(\kappa)} \quad (15a)$$

$$\text{s.t. (9) - (11), and (12b) - (12e),} \quad (15b)$$

which generates $(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)})$ in the next iteration. The computational complexity of the algorithm to solve the convex problem in (15) is $\mathcal{O}(\alpha^2 \beta^{2.5} + \beta^{3.5})$ [74], where $\mathcal{O}(\alpha^2 \beta^{2.5} + \beta^{3.5})$ [74], where $\alpha = 2D + KH(D + 1) + 3$ and $\beta = 2(D + 1)KH + (D + 1)K + D + 3$ are the numbers of decision variables and convex constraints as solving (15), respectively. Note that $(\mathbf{Q}^{(\kappa)}, \tau^{(\kappa)})$ and $(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)})$ are the feasible point and the optimal solution to (15), respectively, we can show that

$$E^{(\kappa)}(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)}) > E^{(\kappa)}(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)}), \quad (16)$$

for any $(\mathbf{Q}^{(\kappa)}, \tau^{(\kappa)}) \neq (\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)})$. It simply means that $(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)})$ is a better point to (12) than $(\mathbf{Q}^{(\kappa)}, \tau^{(\kappa)})$. Considering (12a), it is true that [75]:

$$E(\mathbf{Q}^{(\kappa)}, \tau^{(\kappa)}) = E^{(\kappa)}(\mathbf{Q}^{(\kappa)}, \tau^{(\kappa)}) \quad (17)$$

$$< E^{(\kappa)}(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)}) \quad (18)$$

$$\leq E(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)}). \quad (19)$$

Algorithm 1 OSOW Algorithm For (12)

Initialization: Set $\kappa = 1$;
1: **repeat**
2: Solve the convex problem (15) to obtain $(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)}) = (\mathbf{Q}^*, \tau^*)$;
3: Update $(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)}) := (\mathbf{Q}^*, \tau^*)$
4: Set $\kappa \leftarrow \kappa + 1$
5: **until** Convergence
6: Output $(\mathbf{Q}^{(\kappa)}, \tau^{(\kappa)})$

This clearly shows that the optimal solution $(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)})$ of (12) satisfies the convergence condition: $E(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)}) > E(\mathbf{Q}^{(\kappa)}, \tau^{(\kappa)})$. The sequence $(\mathbf{Q}^{(\kappa+1)}, \tau^{(\kappa+1)})$ converges to a saddle point $(\bar{\mathbf{Q}}, \bar{\tau})$ after a sufficiently large number of iterations [75]. The overall algorithm for the optimization problem in (12) is shown in Algorithm 1. This algorithm is namely OSOW algorithm, meaning that there are only static obstacles in the robot's workspace.

IV. MOVING OBSTACLES SCENARIO

In Section III, we consider the scenario in which the obstacles are static. In realistic scenarios, other than the static obstacles, there are also the obstacles are movable. Thus, in this section, we consider the scenarios with moving obstacles while the robot runs. In particular, we consider the workspace is with the static obstacles (SOs) similar to subsection II-B. First of all, the robot calculates and determines the trajectory to minimize the energy consumption by solve the problem (12). After that, the robot follows the trajectory, i.e., $\mathbf{Q} = [\mathbf{q}_0, \mathbf{q}_1, \dots, \mathbf{q}_D]$. During the robot's movement, at time step $i \in [0, D]$, the robot detects a set $\mathcal{N}_i = \{1, \dots, N_i\}$ of moving obstacles (MOs). Each MO is assumed to be in circle shape. Particularly, the robot observes the MO n_i centered at $\mathbf{q}_{n,i}^O = (x_{n,i}^O, y_{n,i}^O)$ with velocity $\mathbf{v}_{n,i} = (v_{nx_i}, v_{ny_i})$. The collision avoidance requirement is to ensure that the future trajectory of the robot $\mathbf{q}_{i,d_i} = (x_{i,d_i}, y_{i,d_i})$ for $d_i \in [0, D_i]$ does not collide with all obstacles, where D_i is number of steps to overcome N_i MOs. We denote $\mathbf{Q}_i = [\mathbf{q}_{i,0}, \mathbf{q}_{i,1}, \dots, \mathbf{q}_{i,D_i}]$, and $\mathbf{v}_i = [v_{i,1}, v_{i,2}, \dots, v_{i,D_i}]$ be the set of possible coordinates and the set of velocity values of the robot. Accordingly, \mathbf{Q} is called the global trajectory and \mathbf{Q}_i is the local trajectory. For the collision avoidance requirement, the following constraints are satisfied

$$\|\mathbf{q}_{i,d_i} - \mathbf{q}_{n,d_i}^O\| \geq r_{n,i}^O + d_r + \delta, \forall n \in \mathcal{N}, d_i \in [0, D_i], \quad (20)$$

where $r_{n,i}^O$ is the radius of the obstacle n , and $\mathbf{q}_{n,d_i}^O = (x_{n,d_i}^O, y_{n,d_i}^O)$ is the location of the obstacle n at time step d_i . Let $d_{i,r,n} = r_{n,i}^O + d_r + \delta$, the constraint in (20) can be rewritten as

$$\|\mathbf{q}_{i,d_i} - \mathbf{q}_{n,d_i}^O\| \geq d_{i,r,n}, \forall n \in \mathcal{N}, d_i \in [0, D_i]. \quad (21)$$

We now formulate the optimization problem for the MOs scenario, which is similar to (12) with the introduction of the moving obstacle avoidance constraint (20) as follows:

$$\min_{\mathbf{Q}_i, \tau_i, \mathbf{e}} E_i \quad (22a)$$

s.t. (9) - (11), and (21).

$$\mathbf{q}_{i,0} = \mathbf{q}_{i,s}, \quad (22b)$$

$$\mathbf{q}_{i,D_i} = \mathbf{q}_{i,g}, \quad (22c)$$

$$v_{i,d_i} \leq v_{\max}, \forall d_i \in [0, D_i], \quad (22d)$$

$$\tau_{\min} \leq \tau_i \leq \tau_{\max}, \quad (22e)$$

where τ_i is step time, $\mathbf{q}_{i,s}, \mathbf{q}_{i,g}$ are the start point where the robot detects the MOs, and the goal point where the robot overcomes the MOs and backs to the trajectory \mathbf{Q} , respectively, and

$$\begin{aligned} E_i &= E_{k_i} + E_{friction_i} + E_{s_i} \\ &= \frac{1}{2} \sum_{d_i=0}^{D_i-1} m \left(\frac{|\mathbf{q}_{i,d_i+1} - \mathbf{q}_{i,d_i}|}{\tau_i} \right)^2 \\ &\quad + 2\mu mg \sum_{d_i=0}^{D_i-1} |\mathbf{q}_{i,d_i+1} - \mathbf{q}_{i,d_i}| + P_s(D_i - 1)\tau_i \end{aligned} \quad (23)$$

is the total energy consumption for overcoming N_i MOs.

The problem in (22) is nonconvex since the objective (22a) and the constraint in (21) are nonconvex. Similar to Section III, by applying the inequality (A.2) in Appendix A to E_{k_i} , we have

$$\begin{aligned} E_{k_i} &\leq \frac{1}{2} m \sum_{d_i=0}^{D_i-1} \left(\frac{|\mathbf{q}_{i,d_i+1}^{(\kappa)} - \mathbf{q}_{i,d_i}^{(\kappa)}|}{4\tau_i^{(\kappa)}} \left(\frac{|\mathbf{q}_{i,d_i+1} - \mathbf{q}_{i,d_i}|}{|\mathbf{q}_{i,d_i+1}^{(\kappa)} - \mathbf{q}_{i,d_i}^{(\kappa)}|} + \frac{\tau_i^{(\kappa)}}{\tau_i} \right)^2 \right)^2 \\ &= \frac{m}{32} \sum_{d_i=0}^{D_i-1} \frac{|\mathbf{q}_{i,d_i+1}^{(\kappa)} - \mathbf{q}_{i,d_i}^{(\kappa)}|^2}{(\tau_i^{(\kappa)})^2} \left(\frac{|\mathbf{q}_{i,d_i+1} - \mathbf{q}_{i,d_i}|}{|\mathbf{q}_{i,d_i+1}^{(\kappa)} - \mathbf{q}_{i,d_i}^{(\kappa)}|} + \frac{\tau_i^{(\kappa)}}{\tau_i} \right)^4 \\ &\triangleq E_{k_i}^{(\kappa)}. \end{aligned} \quad (24)$$

Therefore, we can approximate the objective in (22a) by the following convex function

$$\begin{aligned} E_i &\leq E_{k_i}^{(\kappa)} + E_{friction_i} + E_{s_i} \\ &\triangleq E_i^{(\kappa)}. \end{aligned} \quad (25)$$

Now, we consider the constraint in (21), which is equivalent to the following expression

$$\begin{aligned} (x_{i,d_i} - x_{n,d_i}^O)^2 + (y_{i,d_i} - y_{n,d_i}^O)^2 &\geq d_{r,n}^2, \\ \forall n \in \mathcal{N}, d_i \in [0, D_i], \end{aligned} \quad (26)$$

where $x_{n,d_i}^O = x_{n,i}^O + v_{nx_i} d_i \tau_i$, $y_{n,d_i}^O = y_{n,i}^O + v_{ny_i} d_i \tau_i$. Let the right-hand-side of (26) be $f(\mathbf{w}_{n,d_i})$, where $\mathbf{w}_{i,d_i} = (\mathbf{q}_{i,d_i}, \mathbf{q}_{n,d_i})$. By taking the Hessian matrix of second partial derivatives of function $f(\mathbf{w}_{n,d_i})$ with respect to x_{i,d_i}, y_{i,d_i} and

τ_i , it is observed that matrix is positive semidefinite. Therefore, $f(\mathbf{w}_{n,d_i})$ is convex. As the function $f(\mathbf{w}_{n,d_i})$ is convex, its gradient, i.e., $\nabla f(\mathbf{w}_{n,d_i}^{(\kappa)})$, is super-gradient [76]. Therefore, we have

$$\begin{aligned} & f(\mathbf{w}_{n,d_i}) \\ & \geq f(\mathbf{w}_{n,d_i}^{(\kappa)}) + \nabla f(\mathbf{w}_{n,d_i}^{(\kappa)})(\mathbf{w}_{n,d_i} - \mathbf{w}_{n,d_i}^{(\kappa)}) \\ & = (x_{i,d_i}^{(\kappa)} - x_{n,d_i}^{O(\kappa)})^2 + (y_{i,d_i}^{(\kappa)} - y_{n,d_i}^{O(\kappa)})^2 + 2(x_{i,d_i}^{(\kappa)} - x_{n,d_i}^{O(\kappa)}) \\ & \quad \times (x_{d_i} - x_{i,d_i}^{(\kappa)}) + 2(y_{i,d_i}^{(\kappa)} - y_{n,d_i}^{O(\kappa)})(y_{i,d_i} - y_{i,d_i}^{(\kappa)}) \\ & \quad - 2d_i(v_{nx}(x_{i,d_i}^{(\kappa)} - x_{n,d_i}^{O(\kappa)}) + v_{ny}(y_{i,d_i}^{(\kappa)} - y_{n,d_i}^{O(\kappa)}))(\tau_i - \tau_i^{(\kappa)}) \\ & \triangleq f^{(\kappa)}(\mathbf{w}_{n,d_i}). \end{aligned} \quad (27)$$

Based on (26) and (27), the nonconvex constraint in (20) is innerly approximated by the following convex constraint

$$f^{(\kappa)}(\mathbf{w}_{n,d_i}) \geq d_{i,r,n}^2. \quad (28)$$

From (25) and (28), instead of (22), we solve the following convex optimization problem

$$\min_{\mathbf{Q}_i, \tau_i, \mathbf{e}} E_i^{(\kappa)} \quad (29a)$$

$$\text{s.t. (9) - (11), (22b) - (22e) and (28),} \quad (29b)$$

which generates $(\mathbf{Q}_i^{(\kappa+1)}, \tau_i^{(\kappa+1)})$ in the next iteration. The computational complexity of the algorithm to solve the convex problem in (15) is $\mathcal{O}(\alpha^2\beta^{2.5} + \beta^{3.5})$ [74], where $\mathcal{O}(\alpha^2\beta^{2.5} + \beta^{3.5})$ [74], where $\alpha = 2D_i + KH(D_i + 1) + 3$ and $\beta = 2(D_i + 1)KH + (D_i + 1)K + D_i + D_i N_i + 3$ are the numbers of decision variables and convex constraints as solving (29), respectively. Similar to Section III, the sequence $(\mathbf{Q}_i^{(\kappa+1)}, \tau_i^{(\kappa+1)})$ converges to a saddle point $(\bar{\mathbf{Q}}_i, \bar{\tau}_i)$ after a sufficiently large number of iterations [75]. To enhance the computational efficiency of the algorithm, it is important to generate a feasible point.

Taking any feasible point $(\mathbf{Q}_i^{(0)}, \tau_i^{(0)})$ for (12b), (12c) and (12e), it follows from (29) that initialized by a feasible point $(\mathbf{Q}_i^{(\kappa)}, \tau_i^{(\kappa)})$ for the convex constraints (12d) and (28), we iterate

$$\max_{\mathbf{Q}_i, \tau_i, \mathbf{e}} \eta \quad (30a)$$

$$\text{s.t. (9) - (11), (22b) - (22c), (22e),} \quad (30b)$$

$$v_{\max} \geq v_{d_i} \eta, \forall d_i \in \{1, 2, \dots, D\}, \quad (30c)$$

for $\kappa = 0, 1, \dots$ until the value of the objective in (30) is great than or equal to 1, making $(\mathbf{Q}_i^{(\kappa)}, \tau_i^{(\kappa)})$ feasible for (29).

As mention early, the robot follows the local trajectory \mathbf{Q}_i to overcome the MOs, and after that, the robot returns the global trajectory \mathbf{Q} . The overall algorithm for the optimization problem with both SOs and MOs is shown in Algorithm 2. This algorithm is namely MSOW algorithm, meaning that there are both moving and static obstacles in the robot's workspace.

Algorithm 2 MSOW Algorithm

- 1: Using Algorithm 1 to generate the solution $(\mathbf{Q}^{(s)}, \tau^{(s)})$ for only static obstacles in workspace.
 - 2: Let $(\mathbf{Q}^{(g)}, \tau^{(g)}) := (\mathbf{Q}^{(s)}, \tau^{(s)})$
 - 3: **repeat**
 - 4: The robot moves base on the global trajectory $(\mathbf{Q}^{(g)}, \tau^{(g)})$.
 - 5: **if** The robot detects N_i MOs at step i **then**
 - 6: Initialization: Generate an initial point $(\mathbf{Q}_i^{(0)}, \tau_i^{(0)})$ for (22b), (22c) and (22e), and iterate (30) for a feasible point $(\mathbf{Q}_i^{(\kappa)}, \tau_i^{(\kappa)})$ for (29). Set $\kappa = 0$;
 - 7: **repeat**
 - 8: Solve the convex problem (29) to obtain $(\mathbf{Q}_i^{(\kappa+1)}, \tau_i^{(\kappa+1)}) = (\mathbf{Q}_i^*, \tau_i^*)$;
 - 9: Update $(\mathbf{Q}_i^{(\kappa+1)}, \tau_i^{(\kappa+1)}) := (\mathbf{Q}_i^*, \tau_i^*)$
 - 10: Set $\kappa \leftarrow \kappa + 1$
 - 11: **until** Convergence
 - 12: Output $(\mathbf{Q}_i^{(\kappa)}, \tau_i^{(\kappa)})$.
 - 13: Using $(\mathbf{Q}_i^{(\kappa)}, \tau_i^{(\kappa)})$ to update $(\mathbf{Q}^{(g)}, \tau^{(g)})$.
 - 14: **end if**
 - 15: **until** The robot is at the goal point.
-

V. PERFORMANCE EVALUATION

In this section, we present the numerical results to demonstrate the effectiveness of the proposed algorithms. We first present simulation results in the scenario with only static obstacles (SOs), and then we present simulation results the scenario with both SOs and moving obstacles (MOs). Consider the first scenario, the number of SOs is set to $K = 5$, which are randomly distributed in a square area of $10\text{m} \times 10\text{m}$ [55]. The friction parameter μ on the flat road is 0.05. Other simulation parameters are listed in Table I. The simulation is implemented by MATLAB, and the optimization problem is solved in MATLAB using the CVX 2.2 [77]–[80] with the optimization solver Mosek 9.1.9 and default precision.

TABLE I. SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
K	5	D	30
d_r	0.3 m	δ	0.1 m
v_{\max}	0.7 m/s	μ	0.05
τ_{\max}	1 s	τ_{\min}	0.01 s
g	9.8m/s ²	m	9 kg [55]
M	100 [71]	P_s	17.8 W
\mathbf{q}_s	[1.5, 1.5]	\mathbf{q}_g	[8, 8]

For the performance comparison, we consider the following algorithms:

- **OSOW:** This is the successive convex approximation algorithm as described in Section III (shown in Algorithm 1), which is proposed to solve the optimization problem given in (12).

- **OSOW with fixed τ** : This algorithm is similar to the OSOW algorithm in which the step time duration τ is fixed.
- **MSOW**: This is the successive convex approximation algorithm as described in Section IV (shown in Algorithm 2), which is proposed to solve the optimization problem given in (22).
- **MSOW with fixed τ** : This is similar to the MSOW algorithm in which the time step duration τ is fixed.

First, we consider the only static obstacles (SOs) scenario. For comparison, we introduce the A* based algorithm [55] as a baseline scheme. It is note that the A* based algorithm is adopted for SOs scenario only, thus the comparison is only in this scenario. Fig. 2(a) shows the optimal trajectory by three algorithms which are the OSOW algorithm, the OSOW algorithm with fixed τ , and the A* based algorithm. It is observed that the trajectory with the OSOW algorithm and the OSOW algorithm with fixed τ are the same, and they are smoother than that with the A* based algorithm. In addition, Fig. 2(b) shows the energy consumption and the travel time of the robot versus its maximum velocity with three algorithms. As seen, the energy consumption and the travel time of the robot obtained by the OSOW algorithm are always the lowest at each value of v_{max} . This result clearly shows that it is important and necessary to optimize the time step duration τ in the OSOW algorithm. This also demonstrates the effectiveness of our proposed algorithm.

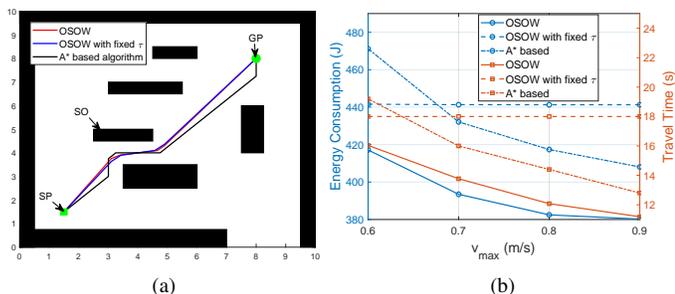


Fig. 2. (a) The optimal trajectory by three algorithms (b) The energy consumption and travel time versus the maximum velocity of the robot.

Next, we discuss the impact of the maximum velocity, i.e., v_{max} , of the robot on the energy consumption obtained by the algorithms in only static obstacles scenario. As shown in Fig. 2, as v_{max} increases, the energy consumption and the travel time of the robot obtained by the OSOW algorithm and the A* based algorithm decrease, and those obtained by the OSOW with fixed τ algorithm keep constant. These results can be explained as follows. With the OSOW algorithm, as v_{max} increases, the totally moving time of the robot decreases that reduces the energy consumption by the onboard computer, sensors, and electric circuit, i.e., E_s . On other hand, as v_{max} increases, the robot moves with higher velocity that increases the energy consumption of the robot’s motion system, i.e., E_{motion} . However,

it is interesting that the total energy consumption decreases. The reason may be that the decrease of E_s is faster than the increase of E_{motion} . The similar reason is explained for down trend of the energy consumption and the travel time of the robot when v_{max} increases. With the OSOW with fixed τ , as v_{max} increases, the travel time keeps no change. The reason is that the location of robot at each step is no change as v_{max} varies. Moreover, the step duration τ is fixed, thus the velocity of the robot is no change. As a result, the total energy consumption seems does not change.

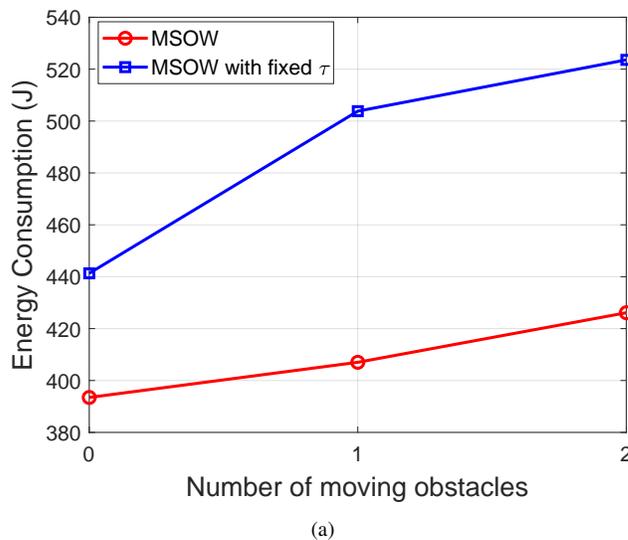


Fig. 3. The energy consumption and travel distance versus number of moving obstacles in the workspace

Now, we consider the scenario with both moving and static obstacles in the working environment of the robot. It is important to show how the energy consumption varies as the number of moving obstacles (MOs) varies. In our simulations, we assume the robot detects one MO at step 3, the MO centered at (1.5, 3), with radius 0.5 m, then the robot detects one more MO at step 20, this MO centered at (6.5, 7.5), with radius 0.3 m. As shown in Fig. 3(a), as the number of MOs increases, the energy consumption of the robot increases. The reason is that as the number of MOs increases, the robot requires longer travel distance and travel time to avoid the collisions. In addition, the energy consumption of the robot obtained by the MSOW algorithm is lower than by the MSOW with fixed τ algorithm. This demonstrates the effectiveness of our proposed algorithm.

Finally, it is worth showing the trajectories of the robot with our proposed algorithms, which are shown in Figs. 4(a), (b), (c), and (d). In the figures, each blue circle and red circles present the robot and the MO at each time step, respectively. Fig. 4(a) shows the trajectory of the robot when there is no MO appears and it is the same as the global trajectory. Figs. 4(b) and (c) shows the trajectory of the robot in scenarios with one and two MOs, respectively. It is observed that the trajectory of

the robot is changed to avoid the MOs. Fig. 4(d) also shows that by adopting our algorithms, the trajectory is smooth and the collision avoidance requirement is satisfied with minimum energy consumption.

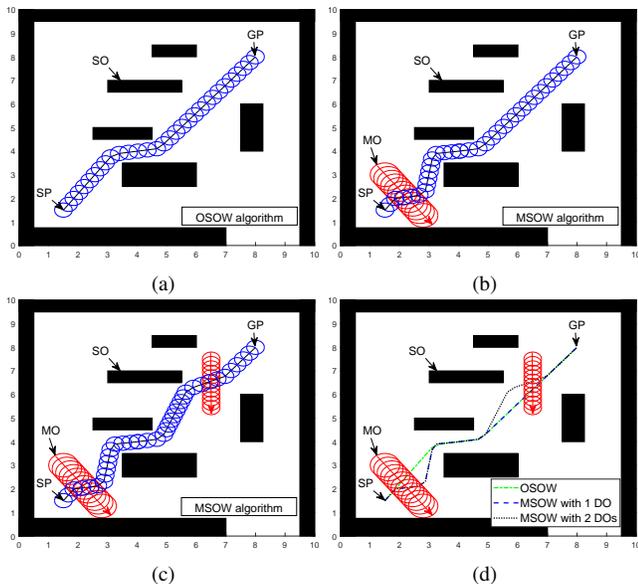


Fig. 4. The optimal trajectory of the robot for minimizing the energy consumption in (a) OSOW scenario (b) MSOW scenario with 1 MO (c) MSOW scenario with 2 MOs and (d) 3 scenarios.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have investigated the energy consumption of the AMR with differential drive wheels. In particular, we aim to design the trajectory and the time step duration for the robot to minimize the energy consumption. The robot was limited by the velocity and the workplace layout. Therefore, we have formulated an optimization problem that minimizes the energy consumption constrained to a maximum velocity and obstacles. We consider two scenarios: the first scenario with static obstacles and the second scenario with both static and moving obstacles. The optimization problem is nonconvex, and we have proposed the successive convex approximation algorithms to solve it. Simulation results have shown the effectiveness of the proposed algorithms compared with the A* algorithm. In addition to the path planning, path tracking as a motion planning can help to improve the performance of the mobile robot. In future work, we will investigate the energy consumption minimization of the mobile robots by jointly optimizing the path planning and path tracking.

APPENDIX A: FUNDAMENTAL INEQUALITIES

Using the Cauchy-Schwarz inequality, we have the following inequality

$$2xy \leq \frac{\bar{x}\bar{y}}{2} \left(\frac{x}{\bar{x}} + \frac{y}{\bar{y}} \right)^2, \forall (x, y) \in \mathbb{R}_+^2 \text{ and } (\bar{x}, \bar{y}) \in \mathbb{R}_+^2 \quad (\text{A.1})$$

Substituting $y \rightarrow 1/y$ and $\bar{y} \rightarrow 1/\bar{y}$, we have the following inequality

$$2\frac{x}{y} \leq \frac{\bar{x}}{2\bar{y}} \left(\frac{x}{\bar{x}} + \frac{\bar{y}}{y} \right)^2, \forall (x, y) \in \mathbb{R}_+^2 \text{ and } (\bar{x}, \bar{y}) \in \mathbb{R}_+^2 \quad (\text{A.2})$$

ACKNOWLEDGMENT

This research was funded by Vietnam's National project "Research, develop an intelligent mobile robot using different types of sensing technology and IoT platform, AI, and implemented in radioactive environment monitoring application," code: DTDLCN.19/23 of the CT1187 Physics development program in the period 2021- 2025.

REFERENCES

- [1] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419839596, 2019.
- [2] M. T. Ballestar, Ángel Díaz-Chao, J. Sainz, and J. Torrent-Sellens, "Knowledge, robots and productivity in smes: Explaining the second digital wave," *Journal of Business Research*, vol. 108, pp. 119–131, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0148296319306861>
- [3] A. Yang, J. Pu, C. B. Wong, and P. Moore, "By-pass valve control to improve energy efficiency of pneumatic drive system," *Control Engineering Practice*, vol. 17, no. 6, pp. 623–628, 2009.
- [4] R. Saidur, "A review on electrical motors energy use and energy savings," *Renewable and sustainable energy reviews*, vol. 14, no. 3, pp. 877–898, 2010.
- [5] R. Grebers, M. Gadaleta, A. Paugurs, A. Senfelds, A. Avotins, and M. Pellicciari, "Analysis of the energy consumption of a novel dc power supplied industrial robot," *Procedia Manufacturing*, vol. 11, pp. 311–318, 2017.
- [6] F. Lu, H. Zhang, C. Zhu, L. Diao, M. Gong, W. Zhang, and C. C. Mi, "A tightly coupled inductive power transfer system for low-voltage and high-current charging of automatic guided vehicles," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 9, pp. 6867–6875, 2018.
- [7] F. Lu, Y. Zhang, H. Zhang, C. Zhu, L. Diao, M. Gong, W. Zhang, and C. Mi, "A low-voltage and high-current inductive power transfer system with low harmonics for automatic guided vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3351–3360, 2019.
- [8] C. Jiang, K. Chau, C. Liu, C. H. Lee, W. Han, and W. Liu, "Move-and-charge system for automatic guided vehicles," *IEEE Transactions on Magnetics*, vol. 54, no. 11, pp. 1–5, 2018.
- [9] Y.-M. Yang, S.-J. Huang, J.-Y. Chen, S.-H. Dai, and T.-S. Lee, "Auxiliary magnetic repeater-aided dual-output wireless charging with coupling analysis and power transfer capability considerations," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, vol. 1. IEEE, 2019, pp. 3450–3455.
- [10] M. Sugino and T. Masamura, "The wireless power transfer systems using the class e push-pull inverter for industrial robots," in *2017 IEEE Wireless Power Transfer Conference (WPTC)*. IEEE, 2017, pp. 1–3.
- [11] Y. Hao, J. Wang, and Y. Liu, "Research on wireless power transfer system of automated guided vehicle based on magnetic coupling resonance," in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*. IEEE, 2019, pp. 1–4.
- [12] M. Putz and A. Schlegel, "Scheduling charging operations of autonomous agvs in automotive in-house logistics."
- [13] X. Zhan, L. Xu, J. Zhang, and A. Li, "Study on agvs battery charging strategy for improving utilization," *Procedia CIRP*, vol. 81, pp. 558–563, 2019.
- [14] W. Chen, J. Liu, S. Chen, and L. Zhang, "Energy shaping control for wireless power transfer system in automatic guided vehicles," *Energies*, vol. 13, no. 11, p. 2959, 2020.

- [15] C. Xiao, N. D. Naclerio, and E. W. Hawkes, "Energy harvesting across temporal temperature gradients using vaporization," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7170–7175.
- [16] Y. Zhai, Z. Ding, Y. Liu, S. Jin, and L. Kang, "Research on novel spatial structure of power generation of spherical robot," *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 234, no. 6, pp. 600–612, 2020. [Online]. Available: <https://doi.org/10.1177/0954408920932358>
- [17] M. T. Ajala, M. R. Khan, M. Salami, A. Shafie, M. O. Oladokun, and M. I. M. Nor, "Pneumatic actuation of a firefighting robot: A theoretical foundation and an empirical study," in *2019 7th International Conference on Mechatronics Engineering (ICOM)*, 2019, pp. 1–6.
- [18] T. Maeda, *Fluid Power: Proceedings of the Second JHPS International Symposium on Fluid Power*. CRC Press, 1993.
- [19] Y. Mei, Y.-H. Lu, Y. Hu, and C. Lee, "A case study of mobile robot's energy consumption and conservation techniques," in *ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, 2005.*, 2005, pp. 492–497.
- [20] Angelina, Seilla, Afifah, Shofuro, Susamti, Paramadina, Ardianto Priramadhi, Rizki, and Darlis, Denny, "Efficient energy consumption for indoor mobile robot prototype under illumination," *MATEC Web Conf.*, vol. 197, p. 11016, 2018. [Online]. Available: <https://doi.org/10.1051/mateconf/201819711016>
- [21] H. Kim and B.-K. Kim, "Minimum-energy translational trajectory planning for battery-powered three-wheeled omni-directional mobile robots," in *Proc. of the 10th International Conference on Control, Automation, Robotics and Vision*, pp. 1730–1735, 2008.
- [22] Y. Mei, Y.-H. Lu, C. G. Lee, and Y. C. Hu, "Energy-efficient mobile robot exploration," in *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 505–511, 2006.
- [23] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5. IEEE, 2004, pp. 4344–4349.
- [24] L. Xin, Q. Wang, J. She, and Y. Li, "Robust adaptive tracking control of wheeled mobile robot," *Robotics and Autonomous Systems*, vol. 78, pp. 36–48, 2016.
- [25] M. Yue, L. Wang, and T. Ma, "Neural network based terminal sliding mode control for wmsr affected by an augmented ground friction with slippage effect," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 498–506, 2017.
- [26] S. Peng and W. Shi, "Adaptive fuzzy output feedback control of a nonholonomic wheeled mobile robot," *IEEE Access*, vol. 6, pp. 43 414–43 424, 2018.
- [27] P. Petrov and V. Georgieva, "Adaptive velocity control for a differential drive mobile robot," in *2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA)*. IEEE, 2018, pp. 1–4.
- [28] B. B. Mevo, M. R. Saad, and R. Fareh, "Adaptive sliding mode control of wheeled mobile robot with nonlinear model and uncertainties," in *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*. IEEE, 2018, pp. 1–5.
- [29] F.-G. Rojas-Contreras, A.-I. Castillo-Lopez, L. Fridman, and V.-J. Gonzalez-Villela, "Trajectory tracking using continuous sliding mode algorithms for differential drive robots," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 6027–6032.
- [30] Y. Jinhua, Y. Suzhen, and J. Xiao, "Trajectory tracking control of wmr based on sliding mode disturbance observer with unknown skidding and slipping," in *2017 2nd International Conference on Cybernetics, Robotics and Control (CRC)*. IEEE, 2017, pp. 18–22.
- [31] J. Meng, A. Liu, Y. Yang, Z. Wu, and Q. Xu, "Two-wheeled robot platform based on pid control," in *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2018, pp. 1011–1014.
- [32] D. Diaz and R. Kelly, "On modeling and position tracking control of the generalized differential driven wheeled mobile robot," in *2016 IEEE International Conference on Automatica (ICA-ACCA)*. IEEE, 2016, pp. 1–6.
- [33] N. V. Tinh, N. T. Linh, P. T. Cat, P. M. Tuan, M. N. Anh, and N. P. Anh, "Modeling and feedback linearization control of a nonholonomic wheeled mobile robot with longitudinal, lateral slips," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2016, pp. 996–1001.
- [34] M. Yallala and S. Mija, "Path tracking of differential drive mobile robot using two step feedback linearization based on backstepping," in *2017 International Conference on Innovations in Control, Communication and Information Systems (ICICCI)*. IEEE, 2017, pp. 1–6.
- [35] X. Zhou, B. Ma, and L. Yan, "Adaptive output feedback tracking controller for wheeled mobile robots with unmeasurable orientation," in *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018, pp. 412–417.
- [36] A. Stefek, T. V. Pham, V. Krivanek, and K. L. Pham, "Energy comparison of controllers used for a differential drive wheeled mobile robot," *IEEE Access*, vol. 8, pp. 170 915–170 927, 2020.
- [37] A. Štefek, V. T. Pham, V. Krivanek, and K. L. Pham, "Optimization of fuzzy logic controller used for a differential drive wheeled mobile robot," *Applied Sciences*, vol. 11, no. 13, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/13/6023>
- [38] S. H. Manoharan and W.-Y. Chiu, "Consensus based formation control of automated guided vehicles using dynamic destination approach," in *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. IEEE, 2019, pp. 902–907.
- [39] M. Wang and Y. Zhou, "Scheduling for an automated guided vehicle in flexible machine systems," in *2015 winter simulation conference (WSC)*. IEEE, 2015, pp. 2908–2916.
- [40] Z. Yang, C. Li, and Q. Zhao, "Dynamic time estimation based agv dispatching algorithm in automated container terminal," in *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018, pp. 7868–7873.
- [41] X. Tang, T. Zhou, J. Yu, J. Wang, and Y. Su, "An improved fusion algorithm of path planning for automated guided vehicle in storage system," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*. IEEE, 2018, pp. 510–514.
- [42] V. Bobanac and S. Bogdan, "Routing and scheduling in multi-agv systems based on dynamic banker algorithm," in *2008 16th Mediterranean conference on control and automation*. IEEE, 2008, pp. 1168–1173.
- [43] S. Uttendorf, B. Eilert, and L. Overmeyer, "A fuzzy logic expert system for the automated generation of roadmaps for automated guided vehicle systems," in *2016 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2016, pp. 977–981.
- [44] Q. Sun, H. Liu, Q. Yang, and W. Yan, "On the design for agvs: Modeling, path planning and localization," in *2011 IEEE International Conference on Mechatronics and Automation*. IEEE, 2011, pp. 1515–1520.
- [45] Z. Zhang, Q. Guo, and P. Yuan, "Conflict-free route planning of automated guided vehicles based on conflict classification," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 1459–1464.
- [46] G. Qing, Z. Zheng, and X. Yue, "Path-planning of automated guided vehicle based on improved dijkstra algorithm," in *2017 29th Chinese control and decision conference (CCDC)*. IEEE, 2017, pp. 7138–7143.
- [47] S. Kim, H. Jin, M. Seo, and D. Har, "Optimal path planning of automated guided vehicle using dijkstra algorithm under dynamic conditions," in *2019 7th International Conference on Robot Intelligence Technology and Applications (RiTA)*. IEEE, 2019, pp. 231–236.
- [48] C. Liu, J. Tan, H. Zhao, Y. Li, and X. Bai, "Path planning and intelligent scheduling of multi-agv systems in workshop," in *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017, pp. 2735–2739.
- [49] L. Guo, Q. Yang, and W. Yan, "Intelligent path planning for automated guided vehicles system based on topological map," in *2012 IEEE conference on control, systems & industrial informatics*. IEEE, 2012, pp. 69–74.
- [50] Y. D. Setiawan, P. S. Pratama, J. W. Kim, D. H. Kim, Y. S. Jung, S. B. Kim, S. M. Yoon, T. K. Yeo, and S. Hong, "Path replanning and controller design for trajectory tracking of automated guided vehicles," in *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2014, pp. 771–777.
- [51] S. L. Francis, S. G. Anavatti, and M. Garratt, "Real time cooperative path planning for multi-autonomous vehicles," in *2013 International*

- Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2013, pp. 1053–1057.
- [52] D. Herrero-Perez and H. Matinez-Barbera, “Decentralized coordination of autonomous agvs in flexible manufacturing systems,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 3674–3679.
- [53] P. S. Pratama, T. H. Nguyen, H. K. Kim, D. H. Kim, and S. B. Kim, “Positioning and obstacle avoidance of automatic guided vehicle in partially known environment,” *International Journal of Control, Automation and Systems*, vol. 14, no. 6, pp. 1572–1581, 2016.
- [54] I. Duleba and J. Z. Sasiadek, “Nonholonomic motion planning based on newton algorithm with energy optimization,” *IEEE transactions on control systems technology*, vol. 11, no. 3, pp. 355–363, 2003.
- [55] S. Liu and D. Sun, “Minimizing energy consumption of wheeled mobile robots via optimal motion planning,” *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 2, pp. 401–411, 2013.
- [56] A. Pal, R. Tiwari, and A. Shukla, *Modified A* Algorithm for Mobile Robot Path Planning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 183–193. [Online]. Available: https://doi.org/10.1007/978-3-642-25507-6_16
- [57] K.-w. Jeong, J. Lee, and M. C. Lee, “Energy-saving trajectory planning for an inverse ball drive robot with mecanum wheels,” *International Journal of Control, Automation and Systems*, vol. 15, no. 2, pp. 752–762, 2017.
- [58] W. Rahmaniar and A. E. Rakhmania, “Mobile robot path planning in a trajectory with multiple obstacles using genetic algorithms,” *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 1–7, 2022.
- [59] S. Dogru and L. Marques, “Towards fully autonomous energy efficient coverage path planning for autonomous mobile robots on 3d terrain,” in *2015 European Conference on Mobile Robots (ECMR)*, 2015, pp. 1–6.
- [60] M. F. Jaramillo-Morales, S. Dogru, and L. Marques, “Generation of energy optimal speed profiles for a differential drive mobile robot with payload on straight trajectories,” in *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2020, pp. 136–141.
- [61] C. H. Kim and B. K. Kim, “Energy-saving 3-step velocity control algorithm for battery-powered wheeled mobile robots,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 2375–2380.
- [62] J. Velagić, L. Vuković, and B. Ibrahimović, “Mobile robot motion framework based on enhanced robust panel method,” *International Journal of Control, Automation and Systems*, vol. 18, no. 5, pp. 1264–1276, 2020.
- [63] Y. Liu, Z. Chen, Y. Li, M. Lu, C. Chen, and X. Zhang, “Robot search path planning method based on prioritized deep reinforcement learning,” *International Journal of Control, Automation and Systems*, vol. 20, no. 8, pp. 2669–2680, 2022.
- [64] X. Xue, Z. Li, D. Zhang, and Y. Yan, “A deep reinforcement learning method for mobile robot collision avoidance based on double dqn,” in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*. IEEE, 2019, pp. 2131–2136.
- [65] L. Tai, G. Paolo, and M. Liu, “Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 31–36.
- [66] J. Gao, W. Ye, J. Guo, and Z. Li, “Deep reinforcement learning for indoor mobile robot path planning,” *Sensors*, vol. 20, no. 19, p. 5493, 2020.
- [67] J. Xin, H. Zhao, D. Liu, and M. Li, “Application of deep reinforcement learning in mobile robot path planning,” in *2017 Chinese Automation Congress (CAC)*. IEEE, 2017, pp. 7112–7116.
- [68] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, “Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.
- [69] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, “Deployment of mobile robots with energy and timing constraints,” *IEEE Transactions on robotics*, vol. 22, no. 3, pp. 507–522, 2006.
- [70] L. Hou, L. Zhang, and J. Kim, “Energy modeling and power measurement for mobile robots,” *Energies*, vol. 12, no. 1, p. 27, 2018.
- [71] H. Ding, G. Reißig, D. Groß, and O. Stursberg, “Mixed-integer programming for optimal path planning of robotic manipulators,” in *Proc. of the IEEE International Conference on Automation Science and Engineering*, pp. 133–138, 2011.
- [72] H. P. Williams, *Model building in mathematical programming*. John Wiley & Sons, 2013.
- [73] M. Razaviyayn, “Successive convex approximation: Analysis and applications,” Ph.D. dissertation, University of Minnesota, 2014.
- [74] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization*. Philadelphia: MPS-SIAM Series on Optim., SIAM, 2001.
- [75] A. Beck, A. Ben-Tal, and L. Tretuashvili, “A sequential parametric convex approximation method with applications to nonconvex truss topology design problems,” *J. Global Optim.*, vol. 47, no. 1, pp. 29–51, May 2010.
- [76] H. Tuy, *Convex analysis and global optimization*. Springer, 1998.
- [77] M. Grant and S. Boyd, “Cvx: Matlab software for disciplined convex programming, version 2.1,” 2014.
- [78] —, “CVX: Matlab software for disciplined convex programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.
- [79] —, “Graph implementations for nonsmooth convex programs,” in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.
- [80] D. Adionel Guimaraes, G. H. Faria Floriano, and L. Silvestre Chaves, “A tutorial on the cvx system for modeling and solving convex optimization problems,” *IEEE Latin America Transactions*, vol. 13, no. 5, pp. 1228–1257, 2015.