Robotics and Autonomous Systems 162 (2023) 104366

Contents lists available at ScienceDirect

Robotics and Autonomous Systems

journal homepage: www.elsevier.com/locate/robot

# Energy efficient path planning for autonomous ground vehicles with ackermann steering☆

Haojie Zhang [a,*], Yudong Zhang [a], Chuankai Liu [b,c], Zuoyu Zhang [b,c]

[a] School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing, 100083, China
[b] Beijing Aerospace Control Center, Beijing, 100094, China
[c] State Key Laboratory of Science and Technology on Space Flight Dynamics, Beijing, 100094, China

ARTICLE INFO

*Article history:*
Received 26 September 2022
Received in revised form 10 December 2022
Accepted 10 January 2023
Available online 16 January 2023

*Keywords:*
Energy efficient path planning
Ackermann steering
Energy cost model
Motion primitive

ABSTRACT

The autonomous ground vehicles have attracted a great deal of attention as viable solutions to a wide variety of military and civilian applications. However, the energy consumption plays a major role in the navigation of autonomous ground vehicles in challenging environments, especially if they are left to operate unattended under limited on-board power, such as planetary exploration, border patrol, etc. The autonomous ground vehicles are expected to perform more tasks more efficiently with limited power in these scenarios. Although plenty of research has developed an effective methodology for generating dynamically feasible and energy efficient trajectories for skid steering or differential steering vehicles, few studies on path planning for ackermann steering autonomous ground vehicles are available. In this study, an energy efficient path planning method with guarantee on completeness is proposed for autonomous ground vehicle with ackermann steering which is based on $A^*$ search algorithm. Firstly, the energy cost model is established for the autonomous ground vehicle using its kinematic constraints. Then, given the start and goal states, the energy-aware motion primitives are generated offline using the energy cost model to calculate the cost of each primary trajectory. Lastly, the energy efficient path planner is proposed and the analysis for completeness properties is given. The effectiveness of the proposed energy efficient path planner is verified by simulation over 150 randomly generated maps and real vehicle tests. The results show that a small increase in the distance of a path over the distance optimal path can result in a reduction of energy cost by nearly 26.9% in simulation and 21.09% in real test scenario for autonomous ground vehicles with ackermann steering.


## 1. Introduction

Autonomous ground vehicle is a smart vehicle capable of doing tasks without the need of a human operator. As improving the automation level and vehicle intelligence, it is widely used in military and civilian fields [1,2], such as planetary exploration, detecting bombs, border patrol, carrying cargos, search and rescue, etc. All-electric vehicles are a better choice for development in fully autonomous vehicle system since zero emissions, low noise, high usage rates of energy and simply accurate control [3]. Currently, most of all-electric vehicles are powered by rechargeable lithium-ion batteries. How long an all-electric vehicle can work on a single charge is largely based on the energy density. However, it is difficult to achieve rapid recharging in some complex scenarios where autonomous ground vehicles perform tasks, such as Mars exploration, constantly border patrol in all weather conditions. Meanwhile, it is expected that the autonomous ground vehicle can work for a long time on a single charge and reduce the frequency of recharges, such as precise farming for area coverage operations in agriculture, cargo transportation and scheduling in industry, collaborative exploration for map construction in unknown outdoor area, etc. Therefore, it becomes more important for autonomous ground vehicle to perform more tasks under the limited energy supply.

The energy consumption during autonomous ground vehicle locomotion can be reduced by selecting high efficiency motors, low power sensors, advanced energy management systems [4,5], etc. However, once these firmware are confirmed, the energy consumption is significantly impacted by autonomous planning system [6]. The optimization methods of energy consumption has been widely used for path planning for unmanned aerial vehicles (UAVs) [7] and unmanned underwater vehicles (UUVs) [8], which

☆ This document is the results of the research project funded by the State Key Laboratory, China (Grant No. KJW6142210210308) and National Natural Science Foundation of China (Grant No. 61806183, 61972020, 62003025).
* Corresponding author.
*E-mail addresses:* haojie.bit@gmail.com (H. Zhang), yudong.zhang@foxmail.com (Y. Zhang), ckliu2005@126.com (C. Liu), zyzhang1002@163.com (Z. Zhang).

https://doi.org/10.1016/j.robot.2023.104366
0921-8890/

are subject to numerous disturbances by wind or water flow during the movement. These disturbances can serve to help the UAVs and UUVs mission by planning more energy efficient paths [9]. With the increasing demand for autonomous ground vehicles in both civilian and military applications, the path planning problems considering energy consumption optimization have become a hot topic for them [10].

The earliest energy efficient path planning method for autonomous ground vehicle was proposed in 1990s, which utilized a physical energy cost model in planning process. The developed energy cost model is typically formulated as the function of gravity and friction forces operating on the vehicle. The cost between nodes in search state space is defined as the energy consumption to overcome the effects of gravity and friction forces [11]. Therefore, the energy cost of a path is the integral over the path. After that, a series of energy efficient path planning methods are proposed for autonomous ground vehicles based on this idea. An energy-minimizing path planning method is presented for autonomous ground vehicle on terrains by interconnecting the nodes by edges with appropriate weights. With some additional assumptions, some upper and lower bound path results are given on terrains [12]. However, in order to reduce the computational complexity, the uneven terrain is approximate as the flat terrain which leads to a decrease in the accuracy of the generated path. Besides, it cannot guarantee to get the global energy efficient path via this method since the elevation change of the terrain is not considered. Therefore, an energy cost model is established to calculate the autonomous ground vehicle's energy consumption, which integrates the terrain's angle of declination with the gravity and wheel-terrain friction forces [13]. Then, a variant Dijkstra algorithm is created for path planning based on the energy cost model. Unfortunately, obtaining such an energy cost model is difficult which prevents the practical applicability of path planning algorithms for energy optimization. To address this issue, another energy efficient path planning approach is presented based on the assumption that the energy consumption for the autonomous ground vehicle is correlated with ground appearance [14]. The aerial images collected by an UAV is used to generate the energy cost map of a given environment, which is further used for energy efficient coverage path planning for autonomous ground vehicle [15,16].

In other work, a series of energy efficient path planning algorithms are mainly divided into two layers. The global paths are obtained by distance optimal criterion. Then, the local path planning algorithm based on energy cost model is adopted to predict the energy consumption while following the global path. For instance, a power model is derived for skid steering wheeled vehicles, which is based on the torque limitations of the vehicle motors [17]. Then, the power model along with sampling based model predictive optimization are used to develop an effective methodology for generating dynamically feasible, energy efficient trajectories for skid steering autonomous ground vehicles. The dynamic window approach-based local trajectory planner is extended by adding an energy-related criterion for minimizing the power consumption of autonomous Mecanum robot, which evaluates the velocities of the robot [18]. The energy-reduction autonomous navigation is proposed via the combinational cost objectives of low power consumption and high speed. The energy consumption estimation based on this kind of energy cost model is inaccurate on soft terrains. Therefore, an energy-aware trajectory planning method is proposed for a planetary rover which considers vehicle dynamics and energy management of the rover. The energy cost model is established taking into account of the vehicle dynamics and wheel dynamics, such as wheel–soil interaction, reference velocity, slop angle and heading angle. The simulation results indicate that the energy margin could be improved by 13.9% at least [19].

As an important part of bionic technology, some of bionic algorithms are gradually improved to solve the problem of energy efficient path planning, such as particle swarm optimization, ant colony optimization, etc. The energy efficient path planning model is formulated via selecting transport distance and energy consumption as two optimization objectives for automated guided vehicles applied in manufacturing workshops [20]. Further, the two-stage solution method and the particle swarm optimization-based method are put forward to solve the established model for energy efficient paths. However, the energy consumption will be humongous while path planning in a dynamic scenario owing to its uncertain and dynamic nature. Therefore, a dynamic path planning strategy based on fuzzy logic and improved ant colony optimization is proposed to find the most cost-effective path in a road network for unmanned vehicles, which works well in dynamic environments [21]. Motivated by this idea, an efficient gain-based dynamic green ant colony optimization method (GDGACO) has been proposed to reduce the total energy consumed during path planning through an efficient gain function-based pheromone enhancement mechanism [22]. Comprehensive simulation experiments are conducted to demonstrate the efficacy of GDGACO through comparison with other methods in terms of path length, computation time and energy consumed.

The energy cost models are different according to the various drive system of vehicles, such as differential steering, skid steering, omnidirectional steering and ackermann steering, which lead to the variants energy efficient path planning algorithms. Ackermann steering drive, namely car-like drive, is the most common in the real world, which is characterized by a pair of driving wheels and a separate pair of steering wheels. The translation and rotation motion are independent of each other. The intention of ackermann steering is to avoid the need for tires to slip sideways when following the path around a curve. Through the literature review, few works have proposed energy efficient path planning methods for autonomous ground vehicles with ackermann steering. In our previous work [23], an energy cost model was established for differential steering vehicles by considering the mechanical and electrical energy consumption. Based on the energy cost model, we proposed an energy optimal path planning method by extending AD* algorithm. However, while previous works have commonly considered deterministic energy cost models, some parameters of the energy cost model are difficult to obtain and can only be estimated based on empirical analysis. Especially, little attention has been given to the completeness analysis of the energy optimal path planner.

In this paper, our primary contribution is an energy efficient path planning approach for autonomous ground vehicles with ackermann steering. The flow diagram of the energy efficient path planning process is illustrated in Fig. 1. First, the energy cost model is established for ackermann steering vehicle (Section 2). Then, we propose the energy efficient path planner on the basis of the extended A* algorithm and theoretical analysis of completeness is given (Section 3). The experiment is setup to test our planner using both simulation environments and real environments (Section 4). The experiment results are provided and we compare our planner's performance with the distance optimal A* search algorithm (Section 5).

## 2. Kinematic and energy cost model

### 2.1. Kinematic model

The ackermann steering mechanism is a modified four-bar linkage mechanism, which is used in car-like vehicles, as shown in Fig. 2. The basic idea consists of rotating the inner wheel

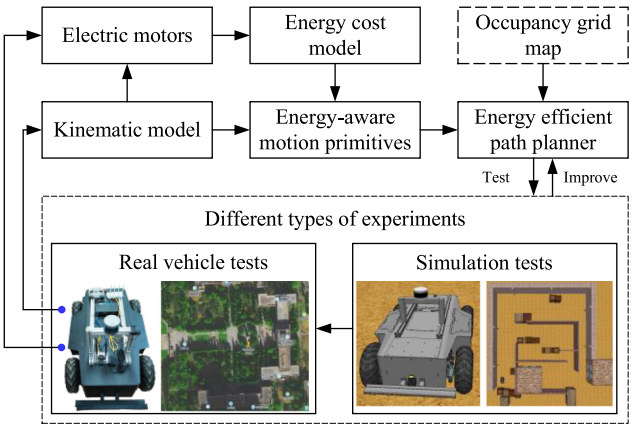



**Fig. 1.** The flow diagram of the energy efficient planner.

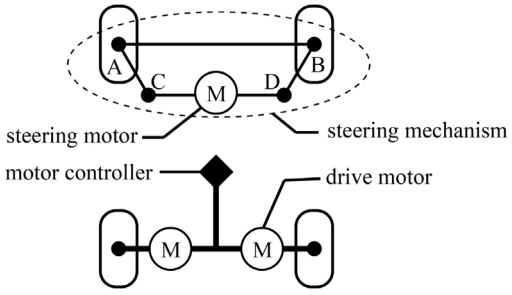



**Fig. 2.** The ackermann steering mechanism.

slightly sharper than the outer wheel to reduce tire slippage. The first two of the four wheels can be used to steer through trapezoidal connecting rod *ACDB* and the next two to drive the vehicle. In the mechanism, an electric motor is used to control the steering angle of two front wheels, and one electric motor per wheel in the back is desirable for driving. The ackermann steering system increases the steering response at high-speed corners by reducing the effect of cornering side forces and is often found in autonomous ground vehicles.

As shown in Fig. 3, we consider that an ackermann steering autonomous ground vehicle is moving about an instantaneous center of rotation (ICR). The global coordinate frame is $X_GOY_G$. The local coordinate frame, which is attached to the central point of the rear wheels axle, is denoted by $X_LM_rY_L$, where $X_L$ is the longitudinal coordinate and $Y_L$ is the lateral coordinate. The ackermann steering model can be simplified as a bicycle model, which combined the front two wheels and rear two wheels to form a virtual front wheel and a virtual rear wheel respectively, as the light gray color wheels shown in Fig. 3. The heading angle of the vehicle is represented by $\theta$ which is defined as the angle between the local longitudinal axes $X_L$ and global longitudinal axes $X_G$. The track width is represented by $d$, which is the lateral wheel separation. The wheel based is $l$, which is the longitudinal wheel separation.

The kinematic bicycle model of an ackermann steering vehicle is given by

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v_r \cos\theta \\ v_r \sin\theta \\ v_r \frac{\tan\varphi}{l} \end{bmatrix} \tag{1}$$

where $(x, y, \theta)$ is the vehicle pose which is taken with respect to the global coordinate frame. $v_r$ is the forward velocity of virtual

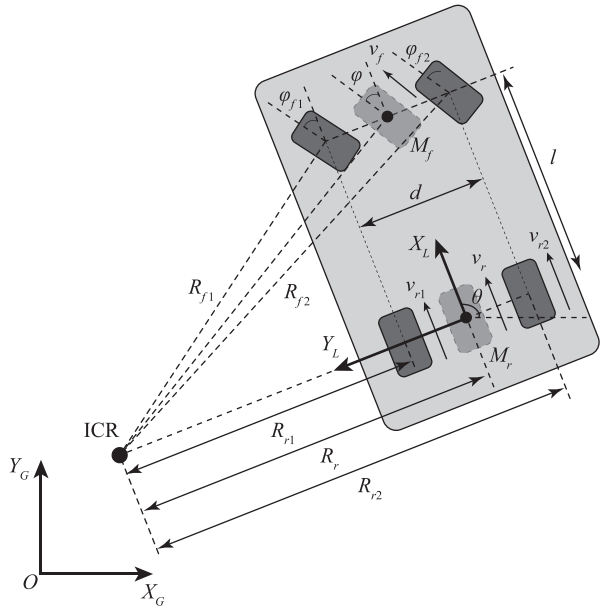



**Fig. 3.** Kinematic model of ackermann steering mechanism.

rear wheel. $\varphi$ is the relative steering angle of the virtual front wheel.

It is assumed that the front two wheels and rear two wheels are in the same motion state when simplifying the ackermann steering to an ideal bicycle model. However, the idea behind the ackermann steering is that the inner wheel should steer for a bigger angle than the outer wheel in order to allow the vehicle to rotate around the middle point between the rear wheel axis. Based on the simplified bicycle model, the ackermann steering equation can then be derivated quite easily by considering the three triangles formed by the wheel base $l$ and the side $R_r$ plus or minus $d/2$. So we get

$$\begin{cases} \cot\varphi = \frac{R_r}{l} \\ \cot\varphi_{f1} = \frac{R_r - d/2}{l} \\ \cot\varphi_{f2} = \frac{R_r + d/2}{l} \end{cases} \tag{2}$$

where $R_r$ is the distance between ICR and the center of virtual rear wheel. $\varphi_{f1}$ is the relative steering angle of the inner wheel. $\varphi_{f2}$ is the relative steering angle of the outer wheel.

Equivalently, the two cotangents can be expressed with angel $\varphi$ as follows

$$\begin{cases} \cot\varphi_{f1} = \cot\varphi - \frac{d}{2l} \\ \cot\varphi_{f2} = \cot\varphi + \frac{d}{2l} \end{cases} \tag{3}$$

These equations have a problem for the case $\varphi$=0, since cot(0) is not defined. However, when considering the fact that $\cot\varphi = \frac{\cos\varphi}{\sin\varphi}$, we can reformulate the equation as

$$\begin{cases} \varphi_{f1} = \tan^{-1}\left(\frac{2l\sin\varphi}{2l\cos\varphi - d\sin\varphi}\right) \\ \varphi_{f2} = \tan^{-1}\left(\frac{2l\sin\varphi}{2l\cos\varphi + d\sin\varphi}\right) \end{cases} \tag{4}$$

Assuming that $v_{r1}$ and $v_{r2}$ are the translational velocities of the left and right wheels respectively. Let $\omega$ be the angular velocity of the vehicle. Then, the inner and outer rear wheels circle around the ICR with the same angular velocity $\omega = v_r/R_r$. Therefore, the rim velocities for them are

$$\begin{cases} v_{r1} = \omega(R_r - \frac{d}{2}) = v_r(1 - \frac{d}{2R_r}) \\ v_{r2} = \omega(R_r + \frac{d}{2}) = v_r(1 + \frac{d}{2R_r}) \end{cases} \tag{5}$$

So, $v_r$ and $R_r$ can be solved from Eq. (5),

$$\begin{cases} v_r = \frac{v_{r1}+v_{r2}}{2} \\ R_r = \frac{d(v_{r1}+v_{r2})}{2(v_{r2}-v_{r1})} \end{cases} \tag{6}$$

Therefore, we can get $\varphi$ according to Eqs. (2) and (6)

$$\varphi = \tan^{-1}\frac{2l(v_{r2}-v_{r1})}{d(v_{r1}+v_{r2})} \tag{7}$$

Substituting Eq. (6) into Eq. (1), we can get the kinematic model of ackermann steering vehicle

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{v_{r1}+v_{r2}}{2}\cos\theta \\ \frac{v_{r1}+v_{r2}}{2}\sin\theta \\ \frac{v_{r1}+v_{r2}}{2}\frac{\tan\varphi}{l} \end{bmatrix} \tag{8}$$

where $\varphi$ can be calculated by Eq. (7) and also used to calculate the final steering angles $\varphi_{f1}$ and $\varphi_{f2}$ according to Eq. (4).

The kinematic model of Eq. (8) is used in the development of energy cost model described in the next section. In addition, for vehicle control the inverse of this model is used to determine the desired left and right wheel velocities $v_{r1}$, $v_{r2}$ and the relative steering angles $\varphi_{f1}$, $\varphi_{f2}$, given the autonomous ground vehicle pose $(x, y, \theta)$.

### 2.2. Energy cost model

Most of the energy is consumed by electric motors for ackermann steering autonomous ground vehicles, which consists of two rear drive motors and one front steering motor. The inner rear wheel angular velocity $\omega_{r1}$ and outer rear wheel angular velocity $\omega_{r2}$ are controlled to achieve the desired vehicle velocity by motor drivers. According to Eq. (2), we can get

$$R_r = \frac{l}{\tan\varphi} \tag{9}$$

Assuming that $r$ is the radius of the rear wheels. Then, we can get $\omega_{r1}$ and $\omega_{r2}$ by substituting Eq. (9) into Eq. (5)

$$\begin{cases} \omega_{r1} = \frac{v_{r1}}{r} = \frac{v_r}{r}(1 - \frac{d\tan\varphi}{2l}) \\ \omega_{r2} = \frac{v_{r2}}{r} = \frac{v_r}{r}(1 + \frac{d\tan\varphi}{2l}) \end{cases} \tag{10}$$

The battery is the power source of the autonomous ground vehicle. Assuming that $U_s$ is the battery voltage and $U_{r1}$, $U_{r2}$ are the armature voltages of the inner drive motor and outer drive motor respectively. Then, $U_{r1}$ and $U_{r2}$ are adjusted by pulse width modulation (PWM) motor driver. The motor speed is controlled by the PWM speed controller. Therefore, the armature voltages of the inner drive motor and outer drive motor can be expressed as

$$\begin{cases} U_{r1} = U_s D_{r1} \\ U_{r2} = U_s D_{r2} \end{cases} \tag{11}$$

where $D_{r1}$ and $D_{r2}$ are the duty cycles of PWM for inner drive motor and outer drive motor.

Most of the armature voltage is converted into back electromotive force, which is used to drive the motor to rotate. A small part of it is used as the voltage of internal resistance. Therefore, the armature voltage of the rear wheel motor is expressed as

$$\begin{cases} U_s D_{r1} = i_{r1}R_{ra} + K_{rb}g\omega_{r1} \\ U_s D_{r2} = i_{r2}R_{ra} + K_{rb}g\omega_{r2} \end{cases} \tag{12}$$

where $i_{r1}$, $i_{r2}$ are the armature currents of the rear wheel motors. $R_{ra}$ is the rear motor internal resistance, $K_{rb}$ is the back electromotive force constant, $g$ is the motor reduction ratio.

To simplify Eq. (12), we can get

$$\begin{bmatrix} D_{r1} \\ D_{r2} \end{bmatrix} = \alpha \begin{bmatrix} \omega_{r1} \\ \omega_{r2} \end{bmatrix} + \boldsymbol{\beta} \tag{13}$$

where,

$$\begin{cases} \alpha = K_{rb}g/U_s \\ \boldsymbol{\beta} = [i_{r1}R_{ra}/U_s \quad i_{r2}R_{ra}/U_s]^T \end{cases}$$

It can be seen from Eq. (13) that $K_{rb}$, $g$, $U_s$, $R_{ra}$ will be constant once the firmware of the ackermann steering vehicle is confirmed, such as battery, motors and reducer, etc. Therefore, $\alpha$ is constant. The motor torque is constant when the fixed load vehicle moves at a constant speed. According to the proportional relationship between the motor torque and the armature current, it can be known that $i_{r1}$ and $i_{r2}$ are also constant. Then, $\boldsymbol{\beta}$ is a constant matrix. Unfortunately, the armature current is difficult to measure. However, it shows that $\alpha$ and $\boldsymbol{\beta}$ can be approximated by least square method in [24].

The energy consumption of path is usually evaluated by the motor power consumption. Therefore, the energy cost model is established by analyzing the energy consumption of two rear drive motors and one front steering motor for ackermann steering autonomous ground vehicles. Assuming that $P_{r1}$ is the power of inner rear drive motor and $P_{r2}$ is the power of outer rear drive motor. Then, $P_{r1}$ and $P_{r2}$ can be expressed by

$$\begin{cases} P_{r1} = U_{r1}i_{r1} = U_s D_{r1} i_{r1} \\ P_{r2} = U_{r2}i_{r2} = U_s D_{r2} i_{r2} \end{cases} \tag{14}$$

$i_{r1}$ and $i_{r2}$ can be solved from Eq. (13)

$$\begin{cases} i_{r1} = (U_s D_{r1} - K_{rb}g\omega_{r1})/R_{ra} \\ i_{r2} = (U_s D_{r2} - K_{rb}g\omega_{r2})/R_{ra} \end{cases} \tag{15}$$

Substituting Eq. (15) into Eq. (14), we can get

$$\begin{cases} P_{r1} = (U_s^2 D_{r1}^2 - U_s D_{r1} K_{rb} g\omega_{r1})/R_{ra} \\ P_{r2} = (U_s^2 D_{r2}^2 - U_s D_{r2} K_{rb} g\omega_{r2})/R_{ra} \end{cases} \tag{16}$$

Therefore, the total energy consumption $E_r$ of two drive motors is

$$\begin{aligned} E_r &= \int (P_{r1} + P_{r2})dt \\ &= \int \frac{U_s^2(D_{r1}^2 + D_{r2}^2) - K_{rb}gU_s(D_{r1}\omega_{r1} + D_{r2}\omega_{r2})}{R_{ra}}dt \end{aligned} \tag{17}$$

Assuming that $\omega_f$ is the constant angular velocity of the front steering motor, $D_f$ is the duty cycles of PWM for it. According to Eq. (16), the power $P_f$ of the front steering motor can be expressed by

$$P_f = (U_s^2 D_f^2 - U_s D_f K_{fb} g\omega_f)/R_{fa} \tag{18}$$

where, $K_{fb}$ is the back electromotive force constant of front steering motor and $R_{fa}$ is the front steering motor internal resistance.

The energy consumption $E_f$ when the front steering motor rotates $\Delta\varphi$ is

$$E_f = P_f \Delta t = P_f \frac{\Delta\varphi}{\omega_f} = \frac{U_s D_f \Delta\varphi(U_s D_f - K_{fb} g\omega_f)}{R_{fa}\omega_f} \tag{19}$$

Therefore, the entire energy cost model of an ackermann steering autonomous ground vehicles is

$$\begin{aligned} E &= E_f + E_r \\ &= \frac{U_s D_f \Delta\varphi(U_s D_f - K_{fb} g\omega_f)}{R_{fa}\omega_f} + \\ &\int \frac{U_s^2(D_{r1}^2 + D_{r2}^2) - K_{rb}gU_s(D_{r1}\omega_{r1} + D_{r2}\omega_{r2})}{R_{ra}}dt \end{aligned} \tag{20}$$

where, $\omega_{r1}$ and $\omega_{r2}$ can be obtained according to Eq. (10) by given the desired forward velocity and relative steering angle of the vehicle.

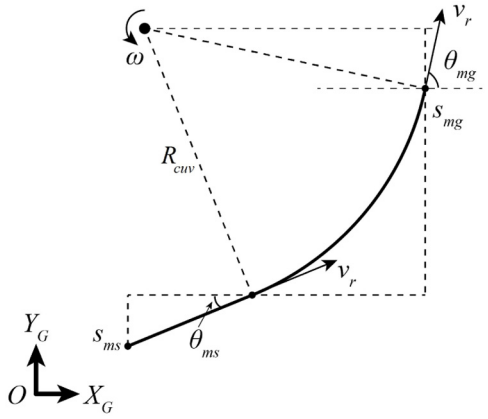



**Fig. 4.** An example of motion primitive.

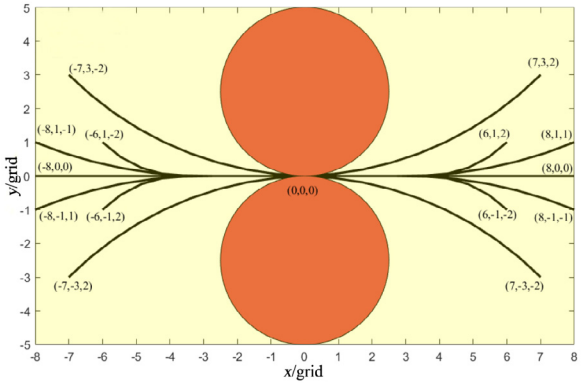



**Fig. 5.** Energy-aware motion primitives with start state (0, 0, 0).

## 3. Energy efficient path planner

Assume that the starting state and goal state of the ackermann steering autonomous ground vehicle are represented by nodes $s_{start}$ and $s_{goal}$, respectively. Then, the path planning is a computational problem to find a minimum energy cost path connecting $s_{start}$ and $s_{goal}$.

### 3.1. Energy-aware motion primitives

Path planning is usually performed in a state space which is a search graph where vertices representing kinematic states of the vehicle. The vertices are connected by motion primitives that satisfy its kinematic constraints. Therefore, planning and cost estimation can be achieved directly over these motion primitives. The motion primitives are generated offline leveraging. To simplify the generation of motion primitives, it is assumed that the translational velocity $v$ ($\sim$1.5 m/s) and angular velocity $\omega$ ($\sim$0.5rad/s) of the vehicle are constant when any motion primitive is generated, and the motion cycle is constant, represented by $T$.

The starting state and goal state of the motion primitives are represented by $s_{ms}(x_{ms}, y_{ms}, \theta_{ms})$ and $s_{mg}(x_{mg}, y_{mg}, \theta_{mg})$ respectively. For each motion primitive, the vehicle firstly performs linear motion for a period of time $t_l$, and then perform curve motion for a period of time $T - t_l$. It is obvious that $0 \leq t_l \leq T$. An example of a motion primitive is shown in Fig. 4. Especially, the motion primitive is a curve when $t_l = 0$, and a straight line when $t_l = T$. Only when $0 < t_l < T$, the motion primitive is a combined trajectory of a straight line and a curve.

According to Eq. (1), the pose of the vehicle in linear motion can be expressed as

$$\begin{cases} x(t) = x_{ms} + v_r t \cos\theta_{ms} \\ y(t) = y_{ms} + v_r t \sin\theta_{ms} \\ \theta(t) = \theta_{ms} \end{cases} \tag{21}$$

where, $0 \leq t \leq t_l$

Assume that the input control commands are constant translational speed $v_r$ and relative steering angle $\varphi$, the radius of curvature $R_{cuv}$ is therefore constant according to Eq. (9). The pose of the vehicle in curve motion can be expressed as

$$\begin{cases} x(t) = x_{ms} + v_r t_l \cos\theta_{ms} + R_{cuv}\sin\theta(t) - R_{cuv}\sin\theta_{ms} \\ y(t) = y_{ms} + v_r t_l \sin\theta_{ms} - R_{cuv}\cos\theta(t) + R_{cuv}\cos\theta_{ms} \\ \theta(t) = \theta_{ms} + v_r(t - t_l)/R_{cuv} \\ R_{cuv} \geq R_{min} \end{cases} \tag{22}$$

where $t_l \leq t \leq T$, $R_{\min}$ is the minimum turning radius.

The goal state of the motion primitive is $s_{mg}(x_{mg}, y_{mg}, \theta_{mg})$ when $t = T$. Therefore, $R_{cuv}, t_l, v_r$ can be solved by substituting the goal state $s_{mg}$ into Eq. (22). If Eq. (22) can be solved, there exists a valid motion primitive between $s_{ms}$ and $s_{mg}$. Otherwise, there is no motion primitive. The motion primitive during period of time $T$ can be generated by substituting $R_{cuv}, t_l, v_r$ into Eq. (21) and (22). It should be noted that multiple motion primitives can be calculated by given different start and goal states. Then, the energy-aware motion primitives can be generated by estimating the energy cost of motion primitives via Eq. (20).

In our application, we define a set of 14 elementary energy-aware motion primitives for each start state with map resolution 0.1 m and heading angle resolution $\pi/8$. An example of energy-aware motion primitives with (0, 0, 0) as start state is shown in Fig. 5. The red circular is the area that the vehicle cannot reach due to the minimum turning radius. The black arcs are the energy-aware motion primitives with the goal state marked.

### 3.2. Energy efficient path planner based on A*

The A* search algorithm is a simple and efficient search optimization method that can be used to find the optimal path between two nodes in a graph. Therefore, it is widely used for the shortest path finding. It is motivated by the simplicity and effectiveness of A* to optimize paths according to cost functions, which make it well-suited to the energy efficient path planning for autonomous ground vehicles. More importantly, A*-like graph search methods have the advantage to guarantee bounded path optimality, and to reduce the computational cost of planning by making use of reasonable heuristics.

The proposed energy efficient path planner is developed from A*, which aims to find a minimum energy cost path from the start state $s_{start}$ to the goal state $s_{goal}$ in a directed graph $G = (S, E)$. $S$ is a set of all states $s(x, y, \theta)$ reachable in the state space. $E$ is called the set of edges of $G$. Similar to A* algorithm, the proposed planner also uses evaluation function to compute the priorities of each node, which is expressed as

$$f(s) = g(s) + h(s) \tag{23}$$

where $g(s)$ is an energy cost of traveling from $s_{start}$ to $s$ which can be calculated using energy-aware motion primitives for each step, and $h(s)$ is a heuristic estimate of energy cost of traveling from $s$ to $s_{goal}$.

The computational cost of the path planner depends on the quality of the heuristics. If the heuristic function always returns zero, then the algorithms reduce to brute-force searches. It is usually expected that the heuristic function could exactly estimate

the cost of reaching goal state from a particular state. Unfortunately, it is unrealistic to find such heuristics in most of the real world problems. Therefore, in order to find an exact heuristic function, Dijkstra's search algorithm is firstly used offline to find the shortest path from $s_{goal}$ to $s_{start}$ in the state space. After the backward search terminates, the $g$ value $g_{bw}$ of any state can be approximated as the $h$ value when planning from $s_{start}$ to $s_{goal}$ for shortest path. Based on the planning results of backward search, we proposed a heuristic energy cost function which can be expressed as

$$h(s) = e \cdot g_{bw}(s) \tag{24}$$

where $e$ is the average energy cost when the vehicle moves 1 m in a straight line with a constant translational speed.

The main routine of the proposed energy efficient planner is explained in Algorithm 1. Similar to A* algorithm, the *OPEN* and *CLOSED* lists are also maintained during the energy efficient path planning. It starts by setting $g(s_{start})$ to 0 and assuming that all the $g$ values of other states are initially $\infty$. All the $h$ values of states are calculated using $g_{bw}(s)$, which is obtained through backward Dijkstra's search algorithm (Algorithm 1, Line 1). It pushes $s_{start}$ to an *OPEN* set and *CLOSED* set is initially $\emptyset$ (Algorithm 1, Line 2). Once the state with lowest $f$ cost value is explored, it will be removed from the *OPEN* set (Algorithm 1, Line 4). According to the connection relationship defined by energy-aware motion primitives, all of explored state's neighbors will update their $g$ cost values if later routes returns a smaller value than the existing one (Algorithm 1, Line 5–7), which is calculated by the cumulative energy cost between two states. Then, the explored state will be pushed to the *CLOSED* set and its updated neighbors which are not already in the *OPEN* set will be inserted in the *OPEN* set(Algorithm 1, Line 8–9). In the next iteration, the state in the *OPEN* set with the lowest $f$ cost value will be explored again unless the current state is the goal state. This idea can be easily implemented using a priority queue.

**Algorithm 1** Pseudocode of proposed energy efficient path planner based on A*

**Require:** $s_{start}$, $s_{goal}$, direct graph $G = (S, E)$
**Ensure:** Dijkstra_computepath($s_{goal}$,$s_{start}$), return $g_{bw}$
1: $g(s_{start}) = 0, \forall s \in S, g(s) = \infty, h(s) = e \cdot g_{bw}(s)$
2: $OPEN = \{s_{start}\}, CLOSED = \emptyset$
3: **while** ($s_{goal}$ is not expanded) **do**
4: remove $s$ with the smallest $f(s)$ from *OPEN*;
5: according to the energy-aware motion primitives, for each successor $s'$ of $s$
6: **if** $g(s') > g(s) + E_{\text{cost}}(s, s')$ **then**
7: $g(s') = g(s) + E_{\text{cost}}(s, s')$
8: move $s$ from *OPEN* into *CLOSED*
9: insert/update $s'$ in *OPEN* with $f(s') = g(s') + h(s')$
10: **end if**
11: **end while**

### 3.3. Completeness analysis

The proposed energy efficient path planner holds several theoretical properties to guarantee the completeness, which means that if at least one solution exists then the algorithm is guaranteed to find a solution in a finite amount of time.

**Definition 1.** For state $s$, $g^*(s)$ is the actual energy cost from $s_{start}$ to $s$, and $h^*(s)$ is the actual energy cost from $s$ to $s_{goal}$. Correspondingly, $f^*(s)$ is the actual energy cost from $s_{start}$ to $s_{goal}$ through state $s$. Therefore, it will hold that $g^*(s) \leq g(s)$ and $h(s) \leq h^*(s)$.

**Definition 2.** $\pi^*$ is the generated path returned from Algorithm 1. It consists of the ordered list of states $\{s_0^*, s_1^*, \ldots, s_n^*\}$ where $s_0^* = s_{start}$ and $s_n^* = s_{goal}$.

**Definition 3.** $Q_1$ is a set of states which belong to $\pi^*$ and *CLOSED* set.

**Lemma 1.** *$\exists$ state $s' \in$ OPEN set such that $g(s') = g^*(s')$*

**Proof.** At the beginning of Algorithm 1, the *OPEN* set only contains $s_{start}$. Since $g(s_{start}) = g^*(s_{start}) = 0$, Lemma 1 holds. Then $s_{start}$ is explored and moved to *CLOSED* set. $\forall\, s_i^* \in Q_1$, it holds that $s_i^* \in \pi^* \cap CLOSED$. Define $s_j^* = \{s | f(s) = \max(f(s_i)), s_i \in Q_1\}$. It is obviously that $s_j^* \neq s_{goal}$ since $s_{goal} \notin CLOSED$. Let $s'$ is the best successor of state $s_j^*$, it holds that $g(s') \leq g(s_j^*) + E_{\text{cost}}(s_j^*, s')$. Then, $s'$ is moved to *OPEN* list when $s_j^*$ is explored. Since $s_j^* \in \pi^*$, it also holds that $g(s_j^*) = g^*(s_j^*)$ and $g^*(s') = g^*(s_j^*) + E_{\text{cost}}(s_j^*, s')$, therefore $g(s') \leq g^*(s')$. According to Definition 1, it satisfies that $g(s') \geq g^*(s')$. Therefore, $g(s') = g^*(s')$. □

**Theorem 1.** *$\forall s \in S$, if the heuristic cost $h(s) = e \cdot g_{bw}(s)$, it will hold that $h(s) \leq h^*(s)$*

**Proof.** The backward Dijkstra's search algorithm is firstly used offline to find the shortest path $\pi_d^*$ from $s_{goal}$ to $s_{start}$. $\forall s \in S$, $g_{bw}(s)$ is the actual shortest distance from $s$ to $s_{goal}$.

- **case i** If $\pi^*$ is same to $\pi_d^*$. According to the definition of $e$, it is known that $e$ is the minimum energy cost when the vehicle moves 1 m since the motion primitives consist of straight and curve lines. The energy cost when vehicle moves 1 m in curve line is more than that when it moves 1 m in curve line. Therefore, $\forall s \in S$, it holds that $e \cdot g_{bw}(s) \leq h^*(s)$, i.e. $h(s) \leq h^*(s)$.
- **case ii** If $\pi^*$ is different with $\pi_d^*$. $\forall s \in S$, let $g_{bw}^*(s)$ is the actual distance from $s$ to $s_{goal}$ on $\pi^*$. Therefore, $h^*(s) \geq e \cdot g_{bw}^*(s) \geq e \cdot g_{bw}(s) = h(s)$, i.e. $h(s) \leq h^*(s)$. □

**Corollary 1.** *If Algorithm 1 is not terminated, $\forall s^* \in \pi^*$, $\exists s' \in$ OPEN such that $f(s') \leq f^*(s^*)$*

**Proof.** From Lemma 1, $\exists$ state $s' \in OPEN$, it satisfies that $g(s') = g^*(s')$. Thus, $s'$ will be a state on $\pi^*$. According to the definition of evaluation function,

$$\begin{aligned} f(s') &= g(s') + h(s') = g^*(s') + h(s') \\ &\leq g^*(s') + h^*(s') = f^*(s') \end{aligned} \tag{25}$$

Since $s^* \in \pi^*$ and $s' \in \pi^*$, therefore $f^*(s') = f^*(s^*)$. Therefore, we get that $f(s') \leq f^*(s^*)$. □

**Theorem 2.** *On a finite graph, the energy efficient path planning algorithm is completeness. The ackermann steering autonomous ground vehicle is guaranteed to reach its goal if any state reachable from the start pose has a feasible path to the goal pose.*

**Proof.** Assume Theorem 2 does not hold, then the algorithm cannot find an energy efficient path from $s_{start}$ to $s_{goal}$. One of the following conditions holds:

- **case i** The planning search is not terminated at $s_{goal}$, which is a contradiction to the search terminal condition(Line 3, Algorithm 1).
- **case ii** The planner will always keep searching and never find the path. Assume the minimum energy cost of path $\pi^*$ from $s_{start}$ to $s_{goal}$ is $f^*(\pi^*)$. The final path consists of many

**Table 1**
The parameters of the vehicle.

| Parameter | Symbol | Value | Unit |
|---|---|---|---|
| Size of vehicle (length∗width∗height) | $L*W*H$ | 1437∗760∗422 | mm |
| Track | $d$ | 700 | mm |
| Wheelbase | $l$ | 840 | mm |
| Radius of wheel | $r$ | 170 | mm |
| Minimum turning radius | $R_{\min}$ | 1250 | mm |
| Back electromotive force constant of front wheel motor | $K_{fb}$ | 0.0066 | V/r $\min^{-1}$ |
| Back electromotive force constant of rear wheel motor | $K_{rb}$ | 0.0149 | V/r $\min^{-1}$ |
| Internal resistance of front wheel motor | $R_{fa}$ | 0.448 | Ω |
| Internal resistance of rear wheel motor | $R_{ra}$ | 0.321 | Ω |
| Motor reduction ratio | $g$ | 25 | |
| Battery voltage | $U_s$ | 48 | V |
| Average energy cost per 1 m | $e$ | 200 | J |

**Table 2**
The comparison simulation results between energy efficient planner and distance optimal A*.

| Map size | Planner | Avg plan time (s) | Avg path length (m) | Avg energy cost (J) | Avg reduction rate |
|---|---|---|---|---|---|
| 1000 × 1000 | Our planner | 2.19 | 148.16 | 28791 | 26.7% |
| 1000 × 1000 | Dist. opt. A* | 2.09 | 145.46 | 39271 | 26.7% |
| 1500 × 1500 | Our planner | 3.15 | 223.75 | 43333 | 27.6% |
| 1500 × 1500 | Dist. opt. A* | 2.87 | 219.93 | 59889 | 27.6% |
| 2000 × 2000 | Our planner | 3.85 | 297.6 | 56939 | 26.3% |
| 2000 × 2000 | Dist. opt. A* | 4.81 | 292.33 | 77234 | 26.3% |

energy-aware motion primitives. Let $\delta$ be the minimum energy cost of the motion primitives. Then, the maximum number of motion primitives connecting $s_{start}$ and $s_{goal}$ is

$$M = f^*(\pi^*)/\delta \tag{26}$$

$\forall s \in S$, if the number of motion primitives connecting $s_{start}$ and $s$ is more than $M$. Then, we can get that

$$f(s) \geq g(s) \geq g^*(s) > f^*(\pi^*) \tag{27}$$

According to the Corollary 1, $\exists s' \in OPEN$ such that $f(s') \leq f^*(s^*)$, it indicates $f(s') \leq f^*(s^*) = f^*(\pi^*) < f(s)$. Let $S_M$ be the set of state $s'$ such that the number of motion primitives connecting $s_{start}$ and $s'$ is less that $M$ and $\alpha_M$ be the number of states of $S_M$. Thus, it is obvious that $s'$ will be explored, not $s$. The only reason why the planner will always keep searching is that the state $s' \in S_M$ is repeatedly explored. However, the number of path from $s_{start}$ to $s'$ is limited. Let $\rho(M)$ be the maximum number of repeatedly exploring for $S_M$. Thus, all of the states in $S_M$ will moved to *CLOSED* after exploring $\alpha_M \rho_M$. In summary, only state $s' \in S_M$ will be explored at most $\alpha_M \rho_M$ times. Then, the planner will terminate which contradicts the assumption of **case ii**.

- **case iii** The returned path is not the one with minimum energy cost after the Algorithm 1 terminates at $s_{goal}$. Assume **case iii** holds, then $f(s_{goal}) = g(s_{goal}) > f^*(s_{goal})$. According to Corollary 1, $\exists s' \in OPEN$ such that $f(s') \leq f^*(s_{goal})$ before Algorithm 1 terminates. As a result, $f(s') \leq f^*(s_{goal}) < f(s_{goal})$. By line 4 of Algorithm 1, $s'$ will be explored before $s_{goal}$ which contradicts the assumption that the Algorithm 1 terminates at $s_{goal}$. Therefore, the returned path is the minimum energy cost path since the state $s'$ with $f$ value less than $s_{goal}$ will be firstly explored. □

In summary, the Algorithm 1 will terminate on a finite graph with a solution if one exists. Therefore, the energy efficient planner is complete on finite graphs.

## 4. Experimental setup

Three types of experiments are set up to test the effectiveness of the proposed energy efficient planner. We initially tested the planner using 150 randomly generated different sizes of maps where the start and goal states are randomly given. In an offline testing process, the planner is running on these maps to find the optimal path. Then, two special simulation scenarios are designed in CoppeliaSim simulator to test the planner where the simulated environments and ackermann steering vehicle model are imported, such as battle alley and terminal logistics. The start and goal states are pre-settled in the two scenarios. After verifying the planner performed satisfactorily in the simulation environment, we ran it on our real ackermann steering vehicle.

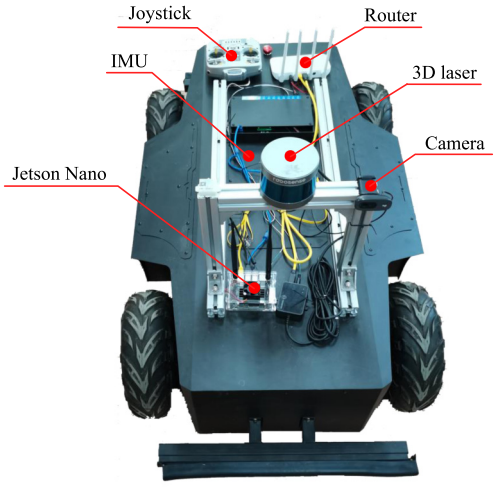



**Fig. 6.** Autonomous ground vehicle with ackermann steering used in the study.

All the simulation tests are performed in a 64-bit Ubuntu 18.04 environment running natively on a laptop with Intel(R) Core i7-8750H 2.5 GHz and 8 GB RAM. The autonomous ground vehicle with ackermann steering used in the experiments is shown in Fig. 6. The mass of the vehicle is 100 kg. The driving power is provided by an integrated 48v 20ah li-ion battery pack. In order to realize the safe autonomous driving system, the vehicle is equipped with the 3D laser range finder, IMU, camera and other attitude measurement sensors for perception.

A complete overview of the key parameters of the vehicle is given in Table 1, which are used in the energy efficient path planning. The efficiency of the electric motor is estimated by

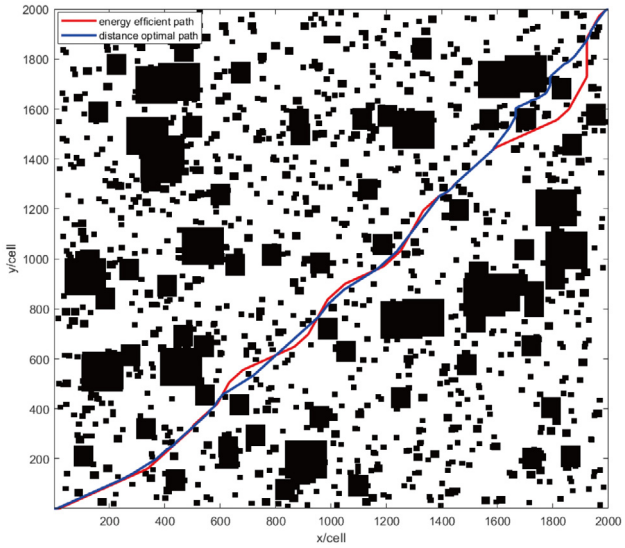



**Fig. 7.** The comparison of path planning results.

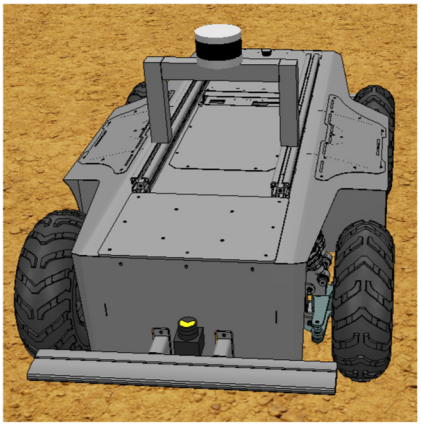
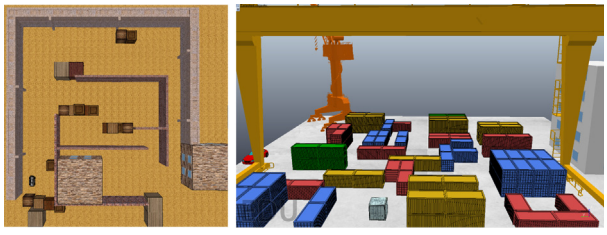


**Fig. 8.** The simulation model of vehicle.



(a) The battle alley (b) The terminal logistics

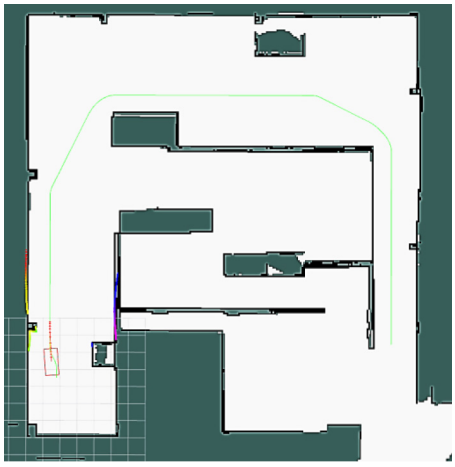
**Fig. 9.** Two simulation scenarios.

taking the ratio of average power output to average power input of the electric motor. The product of the applied voltage and current drawn by the electric motor provided the power input. Therefore, to simplify the operating state of the electric motor, the efficiency of the electric motor is assumed to be constant in this study.

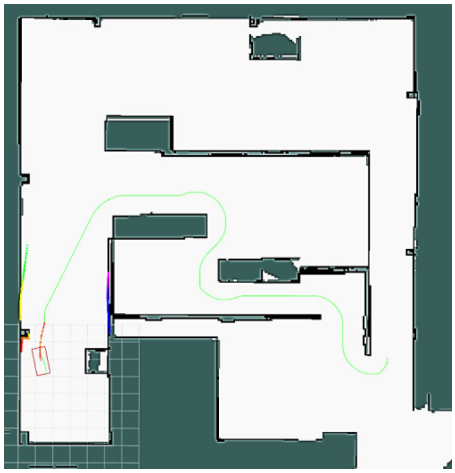
## 5. Experimental results and discussion

### 5.1. Simulation results

In simulation tests, 150 maps of three different sizes were randomly generated, such as 1000× 1000, 1500 ×1500 and 2000 ×



(a) The energy efficient path



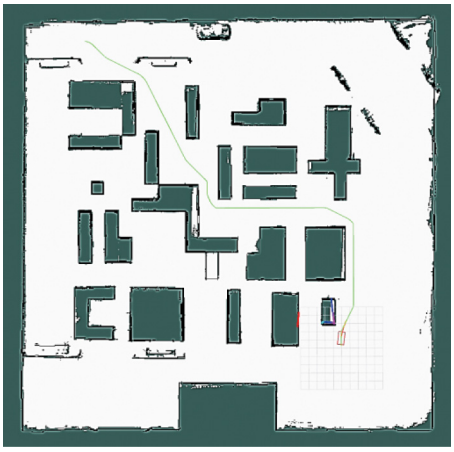
(b) The distance optimal path

**Fig. 10.** The simulation results in battle alley scenario.

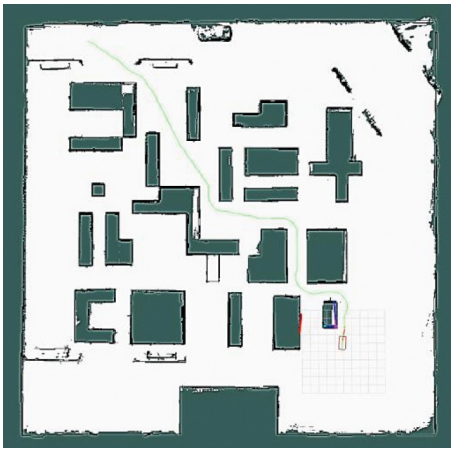
2000. There were many rectangular obstacles and single point obstacles randomly placed in the maps, as shown in Fig. 7. The maps were provided to the vehicle at the start of simulation. For these maps, the start and goal states were placed in diagonally opposite corners with a heading angle of 0°. The proposed energy efficient planner and distance optimal A* were run on each of the maps on $(x, y, \theta)$ state space. The energy cost of the path generated by distance optimal A* is calculated through accumulating the energy cost of motion primitives along it. An example of the generated paths on one map is shown in Fig. 7, where the blue solid line is the distance optimal path and red solid line is the energy efficient path. As can be seen from Fig. 7, the energy efficient path generated by the planner proposed in this study are more inclined to avoid narrow passages between dense obstacles, which aims to reduce the turning frequency and the energy cost.

The average plan time, path length, energy cost and reduction rate of the paths generated by these two planners are shown in Table 2. All statistics are based on the planning results on those maps. As can be seen our planner had comparable path lengths to the distance optimal A* search for each map size, with significantly lower path energy cost. The energy cost of path is reduced nearly 26.9% on average for each map size.

Two special simulation scenarios are designed to test the planners based on CoppeliaSim simulator, with the size of them is 50 m × 50 m. The simulation model of the vehicle used is shown in Fig. 8, which is exactly the same as the real vehicle shown in



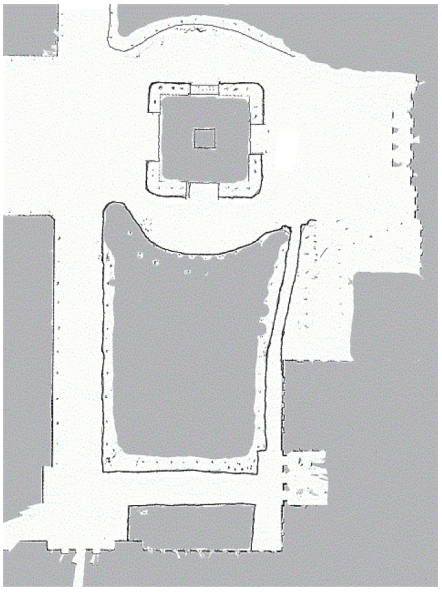
(a) The energy efficient path



(b) The distance optimal path

**Fig. 11.** The simulation results in terminal logistics scenario.

(a) The satellite map of testing place



(b) The grid map of testing place

**Fig. 12.** The test scenario for real vehicle.

Fig. 6. The battle alley scenario consists of semi-enclosed stone walls, randomly placed wooden chests and buildings, as shown in Fig. 9(a). The terminal logistics scenario is more complex, which consists of many container stacking, large gantry cranes and cranes, as shown in Fig. 9(b). Assume that the vehicle performs tasks in these two scenarios, where the start and goal states are pre-settled randomly. The 2D grid maps of the simulation scenarios are firstly built using regular SLAM algorithm and given to the planners. Then, the proposed energy efficient planner and distance optimal A* were run on these two simulation scenarios.

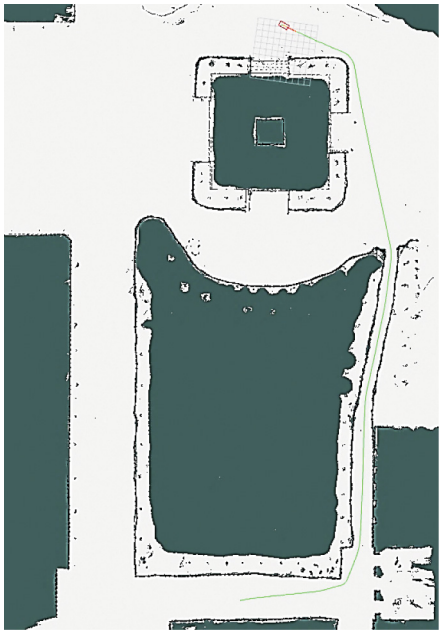
In battle alley simulation scenario, the planning simulation results are shown in Fig. 10. As can be seen, the energy efficient path has less turns than distance optimal path. The length of energy efficient path is 53.1 m, and the energy cost of it is 16,353*J* (See Fig. 10(a)). However, the length of distance optimal path is 39.5 m, and the energy cost of it is 26,786*J* (See Fig. 10(b)). The energy cost of path is reduced 38.9%.

In terminal logistics scenario, the planning simulation results are shown in Fig. 11. As can be seen, the energy efficient path trends to bypass dense obstacle groups, which is more safer. The length of energy efficient path is 56.8 m, and the energy cost of it is 17,926*J* (See Fig. 11(a)). However, the path length of distance optimal path is 53.5 m, and the energy cost of it is 26,734*J* (See Fig. 11(b)). The energy cost of path is reduced 32.9%.

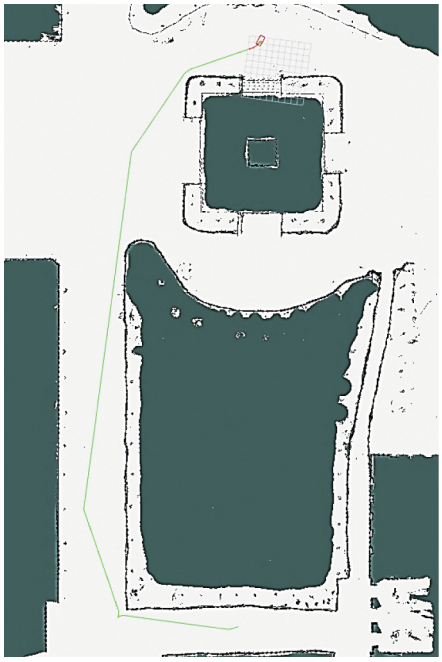
### 5.2. Experimental results for real vehicle

After verifying that the proposed energy efficient planner performed satisfactorily in the simulation environment, we ran it on our ackermann steering autonomous ground vehicle (See Fig. 6). The place for testing is located on the campus of University of Science and Technology Beijing, as shown in Fig. 12. The vehicle is firstly driven along the red route (See Fig. 12(a)) by joystick for building the grid map. A 2D occupancy grid map is generated by fusing the laser range data and odometry data, as shown in Fig. 12(b).

Then, we ran our energy efficient planner and distance optimal A* on the grid map, where the start state is set at the bottom and goal state is set at the top of the map. The path planning results are shown in Fig. 13. The distance optimal path passes through a narrow road with more turns, as shown in Fig. 13(a), which means that more energy will be used. The length of it is 142.8 m, and the energy cost is 29,951*J*. However, the energy efficient path goes through a wider road with less turns, as shown in Fig. 13(b). The path length is 2.2 m longer, but the energy cost is reduced nearly 8007*J*. The energy cost of path is reduced by 21.09% in this real test scenario. Then, after the battery is fully charged at 48v every time, the vehicle will autonomously follow the two paths separately with the maximum velocity over 1.5 m/s. During the movement, the obstacle avoidance is implemented



(a) The distance optimal path



(b) The energy efficient path

**Fig. 13.** The planning results for real vehicle test.

using teb_local_planner [25]. The remaining battery voltage are 46.3v after traversing the energy efficient path and 45.7v after traversing the distance optimal path, which indicates that the actual driving energy cost is reduced for the vehicle. Part of the experiments while the vehicle is moving along the paths are shown in Fig. 14. More vehicle motion video in the simulation and real vehicle experiments can be found at https://youtu.be/U-gzxcEDBwU.

## 6. Conclusion

This paper presented an energy efficient planner for ackermann steering autonomous ground vehicles. The energy cost model was established for the vehicle based on the kinematic constraints. Give a pair of any start state and goal state, the motion primitive between them was generated using the trajectory formation model under the assumption that the velocities of the vehicle are constant. Then, the energy cost model was used to calculate the cost of each generated motion primitives and the energy-aware motion primitives were pre-computed. The energy efficient path planner was proposed based on A* search algorithm. However, unlike A* search, the connection relationship and cost between nodes are defined by energy-aware motion primitives while searching for energy efficient path. More importantly, we can guarantee completeness of the proposed planner while being efficient enough for use as an online planner.

**Fig. 14.** Part of the experiments while the vehicle is moving.

The effectiveness of the energy efficient path planner was verified by simulation over 150 randomly generated different sizes of maps and real vehicle experiments. The results of the tests showed that a small increase in the distance of a path over the distance optimal path can result in a reduction of energy cost by nearly 26.9% in simulation and 21.09% in real test scenario for autonomous ground vehicles with ackermann steering. It also showed that distance optimal path can often cause the vehicle to turn frequently, which can result in poor navigation performance.

This study had several limitations. The established energy cost model is limited to ackermann steering vehicles. However, motivated by this idea, the energy cost models for other kinds of steering mechanisms can be derived from it through simple modification, such as differential steering, skid steering, omnidirectional steering, etc. The energy efficient planner can be also used to find an optimal and collision-free path in a 3D workspace where the uneven terrain should be modeled as 3D grid map, such as octree representation. However, it will demand for excessive computational workload for the real-time implementation in off-road applications. In dynamic environments, the energy efficient planner will replan from scratch when pop-up and moving obstacles appear in the local map. Therefore, the efficiency of the planner can result in a dramatic decrease when there is a large scale environment.

It is worth mentioning that the generated energy efficient path connecting start state and goal state is a global guided path, and the energy cost of the path is the estimated cost while the vehicle executing it. The real energy cost while the vehicle following the path is closely related to the local motion planning and control. In the future research, we would combine the global energy efficient planning with local energy efficient motion planning to reduce the real energy cost for autonomous ground vehicles with ackermann steering.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## References


[1] W. Saloni, The role of autonomous unmanned ground vehicle technologies in defense application, Aerosp. Def. Technol. Mag. (2020).
[2] B. Leonard, A. Tom, S. Henning, State of the art-automated micro-vehicles for urban logistics, Int. Federation Autom. Control 52 (13) (2019) 2455–2462.
[3] V. Algirdas, C. Olegas, M. Vadim, Motion and energy efficiency parameters of the unmanned ground vehicle, Solid State Phenom. 220-221 (2015) 934–939.
[4] E. Valjaots, R. Sell, Energy efficiency profiles for unmanned ground vehicles, Proc. Estonian Acad. Sci. 68 (1) (2019) 55–65.
[5] A. Sharif, H.M. Lahiru, S. Herath, et al., Energy efficient path planning of hybrid fly-drive robot using $A^*$ algorithm, in: Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics, 2018, pp. 211–220.
[6] B.H. Rafael, Design of High-Speed Robots with Drastically Reduced Energy Consumption, ComUE University Bretagne Loire, France, 2019.
[7] D. Hong, S. Lee, Y.H. Cho, et al., Energy-efficient online path planning of multiple drones using reinforcement learning, IEEE Trans. Veh. Technol. 70 (10) (2021) 9725–9740.
[8] N. Yang, D. Chang, M.J. Roberson, et al., Energy-optimal path planning with active flow perception for autonomous underwater vehicles, in: Proceedings of the IEEE International Conference on Robotics and Automation, 2021, pp. 9928–9934.
[9] H. Niu, Y. Lu, A. Savvaris, et al., An energy-efficient path planning algorithm for unmanned surface vehicles, Ocean Eng. 161 (2018) 308–321.
[10] G. Carabin, E. Wehrle, R. Vidoni, A review on energy-saving optimization methods for robotic and automatic systems, Robotics 6 (39) (2017) 1–21.
[11] N.C. Rowe, R.S. Ross, Optimal grid-free path planning across arbitrarily contoured terrain with anisotropic friction and gravity effects, IEEE Trans. Robot. Automat. 6 (5) (1990) 540–553.
[12] Z. Sun, J.H. Reif, On finding energy-minimizing paths on terrains, IEEE Trans. Robot. 21 (1) (2005) 102–114.
[13] M. Saad, A.I. Salameh, S. Abdallah, Energy-efficient shortest path planning on uneven terrains: A composite routing metric approach, in: Proceedings of the IEEE Symposium on Signal Processing and Information Technology, 2019, pp. 1–6.
[14] M.H. Wei, V. Isler, Air to ground collaboration for energy-efficient path planning for ground robots, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2019, pp. 1949–1954.
[15] M.H. Wei, V. Isler, Coverage path planning under the energy constraint, in: Proceedings of the IEEE/RSJ International Conference on Robotics and Automation, 2018, pp. 368–373.
[16] C. Peng, V. Isler, Adaptive view planning for aerial 3D reconstruction, in: Proceedings of the IEEE/RSJ International Conference on Robotics and Automation, 2019, pp. 2981–2987.
[17] N. Gupta, C. Ordonez, Z.G. Collins, Dynamically. feasible, Energy efficient motion planning for skid-steered vehicles, Auton. Robots 41 (2) (2017) 453–471.
[18] L. Xie, C. Henkel, K. Stol, et al., Power-minimization and energy-reduction autonomous navigation of an omnidirectional mecanum robot via the dynamic window approach local trajectory planning, Int. J. Adv. Robot. Syst. 15 (1) (2018) 1–12.
[19] G. Sakayori, G. Ishigami, Energy-aware trajectory planning for planetary rovers, Adv. Robot. 35 (2021) 1302–1316.
[20] Z.W. Zhang, L.H. Wu, W.Q. Zhang, et al., Energy-efficient path planning for a single-load automated guided vehicle in a manufacturing workshop, Comput. Ind. Eng. 158 (2021) 107397, 1-17.
[21] Q. Song, Q. Zhao, S. Wang, et al., Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization, IEEE Access 8 (2020) 62107–62115.
[22] V. Sangeetha, R. Krishankumar, K.S. Ravichandran, et al., Energy-efficient green ant colony optimization for path planning in dynamic 3D environments, Soft Comput.-A Fusion Found., Methodol. Appl. 25 (6) (2021) 4749–4769.
[23] H.J. Zhang, Z.B. Su, B. Su, et al., Energy optimal path planning for mobile robots based on improved AD* algorithm, Trans. Chin. Soc. Agric. Mach. 49 (9) (2018) 19–26.
[24] D.X. Gao, J. Chen, J.F. Bai, Application of least square in test and analysis of motor, Micromotors 50 (3) (2017) 9–13.
[25] C. Rösmann, F. Hoffmann, T. Bertram, Integrated online trajectory planning and optimization in distinctive topologies, Robot. Auton. Syst. 88 (2017) 142–153.


**Haojie Zhang** is an associate professor at the School of Automation and Electrical Engineering, University of Science and Technology Beijing, China. He has received his bachelor degree from Central South University in 2008, and his Ph.D. degree in vehicle engineering from Beijing Institute of Technology in 2013. His research interest covers path planning and tracking control for autonomous ground vehicles.

**Yudong Zhang** is with the Geekplus Corp., Beijing, China. He has received his bachelor degree from Northeastern University at Qinhuangdao in 2019, and his master degree from University of Science and Technology Beijing in 2022. His research interest includes path planning and multirobot scheduling for autonomous ground vehicles.

**Chuankai Liu** is a senior engineer and the director of space teleoperation innovation team with the Beijing Aerospace Control Center, Beijing, China. He is also a supervisor at Jiangxi University of Science and Technology. He received the B.Sc. degree in Measurement, Control Technology and Instrumentations from the University of Electronic Science and Technology of China in 2006, and the Ph.D. degree in robotics and artificial intelligence from the Institute of Automation, Chinese Academy of Sciences in 2011. His current research interests include manipulation planning, vision guided robotics, multi-agent systems and intelligence fusion.

**Zuoyu Zhang** is an engineer with the Beijing Aerospace Control Center, Beijing, China. He received the B.S. degree in measurement and control technology and instruments from the Harbin University of Science and Technology in 2013, and the Ph.D. degree in control science and engineering from the High-Tech Institute of Xian in 2020. His research interests include hyperspectral image processing, computer vision, and machine learning.