

Optimal Motion Planning for Minimizing Energy Consumption of Wheeled Mobile Robots

Roger Datouo¹, Achille Melingui², Othman Lakhal³, Frédéric Biya Motto¹,
Bernard Essimbi Zobo¹, and Rochdi Merzouki³

Achille Melingui, Electrical and Telecommunications Engineering Department, Ecole Nationale Supérieure Polytechnique, the University of Yaoundé 1, P. O. Box: 8390 Yaoundé, Cameroon

Corresponding Author: Roger Datouo

ABSTRACT: This paper presents a novel approach of finding energy-efficient trajectories for mobile robots. The approach integrates new cost and heuristic functions into the conventional A* algorithm while considering ground conditions and obstacle positions. The resulting planner helps to manage obstacle avoidance and to choose intelligent displacements of the robot. A heuristic function with energy-related criterion is defined in order to generate energy-efficient paths. η^3 -Splines continuity property is exploited to generate smoothed energy-paths. The optimal velocity profile for minimum travel time is found by solving Sequential Quadratic Problem (SQP). A series of simulations demonstrate the energy saving efficiency of the proposed method.

Keyword: mobile robots, Minimum energy consumption, motion planning, and A* algorithm.

Date of Submission: 20-10-2019

Date of acceptance: 03-11-2019

I. INTRODUCTION

Energy saving techniques for mobile robots abound in the literature [1, 2, 3, 4, 5]. They empower mobile robots to perform more complex and long-lasting missions. Despite the existence of various energy saving techniques [1, 2, 3, 4], minimizing the energy consumption of mobile robots remains a real challenge. In [3], an energy-saving approach for mobile robots by avoiding torque saturation that generally occurs at the wheels DC motors while climbing hills was proposed. A predictive control was implemented to solve the torque saturation problem. Recently, motion planning has emerged as one of the best ways to minimize energy consumption of mobile robots. As a result, velocity planning which could save battery energy by up 5% compared with the widely used trapezoidal velocity profile was proposed in [4]. [6] compared the energy consumption of different routes at different velocities by considering the energy consumed for accelerations and turns. Other contributions based on the optimal motion planning approach and using various energies criteria were also proposed [4, 2].

Among existing contributions in energy savings, based on effective motion planning, some stand out thanks to the choice of the energy criterion. The shortest path length criterion was used by many researchers to minimize energy consumption [7, 8, 9, 10, 11, 12]. However, the shortest route may not necessarily result in minimum energy consumption. Some factors such as the surface of the navigation or the shape of the planned trajectory might affect significantly the energy consumption. The reduction of the steering actuation was used in [11] as an energy criterion for mobile robots energy minimization. A smoothness criterion depending on the acceleration was used in [12] for energy saving. However, in the aforementioned techniques, a model that can be used to simulate the energy consumption of the mobile robot was not fully investigated. An interesting technique was proposed in [2]. From an energy consumption model of a two-wheeled mobile robot, the authors defined an energy-related criterion. However, the proposed algorithm does not always find the energy-saved path.

To overcome the aforementioned problems, this work integrates new cost and heuristic functions into the

¹Physics Department of Faculty of sciences, the University of Yaoundé 1

²Electrical and Telecommunications Engineering Department of Ecole Nationale Supérieure Polytechnique, the University of Yaoundé 1, Cameroon

³Polytech'Lille, CRISTAL, CNRS-UMR 9189, Avenue Paul Langevin, 59655 Villeneuve d'Ascq, France

conventional A* algorithm. The former helps to manage obstacle avoidance with minimum energy cost, and the last one to better deal with ground conditions. The number of turns of the generated paths is also reduced. The proposed approach can be summarized in three main steps. Energy consumption model of the mobile robot is first developed. Based on the energy model, a new energy-related criterion is defined and collision-free paths are generated, thereafter. Finally, the waypoints are selected and η^3 -splines with optimal shaping parameters are used to smooth the energy-saved paths.

The remainder of the paper is organized as follows. The energy model of a Three-Wheeled Omnidirectional mobile robot (TOMR) is developed in Section II. In Section III, an A* path planner with new cost and heuristic functions is first introduced and η^3 -splines for path smoothing is presented, thereafter. Simulation results are provided in Section IV, and concluding remarks are given in Section V.

II. ENERGY MODEL OF THREE-WHEELED OMNIDIRECTIONAL MOBILE ROBOT

In this section, an energy consumption model for a three-wheeled omnidirectional mobile robot (TOMR) is developed. The mobile robot named Robotino and shown in Figure 1 is used as a case study. It is a holonomic mobile robot with three omnidirectional drive units mounted at an angle of 120 degree to each other. The kinematic equations are first derived and the energy consumption model is developed, thereafter.

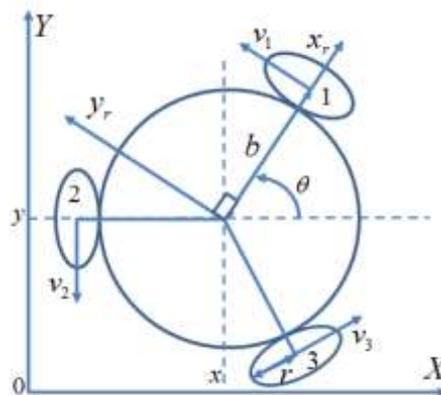


Figure 1: Kinematic model of a TOMR

2.1 Kinematics of the three-wheeled omnidirectional mobile robot

Figure 1 depicts the structure of a TOMR that is driven by three identical DC motors. $q = [x \ y \ \theta]^T$ represents the robot coordinates, where x and y denote the linear coordinates relative to global frame, and θ its orientation with respect to the X axis. The kinematic equations of the TOMR can be expressed as follows:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} -\sin \theta & \cos \theta & b \\ -\sin(\pi/3 - \theta) & -\cos(\pi/3 - \theta) & b \\ \sin(\pi/3 + \theta) & -\cos(\pi/3 + \theta) & b \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (1)$$

where $[V_1 \ V_2 \ V_3]^T$ is the vector of the linear velocities of the wheels. The linear and angular velocities of the robot can be obtained from Figure 1 as:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (2)$$

By reversing (2) and combining with (1), the equation establishing the relationship between the velocities of the wheels and those of the robot is derived.

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 0 & b \\ -\sin(\pi/3) & b \\ \sin(\pi/3) & b \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3)$$

2.2 Energy consumption model of the three-wheeled omnidirectional mobile robot

In TOMR, the motors and other components such as sensors, on-board computers, and electric circuits are the main sources of energy consumption [13]. In the following subsection, the energy consumption model is

derived by considering the losses in DC motor, the losses due to transform back the kinetic energy to the equivalent electric energy, the losses due to friction, and the losses in the electronic components.

2.2.1 Losses in DC motor

The motor purpose is to draw current from the battery and convert the electrical energy to mechanical energy. However, this conversion is not perfect; there is a loss in the armature resistance, friction, windage, etc. The energy loss in the armature resistance being most significant is that it is taken into account. The losses due to friction, windage, etc. are supposed negligible. Thus, the energy loss in motors noted E_m can be expressed as follows:

$$E_m = \frac{1}{R_a} \int_t \left(\sum_{i=1}^3 \left(U_i - \frac{K_b V_i}{r} \right)^2 \right) dt \quad (4)$$

where R_a is the armature resistance, U_i is the motor voltage, K_b is the back electromagnetic force constant, V_i is the linear velocity of the i^{th} wheel, and r the wheel radius. We assume three motors powered with same voltage U , by including (3) in (4), losses in DC motor are related to linear and angular velocities of robot by:

$$E_m = \frac{1}{R_a} \int \left(3U^2 - 6 \frac{bK_b U w(t)}{r} + \frac{K_b^2}{r^2} (3b^2 w(t)^2 + 32v(t)^2) \right) dt \quad (5)$$

2.2.2 Kinetic Energy Losses

A part of the available output energy from motors is used to increase kinetic energy and accelerate the robot. During deceleration phase, the kinetic energy could be transform back, but this is not effective, part of it dissipates into heat. So the kinetic energy loss equation can be expressed by following :

$$E_k = \int_t d(12mv(t)^2 + 12Iw(t)^2) \quad (6)$$

where, m and I are mass and moment of inertia of the robot, respectively, $v(t)$ and $\omega(t)$ are linear and angular velocities, respectively, $d(\cdot)$ is derivative operator considered for it positives values only.

2.2.3 Losses due to frictions

A second part of the available output energy is used to overcome the rolling friction or rolling resistance which is caused by the slight deformation of the ground or the wheel at the point of contact. The powers necessary to overcome the friction of each wheel is expressed as:

$$\begin{bmatrix} P_{F_1} \\ P_{F_2} \\ P_{F_3} \end{bmatrix} = \mu mg \begin{bmatrix} |V_1| \\ |V_2| \\ |V_3| \end{bmatrix} \quad (7)$$

where μ is the rolling friction coefficient that depends on the surface type of the ground, m is the mass of robot, g denotes gravitational acceleration. The total friction power P_f is obtained by summing the individual power. Substituting (3) into (7), the energy loss in friction can be expressed as:

$$\begin{aligned}
 E_F &= \int_t \mu mg \left(\begin{aligned} & \left| b\omega(t) + b\omega(t) - \frac{\sqrt{3}}{2} v(t) \right| \\ & + \left| b\omega(t) + \frac{\sqrt{3}}{2} v(t) \right| \end{aligned} \right) dt \\
 &= \begin{cases} \mu mg \int_t \left(\left| b\omega(t) + 2 \left| \frac{\sqrt{3}}{2} v(t) \right| \right| \right) dt, & \left| \frac{\sqrt{3}}{2} v(t) \right| \geq |b\omega(t)| \\ \mu mg \int_t \left(|b\omega(t)| + 2|b\omega(t)| \right) dt, & \left| \frac{\sqrt{3}}{2} v(t) \right| < |b\omega(t)| \end{cases} \quad (8) \\
 &= \mu mg \int_t \left(|b\omega(t)| + 2 \max \left\{ |b\omega(t)|, \left| \frac{\sqrt{3}}{2} v(t) \right| \right\} \right) dt
 \end{aligned}$$

2.2.4 Losses in the electronics components

A robot system has on-board electronic components which usually include the DC motor drivers, sensors and micro-controllers. The energy consumption of these components within a unit of time is relatively stable and can be presented as the power consumption P_e . The latter can be experimentally measured. So, the energy consumed by on-board electronic components is given by:

$$E_e = \int_t P_e dt \quad (9)$$

2.2.5 Energy consumption model of TOMR

Finally, the energy model of TOMR in terms of energy consumption in motors and electronic components can be expressed as:

$$\begin{aligned}
 E_T &= E_m + E_k + E_F + E_e \\
 &= \frac{1}{R_a} \int \left(\begin{aligned} & 3U^2 - 6 \frac{bK_b U w(t)}{r} \\ & + \frac{K_b^2}{r^2} \left(3b^2 w(t)^2 + \frac{3}{2} v(t)^2 \right) \end{aligned} \right) dt \\
 &+ \int_t d \left(\frac{1}{2} m v(t)^2 + \frac{1}{2} I w(t)^2 \right) dt \quad (10) \\
 &+ \mu mg \int \left(|b\omega(t)| + 2 \max \left\{ |b\omega(t)|, \left| \frac{\sqrt{3}}{2} v(t) \right| \right\} \right) dt \\
 &+ \int P_e dt
 \end{aligned}$$

III. EFFICIENCY MOTION PLANNING FOR ENERGY MINIMIZATION

In this section, the modified A* algorithm with new cost and heuristic functions is first introduced and η^3 -splines with optimal shaping parameters for path smoothing is presented, thereafter.

3.1 Modified A* algorithm

A* algorithm has been widely used in path planning [2, 11, 12]. In robotics, path planning consists of finding successive states (cells) on a grid map that allows the robot to move from an initial state (start) to a final state (goal) by avoiding obstacles [14]. A* algorithm plans the path on a grid map. Each grid constitutes a node that can be free or occupied by an obstacle. The path is planned from a start node (S_{start}) to a goal node (S_{goal}),

with $s_{\text{start}}, s_{\text{goal}} \in S$, and S being the possible set of robot locations [15]. Let s be the actual node, an evaluation function $f(s')$ is used to determine which node should be expanded next. The next node denoted s' is one of the successors of the node s . The actual node has a total of eight successors. The values $f(s')$ of each successor of the node s are calculated and the successor with the smallest value of $f(s')$, denoted s' , is considered. The evaluation function $f(s')$ is the sum of two functions, a heuristic function $h(s', s_{\text{goal}})$ that represents the estimated cost of an optimal path from the node s' to the node s_{goal} , and a function $g(s')$ representing the actual cost of the path from the node s_{start} to the node s' passing through s node [16].

$$f(s') = g(s') + h(s', s_{\text{goal}}) \quad (11)$$

The values of $g(s')$ are derived from the value of $g(s)$ as follows:

$$g(s') = g(s) + c(s, s'), \quad (12)$$

where $c(s, s')$ represents the cost to move from node s to node s' . This cost can be designed to suit the need [15]. The travel distance was widely used as a cost function to minimize energy consumption [8, 9, 10, 11, 12]. To plan energy-saved paths, a valuable contribution was proposed in [2] where, a part of total energy representing loss due to friction zones has been used in path finding. The cost function (13) and the heuristic function (14) were defined, and a penalty factor $\rho(s') \in [0,1]$ was integrated into the cost function to maintain a safety distance from obstacles.

$$c(s, s') = \sqrt{3} \mu_{s,s'} mg \left(\int_{t_s}^{t_{s'}} |v(t)| dt \right) \times \frac{1}{\rho(s')} \quad (13)$$

$$= \sqrt{3} \mu_{s,s'} mg d_{s,s'} \frac{1}{\rho(s')}$$

$$h(s', s_{\text{goal}}) = \sqrt{3} \mu_{s,s'} mg d_{s',s_{\text{goal}}}, \quad (14)$$

Where $\mu_{s,s'}$ and $d_{s,s'}$ are the friction coefficient and distance between the two nodes s and s' , respectively.

The penalty factor related to the distance to the obstacle is given by:

$$\rho(s') = \begin{cases} 1, & \lambda'_s > \lambda_{\text{safe}} \\ \frac{\lambda'_s - b}{\lambda_{\text{safe}} - b}, & b < \lambda'_s \leq \lambda_{\text{safe}} \\ 0, & \lambda'_s \leq b \end{cases} \quad (15)$$

where λ'_s is the distance of the cell s' to the nearest obstacle, λ_{safe} is the safety distance defined for safe motion of the robot [2], b is the half size of the robot. However, the fact that in the heuristic function (14), the friction ($\mu_{s,s'}$) is multiplied by the distance from s' node to goal node s_{goal} suggests that $\mu_{s,s'}$ extends up to goal node; which is not always the case. In this paper, a new heuristic function (20) is defined to better deal with friction zones.

To reduce path turn and numbers of evaluated nodes of traditional A*, we introduce an angular penalty factor in cost function. For this purpose, we investigate three functions name as square (16), sine (17) and gaussian (18). Figures 2,3 and 4, show the effect of these functions in the grid map. The latter passes from a flat surface (node plane) to a hollow surface allowing the robot to move in the same direction. The penalty functions affect higher costs to nodes that are not in the robot direction and lower costs to those are placed in its direction.

$$\phi_1(\alpha) = \frac{19}{20} - \sqrt{1 - \alpha^2} \quad (16)$$

$$\phi_2(\alpha) = \frac{19}{20} - \cos\left(\frac{\pi}{2}\alpha\right) \quad (17)$$

$$\phi_3(\alpha) = \frac{19}{20} - e\left(-\frac{\alpha^2}{8}\right) \quad (18)$$

Where $\alpha \in [-1,1]$ is a normalized angle.

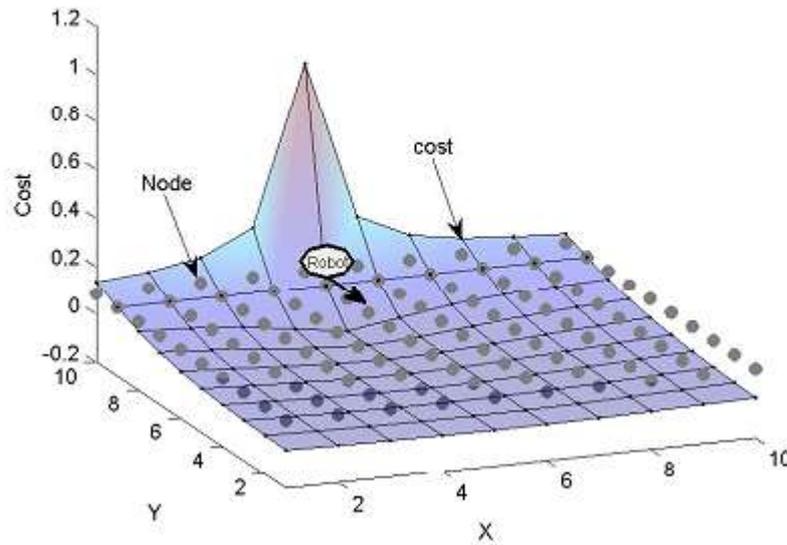


Figure 2: Turning angle penalty Square function (16)

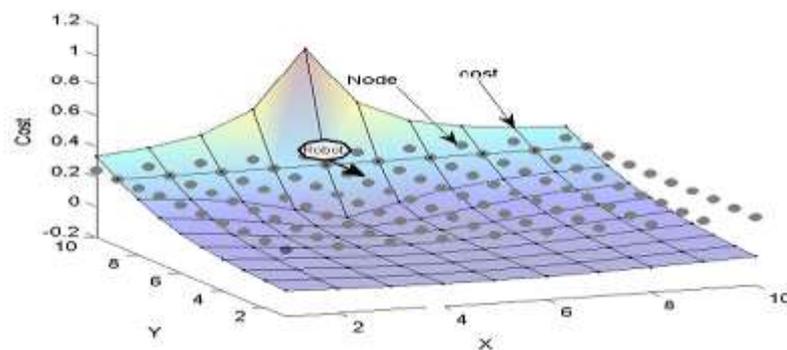


Figure 3: Turning angle penalty Sine function (17)

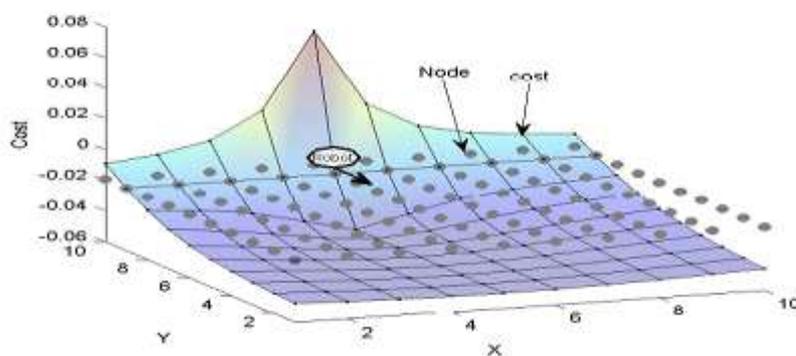


Figure 4: Turning angle penalty Gaussian function (18)

All functions that have same sharp as those presented above can help traditional A* reduce numbers of turn, evaluated nodes as well as simulation time. Simulation results of these function on A* algorithm for minimum distance are shown in section IV. The cost function used in this paper is defined as follows:

$$c(s, s') = \sqrt{3}\mu_{s,s'}mgd_{s,s'} \left(\phi(\alpha) + \frac{1}{\rho(s')} \right) \quad (19)$$

where $\alpha \in [-1,1]$ is the normalized angle formed by the segment s' s and the previous direction of the robot Figure 7, $\rho(s')$ is a penalty factor related to the distance to the obstacles (15).

The heuristic function is modified as follows: the node goal s_{goal} introduced in Liu and Sun's distance $d_{s',s_{goal}}$ is

replaced by a neighbor of the node s' with minimum heuristic value denotes s'' , and its heuristic $h(s'', s_{goal})$ is added to heuristic function (20).

$$h(s', s_{goal}) = \sqrt{3} \mu_{s', s''} \text{mgd}_{s', s''} + h(s'', s_{goal}), \quad (20)$$

where $h(s'', s_{goal})$ is the minimum heuristic function value of the successors of the node s' .

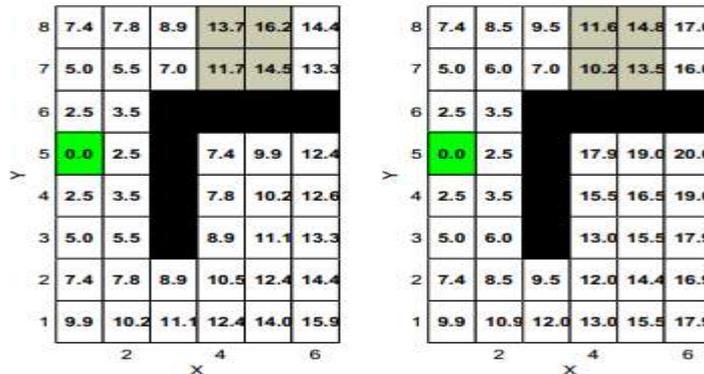


Figure 5: Heuristic value: (left) Liu and Sun (14) and (right) Proposed (20)

Figure 5 represents on a 6x8 grid map, the heuristic values of the proposed heuristic function (right-hand side) and Liu and Sun's heuristic function (left-hand side). In both diagrams, grey spaces represent friction zones, the black boxes are the obstacles, and the green circle is the goal node. We notice that the values obtained from Liu and Sun's heuristic function are very large for friction zones, which discriminate passages through the latter, even if passages through them yield a minimum energy consumption. However, we observe reasonable values in the proposed heuristic function giving a good estimation of friction zones.

In short, an optimal path with minimum energy consumption can be obtained by using the proposed cost function (19) with complete Algorithm 1.

Algorithm 1 Modified A* algorithm

```

procedure MAIN
    DEFINEHEURISTIC( $s_{goal}$ )           Start at  $s_{goal}$  with
     $h(s_{goal}, s_{goal}) = 0$  and extend to its neighbors (20)
    until map are covered.
     $g(s \in S) \leftarrow \infty$ ;
     $g(s_{start}) \leftarrow 0$ ;
    OPEN  $\leftarrow \emptyset$ ;
    insert  $s_{start}$  into OPEN ;           With value
    ( $g(s_{start}) + h(s_{start}, s_{goal})$ )
    COMPUTEOPTIMALPATH;
end procedure
function COMPUTEOPTIMALPATH
    while  $argmin_{s \in OPEN} (g(s) + h(s, s_{goal})) \neq$ 
     $s_{goal}$  do
        remove state  $s$  from the front of OPEN;
        find Succ( $s$ ) and get associated cost (19);
        for all  $s' \in Succ(s)$  do
            if  $g(s') > g(s) + c(s, s')$  then
                 $g(s') \leftarrow g(s) + c(s, s')$ ;
                insert  $s'$  into OPEN;     With value
                ( $g(s') + h(s', s_{goal})$ )
            end if
        end for
    end while
    return
end function
    
```

3.2 η^3 -Splines smooth path with minimum energy

This subsection focuses on the smoothing of the generated paths while minimizing energy consumption. The logical step following the path generation is the tracking of the latter with a minimum time. Thus, in order to obtain continuous paths, η^3 -Splines, thanks to its third order geometric continuity with continuous tangent vector, curvature, and curvature derivative along the arc length, was used for path smoothing. The smoothing can be summarized in three steps (see Algorithm 2): the selection of the Waypoints along the path generated, the selection of the optimal shaping parameters via an optimization problem, and the tracking of the smoothed path with minimum time through another optimization problem.

The Waypoints are selected on the generated path as follows: The start and goal positions as well as closed neighbors of knee points along the path are selected. Two neighboring waypoints are combined if they are close enough, and new waypoints may be inserted if they are extremely distant. The orientation $\theta \in [0, 2\pi)$ at each waypoint is set along the next segment of the generated path Figure 6; concerning the endpoint, the robot orientation is set to the previous orientation.

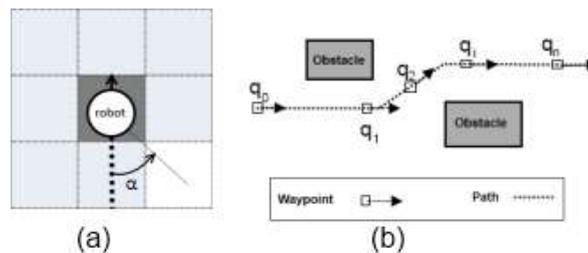


Figure 6: (a) Turning angle: gray cell s , white cell s' , dot line is the previous direction; (b): Waypoints selection

η^3 -Splines are seventh order polynomial curves which smoothly connect two arbitrary points $q_{i-1} = [X_{i-1} \ Y_{i-1} \ \theta_{i-1}]^T$ and $q_i = [X_i \ Y_i \ \theta_i]^T$ where $[X, Y]$ and θ are the robot position and orientation, respectively. η^3 -Splines is parameterized by: $q(u) = [x(u) \ y(u)]^T, u \in [0, 1]$ with $x(u)$ and $y(u)$ defined as:

$$x(u) = a_0 + a_1u + a_2u^2 + a_3u^3 + a_4u^4 + a_5u^5 + a_6u^6 + a_7u^7 \quad (21)$$

$$y(u) = b_0 + b_1u + b_2u^2 + b_3u^3 + b_4u^4 + b_5u^5 + b_6u^6 + b_7u^7, \quad (22)$$

where the polynomial coefficients $a_i, b_i, i = 0, \dots, 7$ depend on six shaping parameters $\eta_i, i = 1, \dots, 6$ [17].

These parameters influence only the path shape; the endpoints interpolating conditions i.e. the position, the unit tangent vector, the curvature, and the curvature derivative, remain unchanged.

To maintain a safe distance to the obstacle, the curvatures of η^3 -splines curves are constrained to the maximal bound k_{max} while the path length is minimized [18]. This constitutes the first optimization problem, defined as follows:

$$\min_{\eta \in \mathbb{R}^6} s_q \quad (23)$$

$$|k(u)| \leq k_{max}, u \in [0, 1], \quad (24)$$

where s_q is the arc length of a spline curve between two endpoints. The curvature $k(u)$ with respect to u is described in (25)

$$k(u) = \frac{(x\ddot{y} - \dot{x}\dot{y})}{(x^2 + y^2)^{3/2}} \quad (25)$$

The above problem is a sequential quadratic problem (SQP), its resolution leads to the selection of optimal shaping parameters.

The second optimization problem concerns the tracking of the smoothed paths with minimum time. Let $d(t)$, the arc length measured along the smoothed path at time t , d_f the total length of the smoothed path, and t_f the time required to travel the smoothed path. To minimize energy along this path, t_f need to be minimal while satisfying the velocity and acceleration constraints. This problem has been addressed in detail in [19]. The smoothed path is divided in $N - 1$ elementary equal parts of length $l = \frac{d_f}{N-1}$. Velocity along the j -thelementary part is considered constant and set as the average $\frac{v_j + v_{j+1}}{2}$ of the velocities of the two endpoints. The total time t_f is the sum $\frac{2l}{v_j + v_{j+1}}$ of the time taken to cover the elementary parts.

$$t_f = 2l \sum_{j=1}^{N-1} \frac{1}{v_j + v_{j+1}} \quad (26)$$

This optimization problem is defined as follows:

Find the velocity sequence $v := (v_1, \dots, v_N) \in \mathbb{R}^N$ which $\min_{v \in \mathbb{R}^N} 2l \sum_{j=1}^{N-1} \frac{1}{v_j + v_{j+1}}$ subject to constraints (27-30).

$$v_1 = v_{start}, v_N = v_{goal} \quad (27)$$

$$0 \leq v_j \leq v_{max}, j = 1, \dots, N \quad (28)$$

$$a_{min}^h \leq a_j^h \leq a_{max}^h, j = 1, \dots, N - 1 \quad (29)$$

$$v_j^2 |k_j| \leq a_{max}^n, j = 1, \dots, N. \quad (30)$$

Where a^h and a^n are the longitudinal and normal accelerations, and k_j the path curvature of the j^{th} elementary part.

Finally, after the selection of shaping parameters $[\eta_1 \ \eta_2 \ \eta_3 \ \eta_4 \ \eta_5 \ \eta_6]^T$ and the velocities (v_1, \dots, v_N) along the smoothed trajectory, the energy model (10) is reformulated as follows:

$$E_T = \sum_{j=1}^{N-1} E_j, \quad (31)$$

with E_j the energy consumed for $j - th$ part of the elementary path, and defined as follows:

$$E_j = \frac{1}{R_a} \left(\frac{3U^2 - 6 \frac{bK_b U K_j V_j}{r}}{+ \frac{K_j^2}{r^2} \left(3b^2 K_j^2 V_j^2 + \frac{3}{2} V_j^2 \right)} \right) \times t_j + \frac{1}{2} \max\{m(V_j^2 - V_{j-1}^2) + I(k_j^2 V_j^2 - k_{j-1}^2 V_{j-1}^2), 0\} + \mu mg \left(|bk_j V_j| + 2 \max\left\{ |bk_j V_j|, \left| \frac{\sqrt{3}}{2} V_j \right| \right\} \right) \times t_j + P_e t_j \quad (32)$$

where $V_j = \frac{v_j + v_{j+1}}{2}$, k_j , and t_j are the average velocity, the path curvature, and the travel time of the j^{th} elementary part, respectively.

Algorithm 2 Smooth path algorithm

```

procedure MAIN
    SELECTWAYPOINTS(path);           Get waypoints on
    generated path with orientation.
    SMOOTHPATH(waypoints);
    VELOCITYPROFILE(smoothpath);
end procedure
procedure SMOOTHPATH(waypoints)
    for  $i \leftarrow 2, M$  do           By solving first SQP
        Find optimal  $[\eta_1 \ \eta_2 \ \eta_3 \ \eta_4 \ \eta_5 \ \eta_6]^T$  for
        waypoints  $i - 1$  and  $i$ ;
        Get smoothed curve between waypoints  $i - 1$  and
         $i$ ;
    end for
    Joint smoothed curves;
return
end procedure
procedure VELOCITYPROFILE(smoothpath)
    SQP2: Find optimal velocity profile to travel
    smoothed path in minimal time.
    Divide smoothpath in  $N$  arc with equal length;
    Solve SQP2;
    Compute energy  $E_T$  with (31);
return ;
end procedure
    
```

3.3 Outline of the proposed motion planning

An outline of the proposed motion planning for energy minimization is given here. Initially, map is scanned to get start, goal and obstacles position with ground characteristic. Heuristic value of all cells is evaluated and modified A* algorithm is run to find path (Algorithm 1) on which waypoints are selected for smooth trajectory generation. Orientation at each waypoint is set, first SQP is solved to find shaping parameters of η^3 -splines and smooth path is obtained. The later is divided in N-1 arc with equal length, second SQP is solved to get velocity profile along smoothed path and total energy consumption of trajectory is estimated (Algorithm 2).

IV. MODEL VALIDATION AND SIMULATION RESULTS

In this section, simulations are conducted to demonstrate the effectiveness of the proposed algorithm. The section starts by the identification and validation of the energy model, and ends by simulation results and a discussion.

4.1 Identification and validation of the energy model of the Robotino mobile robot

The Robotino robot is supplied by two 12V batteries which permit a running time of up to two hours. The robot's dimensions are 350mm in diameter and 210mm in height with an overall weight of approximately 11kg. The platform embeds numerous application programming interfaces layers. While the Linux layer provides standard user space, the platform can also be controlled from an external PC via the wireless communication, by using the Real Time Linux layer. Experimental tests were performed to identify the modeling parameters μ and P_e for the Robotino robot.

The ground friction μ was identified by driving robot on a carpet surface. In the experiment, the robot accelerated from zero to the desired velocity and maintains it for 5s. The robot runs freely until it stops, thereafter. The parameter μ was estimated by following the coast-down equation [13]:

$$\mu = \frac{v_i}{gt} \quad (33)$$

where g represents gravity. The relationship between the elapsed time t and the robot velocity v could be obtained by fitting the curve. For a carpet surface, the value of $\mu \approx 0.013 \pm 0.003$ was obtained.

The energy consumed by the electronic components was identified by letting the robot stopped at fixed position while turning on the power to control the current drawn from the battery. In this situation, the power consumption of the motors was zero and the energy consumption for electronic components was counted. The battery current was stable, and parameter P_e was calibrated (1.46 ± 0.05)W.

Finally, the energy model of the Robotino robot was calibrated as follows:

$$E_T = \frac{1}{7.9} \int \left(3U^2 - 0.656Uw(t) + 0.39(0.09w(t)^2 + 1.5v(t)^2) \right) dt + \int_t d(5.5v(t)^2 + 0.08w(t)^2) + 1.43 \int \left(|0.175w(t)| + 2 \max \left\{ |0.175w(t)|, |0.866v(t)| \right\} \right) dt + 1.46t \quad (34)$$

4.2 Simulations results and discussion

The effectiveness of the proposed motion planning was demonstrated by conducting a series of simulations. Based on the energy model (34), simulations were performed in an environment containing 50×50 grids with wall-like obstacles. The grid size is set based on the robot and obstacles dimensions. It is considered as a square of 350mm of side. In each environment, grey spaces represent friction zones, the black boxes are the obstacles, the red circle is the start node and the green circle is the goal node. Table 1 lists the parameters used in simulations.

Table 1: Robotino parameters

Parameters	Value	Parameters	Value
Wheel radius	40mm	Battery voltage	24V
Robot radius	175mm	Robot mass	11kg
M. of inertia	$0.16245kg.m^2$	Max velocity	$1.325m.s^{-1}$
Back emf.	$0.025V.(rad.sec)$	Resistance R_a	7.9Ω

The performance of the proposed energy saving method was assessed as well in cluttered environments that those containing different friction areas. Regarding comparison with existing contributions, the proposed energy saving approach is compared with the Liu & Sun's method which showed satisfactory performance over existing approaches [2, 11, 12]. In order to preserve the main property of the A* algorithm, our method is compared with a motion planning method based on the Newton algorithm [8]. Computational costs were

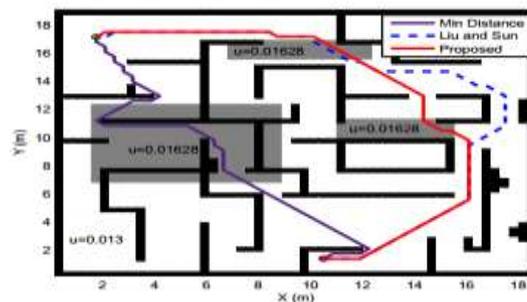
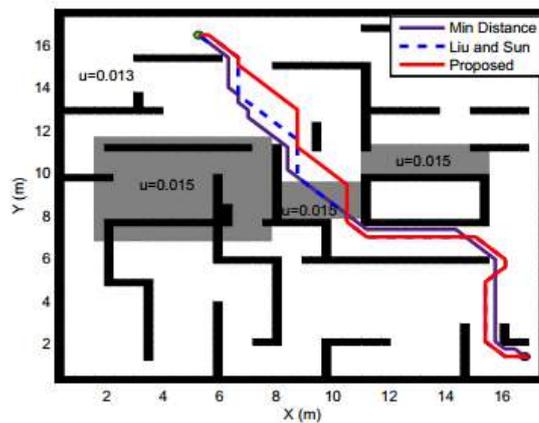
measured in MATLAB under a personal computer with a 1.5GHz Intel Pentium M CPU processor and 1.00Go Random Access Memory. Regarding the simulation results, four scenarios were considered. The ability to generate energy-saving paths in a cluttered environment was assessed in the first and second scenarios while the behavior in environments containing varied frictions zones was assessed in third and fourth scenarios.

4.2.1 Generated paths

To select a function on the three proposed above, simulations are done with these functions integrated in A* cost function for minimum distance as $c(s, s') = d_{s,s'}(\phi(\alpha) + 1)$. Table 2 shows simulation times, path's shape turn and evaluated nodes. These functions shorten evaluated nodes as well as simulation time, and decrease path turn where possible. Its find same path or different paths with equal length. Regarding results, sine function is selected as it has low simulation time and better reduces path turn with minimum decrease of nodes evaluation (More diminution of evaluated nodes can affect path quality). The simulation times, the number of shape turn of generated path, and the path lengths of the fourth scenarios were computed and compared, Table 3. We indicate in bracket the time taking to evaluate heuristic values, simulation time for proposed method integrate this time.

Table 2: Simulation results: simulation time (average of 5 run), number of turn,evaluated nodes

Gridmap	Functions	Time (s)	Turn	Nodes
Scenario 1	None	0.406	14	685
	Gaussian	0.346	10	485
	Square	0.277	8	490
	Sine	0.276	8	505
Scenario 2	None	0.245	10	354
	Gaussian	0.202	9	104
	Square	0.206	9	118
	Sine	0.214	9	173
Scenario 3	None	0.379	22	787
	Gaussian	0.367	17	694
	Square	0.359	13	705
	Sine	0.357	14	723
Scenario 4	None	0.300	6	432
	Gaussian	0.271	6	355
	Square	0.269	6	338
	Sine	0.255	6	338



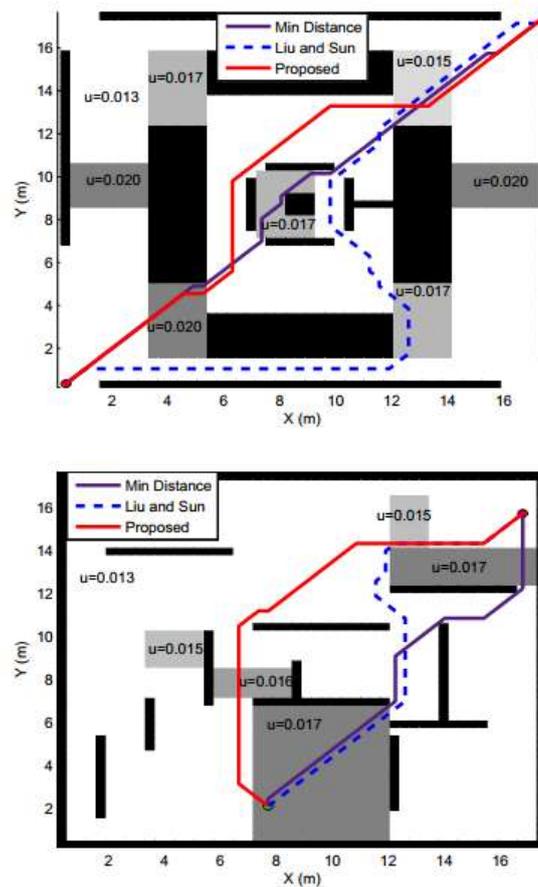


Figure 7: Generated paths

Friction zones with rolling coefficient of 0.015 were set while flat zone has a rolling coefficient of 0.013 in first scenario, Figure 7-(a). The basic A* algorithm for minimum travel distance found the optimal path in approximately 0.4s with 14 turns Table 3. The path generated based on Liu & Sun’s method are found in approximately 0.44s with 15 turns and measure 23.28m. The optimal path generated by the proposed cost function (19) with (17) and heuristic function (20) is found in approximately .37s with 14 turns. In the second scenario, friction zones have rolling coefficient of 0.01628, Figure 7-(b). The basic A* algorithm use 0.39s to get the optimal path having 22 turns. Liu & Sun’s method generates optimal path of 33.05m in 0.75s. The optimal path generated by the proposed method is found in approximately 0.4s with 30.25m of length, path passed through two frictions zones. Friction zones with rolling coefficient varying from 0.015 to 0.020 were set in third scenario, Figure 7-(c). A* algorithm for minimum travel distance found the optimal path in approximately 0.28s with 25.07m of length, path passed in three high frictions. The path generated based on Liu & Sun’s method are found in approximately 0.71s with 14 turns and proposed method found path of 26.71m with 6 turns in approximately 0.39s. For fourth scenario, rolling coefficient of friction zones varying from 0.015 to 0.017, Figure 7-(d). Minimum travel distance method found the optimal path in approximately 0.32s with 6 turns. The path generated based on Liu & Sun’s method are found in approximately 0.6s with 8 turns. Proposed method generates path of 21.16m length in approximately 0.42s.

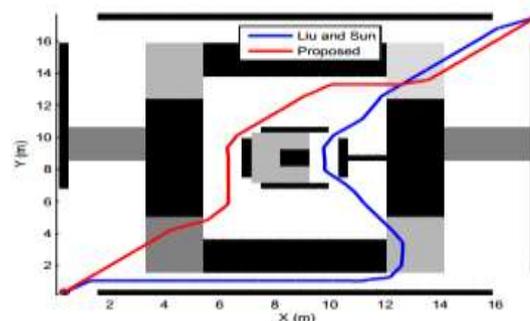
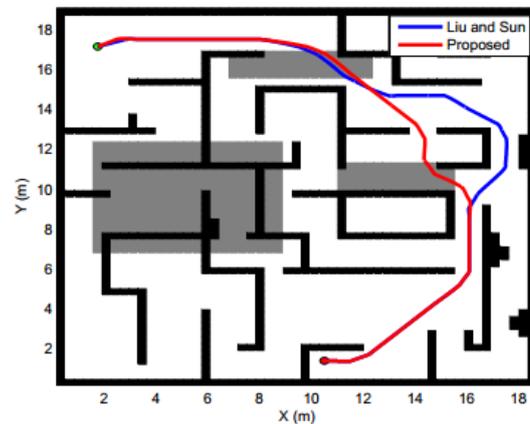
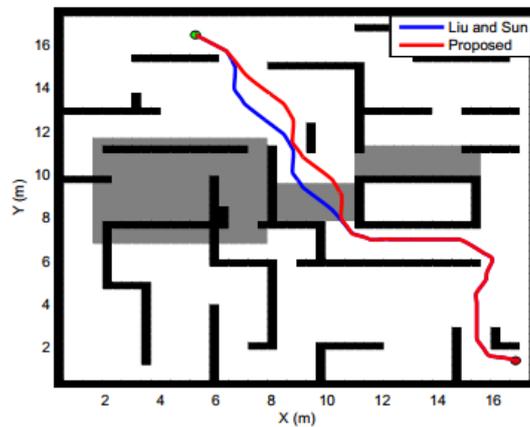
Table 3: Simulation results: simulation time (average of 5 run), number of turn, and path length

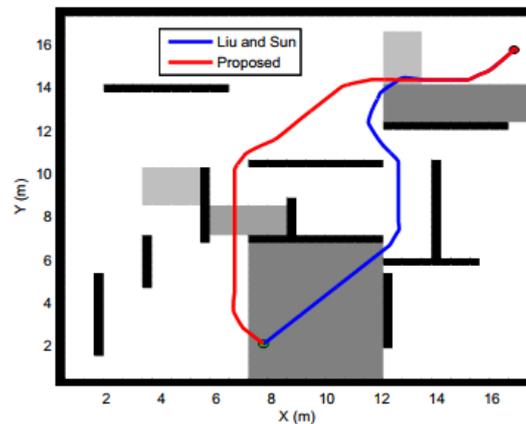
Gridmap	Methods	Time (s)	Turn	Length(m)
Scenario 1	Min dist	0.404	14	21.88
	Liu & Sun	0.444	15	23.28
	Proposed	0.366(0.162)	14	23.28
Scenario 2	Min dist	0.394	22	25.72
	Liu & Sun	0.746	16	33.05
	Proposed	0.398(0.184)	11	30.25
Scenario 3	Min dist	0.276	10	25.07

	Liu & Sun	0.709	14	33.34
	Proposed	0.389(0.188)	6	26.71
Scenario 4	Min dist	0.317	6	18.24
	Liu & Sun	0.597	8	20.13
	Proposed	0.419(0.193)	6	21.16

4.2.2 Smoothed paths results

To estimated the energy saving aspect of the proposed method, the generated paths are smoothed using η^3 -splines while considering the energy model (31) and (32). The maximum curvature has been fixed as the inverse of the half size of the robot dimension for first SQP problem, while curvature at each waypoint has been set to $-\frac{\theta}{4\pi}$, with θ the robot orientation, and the curvature derivative set to zero. The segment joining two waypoints was divided into 1000 pieces for a smoothed path. The smoothed path has been divided into 100 elementary parts for second SQP problem and the following parameters were used to find the optimal travel time: $-0.5\text{m}\cdot\text{s}^{-2}$ and $0.5\text{m}\cdot\text{s}^{-2}$ for minimal and maximal longitudinal accelerations, respectively, and $0.3\text{m}\cdot\text{s}^{-2}$ for maximal normal acceleration. Table 4 shows the travel time, the travel distance and the energy consumption along smoothed paths.





Smoothed paths

Table 4: Smooth path results

Gridmap	Methods	Travel time (s)	Travel distance (m)	Consumedenergy(J)
Scenario 1	Liu & Sun	37.49	22.49	8358.8
	Proposed	36.84	22.50	8213.5
Scenario 2	Liu & Sun	42.52	32.25	9486.7
	Proposed	38.37	29.69	8563.1
Scenario 3	Liu & Sun	40.03	32.54	8945.7
	Proposed	33.23	26.34	7439.0
Scenario 4	Liu & Sun	27.42	19.67	6130.4
	Proposed	27.12	20.86	6058.7

Paths generated by A* algorithm for minimum distance are too close to surrounding obstacles, robot moves on those paths will knock obstacles and energy loss by this event couldn't be estimated. Simulation results of two others methods are provided with trajectories show in Figure 8. For the paths generated by Liu & Sun's method, robot needs 37.5, 42.5, 40 and 27.4seconds, respectively, to join goals positions for the four scenarios. Consumed energy are estimated to 8358, 9486, 8945 and 6130 Joules, respectively. Travel distance of smooth trajectories are less than length of paths generated. For proposed method, robot needs more than 36s to follow generated paths in first two scenarios, while it consumes over 8kJ. In third and fourth scenario consumed energy are estimated to 7439 and 6058 Joules, while travel distances are 26.34m and 20.86m, respectively.

4.2.3 Discussion

Traditional A* algorithm for minimum distance generates shortest paths in small time, Table 3, but doesn't maintain a safety distance to obstacles, and passed through high friction zones, Figure 7. Robot moving on those paths pass too close to surrounding obstacles, expansively consumes energy and trajectories couldn't be smoothed. Numbers of evaluated nodes before getting the optimal path of this algorithm are reduce by proposed functions and thus further minimize simulations times. For Liu & Sun's method, paths generated maintain safety distance to obstacles and pass on low friction zones except in scenario 4 where destination is located in high friction zone, Figure 7-(d). Paths obtained are too long as well as simulation times, robot needs more time and energy to follow smoothed trajectories, Table 4. Proposed method takes approximately 0.2s to compute heuristic values, Table 3, but this computation improves visibility on map while finding optimal path. After heuristic values are evaluated, algorithm needs smallest time than previous methods to generate optimal path. Paths obtained are relatively shorter than Liu & Sun's method except in fourth scenario. The smoothed path generated by proposed method has minimal travel time compared to Liu & Sun's method Table 4. The proposed method saved between 1.17%, scenario 4, and 16.84%, scenario 3, of energy over Liu & Sun's method.

V. CONCLUSION

An optimal motion planning aiming to achieve minimum energy consumption was proposed in this paper. A Three-wheeled Omnidirectional Mobile Robot (TOMR) named Robotino was used as a case study. The A* path planner and the heuristic function integrating energy saving criterion were used to generate energy-saved paths. The algorithm integrates previous robot orientations in cost function through a penalty angle function, and as well a new heuristic function for path generation with minimum energy consumption. Three penalty angle functions are proposed to reduce paths turns and simulation times. By using the waypoints of the generated path, the trajectory was smoothed through optimal η^3 -Spline parameters. The velocity profile along the generated path is optimized by solving sequential quadratic problems. The effectiveness of the proposed motion planning was

demonstrated by performing a series of simulations. The proposed energy saving approach showed a better performance compared with the existing methods. Future work will include an extension of the proposed approach to dynamic environments.

REFERENCES

- [1]. Palmieri N, Yang XS, De Rango F et al. Comparison of bio-inspired algorithms applied to the coordination of mobile robots considering the energy consumption. *Neural Computing and Applications* 2017; : 1–24.
- [2]. Liu S and Sun D. Minimizing energy consumption of wheeled mobile robots via optimal motion planning. *IEEE/ASME Transactions on Mechatronics* 2014; 19(2): 401–411.
- [3]. Yacoub MI, Neculescu DS and Sasiadek JZ. Energy consumption optimization for mobile robots motion using predictive control. *Journal of Intelligent & Robotic Systems* 2016; : 1–18.
- [4]. Kim H and Kim BK. Online minimum-energy trajectory planning and control on a straight-line path for three-wheeled omnidirectional mobile robots. *IEEE Transactions on Industrial Electronics* 2014; 61(9): 4771–4779.
- [5]. Zadarnowska K. Switched modeling and task-priority motion planning of wheeled mobile robots subject to slipping. *Journal of Intelligent & Robotic Systems* 2017; 85(3-4): 449–469.
- [6]. Mei Y, Lu YH, Hu YC et al. Energy-efficient motion planning for mobile robots. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5. IEEE, pp. 4344–4349.
- [7]. David Galvão Wall HGKKPS John Economou and Lawrence M. Mobile robot arm trajectory generation for operation in confined environments. *Systems and Control Engineering* 2015; 229(3): 215–234.
- [8]. Duleba I and Sasiadek JZ. Nonholonomic motion planning based on newton algorithm with energy optimization. *IEEE transactions on control systems technology* 2003; 11(3): 355–363.
- [9]. Stentz A. The focussed d* algorithm for real-time replanning. In *Proceedings of the International Joint Conference of Artificial Intelligence*. Morgan Kaufman Publishers, pp. 1652–1659.
- [10]. [10] Sucan IA, Moll M and Kavraki LE. The open motion planning library. *IEEE Robotics & Automation Magazine* 2012; 19(4): 72–82.
- [11]. Yang J, Qu Z, Wang J et al. Comparison of optimal solutions to real-time path planning for a mobile vehicle. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 2010; 40(4): 721–731.
- [12]. Hussein AM and Elnagar A. On smooth and safe trajectory planning in 2d environments. In *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, volume 4. IEEE, pp. 3118–3123.
- [13]. Wahab M, Rios-Gutierrez F and El Shahat A. Energy modeling of differential drive robots. In *SoutheastCon 2015. IEEE*, pp. 1–6.
- [14]. Dalla VK and Pathak PM. Curve-constrained collision-free trajectory control of hyper-redundant planar space robot. *Systems and Control Engineering* 2017; 231(4): 282–298.
- [15]. Hart PE, Nilsson NJ and Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 1968; 4(2): 100–107.
- [16]. Ferguson D, Likhachev M and Stentz A. A guide to heuristic-based path planning. In *Proceedings of the international workshop on planning under uncertainty for autonomous systems, international conference on automated planning and scheduling (ICAPS)*. pp. 9–18.
- [17]. Aurelio Piazzi CGLB and Romano M. η^3 -splines for the smooth path generation of wheeled mobile robots. *IEEE Transactions on Robotics* 2007; 23(5): 1089–1095.
- [18]. Sen Zhang ZCXL Lei Sun and Liu J. Smooth path planning for a home service robot using η^3 -splines. In *International Conference on Robotics and Biomimetics. IEEE*, pp. 1910–1915.
- [19]. L Consolini AMAP M Locatelli. A linear-time algorithm for minimum-time velocity planning of autonomous vehicles. In *24th Mediterranean Conference on Control and Automation. IEEE*, pp. 490–495.

oger Datouo" Optimal Motion Planning for Minimizing Energy Consumption of Wheeled Mobile Robots" American Journal of Engineering Research (AJER), vol. 8, no. 10, 2019, pp 215-229