Research Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
January-February 2018: 1–12

DOI: 10.1177/1729881418754563
journals.sagepub.com/home/arx
SAGE

Power-minimization and energy-reduction autonomous navigation of an omnidirectional Mecanum robot via the dynamic window approach local trajectory planning

Li Xie[1], Christian Henkel[2], Karl Stol[1] and Weiliang Xu[1]

Abstract

To improve the energy efficiency of the Mecanum wheel, this article extends the dynamic window approach by adding a new energy-related criterion for minimizing the power consumption of autonomous mobile robots. The energy consumption of the Mecanum robot is first modeled by considering major factors. Then, the model is utilized in the extended dynamic window approach–based local trajectory planner to additionally evaluate the omnidirectional velocities of the robot. Based on the new trajectory planning objective that minimizes power consumption, energy-reduction autonomous navigation is proposed via the combinational cost objectives of low power consumption and high speed. Comprehensive experiments are performed in various autonomous navigation task scenarios, to validate the energy consumption model and to show the effectiveness of the proposed technique in minimizing the power consumption and reducing the energy consumption. It is observed that the technique effectively takes advantage of the Mecanum robot's redundant maneuverability, can cope with any type of path and is able to fulfil online obstacle avoidance.

Keywords

Autonomous navigation, power optimization, energy reduction, dynamic window approach, omnidirectional Mecanum robot

Date received: 29 May 2017; accepted: 20 December 2017

Topic: Special Issue—Robot Path Planning Design and Implementation in Manufacturing, Healthcare and Service Systems
Topic Editor: Lino Marques
Associate Editor: Hamed Fazlollahtabar

Introduction

The Ilon Mecanum wheel is one of the practical omnidirectional wheel designs utilized in the industry, and it has the advantage of high load capacity over other omnidirectional wheel designs.[1] However, the Mecanum wheel trades off maneuverability against motion efficiency, and its inefficient energy usage for generating omnidirectional motions increases the energy consumption of the robot.[2] Energy-efficient autonomous navigation has been explored very little for the Mecanum robot. A local trajectory planning method that both reduces overall energy consumption in the global sense and reactively avoids unexpected obstacles is essential to realize energy-efficient autonomous navigation. It is both economically valuable to reduce energy consumption of the Mecanum platforms by planning its redundant omnidirectional motion trajectories and

[1] Department of Mechanical Engineering, The University of Auckland (UoA), Auckland, New Zealand
[2] Fraunhofer Institute of Manufacturing and Automation (IPA), Stuttgart, Germany

Corresponding author:
Li Xie, Department of Mechanical Engineering, The University of Auckland (UoA), Auckland 1142, New Zealand.
Email: lxie021@aucklanduni.ac.nz

academically interesting to achieve energy-efficient autonomous navigation via a local planner.

Energy-efficient motion planning methods have been applied to reduce the energy consumption of mobile robots. Mei et al. introduced a method of reducing energy consumption of mobile robots for open area exploration via energy-efficient motion planning.[3] Up to 51% of energy savings were demonstrated for a search of an open area by comparing the energy efficiencies of different motion trajectories. The energy consumption of the robot was modeled by using a sixth-degree polynomial to fit the relationship of motor speed and power consumption. Sun and Reif concentrated on the robotic energy-minimum path planning problem on terrains.[4] The energy cost in their study was based on gravity and friction. Recently, Liu and Sun proposed an experimentally validated energy model for a wheeled mobile robot by considering kinematic energy transformation, overcoming resistive friction and electric supply of onboard devices. Then, a model-based energy criterion was added into the A* algorithm to find the energy-minimum path.[5] To follow the found path, they proposed parameterized smooth trajectory planning using a cubic Bezier curve for minimum energy consumption. In the works of Yang et al.,[6] polynomial parameterization was applied to analytically find the near minimal-energy, real-time, and collision-free path while considering for other criteria for a carlike mobile drive. The modified Newton algorithm was used by Duleba and Sasiadek[7] to optimize the energy consumption in the nonholonomic motion planning. But these methods only apply to nonholonomic mobile robots and do not consider the online local energy-efficient trajectory planning problem.

Kim and Kim studied online minimum-energy trajectory planning of both nonholonomic and holonomic drives on a straight-line path.[8,9] Their energy cost function is the sum energy drawn from the onboard batteries and includes energy dissipation by the motor armature, the energy overcoming frictions, and the kinetic energy of the robot. A closed-form solution of the minimum-energy rotational velocity trajectory was found by using Pontryagin's minimum principle, and the minimum-energy translational velocity trajectory was found using a new researching algorithm. Their results showed that following the same straight-line path via different velocity profiles consumed different amounts of energy. But their studies were restricted to straight-line paths and stationary states in the beginning and at the end, and hence become less practical for autonomous navigation.

Because mobile robots usually source energy from batteries that store limited energy, most of the existing energy-efficient studies aim to minimize the energy consumption instead of the power consumption. The robotic energy consumption to complete a task is the integral of its power consumption over the task duration. The power consumption, which is the energy spent per unit of time, is also a critical criterion to design and control mobile robots with self-sustaining power supplies such as solar power.[10–13] Energy-conservation techniques are proposed by Mei et al. to improve energy efficiency for mobile robots by reducing power consumption of the robotic electronic components.[14] The issue of the power minimization via motion planning for holonomic mobile robots with redundant maneuverability has not been fully studied.

For the purpose of reducing the energy consumption of the autonomous Mecanum robots, an online local trajectory planning method is proposed in this article, aiming for power-minimization and energy-reduction autonomous navigation. The method extends the popular robot operating system (ROS) dynamic window approach (DWA) local trajectory planner by integrating a novel energy consumption model of the Mecanum mobile robot as a new energy-related criterion. The DWA is a well-known reactive collision avoidance navigation concept[15] and it has proven performance over a range of extended applications.[16–18] The contributions of this article include the following two aspects. First, a novel energy consumption model of the four-wheeled Mecanum robot was built, validated by experiments, and in this article, utilized to extend the DWA local trajectory planner. Second, the DWA was extended by means of adding the energy consumption model as a new criterion for minimizing the power consumption of the autonomous Mecanum robot. Based on the proposed extended DWA that can minimize power consumption, energy-reduction autonomous navigation is proposed via the combinational cost objectives of low power consumption and high speed. Comprehensive experiments are performed to validate the energy consumption model and show the effectiveness of the proposed technique in optimizing the power consumption and reducing the energy consumption in various autonomous navigation task scenarios. Comparing to the preliminary publications[19,20] of this article, the major improvements include integrating the novel energy consumption model of the Mecanum platform into the DWA framework, introducing a new cost objective of minimizing the power consumption to the DWA, and conducting more comprehensive experiments.

The reminder of the article is organized as follows. The dynamics and the energy consumption model of the Mecanum robot is reviewed in "Energy consumption model" section. "Extended DWA local trajectory planner" section describes the process of extending the DWA by introducing a new energy-related criterion. Experimental results are discussed to validate the proposed energy consumption model and to investigate the effectiveness of the proposed technique applied as a local trajectory planner in autonomous navigation in "Experiments" section. "Conclusion" section concludes this article.

Energy consumption model

In this study, the four-wheeled Mecanum mobile robot[21] is considered, and its energy consumption is modeled, which

considers the idle energy required for standby and the motional energy consumed for kinetic energy transformation, overcoming robotic traction resistances, electric dissipation in the motor armatures, and mechanical dissipation in the actuators. The main challenge is the complicated dynamics of the omnidirectional Mecanum platform.

The dynamics

Figure 1 shows the schematic diagram and the image of the robot. The front of the robot is defined at the side of the chassis where the second and third wheels are placed. A coordinate system Ox_ry_r is rigidly attached at the mass center of the chassis and is related to the global coordinate system OX_GY_G. Angle $\oslash$ is the angular displacement or orientation of the robot, and angle φ defines the heading direction of the robotic motion, measured anticlockwise from the y_r-axis. Translational displacements X and Y define the center of mass of the robot. Thus, the position and orientation of the robot are represented by $\boldsymbol{q} = [X \quad Y \quad \oslash]^T$ in OX_GY_G. Following the work by Matsinos,[22] the kinematics with the rollers as arranged in Figure 1 is

$$\dot{\boldsymbol{q}} = \frac{r}{4}\boldsymbol{R}(\oslash)\boldsymbol{K}\dot{\boldsymbol{\theta}} \tag{1}$$

in which

$$\boldsymbol{R}(\oslash) = \begin{bmatrix} \cos\oslash & -\sin\oslash & 0 \\ \sin\oslash & \cos\oslash & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$\boldsymbol{K} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ \dfrac{1}{l_x + l_y} & \dfrac{1}{l_x + l_y} & \dfrac{-1}{l_x + l_y} & \dfrac{-1}{l_x + l_y} \end{bmatrix} \tag{3}$$

where r is the radius of the wheel, $\boldsymbol{\theta} = [\theta_1 \ \ \theta_2 \ \ \theta_3 \ \ \theta_4]$ is the rotation of the wheels; l_x and l_y are the distances of the center of mass of the wheel from the y_r-axis and x_r-axis in Ox_ry_r, respectively; and $\boldsymbol{R}$ and $\boldsymbol{K}$ stand for rotation matrix and forward kinematics matrix, respectively.

The dynamics of the robot was modeled in the authors' previous study,[20] repeated in the following

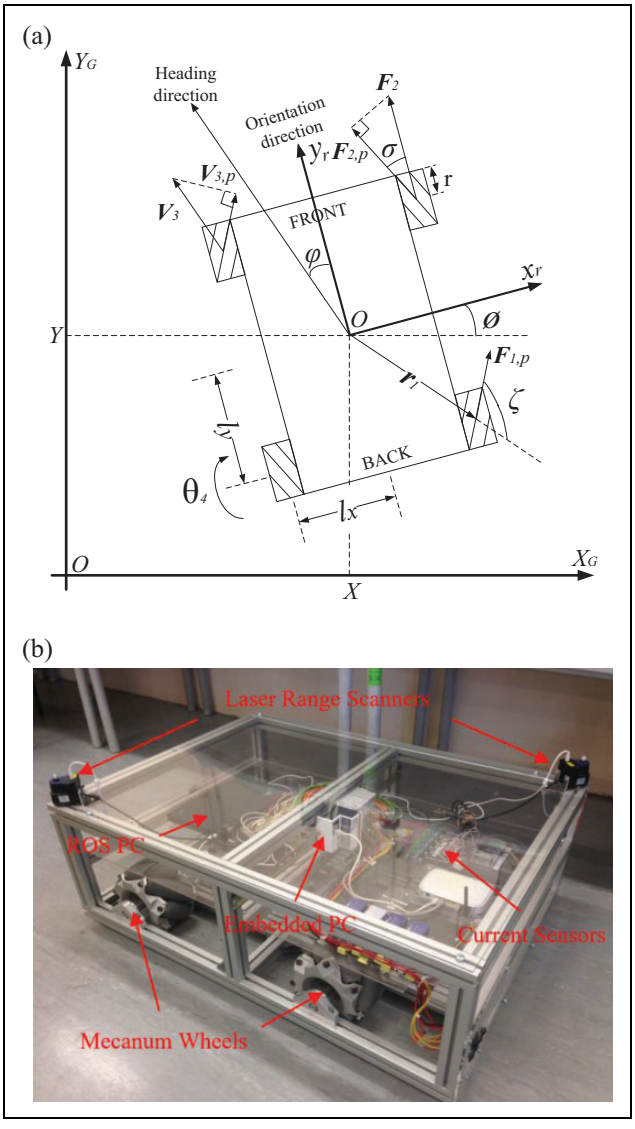


Figure 1. The four-wheeled Mecanum mobile robot. (a) Schematic diagram, viewed from beneath and (b) the built robot, system parameters given in Table 1.

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = \boldsymbol{R}(\oslash)\boldsymbol{C}(\sigma)\boldsymbol{S} \tag{4}$$

in which

$$\boldsymbol{M} = \mathrm{diag}[m \quad m \quad I] \tag{5}$$

$$C(\sigma) = \begin{bmatrix} \cos\sigma \cdot \mathrm{sgn}(\dot{\theta}_1) & -\cos\sigma \cdot \mathrm{sgn}(\dot{\theta}_2) & \cos\sigma \cdot \mathrm{sgn}(\dot{\theta}_3) & -\cos\sigma \cdot \mathrm{sgn}(\dot{\theta}_4) \\ \sin\sigma \cdot \mathrm{sgn}(\dot{\theta}_1) & \sin\sigma \cdot \mathrm{sgn}(\dot{\theta}_2) & \sin\sigma \cdot \mathrm{sgn}(\dot{\theta}_3) & \sin\sigma \cdot \mathrm{sgn}(\dot{\theta}_4) \\ |r_1|\sin\zeta \cdot \mathrm{sgn}(\dot{\theta}_1) & |r_2|\sin\zeta \cdot \mathrm{sgn}(\dot{\theta}_2) & -|r_3|\sin\zeta \cdot \mathrm{sgn}(\dot{\theta}_3) & -|r_4|\sin\zeta \cdot \mathrm{sgn}(\dot{\theta}_4) \end{bmatrix} \tag{6}$$

$$\boldsymbol{S} = \begin{bmatrix} |\boldsymbol{F}_{1,p} + \boldsymbol{f}_{1,s/r} + \boldsymbol{f}_{1,v}| \\ |\boldsymbol{F}_{2,p} + \boldsymbol{f}_{2,s/r} + \boldsymbol{f}_{2,v}| \\ |\boldsymbol{F}_{3,p} + \boldsymbol{f}_{3,s/r} + \boldsymbol{f}_{3,v}| \\ |\boldsymbol{F}_{4,p} + \boldsymbol{f}_{4,s/r} + \boldsymbol{f}_{4,v}| \end{bmatrix} \quad (7)$$

where m is the mass of the robot; I is the moment of inertia for the rotation of the robot; $\mathrm{sgn}(\dot{\theta}_i)$ is the sign of the rotational direction that is defined as positive according to the left-hand rule along the x_r-axis; $\boldsymbol{r}_i$ is the position vector of wheel i in Ox_ry_r; $\boldsymbol{F}_i$ is the driving force proportional to the motor torque at the circumference of the wheel in contact with the ground; $\boldsymbol{F}_{i,p}$ is the effective force of $\boldsymbol{F}_i$ in parallel with the rotational axis of the freely rolling rollers on the wheel; angle σ is between the roller axis and the rotational plane of the wheel; and angle ζ is between $\boldsymbol{r}_i$ and $\boldsymbol{F}_{i,p}$. Matrix $\boldsymbol{C}$ converts the forces of Mecanum wheels $\boldsymbol{S}$ into the net force and torque of the robot. $\boldsymbol{f}_{i,s}$ and $\boldsymbol{f}_{i,r}$ are sliding and rolling frictional forces acting on the wheel, respectively, and $\boldsymbol{f}_{i,v}$ is the analogue viscous resistive friction of the Mecanum wheel,[23] which can be found by

$$\boldsymbol{f}_{i,s} = -\mu_s N_i \cdot \mathrm{sgn}(\boldsymbol{V}_{i,p})\hat{\boldsymbol{u}}_{i,p} \quad (8)$$

$$\boldsymbol{f}_{i,r} = -\mu_r N_i \cdot \mathrm{sgn}(\boldsymbol{V}_{i,p})\hat{\boldsymbol{u}}_{i,p} \quad (9)$$

$$\boldsymbol{f}_{i,v} = -\mu_v N_i \boldsymbol{V}_{i,p} \quad (10)$$

in which N_i is the normal force on the wheel; μ_s, μ_r, and μ_v are the sliding, rolling, and analogue viscous friction coefficients, respectively; and $\boldsymbol{V}_{i,p}$ is the vector projection of the wheel velocity relative to the ground $\boldsymbol{V}_i$ onto the unit vector $\hat{\boldsymbol{u}}_{i,p}$ along the roller parallel direction. $\boldsymbol{f}_{i,s/r}$ means that either $\boldsymbol{f}_{i,s}$ or $\boldsymbol{f}_{i,r}$ takes effect on the wheel at one time.

Energy consumption model

The total energy consumption E_{total} of the robot consists of the idle energy E_{idle} and the motional energy E_{motion}

$$E_{\text{total}} = E_{\text{idle}} + E_{\text{motion}} \quad (11)$$

where E_{idle} is consumed by the idling robot consisting of the idling motors and onboard electric devices, including embedded PC, sensors, electric drives, and other electrical devices, and the idle power P_{idle} is experimentally found to be constant over time t in this study

$$E_{\text{idle}} = P_{\text{idle}}\Delta t \quad (12)$$

where E_{motion} is the energy consumed to attain and sustain robotic motion, and the motional power P_{motion} is motion-dependent

$$E_{\text{motion}} = \int_t P_{\text{motion}} dt = E_k + E_f + E_e + E_m \quad (13)$$

in which E_k, E_f, E_e, and E_m are, respectively, the changes in robot kinetic energy, the frictional dissipation to overcome the traction resistances, the energy dissipation as heat in the armatures of motors, and the mechanical dissipation caused by overcoming the friction torque in the actuators, including motor shafts and bearings, which are expressed by

$$E_k = \int_t (\dot{\boldsymbol{q}}^T \boldsymbol{M}\ddot{\boldsymbol{q}} + J_w \dot{\boldsymbol{\theta}}^T \dot{\boldsymbol{\theta}}) dt \quad (14)$$

$$E_f = \int_t \sum_{i=1}^{4} (|\boldsymbol{f}_{i,s/r} + f_{i,v}| \cdot |\boldsymbol{V}_{i,p}|) dt \quad (15)$$

$$E_e = R_a \int_t \boldsymbol{I}_m{}^T \boldsymbol{I}_m dt \quad (16)$$

$$E_m = M_R \int_t \dot{\boldsymbol{\theta}} dt \quad (17)$$

where J_w is the moment of inertia of the wheel along the rotation axis, R_a is the armature resistance of each motor, $\boldsymbol{I}_m$ is the current vector of four electric motors, and M_R is the static frictional torque in the actuator.

Given an arbitrary velocity profile $\dot{\boldsymbol{q}}(t)$ of the robot, the motor current $\boldsymbol{I}_m$ is required to calculate E_{motion}. According to the brushed direct current motor study in the works of Scott et al.,[24] the motor shaft torque vector $\boldsymbol{T}_m$ is related to $\boldsymbol{I}_m$. In this study, brushless alternating current (AC) servomotors were used, but the torque is proportional to current in both cases

$$\boldsymbol{T}_m = k_t \boldsymbol{I}_m \quad (18)$$

in which k_t is the torque constant. $\boldsymbol{I}_m$ is calculated by using equations (4) and (19)

$$\boldsymbol{I}_m k_t \frac{n}{r} \sin\sigma = [\ |\boldsymbol{F}_{1,p}| \quad |\boldsymbol{F}_{2,p}| \quad |\boldsymbol{F}_{3,p}| \quad |\boldsymbol{F}_{4,p}|\ ]^T \quad (19)$$

in which n is the gear ratio of the motor gear box.

Other energy losses are considered as minor losses and are neglected in this study. The dynamics of the motor is also neglected.

Extended DWA local trajectory planner

The ROS navigation stack[25,26] employs both a global path planner and a local trajectory planner for motion planning. Given the feasible path from the global planner to follow, the local trajectory planner works online to continuously update the local velocity commands for the robot to execute for the next short period of control cycle. This is because the autonomous robot must react to its surrounding environment that is subject to change. Since the global path planning in autonomous navigation is usually reduced to a 2-D representation by assuming the robot to be a point,[27] the proposed energy-efficient path planning methods by

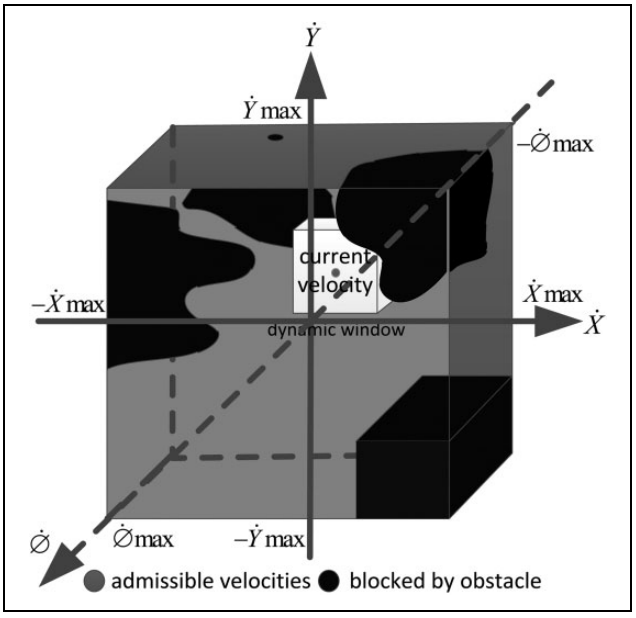


Figure 2. Cuboid search space.

Sun and Reif[4] and Liu and Sun[5] are applicable to both nonholonomic and holonomic mobile robots. Thus, this study only focuses on the autonomous navigation local trajectory planning. An extended DWA local trajectory planner is proposed. The DWA has two steps: search space and optimization. The DWA is extended by additionally employing a new energy consumption cost function to search for the optimal omnidirectional velocity trajectories in a cuboid space.

Cuboid search space

The robot concerned is holonomic with a triplet of omnidirectional velocities $(\dot{X},\ \dot{Y},\ \dot{\varnothing})$. Hence, the cuboid search space in Figure 2 contains all the allowable translational velocities $\dot{X}$ and $\dot{Y}$ and rotational velocities $\dot{\varnothing}$ of the robot. So, the dimension of the space is defined by the maximum velocities $\dot{X}_{\max}$, $\dot{Y}_{\max}$, and $\dot{\varnothing}_{\max}$ in both positive and negative directions. According to the DWA,[15] the cuboid search space distinguishes the admissible velocities that still allow the robot to stop before hitting an obstacle by considering the dynamic limits of the robot. The admissible velocities that are then further restricted within a local dynamic window are defined as reachable admissible velocities. The dimension of the dynamic window or the range of the reachable admissible velocities is determined by the forward-simulation time of the dynamic window and the dynamic constraints of the robot. The center of the dynamic window usually lies at the current velocity in the search space.

Optimization

In this study, the cost function in equation (20), which is based on the ROS dwa_local_planner, is put forward for optimizing the omnidirectional velocities by minimizing the cost

$$\begin{aligned}\text{COST} = {} & p\_bias \cdot path\_distance + g\_bias \cdot goal\_distance \\ & + o\_bias \cdot obstacle + e\_bias \cdot energy\_consumption\end{aligned} \tag{20}$$

Local trajectory planning is aimed to track the given path and approach the local goal while avoiding any unforeseen obstacles. Here, the unforeseen obstacle is defined as the unknown obstacle in the global map and can only be detected by the robot's local sensor readings. Thus, the first three existing terms are associated, respectively, with how much the robot stays close to the given path, how fast the robot approaches the local goal, and how far the robot stays away from an obstacle, and the new fourth term is for how much the robot minimizes its power consumption. *p_bias*, *g_bias*, *o_bias*, and *e_bias* are the tuning parameters.

Optimization of power consumption. Within a local dynamic window, all potential velocity trajectories are over the same forward-simulation time T_{DW}, and consequently, the idle energy consumption is the same and does not need to be included in *energy_consumption*. That, instead, can be calculated by equation (13). The velocity trajectories are evaluated within the local time window that rolls forward. The energy consumption is optimized in the constant-time window and hence contributes to minimizing the overall average power consumption of the resulting trajectories in the global sense for autonomous navigation. (The mathematical proof is presented in Appendix 1.) Thus, *e_bias* determines the bias of optimizing the power consumption against other cost objectives.

Reduction of energy consumption. It is difficult to realize overall energy optimization only via the local trajectory planning. To follow the same global path, the accumulation of the energy optimization in each local window does not necessarily achieve overall energy optimization in the global sense for autonomous navigation. Additionally, it is worth mentioning that the local trajectory planner relies on the given paths from the high-level global path planner. This means that the proposed local trajectory planning method cannot guarantee a global path that leads to minimum energy consumption from a starting pose to a goal pose. It is the responsibility of the global path planner to avoid the areas of high energy cost, such as the ground with high friction in the works of Liu and Sun.[5]

Another aim of this study is to reduce the overall energy consumption via the extended DWA local trajectory planner. Energy is the integral of power over time. Both low average power consumption and low task duration can reduce the overall energy consumption. The overall average power consumption and the task duration in the global sense can both be reduced by the proposed extended DWA. An original cost function term *goal_distance* controls

Table 1. Model parameters of the robot.

Symbol	Quantity	Value
r	Wheel radius	0.11 m
l_x	Distance from the center of robot to the center of Mecanum wheel in x-direction	0.45 m
l_y	Distance from the center of robot to the center of Mecanum wheel in y-direction	0.60 m
m	Mass of robot	94.00 kg
I	Moment of inertia of robot in z-direction	19.40 kg m^2
σ	Roller angle	45.00°
μ_r	Rolling friction coefficient	0.18
μ_s	Sliding friction coefficient	0.90
μ_v	Viscous friction coefficient	0.20
J_w	Moment of inertia of the Mecanum wheel along its rotation axis	0.04 kg m^2
R_a	Armature resistance of the motor	0.80 Ω
M_R	Static friction torque in the actuator	0.02 N m
P_{idle}	Idling robotic power consumption	72.00 W
n	Gear ratio	40.00
k_t	Torque constant	0.14 N m/A

speed by minimizing the distance to the local goal from the endpoint of the trajectory. The combinational cost objectives of low motional power consumption and high speed are proposed to reduce the overall energy consumption via high weighting of *g_bias* and *e_bias*.

Local trajectory planner implementation

The proposed extended DWA technique is applied as a power-optimal local trajectory planner for the energy-efficient autonomous navigation. The conventional DWA has already been implemented in the ROS navigation stack as a popular local trajectory planner. To implement the proposed extended DWA technique, it simply adds a robotic energy consumption cost term into the cost function of the conventional DWA. As equation (20) presents, *energy_consumption* is the added robotic energy consumption cost term that is based on the proposed energy consumption model, while the rest are the original cost terms in the ROS DWA planner. The TrajectoryCostFunction class in the ROS navigation stack base_local_planner program manages the DWA cost function.

Experiments

The Mecanum wheeled robot in Figure 1 was used in this study, which is equipped with an embedded PC CX5020, four AC servomotors AM3121 with 4 × 140 W rated power and ROS Hydro with 2-D navigation stack that was configured for the robot to move toward a goal pose using velocity commands. The key parameters of the robot and their values are given in Table 1. A current sensor was used to measure the current of the robotic battery. Depending on the battery's state of charge, the robotic idle current ranges from 1.4 A to 1.6 A. Thus, the same group of experiments were conducted under the same level of battery's state of charge.

Experimental validation of the energy consumption model for the four-wheeled Mecanum mobile robot

To validate the derived energy consumption models, the actual energy was measured in this group of experiments for the purpose of comparisons when the robot performed three primitive omnidirectional motions that include pure translation in the x_r-direction and y_r-direction, and rotation about the origin of Ox_ry_r. All the motions were tested on concrete at two speeds: ± 0.025 and ± 0.05 m/s for translation in the x_r-direction, 0.05 and ± 0.1 m/s for translation in the y_r-direction, and $\pm 0.0125\pi$ and $\pm 0.025\pi$ rad/s for rotation. Each motion test lasted for 2 s, including 0.15 s acceleration from stationary, 1.75 s constant velocities, and 0.1 s deceleration to stationary. After each run of primitive motion, the robot stayed stationary for 2 s before the next run. Each experiment was repeated three times.

The proposed energy consumption model, implemented in MATLAB, simulates the predicted energy consumptions according to the described motions above. The experimental results are given in Table 2, where it is shown that the robot consumed a consistent amount of energy to perform the same omnidirectional motion under the same surface conditions, and the calculated energy consumption using the energy consumption model in the simulation agrees with the measured energy with an accuracy of 95%. The currents that the robot consumed to perform the set of primitive motions are plotted in Figure 3. The difference between the simulated and experimental current plots is mainly caused without considering the high starting currents in the servomotors.

Experimental validation of the power-optimal local trajectory planning

Given the planned path, the local trajectory planner determines the resulting trajectory for autonomous navigation of the robot by updating the local velocity control commands every control cycle. In the ROS dwa_local_planner, the sim_time is 1.7 s, controller_frequency is 20 Hz, and the sim_granularity is 0.025 m. The sim_time is the entire time of the DWA simulating trajectories within one dynamic window as T_{DW}, the controller_frequency is the frequency at which the dynamic window is called, and the sim_granularity determines the step size between the points in the trajectories. Five types of the planned paths in three task scenarios were tested, including three straight-line paths with different final poses, one curve-line path and one obstacle avoidance path, as shown in Figure 4(a) to (c), respectively.

Since each term of the ROS DWA cost function is mathematically expressed in different units according to equation (20), the tuning parameters are on different scales. For

Table 2. Simulated and experimental energy consumption.

Motions	Schedule (s)	Total energy consumption (J)			
		Experiment 1	Experiment 2	Experiment 3	Simulation
Forward, slow	2–4	323.3	323.8	324.9	321.0
Forward, fast	6–8	338.5	339.5	342.7	340.9
Backward, slow	10–12	320.5	320.6	321.1	321.0
Backward, fast	14–16	341.2	341.3	341.2	340.9
Sideway right, slow	18–20	344.7	346.3	347.2	346.7
Sideway right, fast	22–24	385.0	384.2	384.8	385.8
Sideway left, slow	26–28	346.7	348.3	349.3	346.7
Sideway left, fast	30–32	376.7	377.0	377.0	385.8
Rotation right, slow	34–36	323.9	324.1	323.3	323.3
Rotation right, fast	38–40	337.6	337.0	337.3	335.5
Rotation left, slow	42–44	320.5	321.7	322.1	323.3
Rotation left, fast	46–48	337.0	336.9	337.6	335.5

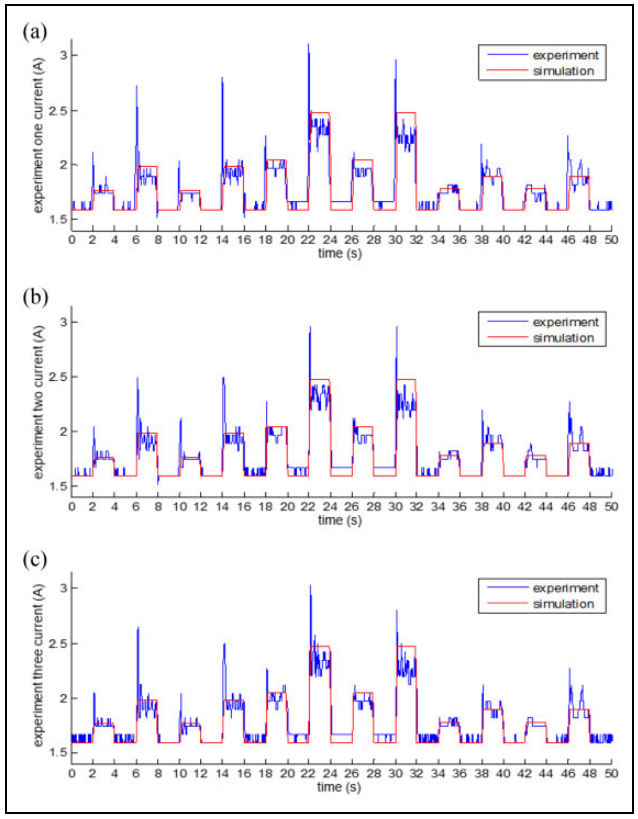


Figure 3. Current plots of experiment and simulation.

the purpose of clear expression, the ratios α, β, γ, and δ are used in the experimental studies to represent the weighting of path closeness, speed, obstacle avoidance, and motional power optimization, respectively, by calculating the corresponding cost percentage

$$\delta = \frac{e\_bias \cdot energy\_consumption}{\text{COST}} \tag{21}$$

Based on the default values given in the ROS dwa_local_planner, the robot was initially tuned by using $p\_bias = 32$, $g\_bias = 50$, $o\_bias = 0.01$, and $e\_bias = 0$. To demonstrate the effectiveness of the proposed energy-optimal omnidirectional velocity search technique in optimizing the power consumption for autonomous navigation, δ was varied from 0.00%, 20.00%, 50.00%, 66.66% to 80.00% by only increasing e_bias. The total time for the robot to complete the planned paths was defined as the task duration. The total energy was calculated by the product of time integral of the measured battery current and battery voltage 48 V. The idle power was calculated to be 72.7 W due to the 1.51–1.52 A observed idle current in this group of experiments. The idle energy was calculated based on the idle power and the task duration. Then, the motional energy was calculated by subtracting the idle energy from the total energy. The motional power was calculated by dividing the motional energy by the task duration.

Task duration, energy, and power of the resulting trajectories are summarized in Table 3. It is shown that the increase in δ reduces both the total power and the motional power of the robot in all the scenarios, which proves the effectiveness and significance of introducing the energy consumption into the cost function for optimizing the robotic power consumption. Reducing the idle power of the robotic electronics components is not within the scope of this study. The motional energy consumptions did not always decrease with the decreasing motional power as shown in Figure 7.

In the experiments, two laser range scanners were used to record the translational and rotational displacements of the robot and detect the unforeseen obstacles near or around the robot. Given the sideways straight-line path in Figure 4(a), two resulting trajectories of the robot are depicted in Figure 5(a) and (b) when $\delta = 20.00\%$ and $\delta = 80.00\%$, respectively, in the cost function. Both resulting trajectories show that the robot reached the goal pose within a reasonable goal tolerance. When $\delta = 80.00\%$, the robot performed a sideways motion with the heading φ angle remaining 90°. Given the reactive obstacle avoidance path in Figure 4(c), two resulting trajectories are depicted in Figure 6(a) and (b) when $\delta = 0.00\%$ and $\delta = 66.66\%$,

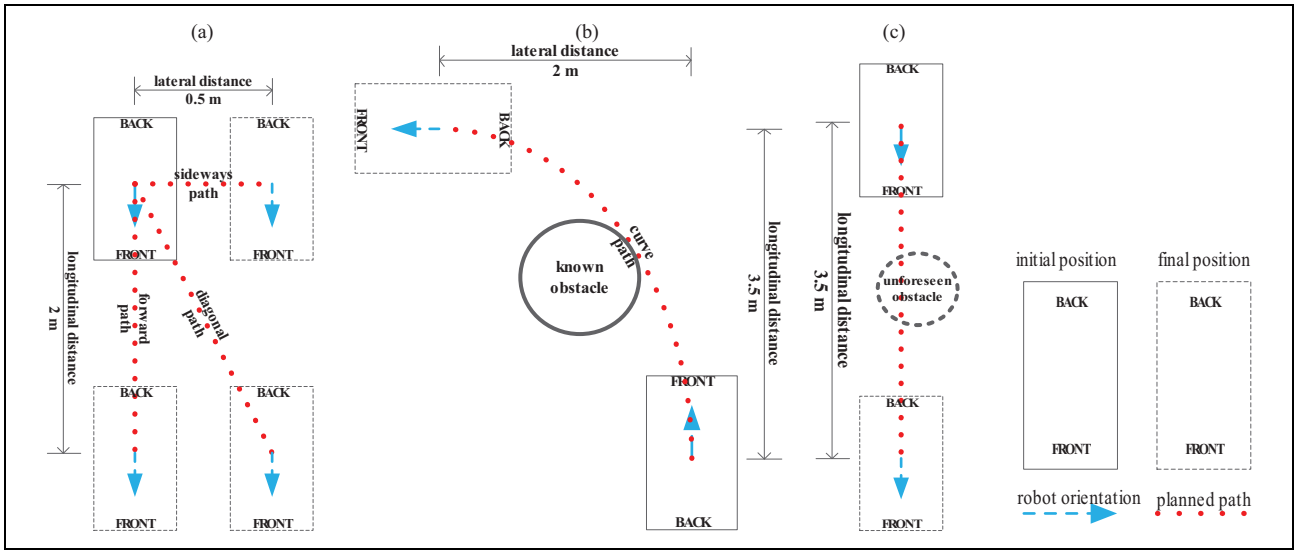


Figure 4. Test paths. (a) Straight-line paths in scenario one; (b) curve-line path in scenario two; and (c) reactive obstacle avoidance path in scenario three.

Table 3. Experimental results of optimizing power consumption.

δ(%)	0.00	20.00	50.00	66.66	80.00
Forward					
Task duration (s)	27.9	26.8	29.9	32.4	49.2
Total energy (J)	2209.9	2110.2	2323.2	2510.7	3745.6
Idle energy (J)	2027.8	1947.8	2171.5	2358.9	3579.2
Motional energy (J)	182.1	162.4	151.7	151.8	166.4
Total power (W)	79.2	78.7	77.7	77.5	76.1
Motional power (W)	6.5	6.1	5.1	4.7	3.4
Sideway					
Task duration (s)	24.5	23.2	19.2	24.5	30.2
Total energy (J)	2004.8	1860.7	1490.1	1879.8	2310.2
Idle energy (J)	1783.1	1690.3	1394.7	1782.9	2194.1
Motional energy (J)	221.7	170.4	95.4	96.9	116.1
Total power (W)	81.8	80.2	77.6	76.7	76.5
Motional power (W)	9.0	7.3	5.0	4.0	3.8
Diagonal					
Task duration (s)	28.2	28.6	30.2	31.6	40.9
Total energy (J)	2271.2	2250.5	2380.3	2481.8	3179.2
Idle energy (J)	2051.8	2082.5	2199.7	2300.2	2977.1
Motional energy (J)	219.4	168.0	180.6	181.6	202.1
Total power (W)	80.5	78.7	78.8	78.5	77.7
Motional power (W)	7.8	5.9	6.0	5.7	4.9
Curve					
Task duration (s)	45.8	46.1	49.7	57.4	70.7
Total energy (J)	3813.4	3803.8	4084.8	4701.6	5676.0
Idle energy (J)	3333.4	3349.7	3612.5	4171.7	5141.3
Motional energy (J)	480.0	454.1	472.3	529.9	534.7
Total power (W)	83.3	82.5	82.2	81.9	80.3
Motional power (W)	10.5	9.9	9.5	9.2	7.6
Obstacle					
Task duration (s)	41.8	44.4	45.6	59.1	90.8
Total energy (J)	3570.7	3740.2	3728.7	4631.0	6974.7
Idle energy (J)	3042.5	3226.9	3318.1	4299.2	6602.5
Motional energy (J)	528.2	513.3	410.6	331.8	372.2
Total power (W)	85.4	84.2	81.8	78.4	76.8
Motional power (W)	12.6	11.6	9.0	5.6	4.1

respectively. The robot successfully avoided the unforeseen obstacle by deviating from the planned path and reached the goal pose. When $\delta = 66.66\%$, the robot performed a diagonal motion to avoid the obstacle. The original ROS dwa_local_planner tends to prefer the trajectories that make the robots face the local path goal or the obstacles. This is because some robots may only be able to observe the environment in front of them by their exteroceptive sensors. It is important to notice that both resulting trajectories with less motional power consumptions are involved with holonomic mobility of the robot such as the sideways motion and the diagonal motion. This is because the local trajectory planner concludes from the cost function that the omnidirectional motions with fewer changes in the orientation of the robot consume less motional power consumption. Thus, the proposed technique effectively takes advantage of the redundant maneuverability of holonomic mobility. In addition, the proposed technique can cope with any type of path and is also able to fulfil online obstacle avoidance as the traditional DWA does.

Experimental validation of energy reduction via the combinational cost objectives

Experiments are conducted to follow the given paths in Figure 4. Again, the ratios α, β, γ, and δ are used in the experimental studies for the purpose of clear expression. To demonstrate the effectiveness in reducing the total energy via the proposed combinational cost objectives of both low motional power consumption and high speed, the weight ratios of α, β, γ, and δ are varied as shown in Table 4.

The experimental results are given in Table 4, where it is found that the task duration is reduced due to the high speed of the motion, by utilizing a higher β. The smaller duration

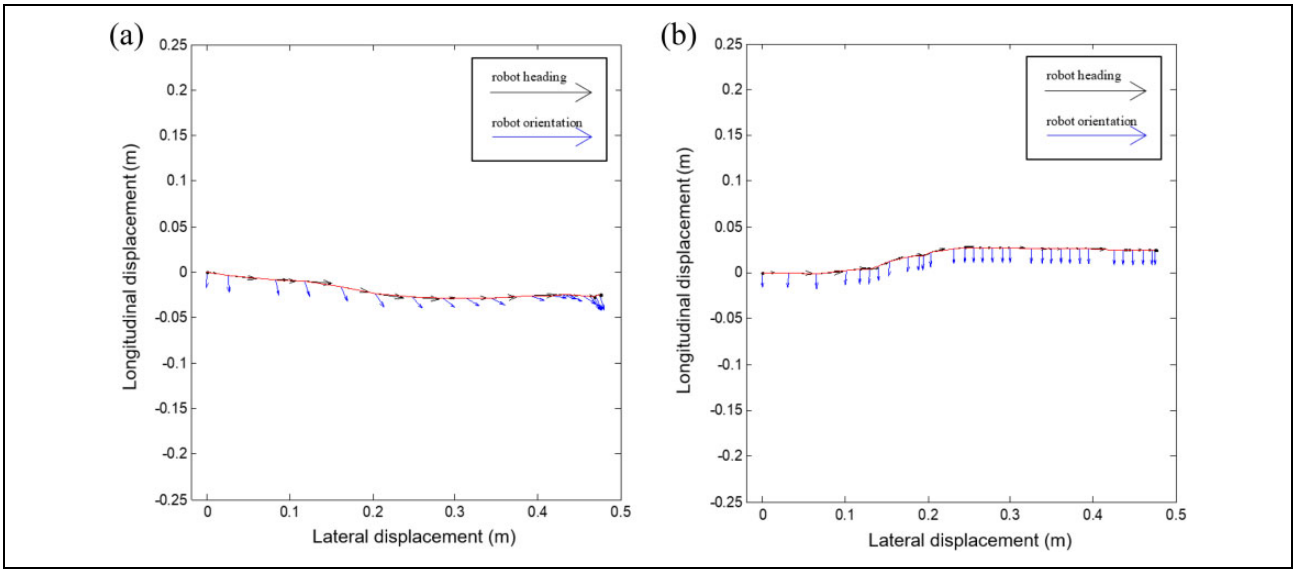


Figure 5. Trajectory of following a sideways straight-line path. (a) Energy weight ratio of 20.00%. (b) Energy weight ratio of 80.00%.

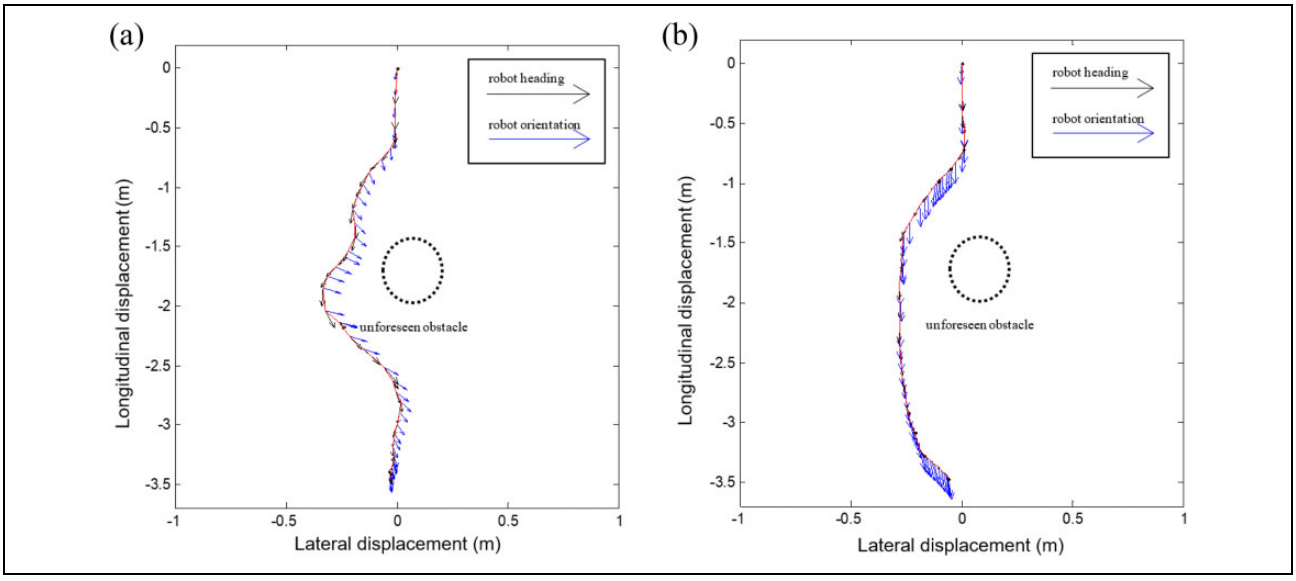


Figure 6. Trajectory of avoiding an unforeseen obstacle. (a) Energy weight ratio of 0.00%. (b) Energy weight ratio of 66.66%.

contributes to less idle energy consumption due to the constant idle power consumption. However, utilizing a high β does not lead to low motional energy due to the high motional power. The use of both higher β and δ reduces the total energy consumption, comparing to the results in Table 3.

In the experiments, it is found that the idle power of the robot is much higher than the motional power. So, the idle energy consumption of the robot is also much higher than the motional energy. β makes a significant contribution to reducing the idle energy consumption of the robot by reducing the task duration. Thus, β is significant, and a higher β can result in less total energy consumption for the experimental paths used in this study. However, it is worth mentioning that both high β and δ result in the trajectories that consume less motional energy to follow the straight-line and curve-line paths than single high β does. In particular, to follow the sideways straight-line path, high β and δ result in the trajectories that consume less idle energy due to the holonomic mobility of the robot.

Discussion

The energy consumption of the mobile robot is a complex problem involving many influence factors. According to the existing methods, building a reliable energy consumption model for the target robot as a solid base is the common approach to solving this problem. The proposed energy consumption model of the four-wheeled Mecanum robot considers the validated factors from the recent energy

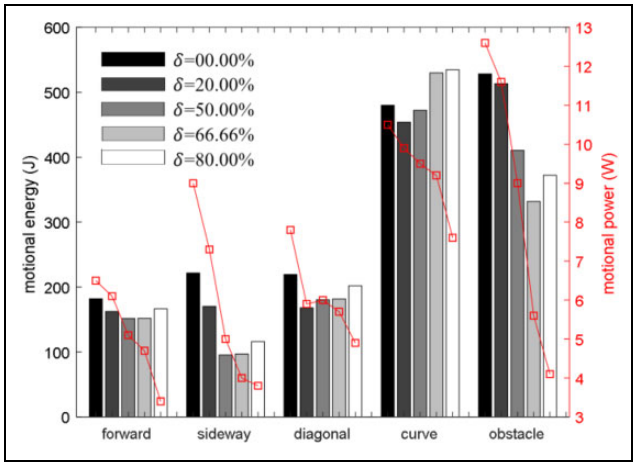


Figure 7. Motional energy and motional power in the experiments.

model works by Liu and Sun[5] and Kim and Kim.[9] The proposed model has been experimentally validated in this study. The experimental energy consumption results in Table 2 show the consistency and accuracy of the model. Thus, the proposed model can be used to predict the robotic energy consumption in DWA. This model is specific to the four-wheeled Mecanum robot by considering the unique dynamics of the Mecanum wheel in omnidirectional motions. Different dynamics should be accommodated in the model in order to cope with other mobile robots. Comprehensive experiments have been performed to show the effectiveness of the proposed technique in optimizing the power consumption and reducing the energy consumption in various autonomous navigation task scenarios. The experiments also prove that the proposed extended DWA local trajectory planning method can cope with any type of path. The experimental results in Table 3 show that the proposed extended DWA-based local trajectory planning method minimizes the robotic power consumption in the global sense for autonomous navigation. It is also observed from Figures 5 and 6 that the proposed technique takes advantages of the redundant maneuverability of holonomic mobility to optimize the power consumption, and it can conduct online obstacle avoidance in the power-optimal way. Comparing the experimental results in Table 4 to the results in Table 3, the overall energy consumption in the global sense is reduced via the combination cost objectives of both low power consumption and high speed. But the overall energy consumption in the global sense is not optimized in this study.

Comparing to the authors' preliminary publication,[19] which proposed a local planning method of considering both local and global energy consumption, this article does not utilize any global information in the local planner. It is difficult to realize absolutely energy-optimal autonomous navigation with only the effort of a local planning method without considering the global information. The accumulation of the energy optimization in each local planning does not necessarily achieve overall energy optimization in the global sense for autonomous navigation. However, the DWA local trajectory planner works in such a way that the optimal velocities of each control cycle are evaluated within each local time window that rolls forward. So, the energy optimization in each constant-time local window achieves the power optimization in the local window and also the overall average power optimization in the global sense. With the power optimization being a new cost objective, the overall energy consumption in the global sense is

Table 4. Experimental results of reducing energy consumption.

Forward			
Weight ratio (%)*	0–100–0–0	0–12–0–88	0–60–0–40
Task duration (s)	24.0	46.1	25.5
Total energy (J)	1920.4	3499.2	2016.0
Idle energy (J)	1743.7	3353.0	1854.7
Motional energy (J)	176.7	146.2	161.3
Total power (W)	80.0	75.9	79.1
Motional power (W)	7.4	3.2	6.3
Sideway			
Weight ratio (%)*	0–100–0–0	10–10–0–80	5–53–0–42
Task duration (s)	22.7	30.2	18.3
Total energy (J)	1861.9	2310.2	1468.8
Idle energy (J)	1654.0	2194.1	1332.1
Motional energy (J)	207.9	116.1	136.7
Total power (W)	82.0	76.5	80.3
Motional power (W)	9.2	3.8	7.5
Diagonal			
Weight ratio (%)*	0–100–0–0	10–10–0–80	5–53–0–42
Task duration (s)	24.5	40.9	25.5
Total energy (J)	1990.1	3179.2	2056.0
Idle energy (J)	1781.1	2977.1	1857.9
Motional energy (J)	209.0	202.1	198.1
Total power (W)	81.2	77.7	80.6
Motional power (W)	8.5	4.9	7.8
Curve			
Weight ratio (%)*	17–49–17–17	6.6–6.6–6.6–80	5–45–5–45
Task duration (s)	43.1	70.7	44.2
Total energy (J)	3572.7	5676.0	3651.2
Idle energy (J)	3132.2	5141.3	3213.9
Motional energy (J)	440.5	534.7	437.3
Total power (W)	82.9	80.3	82.6
Motional power (W)	10.2	7.6	9.9
Obstacle			
Weight ratio (%)*	3–35–35–27	6.6–6.6–6.6–80	5–50–5–40
Task duration (s)	44.9	90.8	46.2
Total energy (J)	3831.5	6974.7	3935.6
Idle energy (J)	3266.8	6602.5	3361.4
Motional energy (J)	564.7	372.2	574.2
Total power (W)	85.3	76.8	85.2
Motional power (W)	12.6	4.1	12.4

*Weight ratio (%): α–β–γ–δ.

reduced via the combinational cost objectives of both low power consumption and high speed.

Conclusion

This article extends the DWA by considering a novel energy consumption model of the Mecanum mobile robot as a new energy-related criterion for the purposes of optimizing power consumption and reducing energy consumption for energy-efficient autonomous navigation. The energy consumption model of the four-wheeled Mecanum robot is built for energy prediction in the DWA. For a given path, the proposed technique optimized the local velocity trajectory based on a cost function involving the energy consumption. The optimization of power consumption and the reduction of total energy consumption were made possible by taking advantage of the holonomic mobility of the Mecanum robot. In addition, the proposed technique can cope with any type of path and is still able to fulfil online obstacle avoidance as the traditional DWA does. Experiments were conducted to validate the energy consumption model and to show the effectiveness of the proposed technique in various autonomous navigation task scenarios.

Acknowledgment

Xie would like to thank the University of Auckland for awarding the Doctoral Scholarship to him.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

1. Adăscăliţei F and Doroftei I. Practical applications for mobile robots based on Mecanum wheels: a systematic survey. *Romanian Rev Precision Mech Opt Mechatronics* 2011; 40: 21–29.
2. Diegel O, Badve A, Bright G, et al. Improved Mecanum wheel design for omni-directional robots. In: *Proceedings Australian conference robotics automation*, Auckland, New Zealand, 27 November 2002, pp. 117–121.
3. Mei Y, Lu Y-H, Hu YC, et al. Energy-efficient motion planning for mobile robots. In: *Proceedings IEEE international conference robotics automation*, 26 April 2004, Vol. 5, pp. 4344–4349. IEEE.
4. Sun Z and Reif JH. On finding energy-minimizing paths on terrains. *IEEE Trans Robot* 2005; 21(1): 102–114.
5. Liu S and Sun D. Minimizing energy consumption of wheeled mobile robots via optimal motion planning. *IEEE/ASME Trans Mechatronics* 2014; 19(2): 401–411.
6. Yang J, Qu Z, Wang J, et al. Comparison of optimal solutions to real-time path planning for a mobile vehicle. *IEEE Trans Syst Man Cybern A Syst Humans* 2010; 40(4): 721–731.
7. Duleba I and Sasiadek JZ. Nonholonomic motion planning based on Newton algorithm with energy optimization. *IEEE Trans Control Syst Technol* 2003; 11(3): 355–363.
8. Kim CH and Kim BK. Minimum-energy translational trajectory generation for differential-driven wheeled mobile robots. *J Intell Robot Syst* 2007; 49(4): 367–383.
9. Kim H and Kim B. Online minimum-energy trajectory planning and control on a straight-line path for three-wheeled omnidirectional mobile robots. *IEEE Trans Ind Electron* 2014; 61(9): 4771–4779.
10. Ray LE, Lever JH, Streeter AD, et al. Design and power management of a solar-powered cool robot for polar instrument networks. *J Field Robot* 2007; 24(7): 581–599.
11. Michaud S, Schneider A, Bertr R, et al. SOLERO: Solar powered exploration rover. In: *The 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, The Netherlands, 2002.
12. Lever JH and Ray LE. Revised solar-power budget for cool robot polar science campaigns. *Cold Regions Sci Technol* 2008; 52: 177–190.
13. Ray L, Price A, Streeter A, et al. The design of a mobile robot for instrument network deployment in Antarctica. In: *Proceedings of the 2005 IEEE international conference on robotics and automation*, Barcelona, Spain, 18 April 2005, pp. 2111–2116. IEEE.
14. Mei Y, Lu YH, Hu YC, et al. A case study of mobile robot's energy consumption and conservation techniques. In: *Proceedings IEEE 12th international conference on advanced robotics*, 18 July 2005, pp. 492–497. IEEE.
15. Fox D, Burgard W and Thrun S. The dynamic window approach to collision avoidance. *IEEE Robot Automat Mag* 1997; 4: 23–33.
16. Brock O and Khatib O. High-speed navigation using the global dynamic window approach. In: *Proceedings IEEE international conference on robotics and automation*, Detroit, MI, 1999, Vol. 1, pp. 341–346. IEEE.
17. Ogren P and Leonard NE. A convergent dynamic window approach to obstacle avoidance. *IEEE Trans Robot Autom* 2005; 21(2): 188–195.
18. Kiss D and Tevesz G. Advanced dynamic window based navigation approach using model predictive control. In: *17th international conference methods and models in automation and robotics*, Miedzyzdroje, Poland, 27 August 2012, pp. 148–153. IEEE.
19. Henkel C, Bubeck A and Xu W. Energy efficient dynamic window approach for local path planning in mobile service robotics. *IFAC-PapersOnLine* 2016; 49(15): 32–37.
20. Xie L, Herberger W, Xu W, et al. Experimental validation of energy consumption model for the four-wheeled omnidirectional Mecanum robots for energy-optimal motion control. In: *IEEE 14th international workshop on advanced motion control*, Auckland, New Zealand, 22 April 2016, pp. 565–572. IEEE.

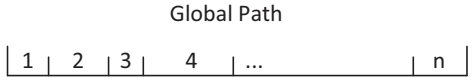
21. Xie L, Scheifele C, Xu W, et al. Heavy-duty omnidirectional Mecanum-wheeled robot for autonomous navigation: system development and simulation realization. In: *Proceedings IEEE international conference on mechatronics*, Nagoya, Japan, 6 March 2015, pp. 256–261. IEEE.
22. Muir PF and Neuman CP. Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In: *Proceedings IEEE international conference on robotics and automation*, March 1987, Vol. 4, pp. 1772–1778. IEEE.
23. Matsinos E. Modelling of the motion of a Mecanum-wheeled vehicle. Institute of Mechatronic Systems, Zurich University of Applied Sciences (ZHAW), Winterthur, Switzerland. Research Article from arXiv, preprint arXiv:1211.2323, 2012 November 10.
24. Scott J, McLeish J and Round WH. Speed control with low armature loss for very small sensorless brushed DC motors. *IEEE Trans Ind Electron* 2009; 56(4): 1223–1229.
25. Quigley M, Gerkey B, Conley K, et al. ROS: An open-source robot operating system. In: *Proceedings open-source software workshop international conference robotics and automation*, Kobe, Japan, 12 May 2009, Vol. 3, p. 5.
26. Marder-Eppstein E, Berger E, Foote T, et al. The office marathon: robust navigation in an indoor office environment. In: *Proceedings IEEE international conference robotics automation*, 3 May 2010, pp. 300–307. IEEE.
27. Siegwart R and Nourbakh IR. *Introduction to autonomous mobile robots*. Cambridge, MA: MIT Press, 2004.

Appendix 1

Overall average power consumption minimization

This section mathematically proves that a sequence of energy-minimization velocity trajectories that are evaluated by the DWA results in the overall average power minimization in the global sense for autonomous navigation.

The DWA local trajectory planner plans the entire robotic velocity trajectory within a local constant-time dynamic window that rolls forward. It is assumed that given a global path to follow, the dynamic window separates the global path into n pieces of path segments after the robot finishes the autonomous navigation task. The velocity trajectory to follow each path segment is planned within one dynamic window. Thus, the time duration t of each velocity trajectory is the same but the length of each path segment is not necessarily the same, as shown below



$$t_1 = t_2 = t_3 = t_4 = \ldots = t_n = t$$

The energy consumptions of the velocity trajectories $e_1, e_2, e_3, e_4, \ldots, e_n$ are positive numbers and are also minimized within each corresponding dynamic window. The overall energy consumption E in the global sense is

$$E = e_1 + e_2 + e_3 + e_4 + \ldots + e_n$$

The overall time T in the global sense is

$$T = t_1 + t_2 + t_3 + t_4 + \ldots + t_n = n \cdot t$$

Thus, the overall average power consumption P in the global sense is

$$P = \frac{E}{T} = \frac{e_1 + e_2 + e_3 + e_4 + \ldots + e_n}{n \cdot t}$$

$$P = \frac{\text{average}(e_1, e_2, e_3, \ldots, e_n)}{t}$$

where t is a constant value, and P is directly proportional to the average energy consumption per number of the dynamic windows $\text{average}(e_1, e_2, e_3, \ldots, e_n)$. Because each energy consumption in $e_1, e_2, e_3, \ldots, e_n$ is the minimized value within each dynamic window, $\text{average}(e_1, e_2, e_3, \ldots, e_n)$ is minimized. Thus, the overall average power consumption is also minimized in the global sense. In addition, the DWA cost objective *energy_consumption* determines the general energy consumption level in every dynamic window. So, *energy_consumption* determines $\text{average}(e_1, e_2, e_3, \ldots, e_n)$ and P.