applied sciences

MDPI

Article

Energy-Aware Navigation Architecture for a Mecanum-Wheeled Robot: Preliminary Energy-Model Sensitivity Analysis

Constantin-Adrian Popescu, Daniela-Mariana Ilie, Ana-Maria Ifrim *, Costel Emil Cotet and Emilia-Maria Popescu

Faculty of Industrial Engineering and Robotics, National University of Science and Technology Politehnica Bucharest, 060042 Bucharest, Romania; cpopescu1107@upb.ro (C.-A.P.); daniela.ilie2607@stud.fiir.upb.ro (D.-M.I.); costel.cotet@upb.ro (C.E.C.); emilia.usurelu@upb.ro (E.-M.P.)
* Correspondence: ana_maria.ifrim@upb.ro

Abstract

Energy-aware navigation requires distinguishing calculated energy from measured savings. This study presents a hierarchical architecture for a mecanum-wheeled robot, combining implemented Adaptive Clearance and Feedback A* (ACF-A*) with Energy- and Clearance-Aware Theta* (ECA-Theta*) and Holonomic Energy-Aware Dynamic Window Approach (H-DWA). The architectural contribution links a common global/local energy formulation to temporary wheel-track traversability memory, so low-progress observations can influence subsequent route evaluation. The quantitative contribution comprises deterministic sensitivity and identifiability analysis, restricted numerical planning and a motor-model comparison. Three prescribed workloads yield 838.1, 1252.0 and 1003.4 J under nominal assumptions. Auxiliary demand contributes approximately 88–90%; 46,875 factorial evaluations and 192 corner evaluations characterize coefficient sensitivity. Yaw and lateral features are confounded, and the two lateral coefficients are not separately identifiable. Nine deterministic kinematic runs execute a four-heading, rest-state ECA-Theta* specialization and H-DWA with exact state feedback, without memory updates. The combined variant changes M2 energy from 1186.7 to 677.5 J, but increases M3 energy from 653.6 to 662.4 J; M1 is unchanged. Reducing the control interval from 0.20 s to 0.10 s changes the M2 energy reduction from 42.91% to 32.97% indicating sensitivity to control discretization. In a matched comparison, the M2 reduction is 45.18% against geometric Theta*/DWA and 41.78% against an energy-aware DWA adaptation; neither M1 nor M3 shows a consistent advantage. A catalog-informed motor replacement lowers nominal auxiliary shares to 75.2–78.0%, showing dependence on model structure. These results establish conditional computational behavior and calibration priorities, not measured savings, adaptive-system performance or physical validation of the complete architecture.

Keywords: autonomous mobile robots; energy-aware path planning; mecanum wheels; energy-efficient navigation; traversability feedback; dynamic obstacle avoidance; intralogistics; ROS 2; sustainable robotics

Check for updates

Academic Editors: Dimitris Vrakas, Dimitrios Bechtsis and Ilias Siniosoglou

Received: 23 October 2026
Revised: 18 November 2026
Accepted: 19 November 2026
Published: 25 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Automation of internal transport in warehouses has evolved from vehicles guided along fixed routes to autonomous mobile robots. These robots estimate their state, interpret maps, and react to local changes along the route. In warehouses, factories, hospitals, and terminals, this transition moves part of the decision-making from central infrastructure

to the robot and increases operational flexibility. However, autonomy is not only a geometric problem. A short mission may include accelerations, in-place rotations, lateral movements, and stops. These maneuvers may consume more energy than a slightly longer route executed smoothly [1–4].

For omnidirectional platforms with mecanum wheels, the trade-off is more pronounced. Holonomic mobility facilitates rack alignment, obstacle avoidance, and docking. However, passive rollers introduce losses. In addition, lateral movement may load the motors differently from longitudinal translation. A planner that treats all segments of equal length as equivalent ignores these differences. Moreover, the energy consumption of the computing electronics and sensors increases with mission duration. Therefore, energy minimization must include both mechanical and auxiliary power [5–8].

The literature includes mature methods for global and local planning, localization and mapping, as well as approaches dedicated to energy-aware planning and control of mobile robots. However, few studies integrate the robot's physical experience and planning within a single loop. Physical experience should subsequently update the cost of the map and route. A geometrically traversable cell may produce wheel slip or limited progress. If the robot returns to the same track, it repeats both the risk and the energy consumption. This observation motivates a temporary traversability memory based on the difference between the commanded kinematics and the encoder response.

The objective is to specify an energy-aware hierarchical architecture and analyze its parametrized energy cost. In this design, safety is a constraint, while energy, distance, orientation, and motion smoothness guide selection among admissible trajectories. Three complementary components are described; the complete architecture is not experimentally validated; a restricted numerical global–local specialization is evaluated separately. ACF-A* (Adaptive Clearance and Feedback A*) is the algorithm implemented in the project and includes clearance, dynamic obstacles, and traversability memory. ECA-Theta* is the proposed global planner and uses any-angle edges and an energy cost. H-DWA (Holonomic Energy-Aware Dynamic Window Approach) is the proposed local controller for holonomic velocities.

Route selection should consider local movement constraints as well as distance. M2 represents a simplified case in which a known region has a restricted travel speed. It tests whether a longer route that avoids this region can reduce modeled mission energy. The test does not simulate the physical cause of the restriction or indicate how often such conditions occur in industrial facilities.

The scientific contribution of this study lies in the formulation and preliminary quantitative assessment of a hierarchical energy-aware navigation architecture for a mecanum-wheeled robot. Building on existing energy-aware navigation approaches, the proposed framework establishes a common energy-cost representation for global route selection and local motion evaluation, while incorporating temporary wheel-track traversability feedback as an architectural extension. The main contributions are as follows:

a. Hierarchical navigation architecture: A framework combining the implemented ACF-A* planner with the proposed ECA-Theta* global planner and H-DWA local controller. The architecture distinguishes obstacle clearance, energy-aware route evaluation, local motion feasibility, and temporary traversability feedback, while explicitly identifying the boundaries between implemented software and numerical prototypes.
b. Energy-aware planning formulation: A parameterized energy-cost model incorporating rolling resistance, acceleration, rotation, auxiliary power, and mecanum-specific lateral motion. The formulation provides consistent energy-related criteria for evaluating global routes and local motion commands, rather than relying exclusively on geometric path length.

c. Numerical evaluation of planning decisions: A restricted ECA-Theta*/H-DWA implementation evaluated through nine deterministic kinematic runs across three obstacle maps. The experiments demonstrate how energy-aware route selection affects calculated mission energy under specified assumptions, including both favorable and unfavorable outcomes, without claiming physical validation or universal performance superiority.
d. Sensitivity, identifiability, and model-structure assessment: A systematic analysis of energy-model parameters using prescribed workloads, factorial and corner evaluations, together with an examination of parameter identifiability and a catalog-informed motor-loss comparison. These analyses identify influential assumptions, structural ambiguities, and priorities for future experimental calibration.

The contributions therefore consist of an integrated architectural formulation and a documented numerical assessment of selected planning and energy-model components. Experimental validation of the complete architecture, including online traversability-memory updates, remains a subject for future research.

Section 2 reviews AMRs and energy-aware planning. Section 3 describes the platform, energy model, and algorithm specifications. Section 4 defines the scope, prescribed workloads, sensitivity procedure, and calibration requirements. Section 5 presents the original workload calculations and the added restricted navigation and motor-model results. Sections 6 and 7 discuss their implications, limitations, and conclusions.

2. State of the Art of Autonomous Mobile Robots

2.1. From AGV to AMR in Intralogistics

AMRs make local navigation decisions, whereas conventional AGVs rely more on prescribed infrastructure and centralized control [1]. This distinction affects task allocation, traffic management and recovery from blockages. Energy-aware navigation must therefore account for industrial operating constraints, not route length alone [2].

Recent intralogistics studies extend this systems view. Chen et al. [9] use semi-open queuing networks checked against simulation to analyze dynamic priorities, throughput and energy in robotic fulfillment systems. Rizqi et al. [10] combine agent-based simulation, efficiency assessment and surrogate optimization to study traffic rules, routing, speed and fleet size. These results motivate explicit waiting and stopping costs, but their fleet-level metrics cannot calibrate this single robot's coefficients.

In robotic fulfillment systems, layout, storage policy, fleet size and congestion jointly affect routing performance [3]. Energy per meter alone is insufficient: a low-energy route may still delay a station or block other robots.

Picking, task assignment and fleet configuration connect routing to warehouse performance [5,6]. This study considers one robot mission, not fleet scheduling.

ISO 3691-4 establishes safety requirements for driverless industrial trucks and their systems [7], including provisions relevant to speed limitation and controlled stopping.

Encoders, IMU and LiDAR provide complementary but imperfect localization signals: slip, inertial drift and repetitive geometry affect the state estimate [11].

ROS 2 [12], navigation frameworks [13] and SLAM tools [14] separate sensing, planning and control. Correct timestamps and coordinate transforms prevent delayed observations from causing false obstacles or unnecessary replanning [15].

Map updates must preserve localization [16]. The architecture distinguishes static occupancy, expiring live obstacles and temporary traversability penalties; a motion penalty is neither an obstacle nor a measured energy value.

Probabilistic localization supplies the uncertainty framework [17]. The software uses footprint inflation and consistency checks, whereas the numerical runs assume exact state feedback.

Dijkstra [18] and A* [19] minimize graph cost; D* Lite [20] reuses search information after changes, and Theta* [21] evaluates line-of-sight shortcuts. Here, shortcuts also require footprint checks and costs for time, acceleration and lateral motion.

PRM [22] and RRT [23] explore feasible states by sampling. This study instead uses a known planar grid and discrete heading labels for deterministic comparison, without claiming optimality for unrestricted energy-dependent motion.

Improved A* and multi-robot planning studies [24,25] balance path length and computation with collision and logistical-conflict avoidance.

ACF-A* uses eight-neighbor A* followed by clearance-aware line-of-sight smoothing. ECA-Theta* instead evaluates any-angle connections during search using the energy model; smoothing an existing path is not equivalent to this search procedure.

The recent literature also emphasizes explicit objective design. Tang and German [26] improve grid-navigation Q-learning through direction-sensitive initialization, annealed exploration and potential-based reward shaping. Iskandar et al. [27] recover reward structures from demonstrations using deep inverse reinforcement learning, evaluated for E-puck swarms in Webots. The former concerns grid learning; the latter concerns collective behavior. Neither is evidence of measured mecanum energy savings or a source of calibrated motor coefficients for this platform.

The Dynamic Window Approach evaluates short trajectories generated by admissible velocities, subject to motion limits and stopping clearance [28]. This permits local obstacle avoidance without rebuilding the global map.

The global dynamic window [29] and convergent formulations [30] address global guidance and local minima. The proposed H-DWA therefore scores route deviation and estimated execution energy alongside clearance and progress.

Global and local planning are increasingly combined [31–33]. These studies motivate consistent costs and constraints across planning levels. The proposed ECA-Theta* and H-DWA share energy terms, but only a restricted kinematic specialization is evaluated here, without the complete sensing and memory loop. A return-to-start leg is included only when explicitly specified as part of a mission; it is not implicit in every route cost.

Predicting obstacle behavior can improve DWA safety in dynamic environments [34]. The current platform uses a more conservative variant. LiDAR points are filtered against the static map and retained with a TTL of 0.90 s. Replanning is triggered if they intersect the 1.35 m route horizon.

2.2. Energy Models and Energy-Aware Planning of an AMR

Energy-aware planning for mobile robots shows that speed profile, acceleration, and activation of computing resources affect autonomy [35–37]. This approach changes the objective from “the shortest route” to “the lowest energy for a safe mission.” The two objectives coincide only under ideal conditions.

Henkel et al. [38] introduced an energy-efficient DWA and reported a 9.79% reduction compared with conventional DWA on their platform. Maidana et al. [39] also considered computing-resource consumption and reported larger reductions when sensors or components can be disabled depending on the route. The results are not directly transferable between robots because energy, mass, speed, and auxiliary power significantly affect the achieved energy performance.

For mecanum-wheeled robots, Xie et al. [40] integrated energy-consumption minimization into the DWA algorithm. Their experiments show that lateral movement may

consume more energy than translational motion. Energy modeling dedicated to four-wheel mecanum platforms [41,42] confirms that holonomic flexibility is obtained with losses dependent on direction, payload, and acceleration. Consequently, the mecanum term in this paper penalizes lateral movement and slip without prohibiting the maneuver when it reduces time or is necessary for docking.

A recent systematic review of energy efficiency in AMRs [43] classifies interventions into energy sources, consumption models, locomotion, hardware, planning, and scheduling. Efficient-motion studies [44] emphasize that stops and restarts can cancel the advantage of a geometrically efficient route. Consequently, the energy consumption of auxiliary subsystems is modeled as the product of baseline power and operating duration. Safety-imposed stops are analyzed separately because they explain part of the energy consumption.

Yao et al. [45] proposed an improved DWA version integrating energy management and fuzzy-logic control. Raj and Kos [46] analyzed the influence of energy-aware planning on mobile-robot autonomy. Recent predictive approaches extend DWA by anticipating the trajectory over multiple steps [47]. Although this strategy can reduce energy consumption, it increases computation time. In real-time control, the absolute execution time is less relevant; the important criterion is completing the computation within the available control-loop interval.

For mecanum platforms, Zakharov et al. [48] study switching between holonomic and nonholonomic motion while considering energy and tracking. Nghiem et al. [49] compare control strategies using a sliding-mode controller and a robot model. Maroșan et al. [50] implement MATLAB trajectory generation and energy estimation for different motion modes. Mokhtari et al. [51] extend energy-aware planning to ambiently powered multi-robot systems, where harvesting changes the objective. These 2025–2026 studies reinforce the need to specify trajectory timing, control assumptions and energy boundaries, rather than compare percentages across unlike platforms.

Energy models may be physics-based, data-driven, or hybrid. The present formulation uses physical terms and assumed effective coefficients. Its coefficients have not been experimentally identified for this platform.

At low speeds, the energy consumption of auxiliary subsystems may become the dominant component of total consumption. For example, for an auxiliary power of 18 W, increasing travel duration by 10 s results in an additional 180 J, independent of route length. This observation highlights that the planner should not favor a significant detour merely to avoid a small-amplitude rotation. Consequently, energy-cost weights must be calibrated together with speed limits and power consumed in idle operation. The proposed global cost function explicitly retains both the distance and energy components, thereby avoiding redundant introduction of the same information through normalization procedures.

2.3. Sustainability, Interoperability, and Research Directions

Lower mission energy may extend operation between charges; battery aging, charging efficiency and fleet usage determine wider sustainability benefits [2,43].

ROS 2 messages and YAML configuration support reusable interfaces, but energy coefficients require platform-specific calibration. The transferable elements are the model structure and calibration protocol.

Nonlinear predictive control for omnidirectional robots [52] handles constraints with a more detailed dynamic model and greater computation. H-DWA offers separately interpretable scoring terms for future ROS 2 integration; that integration is not evaluated here.

Broader path-planning surveys [53] and the experimentally evaluated crawler traction model of Grigore et al. [54] motivate separating navigation choices from platform-specific mechanics. The crawler results cannot provide mecanum coefficients. Grigore et al. [55] ad-

ditionally demonstrate coordination of ground and aerial robots in an emergency-response application; their environmental-impact discussion concerns system integration, not a transferable energy-saving percentage for this robot.

Table 1 synthesizes links between energy estimation, safe clearance, local feasibility, and temporary wheel-level traversability feedback [38–47,52]. The proposed contribution is this integration and its auditable interfaces. The literature review does not establish universal novelty, and the present model calculations do not demonstrate superiority over the cited algorithms.

Table 1. Role of the literature synthesis in the proposed architecture.

Domain	Sources	Dominant Themes	Role
AMR and intralogistics	[1–6]	Planning, control, layout, picking, and fleets	System integration
Safety and perception	[7,8,11–17]	Standards, LiDAR, IMU, SLAM, dynamic maps	Hard constraints
Global planning	[18–25]	Dijkstra, A*, D* Lite, Theta*, PRM, RRT	Path and smoothness
Local planning	[28–34]	DWA, global guidance, fusion, and prediction	Online reaction
Energy	[35–47]	Motion, auxiliary systems, mecanum, fuzzy control, and prediction	Consumption per mission
Holonomic control	[52]	Nonlinear omnidirectional predictive model	Dynamic feasibility

To clarify the distinction from previous energy-aware mecanum studies [40–42], Table 2 compares their contributions with the proposed shared global/local energy formulation and temporary wheel-track feedback. This is an architectural distinction, not demonstrated performance superiority.

Table 2. Comparison with previous energy-aware mecanum robot studies.

Study	Main Focus	Distinction from the Present Work
Reference [40]	Energy-aware DWA for mecanum robots	Proposes a shared energy-cost formulation for both global and local planning levels.
Reference [41]	Energy modeling for mecanum platforms	Integrates energy-cost estimation with route evaluation and temporary traversability feedback.
Reference [42]	Mecanum-specific energy characteristics	Combines energy-cost evaluation with hierarchical planning and wheel-track memory.
Present study	Hierarchical energy-aware architecture and preliminary sensitivity analysis	Integrates implemented ACF-A* traversability feedback with proposed ECA-Theta* and H-DWA extensions, using a common energy-cost formulation.

The comparison highlights the combination of energy accounting and expiring wheel-track costs, rather than energy-aware navigation alone. The M1–M3 evaluation excludes memory updates and does not establish superiority over [40–42].

2.4. Basis for the Matched Navigation Comparison

Xie et al. [40] developed energy-aware local navigation for mecanum robots by extending DWA to three-component holonomic velocity commands. They added an energy term to the path, local-goal and obstacle costs. Their model accounts for idle consumption,

kinetic energy, friction and motor losses, and was evaluated on a physical robot. Holonomic DWA and an energy term in its objective are therefore established approaches. Here, we examine whether energy-aware global planning adds a benefit when an energy-aware local controller would otherwise follow a geometrically selected route.

In offline simulations, Teso-Fz-Betoño et al. [47] report an average energy reduction of approximately 9% relative to their traditional predictive DWA. Their prediction and energy formulation differ from those used in M1–M3, which prevents direct ranking of their published saving against our results.

To compare the methods under the same conditions, we use common maps, robot constraints and one energy model. Basic Theta* [21] and A* generate geometric routes, which are followed by geometric DWA or an energy-aware DWA adaptation based on Xie et al. The following sections describe the changes made to these methods. The resulting comparisons concern their behavior in our simulator, rather than the performance of the original authors' robots.

3. Materials and Methods

3.1. General Architecture

The general AMR architecture separates sensing, state estimation, mapping, planning, control, memory and embedded execution (Figure 1). LiDAR, IMU and encoder observations feed state estimates and cost layers. Implemented ACF-A* uses temporary wheel-track feedback; ECA-Theta* and H-DWA are the energy-aware planning extensions. The complete architecture has not been experimentally integrated or validated.

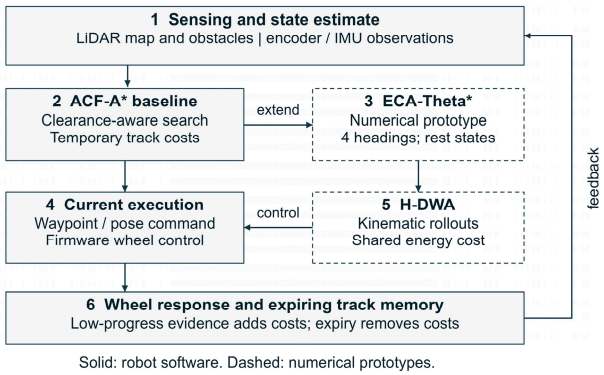

Figure 1. Target architecture and evidence boundary. Solid boxes denote existing robot software; dashed boxes denote separate numerical prototypes. Only the restricted global–local pair is executed here. Connections to physical sensing, firmware and temporary memory describe intended integration, not a complete-system experiment (drawn in PowerPoint).

The global planner generates the route, the local controller selects the velocity, and the firmware commands the motors. Encoder feedback returns both to odometry and to the low-progress detector. Within this target architecture, ECA-Theta* and H-DWA are proposed extensions; their complete physical integration is outside the present analysis. Section 5.6 reports the restricted numerical pair.

ACF-A* is implemented with eight-neighbor search, clearance-aware smoothing and temporary traversability costs. The proposed ECA-Theta* and H-DWA procedures are described in the corresponding algorithm descriptions. Their restricted numerical evaluation in Section 5.6 is separate from W0–W2.

Figure 2 separates route selection from local command selection. Numerical energy is integrated along the exact-state kinematic history; ACF-A* and its memory interface are not executed baselines in M1–M3.

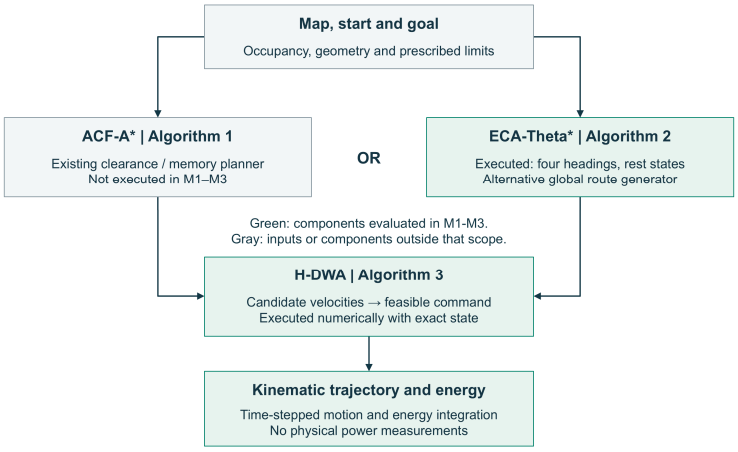

Figure 2. Algorithmic information flow and numerical scope. ACF-A* and ECA-Theta* are alternative global planners, not successive processing stages. The green blocks identify the restricted ECA-Theta*/H-DWA implementation used for M1–M3. Physical sensing, firmware and memory feedback are excluded; Algorithms 1–3 specify the procedures separately (drawn in PowerPoint).

3.2. Robot Platform

The reference platform is a four-mecanum-wheel mobile robot with independently driven wheels and a lifting assembly. A Raspberry Pi 5 runs ROS 2 Jazzy, while a Raspberry Pi Pico 2 W handles low-level control [56]. The platform includes motor encoders, LiDAR, an IMU, ultrasonic sensors, and load cells. Table 3 summarizes the relevant geometric and software parameters. The present numerical analysis uses assumed mass and ideal kinematic feedback; payload sensing and lifting energy are not included in the evaluated model.

Table 3. Platform parameters used in the model and software configuration.

Parameter	Value	Note
Kinematics	4 mecanum wheels	vx, vy, ω
Wheel radius	0.060 m	robot.yaml
Wheelbase	0.400 m	center-to-center
Track width	0.240 m	center-to-center
Encoder	1920 pulses/output revolution	4 channels
Linear velocity	0.30–0.45 m/s	mission profile
Maximum angular velocity	0.45 rad/s	mission profile
PWM	20 kHz	maximum duty 0.55
Cost clearance	0.42 m	weight 1.35
Live obstacles	TTL 0.90 s; 2.8 m	max. 260 points
Traversability memory	TTL 90 s	stall cost 0.35; slip 0.20

Figure 3 documents the reference geometry; construction details appear in [57]. Neither the CAD model nor the inventory constitutes energy calibration.

Table 3 records geometry and selected software settings. The 16.83 kg reference mass in Table 7 is a modeling input; current deployment notes describe an approximately 25 kg prototype. Each physical trial therefore requires an independently measured total mass, including battery, electronics, payload, and attachments.

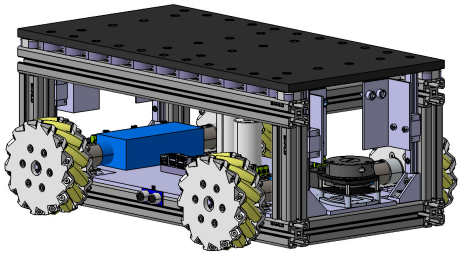

Figure 3. CAD view of the robot platform: mecanum wheels, load-bearing frame, lifting actuator, and upper platform. The image documents the design and is not an experimental record.

3.2.1. Electrical Architecture

Motor and regulated logic supplies are separated to limit startup voltage drops and interference. Protection and conductor sizing require verified current ratings [58]; Figure 4 includes feedback paths.

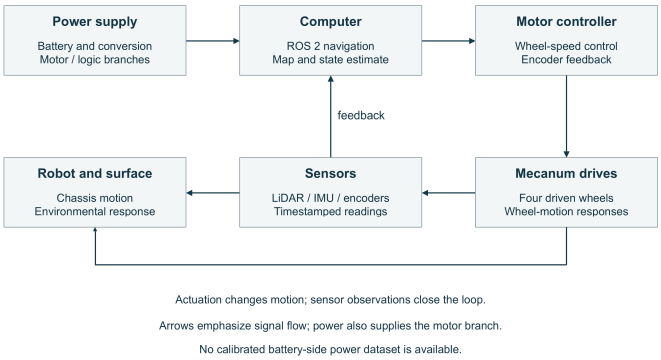

Figure 4. Functional actuation and sensor-feedback paths. The battery supplies the motor branch through protection and the logic branch through conversion. Encoders, IMU and LiDAR return timestamped observations to navigation. This functional diagram does not replace pin-level wiring or demonstrate calibrated electrical measurement (drawn in PowerPoint).

Firmware duty-cycle limits, startup assistance and 100 ms encoder reporting are not represented in the kinematic evaluation or simplified energy equation.

The safety chain must be able to independently bring the enable signals to an inactive state. At the software level, absence of commands, communication loss, or timeout expiration leads to a zero command. At the embedded level, the watchdog stops PWM. At the electrical level, the emergency stop cuts power to the power stages. Recovery is permitted only after a clear reinitialization to avoid unintended restart.

Figure 4 represents a principle diagram of an AMR and does not replace the pin-level wiring diagram. It nevertheless establishes the necessary boundaries: motor current must not flow through the sensor ground, the logic branch must remain stabilized during startup, and the emergency stop must disable power without depending on ROS 2. The Raspberry Pi may remain powered to save logs if the risk analysis permits this separation. Table 4 lists interfaces and safe states; Table 5 identifies representative components rather than certified electrical ratings.

Table 4. Electrical interfaces.

Connection	Domain	Signal	Protection	Safe State
Battery → protection	Power DC	Ubat, total current	fuse + E-stop	unpowered
Protection → drivers	Power DC	4 branches	branch fuse	motor isolated
Pico → drivers	RPWM/LPWM	20 kHz	enable + watchdog	PWM = 0
Encoders → Pico	Digital A/B	4 pairs	pull-up + filtering	encoder fault
DC/DC → Pi	5 V logic	sized current	UVLO + decoupling	shutdown
Pi ↔ Pico	serial/USB	command + state	CRC/timeout	zero command
Pi → LiDAR	USB/serial + 5 V	2D scanning	driver timeout	expired layer
IMU/HX711	I^2C/GPIO	low-level signal	clean ground	invalid data

Table 5. Electrical and electronic components integrated into the structure of the designed mobile robot.

Component	Description	Quantity	Supply/Rating
Li-ion battery	Main source—powers the robot without a cable	1	21.6 V–25.2 V
24 V -> 5 V DC/DC converter	Powers the Pico, encoders, ultrasonic sensors, logic drivers, and level adapters	2	24 V
DC motor driver	Each motor has its own driver. The driver switches 24 V power to the motor and receives PWM signals from the Pico	4	24 V
Pololu DC motor	Each wheel is equipped with a geared DC motor and encoder	4	4 × 60 W
Ultrasonic sensors	Located at the rear and sides of the AMR	3	5 V
LiDAR sensor	2D scanning and mapping. Publishes laser measurements on/scan. Visible obstacles become live red points in the UI	1	2 W
INA219 current sensor	Used to measure current on a selected branch and for energy diagnostics	1	26 V
Raspberry Pi 5	Runs ROS 2, the web application, SLAM, localization, planning, and map services	1	5 V
Load sensor	Mounted under the upper platform of the AMR to measure payload	4	5 V

3.2.2. Sensor Synchronization

LiDAR, encoders, IMU and ultrasonic sensors require timestamps and validated coordinate transforms. M1–M3 instead use exact numerical state feedback.

Data-acquisition frequencies are established according to the dynamics of each signal. In the current implementation, the firmware transmits encoder data every 100 ms. The LiDAR operates at the device's native frequency. The IMU is sampled sufficiently rapidly for robust estimation and filtering of yaw angle. Signals are not concatenated by index but temporally realigned. For future energy measurements, acquisition bandwidth and sampling must resolve the relevant boost and braking transients. The analyzer clock would need synchronization with ROS 2 through a common event; no such synchronized power records are analyzed here.

Figure 5 and Table 6 define the intended acquisition chain. Before a physical campaign, validation must proceed from raw sensor values and axis conventions to timestamps, coordinate transforms, state estimation, and planning. Raw and filtered signals must both be retained. The existence of sensor interfaces or an INA219 component does not establish a calibrated total-system power measurement.

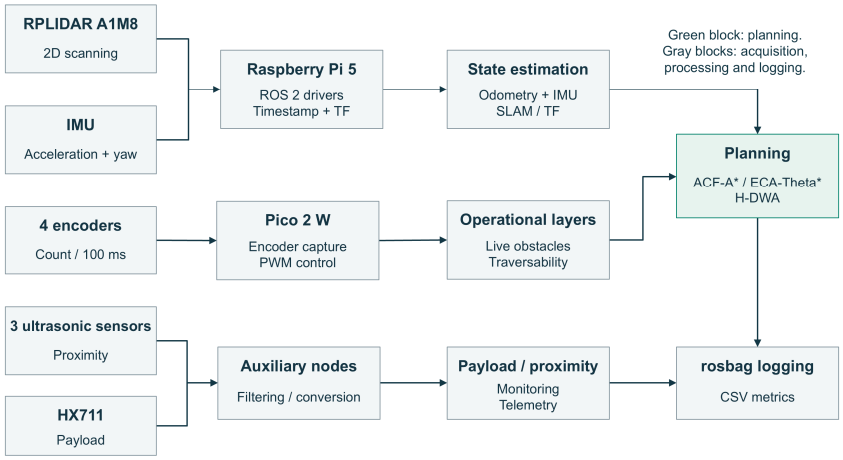

Figure 5. Target acquisition and processing interfaces connecting sensors, state estimation, planning, and monitoring. ECA-Theta* and H-DWA denote the proposed extensions to the navigation architecture (drawn in PowerPoint).

Table 6. Signals, frames, and failure modes of the sensor chain.

Source	Data	Reference	Use	Typical Fault
LiDAR	2D polar scan	laser frame	SLAM, live obstacles	dropout, reflections
IMU	z-axis gyro + acceleration	IMU frame	yaw rate, bias	drift, vibration
Encoders	Δcount FL/FR/RL/RR	base frame	odometry, stall/slip	direction, pulse loss
Ultrasonic sensors	rear/left/right distance	local frames	motion guard	multiple echo
HX711	load and distribution	platform	energy profile	offset, temperature
Power analyzer	U(t), I(t)	acquisition clock	energy and calibration	saturation, offset

3.3. Kinematic Model

The motion of the mecanum platform is modeled using kinematic variables associated with the robot state and velocity.

The planar state is $\eta = [x, y, \psi]^T$ and body velocity $\nu = [v_x, v_y, \omega]^T$. The body x-axis points forward and y-axis left; positive yaw is counterclockwise. r is wheel radius, L the front-to-rear wheel-center spacing, W the left-to-right spacing, and $h = (L + W)/2$. Wheel order is front–left, front–right, rear–left and rear–right. The X-pattern inverse kinematics follow [59–61]:

$$[\omega_{FL}, \omega_{FR}, \omega_{RL}, \omega_{RR}]^T = \frac{1}{r}\begin{bmatrix} 1 & -1 & -h \\ 1 & 1 & h \\ 1 & 1 & -h \\ 1 & -1 & h \end{bmatrix} [v, v, \omega]^T. \quad (1)$$

Let A denote the complete 4×3 inverse-kinematic matrix in Equation (1), including $1/r$; $\Omega = A\nu$. Its entries encode roller signs and $h = (L + W)/2$. Forward kinematics reconstructs vx, vy, and yaw rate from the speeds of the four wheels. The odometry node fuses wheel yaw rate with the IMU using a weight of 0.35 for the inertial signal, after bias calibration and application of a consistency gate. The value 0.35 is used as the initial weight for the inertial channel. Thus, 65% of the estimate remains determined by wheel odometry, while the IMU corrects rapid yaw variations. This choice limits propagation of inertial noise and drift in stationary operation without eliminating the IMU contribution during rotations. The value is not treated as a universal constant. It must be recalibrated by minimizing angular error

against an external reference during rotational maneuvers performed at different speeds and payloads.

Integration into the global frame uses the rotation ψ:

$$x = v_x \cos\psi - v \sin\psi; \quad y = v_x \sin\psi + v \cos\psi; \quad \dot{\psi} = \omega. \tag{2}$$

The kinematic model does not explicitly represent motor dynamics or the forces applied to the wheel rollers. These effects are introduced into the cost through acceleration, rotation, and lateral displacement. This simplified cost is intended to limit computational demand and support coefficient identification from future electrical data without a full rigid-body model. Neither runtime compatibility nor coefficient identification is demonstrated here.

3.4. Edge Energy Model

For an edge between states i and j, the estimated energy is the sum of five components. The separation is useful for both calibration and interpretation of the results:

$$E_{ij} = E_{roll} + E_{acc} + E_{rot} + E_{aux} + E_{mec}. \tag{3}$$

$$E_{roll,ij} = \frac{mgcd_{ij}}{\eta_{drive}}; \quad E_{acc,ij} = \frac{mQ_{ij}}{\eta_{drive}}. \tag{4}$$

$$Q_{ij} = \sum_{k=0}^{N-1} \max(0, \frac{{v_{+1}}^2 - v^2}{2}). \tag{5}$$

The acceleration-energy term sums positive increments in one-half of squared translational speed along each edge, including internal acceleration extrema. Thus, rest-to-rest edges retain their acceleration cost. Deceleration receives no regenerative credit. Summing Q_{ij} over consistently defined edge profiles gives the workload quantity Q in Equation (12).

Here, $v_k = \sqrt{(v_x{}^2 + v_\gamma{}^2)}$ is translational speed, including internal acceleration extrema. Q has units m^2/s^2. A rest-to-rest motion reaching 0.30 m/s contributes $Q = 0.045\ m^2/s^2$; zero endpoint velocities do not eliminate acceleration energy. r, L and W are geometric inputs, while c_r and ηdrive are assumed effective parameters.

$$E_{rot} = k_{rot}\Psi, \quad \Psi = \int |\omega| dt; \quad E_{aux} = P_{aux}T. \tag{6}$$

$$E_{mec} = (k_{lat} + k_{slip})Y, \quad Y = \int |v_y| dt. \tag{7}$$

The rolling and acceleration terms use an assumed effective drivetrain efficiency. The yaw term aggregates rotational losses; it does not represent a separately validated motor model. Auxiliary energy covers the selected navigation-system boundary, excluding lifting unless explicitly included. All coefficients in Table 7 are uncalibrated analysis inputs. Their physical roles motivate the model structure but do not justify the numerical values as measured properties of the prototype.

For a monotonic lateral primitive, |Δy_B| equals the time integral of |v_y|. Equation (7) then depends on k_lat + k_slip, not on two independently observable coefficients. The calculations below use k_lat,eff = 11 J/m and retain separate coefficient perturbations only to expose this redundancy. A future slip term must use an independent wheel–ground motion discrepancy. Command-versus-encoder mismatch alone also cannot establish physical slip: it may indicate stall, actuator saturation, encoder error, or drivetrain response.

Table 7 separates assumptions from measurements: $c_r = 0.035$, ηdrive = 0.70, Paux = 18 W, krot = 1.8 J/rad and the lateral coefficients 8 and 3 J/m are uncalibrated nominal choices. The ±50% ranges (except ηdrive = 0.50–0.90) are deliberately selected stress tests, not literature-derived robot properties or confidence intervals. The reference mass 16.83 kg is not a verified weighing of the assembled prototype.

Table 7. Nominal assumptions, sensitivity bounds, and calibration requirements. Bounds are stress-test choices, not confidence intervals.

Parameter	Nominal Input	Analysis Interval	Basis and Calibration Requirement
m	16.83 kg	16.83–26.83 kg	Reference-model mass, not a verified loaded mass; weigh the complete robot and each payload.
g	9.80665 m/s^2	Fixed	Conventional standard gravity; negligible relative to the stated coefficient uncertainty.
c_r	0.035	0.0175–0.0525	Assumed equivalent rolling resistance; fit from steady straight runs after auxiliary subtraction, jointly with efficiency.
η_drive	0.70	0.50–0.90	Assumed effective efficiency; constrain with electrical/mechanical measurements or an independent calibrated value.
P_aux	18 W	9–27 W	Assumed steady auxiliary load; measure with navigation electronics active and traction inactive.
k_rot	1.8 J/rad	0.9–2.7	Assumed yaw-energy slope; fit residual energy versus accumulated yaw for several rates and payloads.
k_lat	8 J/m	4–12	Assumed lateral-energy slope; only the sum with k_slip is identifiable in the present proxy.
k_slip	3 J/m	1.5–4.5	Legacy lateral-motion proxy; not a measured slip coefficient. Independent ground-motion data are needed for true slip calibration.

3.5. Implemented ACF-A and Proposed Planning Extensions*

These elements form the basis of the ACF-A* formulation, in which route cost is determined by integrating geometric, risk, and direction-change criteria.

The ACF-A* algorithm uses an eight-neighbor grid, and cost estimation is based on the octile heuristic. Edge cost is the geometric step multiplied by cell risk, plus the direction-change penalty. Risk combines unknown space, a quadratic clearance penalty, and traversability cost. Expansion prohibits corner cutting, checks the footprint, and uses strict or relaxed clearance variants only as recovery mechanisms. The route is then smoothed by line of sight with footprint checking.

$$c_{\mathrm{ACF}}(i,j) = d_{\mathrm{ij}}[1 + R_{\mathrm{cell}}(j)] + \lambda_{\mathrm{dir}}\,\Delta dir. \tag{8}$$

$$R_{\mathrm{cell}} = R_{\mathrm{unknown}} + R_{\mathrm{clearance}} + R_{\mathrm{memory}}. \tag{9}$$

Traversability memory is updated when a sufficient command exists but wheel speeds indicate stall or a progress ratio below the threshold. Cells corresponding to wheel tracks receive a cost rather than the entire footprint, and values expire after 90 s. If the layer intersects the active route, replanning is triggered after a 1.50 s cooldown. This granularity reduces the risk that a single observation permanently blocks a corridor.

The main computational steps of implemented ACF-A*, including clearance and temporary traversability costs, are given in Algorithm 1.

Algorithm 1. Readable pseudocode for ACF-A* planning.

```
function PLAN_ACF(start, goal, map, live_obstacles, memory):
    open ← priority queue(start); g[start] ← 0
while open is not empty:
current ← extract node with minimum f(open)
if current = goal: return SMOOTH(RECONSTRUCT(current))
for each valid neighbor in 8-connectivity:
if corner_cut or invalid_footprint(neighbor): continue
    clearance ← distance_to_obstacle(neighbor)
risk ← unknown + clearance_penalty(clearance)
risk ← risk + traversability_cost(neighbor, memory)
edge ← distance(current, neighbor) · (1 + risk)
edge ← edge + direction_change_cost(current, neighbor)
if g[current] + edge < g[neighbor]:
parent[neighbor] ← current; update g and f
return safe fallback or failure
```

The direction-change penalty in ACF-A* depends on the incoming path direction. The implemented search retains only one cost and predecessor per spatial cell; this penalty is evaluated with respect to the currently retained predecessor and should be interpreted as a predecessor-dependent heuristic; and optimality for a direction-dependent path cost is not established. An exact state-augmented treatment would retain distinct arrival-direction states, for example $s_{\mathrm{ACF}} = \left(i_x, i_y, i_\theta\right)$, with separate $g(s_{\mathrm{ACF}})$ and parent(s_{ACF}) values. This state augmentation is identified as a potential refinement, not as a feature already verified in the implemented ACF-A*.

Before the search, the occupancy map is transformed into a collision map and a distance field from obstacles. The rectangular footprint is projected at multiple orientations or conservatively approximated by a circumscribed radius. Occupied cells and cells where the footprint intersects an obstacle are rejected. Cells within 0.42 m receive a gradual cost. The cost is maximum near the obstacle and zero at the edge of the radius.

Dynamic obstacles are not permanently maintained on the map. Each LiDAR point is transformed into the map frame, checked against the static structure, and retained with a timestamp. At each cycle, points older than the TTL are removed, and clearance caches are invalidated only if the layer sequence has changed. This strategy reduces repeated computation and prevents accumulation of temporary objects, contributing to safe AMR movement.

Monitoring robot behavior complements risk evaluation during motion.

The implemented detector compares command-derived wheel speeds with encoder-derived speeds using low-progress thresholds of 0.35 and 0.85, a minimum linear command of 0.12 m/s, and an expected wheel speed of at least 0.8 rad/s. These thresholds are software settings, not calibrated slip measurements. The current configuration permits recording during final-yaw operation. Ground-relative motion from an independent pose reference is needed to distinguish wheel spin from actual chassis progress.

Upon confirmation, the coordinates of the four contact points are calculated from the robot position and L–W geometry. Only tracks associated with the wheels showing low-progress evidence are marked, with a radius of 0.020 m and a cost limited to 1.0. A new observation may increase the cost, while expiration removes it after 90 s. The mechanism is deliberately reversible: the surface may be temporarily contaminated, and a permanent penalty would turn local experience into a false obstacle.

ECA-Theta* compares feasible neighbor transitions with parent-to-neighbor line-of-sight shortcuts. The general formulation depends on arrival heading and body velocity. The executed specialization below retains four headings and fixes terminal velocities to

zero; it is not the unrestricted moving-state planner. Equation (10) defines the general normalized cost.

$$c_{ij} = w_d \frac{d_{ij}}{d_{ref}} + w_E \frac{E_{ij}}{E_{ref}} + w_c C_{ij} + w_\psi \frac{\Psi_{ij}}{\pi} + w_m M_{ij}. \quad (10)$$

All weights are nonnegative and the reference scales are fixed. Occupancy and footprint collision are hard constraints; clearance and memory are dimensionless costs, not joules. The numerical specialization uses $E/60\,\mathrm{J} + 0.02d/\max(c_{min}, 0.05\,\mathrm{m})$, with memory disabled. Its geometric comparator uses $d/(1\,\mathrm{m})$ plus the same clearance term. Section 4.6 gives the exact settings.

To preserve arrival-condition dependence, the general state is $s = (i_x, i_\gamma, i\psi, iv_x, iv_\gamma, i\omega)$, not a spatial cell or scalar speed alone. Cost $g(s)$ and parent(s) are retained separately for each state. The executed restriction keeps four heading labels per cell and sets all three terminal velocity indices to zero. Each transition rotates at rest if needed and then translates with a rest-to-rest profile at fixed chassis heading. Travel angle and chassis heading remain independent. This restricted state space is explicit; no full moving-state implementation or optimality proof is claimed.

Algorithm 2 presents the executed four-heading, rest-state ECA-Theta* specialization.

Algorithm 2. Executed rest-state ECA-Theta* specialization; the unrestricted moving-state design remains future work.

```
INPUT: static map, start/goal pose, energy coefficients and limits.
State s = (ix, iy, heading); all arrival velocities are zero.
Retain separate g[s], parent[s] and depth[s] for four headings.
Initialize OPEN using the rolling/auxiliary energy lower bound.
Pop the lowest score; expand eight neighbors and in-place rotation.
For each terminal heading and origin in {s, parent[s]}:
    Check the complete line against the circular collision envelope.
    Generate a rest-to-rest rotation, then a translation speed profile.
    Integrate distance, time, positive kinetic increments, yaw and lateral travel.
    Evaluate E/60 J + 0.02d/max(cmin, 0.05 m).
    Update/reopen the full heading label when its cost improves.
    On a cost tie, prefer the label with fewer transitions.
Stop at the complete goal pose; reconstruct the retained route.
OUTPUT: route, edge features/costs and expansion trace.
SCOPE: four headings, rest states, static map; no memory update.
```

For a mecanum-wheeled robot, edge direction and chassis heading are independent. ECA-Theta* compares feasible headings, including orientation-preserving lateral or diagonal translation and rotation toward travel. These choices change body-frame displacement, yaw and energy while remaining subject to footprint, clearance and motion constraints. M1 demonstrates lateral translation at unchanged heading.

The reference scales must be determined on calibration workloads separate from evaluation workloads and stored in the configuration. Normalization that changes with the currently observed candidates would change the optimization problem between runs. A reproducible implementation must therefore freeze these scales and record the treatment of unknown cells, diagonal crossings, footprint checks, and goal orientation.

An admissible heuristic is necessary but is not sufficient to guarantee optimality for a heading-dependent any-angle cost. State representation, reopening rules, discretization, and parent relaxation must also be justified. This paper makes no optimality claim for Algorithm 2. A full moving-state version should be checked against exhaustive search on small state-augmented maps before comparing its runtime or energy to baselines.

Calibration should first fix geometry and safety constraints, then identify energy coefficients, and finally tune decision weights on separate training maps. The tuning objective should condition energy reduction on mission success and minimum clearance. No calibrated ECA-Theta* or H-DWA weight set, or completed training/evaluation split, is reported here.

In H-DWA, the velocity window is centered on the estimated actual body velocity, obtained from encoder feedback, rather than on the commanded velocity, and constrained by the control-period acceleration limits.

For numerical M1–M3, the velocity input is the exact simulated state, not an encoder recording. The previous command is used only in the smoothness score. Physical estimation under slip remains unvalidated.

The proposed H-DWA samples triplets (v_x, v_y, ω) within speed and acceleration limits. Its specification propagates candidate trajectories and rejects collisions or infeasible stopping before scoring. Algorithm 3 describes this intended operation. The numerical specialization executes closed-loop kinematics; the general design score below is specialized as stated in Section 4.6. The proposed score is:

$$J = 4D + 0.15H\psi + 0.02C + 0.03S + w_{\mathrm{E}} \frac{E_{\mathrm{H}}}{E_0}. \quad (11)$$

D is the horizon-end distance to the route-lookahead target divided by 1 m; Hψ is the absolute heading error divided by 1 rad; and C = (1 m)/max(cmin, 0.05 m), using minimum rollout clearance. S is the sum of squared command changes normalized by 1 m/s for translation and 1 rad/s for yaw rate. EH is predicted energy over horizon H, and E0 = PauxH = 36 J at the nominal settings. The fixed energy weights wE = 0 and 0.15 isolate the local energy term; the remaining weights are unchanged.

Algorithm 3. Executed H-DWA numerical command selection with ideal state feedback.

```
INPUT: exact simulated pose/velocity, previous command, map and route.
Select a 0.50 m route-lookahead target and its chassis heading.
Build the acceleration window around the current actual velocity.
Sample 5 values per translation axis and 3 yaw rates, plus target commands.
Clip target commands into the window; reject speed-norm violations.
Roll out each candidate for 2 s, with a ramp over the first interval.
Reject footprint collisions, regional speed-limit and braking violations.
Compute the five-term score in Equation (11), including shared energy.
Choose the finite minimum score; apply only the first control interval.
Update ideal kinematics; record state, command, clearance and energy.
Stop on the declared goal tolerances, timeout or no admissible command.
Repeat from the updated state; no perception or memory update is modeled.
OUTPUT: actual numerical trajectory and per-step decision/energy CSVs.
```

The first rollout interval averages current and candidate velocities, representing a bounded acceleration ramp; subsequent intervals retain the candidate. A braking tail after the horizon checks footprint clearance while component speeds decay to zero at the assumed acceleration limits. The plant uses the same midpoint-heading kinematic integration. These checks are numerical constraints, not verified stopping capabilities of the physical platform.

Section 4.6 specifies the numerical settings. Motor voltage/current limits are checked only in post-processing. Sensing latency, communication delay, disturbances and uncertain braking are excluded; real-time and physical safety performance are not established.

The numerical score uses lookahead tracking without separate preferred-speed or wheel-command-derivative penalties. No lateral-energy term is added beyond the shared equation. Weights are neither physically identified nor optimized over these maps. PauxH is

constant across equal-duration candidates and cannot affect local ranking, although global route timing changes mission auxiliary energy. Terminal energy-to-go and variable-duration horizons are not evaluated.

Hysteresis, firmware breakaway thresholds, and final-pose weights are proposed integration requirements. Their influence on the energy model must be recorded explicitly because boost pulses and stiction are not represented by the translational kinetic-energy term alone. No controller timing, convergence, or safety guarantee is inferred from the pseudocode.

The current ACF-A* navigation implementation has event-driven replanning for live obstacles and traversability costs. Table 8 separates this implemented component from the proposed comparison variants. A future H-DWA integration must preserve these interfaces and log every stop, recovery, and replanning event so that its behavior can be reconstructed.

Table 8. Comparative analysis of planning variants.

Method	Space	Dominant Cost	Memory	Role
Classical A*	grid	distance	no	geometric reference
Theta* + DWA	any-angle + local	distance + obstacles	no	smoothness and reaction
A* + DWA	grid + local	distance + obstacles	no	hierarchical baseline
ACF-A*	grid + memory	clearance + traversability	yes	implemented
ECA-Theta* + H-DWA	holonomic any-angle	energy + risk + mecanum	design: yes; numerical tests: no	restricted kinematic prototype; full integration proposed

The existing navigation implementation uses Python 3.12.14 and ROS 2 Jazzy, with configurable motion limits, obstacle handling, and traversability memory. These platform settings provide the architectural context for the energy model. They do not determine the nominal energy coefficients, which remain uncalibrated assumptions as specified in Table 7.

The numerical analysis directly evaluates Equation (12) for prescribed elementary motions and parameter combinations. The workload definitions, timing rules, coefficient ranges, and sensitivity procedure are given in Section 4.3.

4. Preliminary Analysis of the Parametrized Energy Model

4.1. Scope and Assumptions

The W0–W2 analysis concerns the energy required by specified motion profiles under an assumed set of coefficients. It combines an analytically defined motion duration with the energy expression in Section 3.4. It does not model perception, collision avoidance, actuator dynamics, or the closed-loop interaction between global and local planning. The architectural proposal and the numerical model therefore answer different questions: how navigation could incorporate energy costs, and how those costs respond to prescribed motion and parameter changes.

The W0–W2 calculated outputs are energy components, local sensitivities, parameter-dependent bounds, and route-cost thresholds. Navigation success, achieved clearance, planning runtime, and obstacle-avoidance performance are not outputs of this model. This distinction limits the conclusions to the stated workloads and mathematical assumptions.

A separate restricted ECA-Theta*/H-DWA evaluation on M1–M3 produces collision-checked kinematic trajectories, without sensing, slip or full motor-control simulation. S1–S6 remain conceptual scenarios.

4.2. Navigation Context

Figure 6 and Table 9 illustrate the navigation situations that motivate the proposed architecture. S1 represents an open corridor; S2 introduces turns; S3 involves wall proximity; S4 includes a temporary obstacle; S5 introduces reduced traction and repeated traversal; and S6 combines payload and docking requirements. The routes illustrate possible geometry rather than simulated planner outputs. Payload and docking requirements are additional task conditions, not quantities inferred from the drawings.

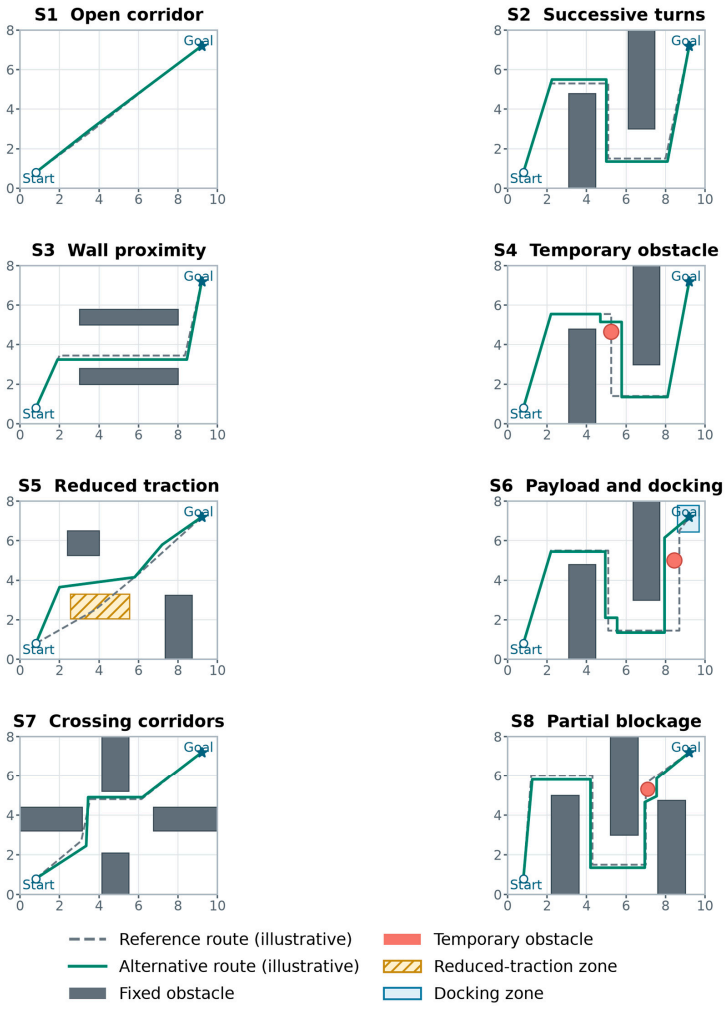

Figure 6. Conceptual navigation scenarios for future evaluation. S1–S6 are the primary design; S7–S8 are optional extensions. Map coordinates and routes are schematic, not outputs from executed algorithms. The reference route precedes adaptation to the illustrated task or surface condition; the alternative route illustrates a possible response (Excel: ScenarioGeometry).

These situations explain why distance alone is insufficient to characterize a mission: turns, waiting, lateral alignment, and surface interaction may change energy demand. The prescribed workloads W0–W2 isolate selected terms of the model. They do not reproduce the maps in Figure 6 or establish the ability of the robot to avoid dynamic obstacles or adapt to reduced traction.

A subsequent navigation comparison could use A* with an explicitly defined tracker, Theta* + DWA, A* + DWA, and the proposed integrated architecture under common maps, footprint, limits, and task conditions. Such a comparison is distinct from the present sensitivity analysis, which remains distinct from the restricted M1–M3 comparison added below.

Table 9. Navigation scenarios proposed for future evaluation. Lengths are illustrative design ranges, not measured or simulated trajectory lengths.

Code	Scenario	Length	Perturbation	Indicators
S1	Open corridor	12–13 m	baseline efficiency	energy, time
S2	Dense turns	13–15 m	repeated direction changes	yaw, energy
S3	Wall proximity	12–13 m	shortcut with low clearance	stops, clearance
S4	Dynamic obstacle	15–16 m	temporary LiDAR obstacle	replanning, success
S5	Reduced traction	13–14 m	stall/slip on wheel tracks	memory, repeatability
S6	Payload and docking	14–15 m	0–10 kg; final orientation	energy, final yaw

4.3. Motion Workloads and Sensitivity Procedure

The reproducible calculation uses three deterministic workloads. W0 is one 12 m forward motion from rest to rest. W1 consists, in order, of 5 m forward, a $\pi/2$ rotation, 1 m lateral, a $\pi/2$ rotation, 5 m forward, a $\pi/2$ rotation, 1 m lateral, and a $\pi/2$ rotation. W2 consists of four 3 m forward motions separated by three 2 s waits. Each translational or rotational primitive starts and ends at rest. These are deliberately prescribed inputs for isolating model terms; they are not alternative routes with a demonstrated common start and goal.

Translation uses a 0.30 m/s peak-speed limit and an assumed 0.30 m/s^2 acceleration magnitude. Rotation uses 0.45 rad/s and an assumed 0.45 rad/s^2. The velocity limits are consistent with the nominal mission settings; the acceleration values define idealized workloads and are not identified motor dynamics. Constant-acceleration ramps and a constant-speed plateau are integrated analytically. For travel magnitude s, speed limit u, and acceleration a, peak speed is $\min(u, \sqrt{(as)})$ and duration is $2u_{peak}/a + \max(0, (s - u_{peak}^2/a)/u_{peak})$. The rule also applies to rotations with angular units.

For a complete workload, let d be total translation, T its duration, Ψ accumulated absolute yaw, Y the integral of absolute body-frame lateral speed, and Q the sum of positive increments in one-half of squared translational speed. The workload energy reduces to Equation (12). Rotational kinetic energy, motor copper loss, breakaway pulses, and braking dissipation beyond the assumed effective coefficients are not resolved.

$$E = \frac{m(g\, c_r\, d + Q)}{\eta_{drive}} + k_{rot}\, \Psi + P_{aux}\, T + (k_{lat} + k_{slip})Y. \quad (12)$$

Table 7 defines all nominal coefficients and bounds. The sensitivity calculation varies six coefficients on five equally spaced levels each, producing $5^6 = 15{,}625$ combinations per workload and 46,875 evaluations in total. At fixed coefficients, payload is varied separately from 0 to 10 kg in 1 kg increments. For each coefficient, an endpoint perturbation is also performed while the other coefficients remain nominal. The six-dimensional coefficient box has $2^6 = 64$ corners per workload; all 192 corner cases are evaluated.

For a coefficient with lower bound a and upper bound b, the five levels are a, $a + (b - a)/4$, $a + (b - a)/2$, $a + 3(b - a)/4$, and b. The order of enumeration does not affect the results. The one-factor analysis contains 36 endpoint evaluations, the payload sweep contains 33 evaluations, and the analytical derivative check covers seven parameters for each workload, giving 21 checks. These counts refer to mathematical evaluations, not repeated robot missions.

The positive model terms are monotonic in c_r, P_aux, k_rot, k_lat, and k_slip, and decrease with η_drive. Therefore, the corner extrema bound the entire stated continuous coefficient box for a fixed workload. The grid verifies the interior calculations but does not define probability distributions. The bounds are epistemic stress tests chosen around the nominal assumptions: ±50% for five coefficients and 0.50–0.90 for efficiency. They are neither empirical uncertainty intervals nor calibrated material-property limits.

Local sensitivity is expressed by the partial derivative of energy with respect to a parameter. Its dimensionless elasticity is the derivative multiplied by the nominal parameter and divided by nominal energy. A small relative parameter change therefore produces approximately the elasticity times that relative change in energy. Derivatives are checked using centered finite differences with a perturbation of 10^{-5} times the larger of the parameter magnitudes and 0.01 in its stated units. The comparison uses a relative tolerance of 10^{-7} and an absolute tolerance of 10^{-6} in the corresponding derivative units.

The calculation uses deterministic parameter levels rather than random sampling. For each workload, the nominal parameter vector is evaluated first. The Cartesian product of the six five-level coefficient sets is then evaluated while holding the workload fixed. Separate calculations perturb one coefficient to each endpoint, evaluate all endpoint combinations, and increase payload by integer kilograms at nominal coefficients. Each evaluation uses the same additive energy expression, without stochastic noise or fitted corrections.

Arithmetic checks give 180 J for a stationary 10 s interval at 18 W auxiliary power and 41 s for a 12 m rest-to-rest translation with the stated speed and acceleration limits. The grid and corner calculations give identical extrema. Replacing the lateral and slip-proxy coefficients of 8 and 3 J/m by 9 and 2 J/m also leaves every workload energy unchanged, confirming that only their sum is identifiable for these motions.

4.4. Scope of Subsequent Navigation Validation

Testing the proposed architecture requires closed-loop navigation under controlled geometry, payload, surface, and initial-pose conditions. Relevant outcomes include task completion, collision-free clearance, duration, and energy over the same mission boundary. Comparisons should use common operating conditions and separately assess the effects of the global planner, local controller, and traversability memory.

The role of traversability memory can be examined by comparing the same system with and without that component while holding the task and operating conditions constant. Such component comparisons concern the executed architecture, whereas W0–W2 perturbations concern the equation alone and M1–M3 exclude memory updates.

Physical validation also depends on repeatable initial conditions. Total mass and load position, battery state of charge, roller condition, sensor alignment, and motor temperature can alter the relation between commanded motion and energy use. These quantities should be controlled or recorded when calibrating the model and comparing navigation strategies.

4.5. Electrical Calibration and Interpretation

Electrical calibration requires a defined battery-side measurement boundary and synchronized voltage and current. Separate motor and logic measurements help identify auxiliary power. The acquisition range must cover startup peaks without saturation, while offset, gain, timing drift, filtering, and internal averaging affect the integrated energy. A 100 Hz output may describe slow power variation but does not resolve 20 kHz PWM switching. Sensor bandwidth and averaging therefore need to be assessed independently of the exported sample rate.

Voltage, current, and robot motion must refer to the same time interval. Synchronization errors are particularly important near acceleration and stopping events, where power can change rapidly. The energy integral in Equation (13) provides the measurement-based quantity against which the parametrized model would be calibrated; it is not the source of the present numerical results.

For sampled voltage and current, Equation (13) defines electrical energy; energy per unit distance is Eelectric/d. Equation (14) defines the relative difference between a baseline and a candidate method. A positive reduction has an operational meaning only for compara-

ble completed tasks, with consistent system boundaries and acceptable clearance. These definitions provide the link between the present energy model and subsequent measurements.

$$E_{\text{electric}} = \sum_{k=1}^{N} U_{\text{k}}\, I_{\text{k}}\, \Delta t_{\text{k}}. \tag{13}$$

$$\Delta E_{\%} = 100 \frac{E_{\text{baseline}} - E_{\text{candidate}}}{E_{\text{baseline}}}. \tag{14}$$

The deterministic sensitivity calculations have no sample standard deviation or statistical confidence interval. Their ranges describe the effects of parameter assumptions. For repeated physical measurements, variability between missions and instrument uncertainty would need to be assessed separately. Paired comparisons under common operating conditions can then distinguish a systematic energy difference from environmental variation.

Measurement uncertainty includes calibration offsets and gains, integration error, timing mismatch, and correlated sampling errors. Independent contributions can be combined by root-sum-square propagation, whereas shared errors require covariance terms and do not disappear with more samples. The coefficient-box bounds in this article describe model sensitivity, not electrical measurement uncertainty.

4.6. Computational Scope and Reproducibility of the Preliminary Analysis

The numerical analysis reported in the following section is based on deterministic evaluation of the energy model defined in Equation (12), using the prescribed workloads and parameter intervals specified above. No stochastic sampling is employed; therefore, random seeds are not applicable to the present analysis. For each workload, the nominal parameter set is evaluated first, followed by the one-factor endpoint perturbations, the five-level full-factorial coefficient evaluation, the corner evaluation of the parameter domain, and the payload sweep described in this section.

The resulting 46,875 factorial evaluations and 192 corner evaluations are mathematical evaluations of the parametrized energy expression rather than repeated robot missions or navigation simulations. Consequently, differences between the calculated energy values quantify the behavior and sensitivity of the stated model under the prescribed assumptions; they do not constitute experimentally measured energy savings.

The W0–W2 results do not depend on planner execution. Separate research code executes a restricted ECA-Theta* search and H-DWA kinematic controller, outside ROS 2 and firmware. Three variants per map are compared: geometric Theta*-type route with local energy weight wE = 0; ECA route with wE = 0; and ECA route with wE = 0.15. The geometric comparator fixes start heading; ECA considers four headings, but all selected routes in these maps retain the same zero heading. These nine runs do not test the complete architecture or memory.

Reproducibility of the reported calculations is based on the explicit specification of the workload definitions, nominal parameter values, parameter intervals, evaluation levels, and energy equations provided in this section. Because the evaluation is deterministic and contains no fitted stochastic component, identical inputs to the stated model produce identical numerical outputs. Experimental identification of the coefficients and validation of the complete navigation architecture remain separate subsequent stages.

The grid spacing is 0.25 m; headings are 0, $\pi/2$, π and $3\pi/2$. A 0.30 m circular collision envelope includes the nominal chassis. Translation and yaw limits are 0.30 m/s and 0.45 rad/s; component acceleration limits are 0.30 m/s^2 and 0.45 rad/s^2. Global speed profiles use 0.05 m sampling; collision lines use 0.025 m sampling plus half-step clearance padding. Improved heading labels are reopened, equal-cost ties favor fewer transitions, and the expansion limit is 15,000.

Coordinates are in meters and headings in radians. M1 is an empty 6 × 5 m map with start (1,1,0) and goal (1,4,0), testing lateral motion without yaw. M2 is 8 × 5 m, with start (0.75,2.75,0), goal (7.25,2.75,0), an obstacle [3,1.5,4.75,3], and a prescribed slow rectangle [2.75,3.25,5,4.25] limited to 0.08 m/s. Rectangle entries are [xmin,ymin,xmax,ymax]. M3 is 8 × 5 m, with start (0.75,2.5,0), goal (7.25,2.5,0), and staggered obstacles [2,0,2.75,2.5] and [4.75,2.5,5.5,5]. Map boundaries are also obstacles. The slow zone is known a priori; it is neither measured friction nor a learned memory patch.

H-DWA uses a 2 s horizon, a 0.20 s control interval and a 0.50 m lookahead. Each axis includes five translational or three yaw samples and the target-directed command clipped to the acceleration window. The score in Equation (11) uses fixed coefficients 4, 0.15, 0.02 and 0.03 for normalized target distance, heading error, inverse clearance and command changes, plus wE EH/36 J. Completion requires position error <0.08 m, heading error <0.10 rad and velocity norm <0.035 in the stated unit scales; timeout is 180 s. No admissible command is recorded as failure. No sensor noise or random sampling is added; seed is null.

Global edges use rest-to-rest speed profiles, including the known regional limit. H-DWA tracks the route continuously and may smooth or retime it, so planned edge energy and integrated local trajectory energy are different outputs. Candidate rollout includes an initial acceleration ramp and a braking-tail collision check. Perfect kinematics do not establish physical stopping performance. The reported M1–M3 results were generated using the implemented planning and control algorithms. The Supplementary Materials contains the resulting numerical data, recorded trajectories, simulation configurations, and supporting energy calculations.

Ten checks pass; a second process reproduces the nominal summaries except wall-clock time. Further runs halve the control interval and global profile step.

Matched Navigation Protocol

On the unchanged M1-M3 maps, we compared Basic Theta* with geometric DWA (Theta/DWA-G), eight-connected A* with geometric DWA (A*/DWA-G), and Basic Theta* with energy-aware DWA (Theta/DWA-E). Basic Theta* uses Euclidean edge costs and parent-based line-of-sight relaxation [21]. A* uses the same grid and Euclidean neighbor costs, without any-angle parent relaxation. Both use the inflated circular footprint of ECA-Theta*. The original clearance-penalized geometric planner remains a separate internal baseline, distinct from Basic Theta*.

ECA-Theta* retains its energy cost, soft-clearance penalty and four chassis-heading states. The geometric references use Euclidean costs and a fixed chassis-heading label. The comparison therefore evaluates the whole global stage, rather than isolating a single energy coefficient.

DWA-G adapts the ROS-style objective in [40] to holonomic motion, following the dynamic-window principles in [28]. Its score is independent of H-DWA. DWA-E follows the four-term structure of Equation (20) in [40], replacing the calibrated drivetrain model with our parametrized model. Both references share the terminal chassis-heading criterion and exclude the H-DWA command-change term. DWA-E is thus inspired by Xie et al., rather than an exact reproduction.

For each admissible candidate, the minimized reference-controller score is

$$J = w_p d_{path} + w_g d_{goal} + \frac{w_c}{max(c,\ 0.05\ \mathrm{m})} + 0.15\, e_{heading} + w_e \frac{E_H - P_{aux} H}{P_{aux} H}$$

Here, d_{path} is the endpoint distance to the global polyline, d_{goal} is the endpoint distance to the local target, c is the minimum predicted footprint clearance, $e_{heading}$ is the chassis-heading error in radians, and E_H is predicted energy over H = 2 s. At nominal P_{aux} = 18 W, the energy normalization is 36 J. Fixed-horizon auxiliary energy is candidate-independent;

subtracting it does not change the ranking. Distance and inverse-distance terms retain their declared units, so their weights are scale-dependent, not universal constants.

All controllers share the geometry, motion limits, horizon, lookahead, velocity sampling, collision, braking, slow-zone and terminal conditions specified above. The inherited stopping test uses the norm of the three numerical command components, with threshold 0.035; this mixed-unit convention is preserved for comparison, not interpreted as a physical speed norm.

We selected the reference-controller settings on three separate maps, C1–C3. The geometric controller used four path/clearance weight combinations; the energy-aware controller used twelve, including three positive energy weights. Selection favored completed missions first, then lower mean calculated energy. The selected weights were fixed before M1–M3 and were unchanged for the 0.10 s check. H-DWA retained its submitted parameters. The selected weights (w_p, w_g, w_c, w_e) are $(0.5, 4, 0.02, 0)$ for DWA-G and $(0.5, 4, 0.05, 0.5)$ for DWA-E.

Eight configurations on three maps at control intervals of 0.20 s and 0.10 s give 48 deterministic runs. ECA/DWA-G and ECA/DWA-E keep the reference local controller fixed to assess global route selection. Internal G/H-DWA(0) and ECA/H-DWA(0) retain the original component comparisons. Forty-eight calibration runs are provided separately. No random noise or sampling is used, so the seed is null. Identical reruns check reproducibility; they are not independent statistical samples.

The slow-zone check was corrected to test each complete predicted and braking segment, rather than sampled positions alone. All methods use this correction; maps, robot parameters and energy coefficients are unchanged. Independent integration of the saved trajectories reproduces the energy totals. Dense checks of all 48 comparison trajectories found no obstacle penetration, zone-speed violation or acceleration-limit violation under the numerical model.

The supplementary AMR_Matched_Benchmarks.xlsx workbook links raw trajectories (Trace) to run totals (Runs), percentage comparisons (Comparison) and independent checks (Audit).

4.7. Motor-Model Structural Comparison

To test omitted electrical losses, identical motion histories are replayed through an effective geared-DC-motor model. The project specification identifies four Pololu 24 V, 30:1 gearmotors, item 4692 [62]. Catalog values are 330 rpm and 0.10 A at no load, and extrapolated stall quantities are 3 A and 15 kg·cm. They yield $R = 24/3 = 8\ \Omega$, $Kt = 15 \times 0.0980665/(3 - 0.1) = 0.507241$ N·m/A and $Ke = (24 - 8 \times 0.1)/(330 \times 2\pi/60) = 0.671344$ V·s/rad, expressed as output-shaft equivalents. These are not identified robot constants; catalog-based motor calculations are approximate [63].

With inverse-kinematic matrix A from Equation (1), $\Omega = A\nu$. Let $f = [m(\tilde{v}_x - \omega v_\gamma), m(\tilde{v}_\gamma + \omega v_x), Jz\dot{\omega}]^T$ and $Jz = m(L^2 + W^2)/12$, a uniform-body assumption. Minimum-norm inertial torque allocation plus equal rolling resistance gives:

$$\tau = A(AA)^{-1}f + \frac{cmgr}{4}\,\mathrm{sign}(\Omega). \tag{15}$$

For each wheel, torque and no-load current determine current; back electromotive force and resistance determine terminal voltage:

$$I_i = \frac{\tau_i}{K_t} + I_0\,\mathrm{sign}(\Omega_i), \quad U_i = K_e\Omega_i + RI_i. \tag{16}$$

The alternative total replaces the base rolling/acceleration efficiency terms and yaw penalty, rather than adding a complete motor model to them. It retains auxiliary demand and the uncalibrated lateral proxy:

$$E_{\text{motor}} = P_{\text{aux}}T + \int \sum_{i=1}^{4} \max(0, U_i I_i)dt + (k_{\text{lat}} + k_{\text{slip}})Y. \quad (17)$$

Copper loss $\int \Sigma R I_i^2 dt$ is recorded diagnostically and is already included in UI; it is not added twice. W0–W2 profiles use 0.02 s sampling with ramp-switch instants inserted. Twenty-seven combinations of R = 6/8/10 Ω, Kt = 0.8/1/1.2 times nominal and I0 = 0.05/0.10/0.20 A are evaluated per workload, keeping Ke fixed. These 81 cases are assumed stress ranges, not manufacturer tolerances. Current and voltage checks against 3 A and 24 V are not thermal certification. Inductance, converter behavior, roller contact, battery dynamics and calibration remain unresolved.

5. Results of the Preliminary Model Analysis

The results presented in this section quantify the deterministic behavior and sensitivity of the parametrized energy model under the prescribed workloads and assumptions; they do not represent measured robot energy consumption or physical navigation performance. Sections 5.6 and 5.7 report separate numerical extensions.

The results quantify the response of the parametrized energy model to the workloads defined in Section 4.3. Tables 10–12 provide the nominal features, component contributions, and bounds used to interpret the plots. These are calculated energy relationships, not measured savings or performance estimates for a navigation algorithm.

5.1. Workload Features and Nominal Energy

Table 10 gives the features obtained by analytical integration of the declared rest-to-rest primitives. Each workload travels 12 m, but the prescribed motion sequences differ. W1 includes four quarter-turns and 2 m of lateral travel. W2 includes three waiting intervals and four accelerations from rest. W0–W2 are diagnostic workloads; they do not constitute a route-planning benchmark or a common-goal algorithm comparison.

Table 10. Deterministically calculated features of prescribed workloads. Q is the positive specific kinetic-energy increment.

Workload	d [m]	T [s]	Ψ [rad]	Y [m]	Q [m²/s²]
W0	12.0	41.000	0.000	0.0	0.045
W1	12.0	61.963	6.283	2.0	0.180
W2	12.0	50.000	0.000	0.0	0.180

Only W1 has nonzero yaw and lateral travel, with $\Psi = 2\pi$ rad and Y = 2 m; therefore, $\Psi = \pi Y$ across W0–W2. Their separate calculated contributions are not independently identifiable from workload totals. Rotation-only and lateral-only calibration workloads are needed to separate yaw from combined lateral cost. Independent measurements are still required to separate lateral resistance and slip.

Writing E = Xβ with feature rows (d,Q,Ψ,T,Y,Y) gives rank(X) = 3 and nullity of 3 for six linear coefficients. Both identical lateral columns and $\Psi = \pi Y$ prevent separate identification. Compensating perturbations satisfy:

$$\pi \Delta k_{\text{rot}} + \Delta k_{\text{lat}} + \Delta k_{\text{slip}} = 0. \quad (18)$$

Figure 7 and Table 11 show the exact component sums at the nominal coefficients. For W0, rolling contributes 99.03 J, acceleration 1.08 J, and auxiliary operation 738.00 J, totaling

838.11 J. W1 totals 1251.99 J and W2 totals 1003.36 J. Rounding in the displayed tables is for readability, not an estimate of measurement precision.

The prescribed W0 profile comprises 1 s of acceleration, 39 s at 0.30 m/s, and 1 s of deceleration; these durations are analytical inputs, not observed robot motion. The total duration is therefore 41 s, with one positive specific kinetic-energy increment of 0.045 m^2/s^2. Substitution in Equation (12) gives rolling energy $16.83 \times 9.80665 \times 0.035 \times 12/0.70 = 99.03$ J, acceleration energy $16.83 \times 0.045/0.70 = 1.08$ J, and auxiliary energy $18 \times 41 = 738.00$ J. Rotation and lateral terms are zero. This worked calculation illustrates how the tabulated inputs determine the total.

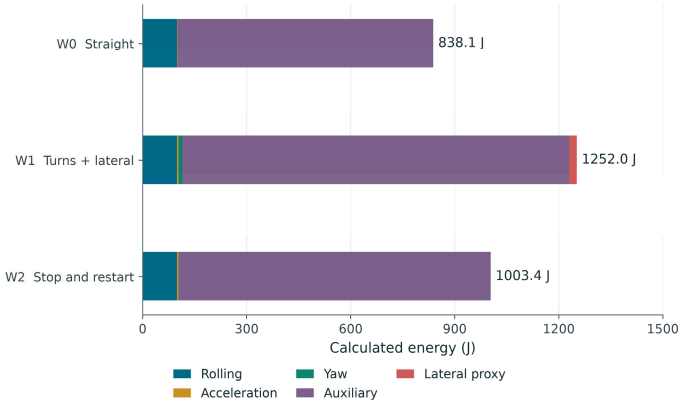

Figure 7. Calculated energy components for W0–W2 at the assumed nominal coefficients. Workload labels do not denote algorithms. There are no measured samples or statistical error bars (Python: plot_workloads.py; Excel: Results).

Table 11. Component-wise calculation results in joules. Lateral is the combined proxy (k_lat + k_slip)Y.

Workload	Rolling	Accel.	Yaw	Auxiliary	Lateral	Total
W0	99.03	1.08	0.00	738.00	0.00	838.11
W1	99.03	4.33	11.31	1115.33	22.00	1251.99
W2	99.03	4.33	0.00	900.00	0.00	1003.36

Auxiliary operation represents 88.1% of W0, 89.1% of W1, and 89.7% of W2. This dominance follows from the assumed 18 W baseline and the prescribed durations. It is not a measured power breakdown. W2 exceeds W0 by 165.25 J: 162.00 J comes from nine additional seconds, and 3.25 J comes from three additional translational accelerations. This exact decomposition illustrates the model mechanism without attributing it to a controller.

5.2. Coefficient Sensitivity and Global Bounds

Figure 8 varies one coefficient at a time for W1. Changing P_aux from 18 W to 9 or 27 W changes calculated energy by −44.54% or +44.54%. The ±50% rolling-coefficient range produces ±3.95%. Efficiency changes from 0.70 to 0.50 and 0.90 produce +3.30% and −1.83%, respectively. The yaw, lateral, and slip-proxy ranges each produce changes below 0.65% under this particular workload and nominal weighting.

The analytical elasticities support this ranking. For W1, S_Paux = 0.8908, S_cr = 0.0791, S_eta = −0.0826, and S_krot = 0.0090. The lateral elasticities are 0.0128 and 0.0048. A local 1% coefficient change therefore has an approximately proportional percentage effect on model energy. These values are workload-specific; a small yaw coefficient does not establish that turning is physically inexpensive because turn duration also changes auxiliary energy.

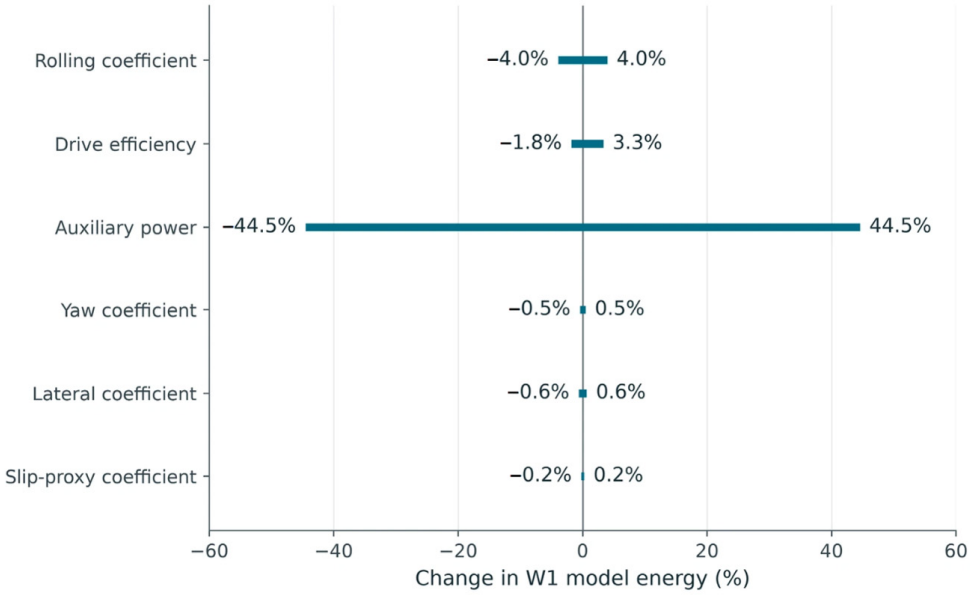

Figure 8. One-factor endpoint sensitivity of W1. The horizontal spans connect deterministic coefficient perturbations and are not confidence intervals. The efficiency range is 0.50–0.90; the other ranges are ±50% around their nominal values (Python: plot_workloads.py; Excel: Sensitivity).

Figure 9 shows the joint dependence on auxiliary power and efficiency. Table 12 reports the extrema obtained by evaluating every corner of the six-parameter box. The 46,875-point calculation grid agrees with these corner extrema. Since all terms have known monotonic signs for a fixed workload, the bounds also hold between grid points within the declared box.

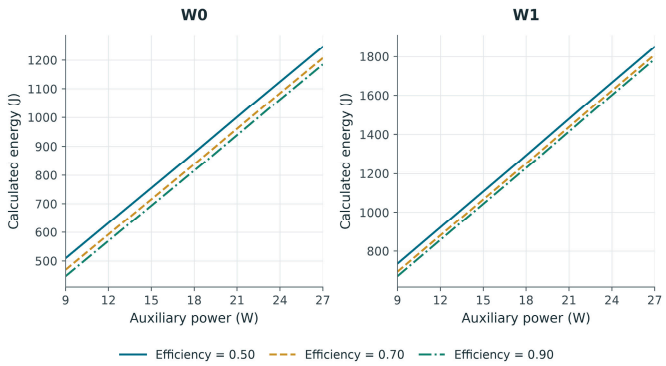

Figure 9. Calculated energy for W0 and W1 over auxiliary-power and efficiency assumptions. The other coefficients remain nominal. Lines are evaluations of Equation (12), not fits to robot measurements (Python: plot_workloads.py; Excel: FigureSamples).

Table 12. Exact coefficient-box bounds for each fixed workload. These are conditional model envelopes, not probabilistic uncertainty intervals.

Workload	Nominal [J]	Minimum [J]	Maximum [J]
W0	838.11	408.35	1316.47
W1	1251.99	616.20	1936.97
W2	1003.36	491.88	1564.02

The broad envelopes show why uncalibrated absolute joule values must be treated cautiously. Combining separate workload minima and maxima would also be an invalid

paired comparison: two routes on the same robot share the same coefficient vector. A defensible ranking requires comparing both routes under each common vector and then checking whether the sign of the difference is preserved. Even a robust fixed-route ranking would not establish that a planner generates the preferred route.

Coefficient sensitivity does not test omitted motor losses. Its rankings remain conditional on the model structure. Sections 4.7 and 5.7 examine this issue through a motor-model comparison, without claiming electrical calibration.

5.3. Payload and Parameter Identifiability

Figure 10 varies only the assumed mass while holding motions and other coefficients fixed. Added mass increases W0 energy by 5.95 J/kg and W1/W2 energy by 6.14 J/kg. Adding 10 kg therefore adds 59.48 J to W0 and 61.41 J to W1/W2. The linear response is a consequence of the model, not experimental confirmation that a loaded robot preserves the same speed, slip, current, or efficiency.

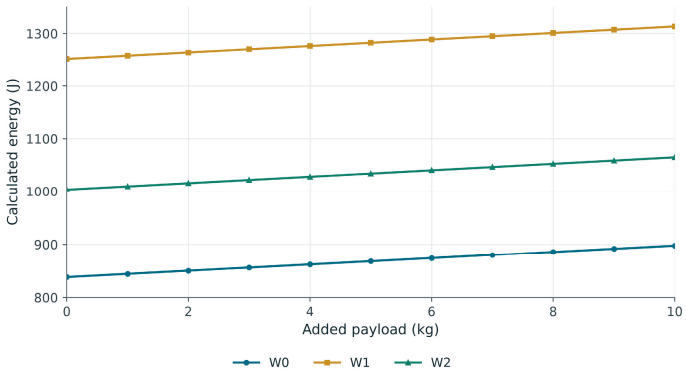

Figure 10. Conditional payload sweep from 0 to 10 kg above the 16.83 kg reference mass. Motion profiles and all other coefficients are held fixed. The curves are not measured payload tests (Python: plot_workloads.py; Excel: Payload).

Two calibration ambiguities are exposed. First, k_lat and k_slip multiply the same lateral-distance feature for these motions. Any change that leaves their sum fixed leaves every result unchanged. Second, steady straight runs identify the effective ratio c_r/η_drive rather than both coefficients separately. Distinct acceleration workloads can help separate effective terms within the simplified model, but independent efficiency evidence and checks for omitted motor losses remain necessary.

A practical calibration sequence should measure total mass, then auxiliary power with the relevant computing load, and then steady longitudinal motion, controlled acceleration, in-place rotations, and lateral motion. Multiple speeds and payloads are needed to test whether constant coefficients are adequate. A separate ground-relative motion reference is required before fitting a physical slip-loss term. The rank and conditioning of the calibration design matrix should be reported; fitting two identical columns produces non-unique coefficients even if residual error is small.

5.4. Conditional Route-Cost Threshold

For two already specified feasible trajectories A and B, the model difference is $\Delta E = m(g\, c_r\, \Delta d + \Delta Q)/\eta_drive + k_rot\, \Delta\Psi + P_aux\, \Delta T + (k_lat + k_slip)\Delta Y$. Route B is lower in modeled energy only when this expression is negative. This relationship provides a testable decision condition, without asserting which trajectory the proposed algorithms select.

Table 13 compares the executed M2 global-search profiles from Section 4.6. Both connect the same start and goal at zero heading; their rest-state profiles differ from the continuously tracked trajectories in Section 5.6.

Table 13. Calculated M2 global route profiles; the geometric and energy searches use the same obstacle map.

Parameter	Route A	Route B
Travel distance (m)	6.777	7.570
Travel time (s)	49.475	28.238
Accumulated rotation (rad)	0.000	0.000
Lateral displacement (m)	1.500	3.500
Estimated energy (J)	968.106	612.498
Route selection criterion	Distance plus clearance	Normalized energy plus clearance

Figure 11 isolates a 1 m detour with unchanged acceleration, yaw, and lateral features. Its additional rolling cost must be offset by a reduction in duration. At the nominal 16.83 kg mass and 18 W auxiliary power, the necessary time saving exceeds 0.458 s. At 26.83 kg, it exceeds 0.731 s. This time refers to the net total-duration reduction after accounting for the extra travel; it cannot be assumed merely because a route is smoother.

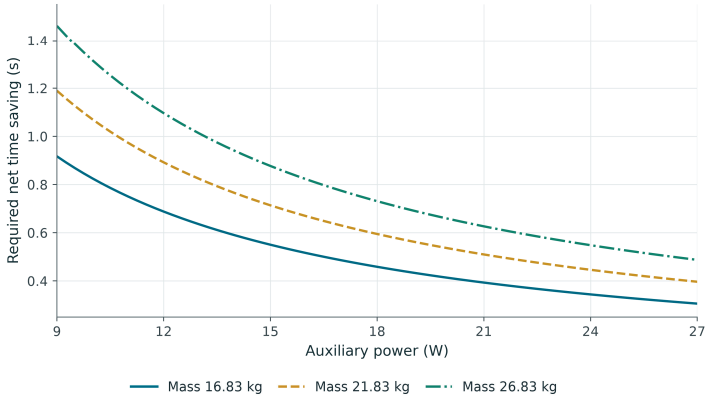

Figure 11. Calculated break-even time saving for an additional 1 m of travel, with all other motion features fixed. The threshold is m g c_r/(η_drive P_aux). It is a conditional cost test, not an observed detour benefit (Python: plot_workloads.py; Excel: FigureSamples).

For arbitrary trajectories, the yaw, acceleration, and lateral differences may reverse this conclusion. The same calculation must be repeated for every plausible coefficient vector and for the actual logged motion. Feasibility, clearance, stopping distance, and task completion remain independent constraints and cannot be inferred from a negative energy difference.

5.5. Sensitivity Across the Three Workloads

Table 14 compares dimensionless local elasticities at the nominal coefficients. For example, a 1% increase in auxiliary power produces an approximately 0.881% increase in W0 energy when the motion profile is fixed. Zero yaw and lateral elasticities in W0 and W2 arise because these workloads contain neither rotation nor lateral travel. They do not imply that those effects are absent from other robot missions.

Table 14. Dimensionless local energy elasticities at the nominal assumptions. Motion features remain fixed.

Parameter	W0	W1	W2
Mass	0.1194	0.0826	0.1030
Rolling coefficient	0.1182	0.0791	0.0987
Drive efficiency	−0.1194	−0.0826	−0.1030
Auxiliary power	0.8806	0.8908	0.8970
Yaw coefficient	0.0000	0.0090	0.0000
Lateral coefficient	0.0000	0.0128	0.0000
Slip-proxy coefficient	0.0000	0.0048	0.0000

Auxiliary-power elasticity remains between 0.881 and 0.897 across the three workloads. The mass and efficiency elasticities have equal magnitudes and opposite signs because the same rolling and translational-acceleration terms scale with mass divided by efficiency. Their contribution is proportionally smaller in W1, where additional rotation time and lateral movement increase other terms. The ranking is conditional on the chosen coefficients and profiles, but it identifies which calibration errors most affect these particular predictions.

5.6. Restricted ECA-Theta and H-DWA Numerical Results*

Figure 12 shows actual global routes on M2 and M3, not the conceptual S1–S6 maps. On M1, the global path translates 3 m laterally with zero yaw. On M2, the energy-aware route bypasses the known slow zone above: planned distance increases from 6.777 to 7.570 m, but planned duration falls from 49.475 to 28.238 s and energy from 968.1 to 612.5 J. The known speed constraint is not learned friction or a traversability-memory observation.

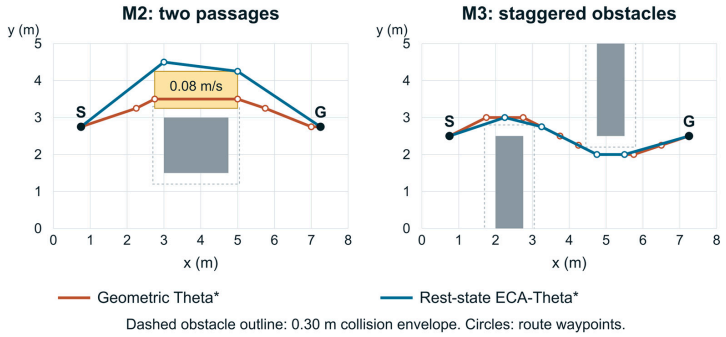

Figure 12. Executed global-search outputs. Gray marks obstacles; yellow denotes the prescribed 0.08 m/s region. Waypoints and collision-envelope outlines are shown. These numerical maps are distinct from conceptual scenarios S1–S6 (Python: figures.py; Excel: Routes, Config).

Table 15 and Figure 13 integrate the actual numerical trajectories. Changing only the global M2 route lowers energy from 1186.7 to 708.9 J; adding the local energy score gives 677.5 J, a further 4.42% change. The combined reduction is 42.91%. M1 is unchanged. M3 instead rises from 653.6 to 662.4 J, an increase of 1.36%, despite a shorter executed path. The nine successful terminations under ideal state feedback are not a statistical physical success rate.

Table 15. Nine nominal kinematic runs; G denotes geometric global search, E restricted ECA-Theta*. All reach the stated goal tolerances. Clearance denotes the minimum free gap beyond the 0.30 m circular collision envelope, accounting for obstacles and map boundaries.

Map/Global	wE	Time (s)	Distance (m)	Energy (J)	Clearance (m)
M1/G	0	14.2	2.925	312.6	0.700
M1/E	0	14.2	2.925	312.6	0.700
M1/E	0.15	14.2	2.925	312.6	0.700
M2/G	0	61.4	6.766	1186.7	0.150
M2/E	0	33.6	7.491	708.9	0.299
M2/E	0.15	31.8	7.436	677.5	0.303
M3/G	0	31.2	6.872	653.6	0.069
M3/E	0	31.8	6.774	655.0	0.043
M3/E	0.15	32.4	6.780	662.4	0.075

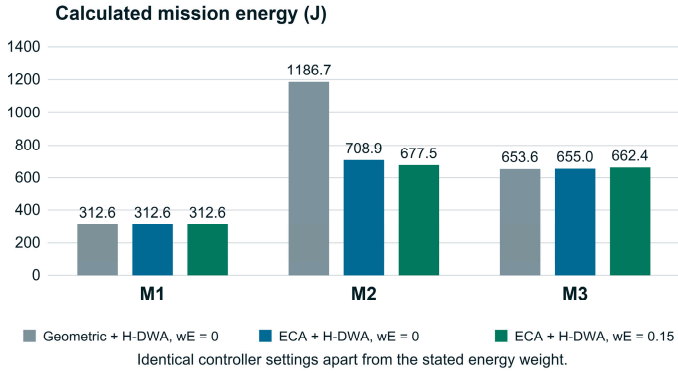

Figure 13. Calculated energy from the nine closed-loop kinematic histories. M1 shows no improvement and M3 favors the geometric baseline. Bars are deterministic calculations without statistical error bars (Python: plot_controller_energy.py; Excel: AMR_Matched_Benchmarks, Runs).

As a control-discretization sensitivity check, reducing the H-DWA control interval from 0.20 s to 0.10 s changes the M2 baseline/combined energies from 1186.7/677.5 J to 1027.3/688.6 J, respectively. The corresponding energy reduction decreases from 42.91% to 32.97%, while M3 changes to 611.1/630.8 J. This change affects both the command-update frequency and the acceleration window, and therefore should not be interpreted as a numerical convergence test.

Halving the global profile step to 0.025 m preserves all six routes, with a maximum energy change of about 1.2%. A nine-case global sweep of auxiliary power 9/18/27 W and slow-zone limit 0.08/0.15/0.30 m/s changes the route preference. At 0.08 m/s, the 7.570 m bypass remains; at 0.30 m/s, a route near 6.73–6.74 m is selected. These results establish conditional computational behavior, not optimality or measured savings.

5.7. Motor-Model Comparison

Using the identical W0–W2 histories, the catalog-informed replacement gives 981.1, 1458.3 and 1154.4 J, with auxiliary shares of 75.2%, 76.5% and 78.0%. Figure 14 shows the 27 assumed parameter combinations per workload. The ranges are 878.5–1243.3 J, 1316.0–1831.7 J and 1045.4–1434.2 J; auxiliary fractions span 59.4–84.0%, 60.9–84.8% and 62.8–86.1%. Thus, the original 88–90% contribution is not invariant to model structure.

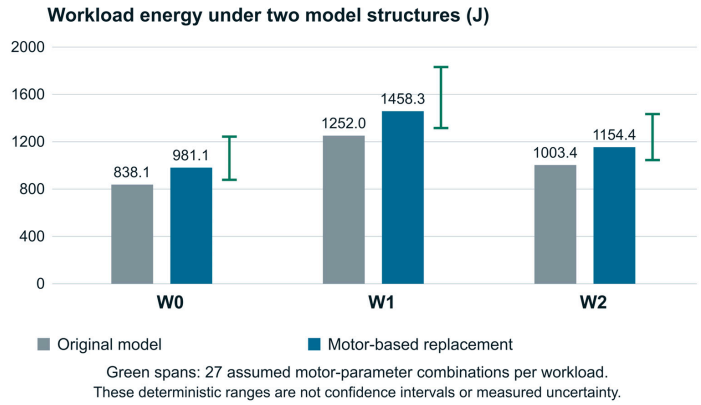

Figure 14. Original and motor-replacement workload models. Green spans are deterministic ranges over 27 assumed motor-parameter combinations, not measured uncertainty or confidence intervals (Python: figures.py; Excel: MotorModel, Robustness).

The nominal workload replay reaches about 0.420 A and 6.684 V per motor; nominal navigation traces reach 0.582 A and 8.314 V. These algebraic limits do not demonstrate continuous thermal feasibility. Replaying the same controller histories preserves the map-dependent ranking: M2 baseline/combined energy is 1350.4/766.3 J, and M3 is 742.4/758.8 J. The planners are not rerun under this alternative objective; this is a fixed-history structural check, not optimization or physical validation.

5.8. Matched Navigation Results

Table 16 presents the matched-condition navigation results for the evaluated methods.The largest improvement occurs in M2. The proposed method requires 677.5 J, compared with 1235.8 J for Theta/DWA-G, 1203.5 J for A*/DWA-G and 1163.6 J for Theta/DWA-E. The corresponding reductions are 45.18%, 43.71% and 41.78%. Although the route is 10.72% longer than that of Theta/DWA-G, mission time falls from 63.8 to 31.8 s. Auxiliary energy falls by 576.0 J, while the other modeled energy terms increase by 17.7 J, giving a net reduction of 558.3 J. The main benefit is therefore the shorter time spent completing the mission after avoiding the known slow zone. These results do not show an improvement in physical traction efficiency.

Table 16. Navigation results at a 0.20 s control interval. DWA-G is the geometric holonomic reference; DWA-E is the Xie-inspired adaptation using the common energy model. Clearance is the distance from the robot center to an obstacle or map boundary, minus the 0.30 m footprint radius. All twelve runs reached the numerical goal.

Map	Method	Energy J	Distance m	Time s	Clearance m
M1	Theta/DWA-G	316.2	2.925	14.4	0.700
M1	A*/DWA-G	316.2	2.925	14.4	0.700
M1	Theta/DWA-E	316.2	2.925	14.4	0.700
M1	ECA/H-DWA	312.6	2.925	14.2	0.700
M2	Theta/DWA-G	1235.8	6.716	63.8	0.036
M2	A*/DWA-G	1203.5	6.894	62.4	0.150
M2	Theta/DWA-E	1163.6	6.689	60.4	0.110
M2	ECA/H-DWA	677.5	7.436	31.8	0.303
M3	Theta/DWA-G	615.2	6.717	29.6	0.061
M3	A*/DWA-G	672.9	6.974	32.4	0.081
M3	Theta/DWA-E	615.1	6.754	29.6	0.091
M3	ECA/H-DWA	662.4	6.780	32.4	0.075

In M1, the proposed method uses 312.6 J, compared with 316.2 J for each reference at 0.20 s. The 1.14% difference corresponds to one control interval of auxiliary consumption; the executed distance is the same. At 0.10 s, all methods require approximately 309.2 J and this advantage disappears. M1 therefore provides no evidence of a robust energy benefit.

M3 shows the opposite result. The proposed method requires 662.4 J, compared with 615.2 J for Theta/DWA-G and 615.1 J for Theta/DWA-E, increases of 7.67% and 7.69%. It still uses 1.55% less energy than A*/DWA-G, which requires 672.9 J. The proposed method takes 32.4 s, whereas both Theta-based references take 29.6 s. Its clearance of 0.075 m lies between their values of 0.061 m and 0.091 m. The higher energy use therefore cannot be explained as a consistent gain in clearance.

Global and Local Contributions and Control-Interval Sensitivity

Table 17 summarizes the comparative energy results for the evaluated planner–controller configurations and control intervals.Keeping the local controller fixed helps identify how much of the M2 benefit comes from global planning. Replacing Basic Theta* with ECA-Theta* reduces energy from 1235.8 to 709.2 J with DWA-G, and from 1163.6 to 688.7 J with DWA-E. Most of the saving is therefore retained without the proposed local score. At 0.20 s, the full proposed configuration uses 4.47% less energy than ECA/DWA-G and 1.63% less than ECA/DWA-E. At 0.10 s, the ordering changes: ECA/DWA-E requires 675.8 J, compared with 688.6 J for the proposed configuration. The proposed local controller does not consistently outperform the energy-aware reference.

At a control interval of 0.10 s, the M2 energy reductions are 33.61%, 34.41% and 33.44% relative to Theta/DWA-G, A*/DWA-G and Theta/DWA-E. In M3, the proposed method still uses more energy than the two Theta-based references, by 2.72% and 2.57%. Thus, the M2 advantage remains when the interval is shortened, but its size changes appreciably. Testing two intervals shows this sensitivity; it does not establish numerical convergence.

Table 17. Relative reduction in calculated energy of ECA-Theta*/H-DWA against each matched reference. Positive values indicate lower energy; negative values indicate higher energy. The two control intervals test sensitivity, not numerical convergence.

Map	Reference	Reduction at 0.20 s	Reduction at 0.10 s
M1	Theta/DWA-G	1.14%	−0.01%
M1	A*/DWA-G	1.14%	−0.01%
M1	Theta/DWA-E	1.14%	−0.02%
M2	Theta/DWA-G	45.18%	33.61%
M2	A*/DWA-G	43.71%	34.41%
M2	Theta/DWA-E	41.78%	33.44%
M3	Theta/DWA-G	−7.67%	−2.72%
M3	A*/DWA-G	1.55%	2.08%
M3	Theta/DWA-E	−7.69%	−2.57%

In the internal component comparison, the added H-DWA energy term gives a 4.42% benefit at 0.20 s but a 0.38% penalty at 0.10 s. This also points to global route selection, rather than a consistently better local controller, as the main source of the M2 improvement.

All comparisons use the same five energy terms: rolling resistance, positive kinetic-energy increments, yaw cost, auxiliary power multiplied by duration, and a lateral/slip proxy. The sum of the nonauxiliary terms should not be interpreted as measured traction energy, since the coefficients have not been identified from robot power measurements. These deterministic runs are not assigned confidence intervals or significance tests. All eight configurations completed all three maps at both intervals, but this count does not estimate the probability of success in deployment.

6. Discussion

Despite its preliminary and model-based character, the present analysis provides several results that are relevant to the subsequent development and validation of energy-aware navigation for mecanum-wheeled robots. First, it identifies which model parameters have the strongest influence on the calculated energy demand under the prescribed motion workloads, thereby indicating which quantities require priority during experimental calibration. Second, the analysis shows that auxiliary power can dominate the calculated energy budget under the considered operating conditions, highlighting the importance of mission duration in addition to geometric path length when energy-aware planning objectives are formulated. Third, the analysis exposes structural parameter-identifiability limitations, particularly for the lateral-motion terms, and therefore prevents unsupported interpretation of individual coefficients before experimental calibration. Finally, the deterministic factorial and corner evaluations establish the range of model responses that can arise solely from the stated parameter assumptions. These results provide a quantitative basis for designing subsequent experiments and for deciding which parameters and measurements should receive priority during physical validation.

The analysis supports a limited but useful conclusion: under the declared model assumptions, mission duration and auxiliary power dominate the prescribed low-speed workloads. Consequently, electrical calibration of the computer, sensors, converters, and communication load is at least as important as tuning a geometric route cost. The result is an implication of an explicit model and should be tested against synchronized power measurements before guiding deployment.

The identified lateral-term redundancy is a structural issue rather than a small parameter uncertainty. Increasing the number of trials with the same collinear features cannot distinguish k_lat from k_slip. Similarly, an encoder-based low-progress event does not uniquely diagnose ground slip. A stronger model needs independent motion evidence and calibration motions that excite distinct loss mechanisms.

The proposed architecture specifies how temporary wheel-track experience could influence global and local decisions. The model analysis complements this design by identifying dominant energy terms and calibration ambiguities. The matched M1–M3 simulations further evaluate the route-selection outcomes of the proposed planner against reference navigation methods under common numerical conditions, while experimental validation of the complete integrated architecture remains a subject for future work.

The restricted M1–M3 runs exclude sensor uncertainty, drivetrain control and memory updates. Mixed outcomes and time-step sensitivity preclude universal superiority. The motor replay tests another model structure, not ground truth; a controlled memory ablation remains outstanding.

Physical validation remains necessary to establish predictive accuracy and measurable navigation-energy savings.

Published energy-aware navigation studies [38–42] use different platforms, operating conditions and energy boundaries. Their reductions cannot be ranked against the present W0–W2 totals or map-specific calculations. A direct comparison requires equivalent tasks on the same platform and measurement boundary.

For quantitative published context, Zhang et al. [42] report model-prediction accuracy above 95% following physical verification, whereas Rizqi et al. [10] report 3.41% lower energy and 26.07% higher throughput for a traffic-priority policy in a different fleet simulation. These metrics have different denominators and boundaries from W0–W2 joules or the M2 numerical percentage; they are not a cross-platform ranking.

Sustainability implications are conditional. If a future controlled comparison establishes a positive measured saving ΔE_meas in joules per mission, annual grid-energy savings

would be N_day × D_year × ΔE_meas/(3.6 × 10^6 × η_charge) kWh, under a fixed utilization and charging-efficiency assumption. Neither the present workload totals nor their differences supply ΔE_meas for the proposed planner. Battery lifetime, wear, emissions, and fleet availability require additional evidence [55].

Sensitivity to coefficient assumptions is not a substitute for validating the model structure. The model omits speed-dependent motor and gearbox efficiency, stiction and boost pulses, thermal effects, roller deformation, and regeneration. Payload can change feasible motion, friction, and slip rather than only the mass term. The fixed-workload bounds therefore apply only inside the stated mathematical model and should not be interpreted as bounds on the real robot's energy.

Future work should connect the proposed planners to the robot's navigation loop and calibrate the energy model using independent electrical measurements. Evaluation under varied payloads, surfaces, and obstacle conditions can then establish whether the predicted cost relationships remain useful when perception, actuation, and replanning influence the executed motion.

Interpretation of the Matched Comparison

The simulations show a benefit when the global planner can avoid a short route that would require prolonged travel through a known slow zone. Energy-aware local control alone may not recover this benefit if it follows a geometric route through that zone. In M2, the saving persists when ECA-Theta* is combined with either reference local controller. However, the smaller and sometimes reversed differences between local controllers, together with the M3 results, do not support a general advantage for H-DWA. The contribution demonstrated here is the benefit of coordinating global and local energy costs under these conditions. Energy-aware DWA and hybrid navigation are established methods, and these tests do not establish a universally optimal energy route.

The energy-aware local reference uses the objective structure of Xie et al. [40], adapted to our energy model and numerical constraints. It provides a comparison within the same simulator, rather than a reproduction of their physical experiments. The predictive method in [47] is discussed but not reimplemented. The M2 results establish an advantage over the references tested here, without establishing superiority over all current methods. Percentages obtained with other robots or different definitions of total energy cannot be ranked directly against these results.

The practical lesson from M2 is that a shorter route can consume more energy when it requires sustained low-speed travel. This is relevant to facilities where permitted or feasible speeds vary between known areas. In our simulation, both the zone and its speed limit are specified in advance. Floor friction, wheel-slip dynamics, online detection and updates to the traversability memory are not evaluated. Unevenness and low adhesion are different physical conditions; a prescribed speed limit does not validate operation on either type of surface. Most of the calculated benefit comes from completing the mission sooner and reducing auxiliary energy, rather than improving traction efficiency. The same mechanism may benefit other drive types, so the result cannot yet be regarded as specific to mecanum wheels. M1 shows no robust gain, and the two Theta-based references perform better in M3. Together, these results indicate where the proposed method helps and where it does not.

The simulations show that the implemented algorithms run and allow their energy estimates to be compared under common conditions. They do not validate the model coefficients, measured energy savings, real-world safety, temporary-memory effectiveness or real-time operation on the robot. Testing those aspects requires synchronized measurements of battery power and robot pose, including tests with different payloads, floor conditions,

sensing errors and actuator dynamics. The effect of the control interval also needs further study before the saving can be described as numerically converged.

7. Conclusions

This study presents an energy-aware navigation architecture for a mecanum-wheeled robot and a preliminary analysis of its parametrized energy model. The architecture combines implemented ACF-A* and temporary traversability memory with proposed ECA-Theta* and H-DWA extensions. The numerical contribution includes the original energy-model analysis and a restricted global–local kinematic evaluation, not physical validation of the complete system.

The study connects energy-model sensitivity with numerical planning decisions and identifies priorities for experimental calibration. Its principal findings concern the influence of model parameters, the conditional effects of global and local planning choices, and the dependence of energy estimates on motor-model structure. The analysis identifies the parameters that dominate the calculated energy response, reveals limitations in parameter identifiability, and distinguishes model-level energy predictions from experimentally measurable performance. In this sense, the proposed framework provides both an architectural basis for subsequent implementation of energy-aware planning and a quantitative basis for prioritizing model calibration and physical energy measurements.

The deterministic energy-model analysis gives 838.1, 1252.0 and 1003.4 J for W0–W2, with auxiliary shares of 88–90%. Varying auxiliary power from 9 to 27 W changes W1 energy by ±44.5%. The lateral-motion and slip-proxy coefficients are not separately identifiable. These findings guide calibration, not experimentally validated savings. Restricted ECA-Theta*/H-DWA matches the geometric baseline on M1, reduces M2 energy by 42.91% at a 0.20 s control interval, and increases M3 energy by 1.36%. Reducing the control interval to 0.10 s changes the M2 reduction to 32.97%, demonstrating control-discretization sensitivity rather than numerical convergence. For M2, most of the calculated energy reduction results from global route selection that avoids the predefined slow-speed region; the local energy term provides an additional 4.42% reduction at 0.20. The M1–M3 experiments do not evaluate online traversability-memory updates and therefore do not establish an energy benefit for the complete architecture.

With the same corrected kinematic protocol applied to every method, the proposed configuration reduces M2 modeled energy by 45.18% relative to Theta/DWA-G and 41.78% relative to the Xie-inspired Theta/DWA-E adaptation at 0.20 s. At 0.10 s, these reductions are 33.61% and 33.44%. M1 shows no robust advantage, and both Theta-based references perform better in M3. The results support energy-aware global route selection when a known slow zone makes a geometric route costly in time. They do not establish superiority across all conditions or measured savings on the robot.

The motor replacement gives 981.1, 1458.3 and 1154.4 J and auxiliary shares of 75.2–78.0%. All results depend on uncalibrated coefficients and numerical settings; no physical experiments were performed.

The study provides an explicit connection between motion composition, assumed energy coefficients, and calculated mission cost. Its main practical value is to guide calibration and identify when a geometric route improvement could or could not reduce modeled energy. Physical energy savings and integrated navigation performance remain subjects for subsequent validation.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/app162311699/s1. AMR_Energy_Calculations.xlsx, containing energy-model calculations, sensitivity analyses, numerical navigation results, and supporting data;

and AMR_Matched_Benchmarks.xlsx, containing matched-condition navigation benchmark results, simulation parameters, trajectory records, and supporting calculations.

Author Contributions: Conceptualization, C.-A.P. and D.-M.I.; methodology, C.-A.P., C.E.C. and D.-M.I.; validation, C.E.C., C.-A.P. and A.-M.I.; formal analysis, D.-M.I., C.-A.P. and E.-M.P.; resources, C.-A.P., D.-M.I., A.-M.I. and C.E.C.; writing—original draft preparation, C.-A.P., D.-M.I., A.-M.I. and E.-M.P.; writing—review and editing, D.-M.I., C.-A.P., A.-M.I. and E.-M.P. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was funded by the National University of Science and Technology POLITEHNICA Bucuresti. The authors acknowledge the support of the PubArt Program from the National University of Science and Technology POLITEHNICA Bucuresti.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The numerical results and supporting calculation data are provided in the Supplementary Materials. Additional research data may be made available by the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACF-A*	**Adaptive Clearance and Feedback A***
AGV	Automated Guided Vehicle
AMR	Autonomous Mobile Robot
DWA	Dynamic Window Approach
ECA-Theta	Energy- and Clearance-Aware Theta*
H-DWA	Holonomic Energy-Aware Dynamic Window Approach
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
ROS 2	Robot Operating System 2
SLAM	Simultaneous Localization and Mapping
TTL	Time to Live

References

1. Fragapane, G.; de Koster, R.; Sgarbossa, F.; Strandhagen, J.O. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *Eur. J. Oper. Res.* **2021**, *294*, 405–426. [CrossRef]
2. Lackner, T.; Hermann, J.; Kuhn, C.; Palm, D. Review of Autonomous Mobile Robots in Intralogistics: State-of-the-Art, Limitations and Research Gaps. *Procedia CIRP* **2024**, *130*, 930–935. [CrossRef]
3. Barros, I.R.C.; Nascimento, T.P. Robotic Mobile Fulfillment Systems: A survey on recent developments and research opportunities. *Robot. Auton. Syst.* **2021**, *137*, 103729. [CrossRef]
4. Bernardo, R.; Sousa, J.M.C.; Gonçalves, P.J.S. Survey on robotic systems for internal logistics. *J. Manuf. Syst.* **2022**, *65*, 339–350. [CrossRef]
5. Pugliese, G.; Chou, X.; Loske, D.; Klumpp, M.; Montemanni, R. AMR-Assisted Order Picking: Models for Picker-to-Parts Systems in a Two-Blocks Warehouse. *Algorithms* **2022**, *15*, 413. [CrossRef]
6. Cai, J.; Li, X.; Liang, Y.; Ouyang, S. Collaborative Optimization of Storage Location Assignment and Path Planning in Robotic Mobile Fulfillment Systems. *Sustainability* **2021**, *13*, 5644. [CrossRef]
7. *ISO 3691-4:2023*; Industrial Trucks—Safety Requirements and Verification—Part 4: Driverless Industrial Trucks and Their Systems. ISO: Geneva, Switzerland, 2023.
8. Kubasakova, I.; Kubanova, J.; Benco, D.; Kadlecová, D. Implementation of Automated Guided Vehicles for the Automation of Selected Processes and Elimination of Collisions between Handling Equipment and Humans in the Warehouse. *Sensors* **2024**, *24*, 1029. [CrossRef]
9. Chen, W.; Wu, P.; Gong, Y.; Zhang, Z.; Wang, K. The Role of Energy Consumption in Robotic Mobile Fulfillment Systems: Performance Evaluation and Operating Policies with Dynamic Priority. *Omega* **2025**, *130*, 103168. [CrossRef]

10. Rizqi, Z.U.; Chou, S.Y.; Oscar, A.D. Towards Energy-Efficient Robotic Mobile Fulfillment System: Hybrid Agent-Based Simulation with DEA-Based Surrogate Machine Learning. *Appl. Soft Comput.* **2025**, *176*, 113141. [CrossRef]
11. Alatise, M.B.; Hancke, G.P. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access* **2020**, *8*, 39830–39846. [CrossRef]
12. Macenski, S.; Foote, T.; Gerkey, B.; Lalancette, C.; Woodall, W. Robot Operating System 2: Design, architecture, and uses in the wild. *Sci. Robot.* **2022**, *7*, eabm6074. [CrossRef]
13. Macenski, S.; Moore, T.; Lu, D.V.; Merzlyakov, A.; Ferguson, M. From the desks of ROS maintainers: A survey of modern and capable mobile robotics algorithms in ROS 2. *Robot. Auton. Syst.* **2023**, *168*, 104493. [CrossRef]
14. Macenski, S.; Jambrecic, I. SLAM Toolbox: SLAM for the dynamic world. *J. Open Source Softw.* **2021**, *6*, 2783. [CrossRef]
15. Gatesichapakorn, S.; Takamatsu, J.; Ruchanurucks, M. ROS based Autonomous Mobile Robot Navigation using 2D LiDAR and RGB-D Camera. In Proceedings of the ICA-SYMP 2019, Bangkok, Thailand, 16–18 January 2019. [CrossRef]
16. Stefanini, M.; Luperto, M.; Basilico, N.; Amigoni, F. Safe Map Updating for Long-Term Operations of Mobile Robots in Dynamic Environments. *Sensors* **2023**, *23*, 6066. [CrossRef]
17. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
18. Dijkstra, E.W. A Note on Two Problems in Connexion with Graphs. *Numer. Math.* **1959**, *1*, 269–271. [CrossRef]
19. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
20. Koenig, S.; Likhachev, M. D* Lite. In Proceedings of the AAAI-02, Edmonton, AB, Canada, 28 July–1 August 2002; pp. 476–483.
21. Nash, A.; Daniel, K.; Koenig, S.; Felner, A. Theta*: Any-Angle Path Planning on Grids. In Proceedings of the AAAI-07, Vancouver, BC, Canada, 22–26 July 2007; pp. 1177–1183.
22. Kavraki, L.E.; Švestka, P.; Latombe, J.-C.; Overmars, M.H. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. *IEEE Trans. Robot. Autom.* **1996**, *12*, 566–580. [CrossRef]
23. LaValle, S.M. *Rapidly-Exploring Random Trees: A New Tool for Path Planning*; Technical Report 98-11; Iowa State University: Ames, IA, USA, 1998.
24. Xiang, D.; Lin, H.; Ouyang, J.; Huang, D. Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot. *Sci. Rep.* **2022**, *12*, 13273. [CrossRef]
25. Lin, S.; Liu, A.; Wang, J.; Kong, X. A Review of Path-Planning Approaches for Multiple Mobile Robots. *Machines* **2022**, *10*, 773. [CrossRef]
26. Tang, Y.; German, Y.O. Enhanced Q-Learning for Grid Navigation: Direction-Sensitive Priors, Annealed Exploration, and Potential-Based Reward Optimization. *Inform. Autom.* **2026**, *25*, 1235–1259. [CrossRef]
27. Iskandar, A.; Hammoud, A.; Kovács, B. Implicit Understanding: Decoding Swarm Behaviors in Robots through Deep Inverse Reinforcement Learning. *Inform. Autom.* **2024**, *23*, 1485–1504. [CrossRef]
28. Fox, D.; Burgard, W.; Thrun, S. The Dynamic Window Approach to Collision Avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [CrossRef]
29. Brock, O.; Khatib, O. High-Speed Navigation Using the Global Dynamic Window Approach. In Proceedings of the IEEE ICRA, Detroit, MI, USA, 10–15 May 1999; pp. 341–346.
30. Ögren, P.; Leonard, N.E. A Convergent Dynamic Window Approach to Obstacle Avoidance. *IEEE Trans. Robot.* **2005**, *21*, 188–195. [CrossRef]
31. Li, Y.; Zhao, J.; Chen, Z.; Xiong, G.; Liu, S. A Robot Path Planning Method Based on Improved Genetic Algorithm and Improved Dynamic Window Approach. *Sustainability* **2023**, *15*, 4656. [CrossRef]
32. Hou, J.; Jiang, W.; Luo, Z.; Yang, L.; Hu, X.; Guo, B. Dynamic Path Planning for Mobile Robots by Integrating Improved Sparrow Search Algorithm and Dynamic Window Approach. *Actuators* **2024**, *13*, 24. [CrossRef]
33. Liu, Y.; Wang, C.; Wu, H.; Wei, Y. Mobile Robot Path Planning Based on Kinematically Constrained A-Star Algorithm and DWA Fusion Algorithm. *Mathematics* **2023**, *11*, 4552. [CrossRef]
34. Hahn, B. Enhancing Obstacle Avoidance in Dynamic Window Approach via Dynamic Obstacle Behavior Prediction. *Actuators* **2025**, *14*, 207. [CrossRef]
35. Mei, Y.; Lu, Y.-H.; Hu, Y.C.; Lee, C.S.G. Energy-Efficient Motion Planning for Mobile Robots. In Proceedings of the IEEE ICRA, New Orleans, LA, USA, 26 April–1 May 2004; pp. 4344–4349.
36. Mei, Y.; Lu, Y.-H.; Hu, Y.C.; Lee, C.S.G. A Case Study of Mobile Robot's Energy Consumption and Conservation Techniques. In Proceedings of the ICAR, Seattle, WA, USA, 18–20 July 2005; pp. 492–497.
37. Liu, S.; Sun, D. Optimal Motion Planning of a Mobile Robot with Minimum Energy Consumption. *IEEE/ASME Trans. Mechatron.* **2014**, *19*, 492–503. [CrossRef]
38. Henkel, C.; Bubeck, A.; Xu, W. Energy Efficient Dynamic Window Approach for Local Path Planning in Mobile Service Robotics. *IFAC-PapersOnLine* **2016**, *49*, 32–37. [CrossRef]

39. Maidana, R.; Granada, R.; Jurak, D.; Magnaguagno, M.; Meneguzzi, F.; Amory, A. Energy-Aware Path Planning for Autonomous Mobile Robot Navigation. In Proceedings of the FLAIRS-33, North Miami Beach, FL, USA, 17–20 May 2020; pp. 362–367.
40. Xie, L.; Henkel, C.; Stol, K.; Xu, W. Power-Minimization and Energy-Reduction Autonomous Navigation of an Omnidirectional Mecanum Robot via the Dynamic Window Approach Local Trajectory Planning. *Int. J. Adv. Robot. Syst.* **2018**, *15*. [CrossRef]
41. Xie, L.; Herberger, W.; Xu, W.; Stol, K.A. Experimental Validation of Energy Consumption Model for the Four-Wheeled Omnidirectional Mecanum Robots for Energy-Optimal Motion Control. In Proceedings of the IEEE 14th International Workshop on Advanced Motion Control, Auckland, New Zealand, 22–24 April 2016; pp. 565–572. [CrossRef]
42. Zhang, L.; Kim, J.; Sun, J. Energy Modeling and Experimental Validation of Four-Wheel Mecanum Mobile Robots for Energy-Optimal Motion Control. *Symmetry* **2019**, *11*, 1372. [CrossRef]
43. Wu, M.; Yeong, C.F.; Su, E.L.M.; Holderbaum, W.; Yang, C. A Review on Energy Efficiency in Autonomous Mobile Robots. *Robot. Intell. Autom.* **2023**, *43*, 648–668. [CrossRef]
44. Mohammadpour, M.; Zeghmi, L.; Kelouwani, S.; Gaudreau, M.-A.; Amamou, A.; Graba, M. An Investigation into the Energy-Efficient Motion of Autonomous Wheeled Mobile Robots. *Energies* **2021**, *14*, 3517. [CrossRef]
45. Yao, M.; Deng, H.; Feng, X.; Li, P.; Li, Y.; Liu, H. Improved Dynamic Windows Approach Based on Energy Consumption Management and Fuzzy Logic Control for Local Path Planning of Mobile Robots. *Comput. Ind. Eng.* **2024**, *187*, 109767. [CrossRef]
46. Raj, R.; Kos, A. A Comprehensive Study of Mobile Robot Navigation Techniques. *Appl. Sci.* **2023**, *13*, 9667. [CrossRef]
47. Teso-Fz-Betoño, D.; Aramendia, I.; Ramos-Hernanz, J.A.; Caballero-Martin, D.; Affou, H.; Lopez-Guede, J.M. Optimization of Energy Efficiency with a Predictive Dynamic Window Approach for Mobile Robot Navigation. *Sustainability* **2025**, *17*, 4526. [CrossRef]
48. Zakharov, D.N.; Panin, A.D.; Iaremenko, A.M.; Aliev, D.R.; Borisov, O.I. Quasi-Optimal Control of a Reconfigurable Mecanum-Wheeled Robot for Energy-Efficient and Accurate Trajectory Tracking. *IFAC-PapersOnLine* **2025**, *59*, 110–115. [CrossRef]
49. Nghiem, G.-K.; Le, T.-L.; Phung, T.C.; Dinh, Q.H. A Study on the Optimal Control Strategy Using Sliding Mode Controller for Mecanum-Wheeled Omnidirectional Mobile Robot. *Measurement* **2025**, *255*, 118113. [CrossRef]
50. Maroșan, I.-A.; Gîrjob, C.-E.; Crenganiș, M.; Biriș, C.-M. Trajectory Simulation and Energy Consumption Evaluation for Mobile Robots Equipped with Mecanum Wheels Using MATLAB. *Acta Tech. Napoc. Appl. Math. Mech. Eng.* **2025**, *68*, 673–684.
51. Mokhtari, M.; Vanderborght, B.; Famaey, J. Energy Harvesting Aware Path Planning for Ambiently-Powered Multi-Robot Systems. *Robot. Auton. Syst.* **2026**, *197*, 105260. [CrossRef]
52. Galati, R.; Mantriota, G. Motion Planning and Control of an Omnidirectional Mobile Robot Based on Nonlinear Model Predictive Control. *Robotics* **2023**, *12*, 78. [CrossRef]
53. Patle, B.K.; Babu, L.G.; Pandey, A.; Parhi, D.R.K.; Jagadeesh, A. A Review: On Path Planning Strategies for Navigation of Mobile Robot. *Def. Technol.* **2019**, *15*, 582–606. [CrossRef]
54. Grigore, L.Ș.; Oncioiu, I.; Priescu, I.; Joița, D. Development and Evaluation of the Traction Characteristics of a Crawler EOD Robot. *Appl. Sci.* **2021**, *11*, 3757. [CrossRef]
55. Grigore, L.S.; Priescu, I.; Joita, D.; Oncioiu, I. The Integration of Collaborative Robot Systems and Their Environmental Impacts. *Processes* **2020**, *8*, 494. [CrossRef]
56. Nwankwo, L.; Fritze, C.; Bartsch, K.; Rueckert, E. ROMR: A ROS-Based Open-Source Mobile Robot. *HardwareX* **2023**, *14*, e00426. [CrossRef]
57. Ilie, D.-M.; Popescu, C.-A.; Popescu, E.-M.; Cotet, C.E. Design and development of an autonomous mobile platform for optimizing logistics processes. *Proc. Manuf. Syst.* **2026**, *21*, 99–109. [CrossRef]
58. Corke, P. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, 2nd ed.; Springer: Cham, Switzerland, 2017. [CrossRef]
59. Muir, P.F.; Neuman, C.P. Kinematic Modeling of Wheeled Mobile Robots. *J. Robot. Syst.* **1987**, *4*, 281–340. [CrossRef]
60. Campion, G.; Bastin, G.; D'Andrea-Novel, B. Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots. *IEEE Trans. Robot. Autom.* **1996**, *12*, 47–62. [CrossRef]
61. Ilon, B.E. Wheels for a Course Stable Selfpropelling Vehicle Movable in Any Desired Direction on the Ground or Some Other Base. U.S. Patent 3,876,255, 8 April 1975.
62. Pololu. 30:1 Metal Gearmotor 37D × 68L mm, 24 V, with 64 CPR Encoder (Helical Pinion), Item 4692. Specifications. Available online: https://www.pololu.com/product/4692/specs (accessed on 9 November 2026).
63. Schmid, W. Motor Data and Simulation. Maxon Support, Updated 29 August 2024. Available online: https://support.maxon-group.com/hc/en-us/articles/360013761160-Motor-data-and-simulation (accessed on 9 November 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.