Low-power Energy Management Module for Ambiently Powered Robotic Swarms

Mohmmadsadegh Mokhtari*, Parham Haji Ali Mohammadi†, Dragan Subotic*, Ritesh Kumar Singh*,
Bram Vanderborght†, Maarten Weyn*, Jeroen Famaey*
*University of Antwerp – imec, Antwerp, Belgium
†Vrije Universiteit Brussel – imec, Brussels, Belgium

***Abstract*—This paper presents an Energy Management Module (EMM) designed for low-power ambiently powered robotic swarms that harvest energy from the environment in addition to the conventional charging methods. The EMM is designed to cooperate with a scheduling architecture that coordinates tasks across the Ambiently Powered Swarm Robots (APSRs). This module helps the scheduler to have more flexible control over the robot's energy storage and consumption.**

Key functionalities of the proposed EMM include real-time monitoring of energy level, low-power schedulable idle mode, monitoring of efficient energy harvesting mechanisms, dead-battery robot recovery, and providing hard reset ability of the robots. The module provides an extra low-power wireless connection to the robots. In collaboration with the EMM, the scheduler manages energy consumption across the swarm and prevents individual robots from exhausting their energy reserves in idle mode, dead battery mode, and malfunctioning.

To validate the effectiveness of the EMM, experiments are conducted in both simulated and real-world environments, demonstrating improvements in mission endurance, task completion rates, and overall swarm performance. Results indicate that the EMM effectively extends mission duration and enhances operational efficiency compared to baseline approaches.

***Index Terms*—Energy Management Module, Ambiently Powered Robot Swarm, Energy-Aware Scheduling, Robot Recovery**

I. Introduction

Optimization of energy consumption in a swarm of robots can reduce operating costs and improve performance. Additionally, batteries are the main energy storage technology for untethered robots [1], and the energy consumption optimization not only improves the robot's long-duration autonomy but also enhances the overall sustainability of the robot by increasing its battery lifespan.

Looking deeply into the energy optimization of robots opens up two main branches: component-based energy optimization and time-base energy optimization [2]. In component-related efficiency, researchers look into the main energy consumer efficiency of the robot, including mechanical mechanisms, motor drives, control systems, and computational units. On the other hand, there are other factors like efficient routing, navigation, scheduling, and swarm planning [3], which are time-related efficiency factors and optimize the time that the swarm uses a specific robot. Indeed time-related objective factors are important since optimizing the time that a swarm uses a specific robot causes energy consumption reduction and in some cases task accomplishment efficiency.

Generally, optimizing a robot's energy consumption involves several key components that are crucial for maximizing efficiency and prolonging the operational lifespan. These can be broadly categorized into three main aspects: active time, idle time, and charging time [4].

Ambiently Powered Swarm Robots (APSRs) are currently gaining traction [5] due to their potential applications in various fields such as mobile sensing, search and rescue operations, agriculture, and surveillance. APSRs are swarm robotic systems designed to harvest energy from ambient sources such as solar energy.

In this paper, an Energy Management Module (EMM) based on an ultra-lower power microcontroller unit (MCU) is proposed that uses a Bluetooth Low Energy (BLE) connection to maintain perpetual connectivity between the central controller and the APSRs, even when the available energy is severely constrained. This enables an energy-aware scheduler to flexibly manage the swarm remotely and brings about advantages for swarm robot management applications. The first advantage of this method is managing energy consumption in idle time. This plug-in module wakes up the robot when necessary and at the right time and enables the energy-aware scheduler to manage the robot's functionality. The second benefit of the EMM is enabling the scheduler to recover APSRs with dead batteries using a low-power extra connection that the EMM provides. Additionally, the scheduler can calculate the recovery time and recovery energy needed to recover a robot with a dead battery and schedule the tasks more flexibly. The third advantage of this method is to be able to hard reset the robot during malfunctions using the electronic switches which increases the reliability of the robots [6]. To this end, this paper provides a practical research effort to provide a module that enables low-power communication, wake-up, and monitoring of energy harvesting that enables the robot's recovery and facilitates its integration and collaboration with the swarm.

II. Related Work

Optimizing the energy consumption of swarm robotics has gained significant attraction due to the futuristic view of this field. Paryanto *et al.* [7] investigated the power loss points of robots from the hardware layer of mechanical components towards the software layers of planning and management. They stated that existing methods for reducing the energy consumption in robots are energy-efficient motion planning, optimizing operating parameters, and scheduling. This work emphasizes that reducing operation time and idle time is an important method for optimizing energy consumption. Nilakantan *et al.* [8] researched minimizing cycle time and total energy consumption in robotic assembly line systems. This study specifically highlighted standby energy consumption as one of the most significant points of energy loss. Li *et al.* [6] investigated minimizing energy consumption and cycle time.

To overcome the idle mode energy management, Benedikt *et al.* [9] presented a novel scheduling approach that aims to minimize the energy consumption of a machine during its idle periods. In the scheduling domain, it calls the idle management model of the machine a ”common” way by defining a set of machine modes, e.g., “on”, “off” and “standby”. They argue that this model might be too restrictive for some types of machines (e.g. furnaces). For such machines, they propose to employ the complete time-domain dynamics and integrate it into an idle energy function. Since normally swarm robots are not that complicated we considered a ”common” model to manage idle mode. Abikarram *et al.* [10] presented energy cost minimization for unrelated parallel machine scheduling which considered this ”common” idle mode scheduling.

The novelty of our work lies in using a low-power module as a redundant communication system, which enables the scheduler and other monitoring units to have reliable additional access to ambiently powered swarm robots. This system is specifically designed for swarm robot applications in hazardous, inaccessible, or hard-to-access use cases. Unlike other works, our research focuses on ambiently powered robots, reducing reliance on mains-powered charging infrastructure.

III. System Description

A. Introduction

The EMM has important functions that help optimize the operation of swarm robotic systems by monitoring energy consumption, facilitating dead-battery recovery, and ensuring efficient task scheduling.

In this regard, the proposed system shown in Figure 1 is presented. This system consists of three main components:

- The **EMM**, mounted directly on the robot, serves as the primary interface for the scheduler. It includes low-power communication modules for data transmission, energy monitoring, and switches to handle different power states. The EMM supports lower-power communication like BLE and allows the scheduler to wake up and power off the robot during different states.
- The **scheduler** is an external system responsible for planning and managing the robot’s tasks and charging schedule, optimizing the robot’s energy consumption by controlling its operational modes. The proposed scheduler is aware of the robot’s energy model and using the EMM it can manage the robot’s energy consumption efficiently.

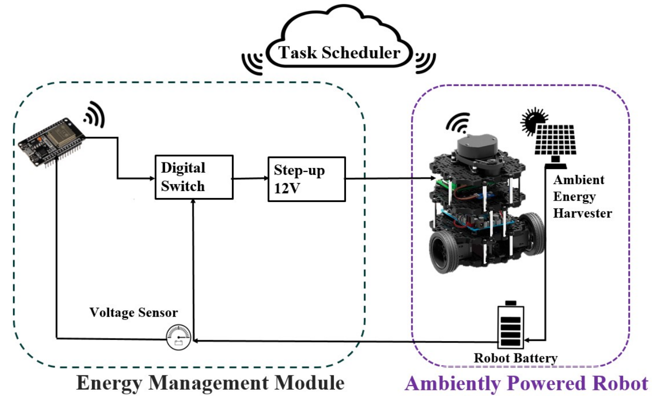

Fig. 1: This figure illustrates the implementation model of an energy management module integrated with an ambiently powered robot and a scheduler unit.

- The **ambiently powered robot**, is equipped with various sensors and actuators. It harvests ambient energy. The robot communicates to the scheduler using its own wireless interface (e.g., Wi-Fi based) during “Standstill” and “Active” modes.

The robot’s five operational modes can be summarized as follows:

- **Active Mode**: The robot is fully operational and performs tasks as scheduled.
- **Standstill Mode**: The robot is in a standstill state, conserving energy while allowing it to transition back to Active or Off mode when necessary.
- **Off Mode**: The robot is fully powered down, and the EMM enters standby mode. In this state, the scheduler can still monitor the robot via the EMM. The scheduler can activate the EMM to turn the robot into Standstill mode.
- **Dead Battery Mode**: The robot battery is drained and is not able to switch from Off mode into Standstill mode.
- **Malfunction Mode**: The robot is unable to continue in active mode due to software or hardware problems and needs a hard reset.

The idle period is when the robot has no assigned task for a specific time. In this period robots can be in either ”Off”, ”Standstill” or a combination of these two states. Figure 2 shows the state flow of the different operational modes of the robot. The arrow’s direction shows the state transition flow.

B. System Definitions and Notations

Let us consider a swarm of robots $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$, a set of tasks $\mathcal{T}_r = \{t_1, t_2, \ldots, t_m\}$ assigned to each robot $r \in \mathcal{R}$ by the scheduler, and a set of stationary charging stations $\mathcal{P} = \{p_1, p_2, \ldots, p_s\}$.

The transition time of a robot $r \in \mathcal{R}$ from the Off to the Standstill mode is defined as T_r^{startup} and the energy consumption in this period is E_r^{startup}. At time τ, the location of robot r is defined as $L_r[\tau]$, and its energy level as $E_r[\tau]$.

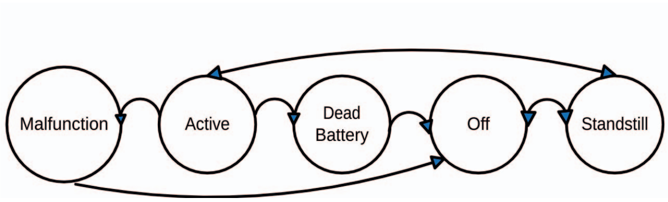

Fig. 2: This figure shows the state flow defined for the robots in the swarm. The direction of the arrows shows the operational state flow.

L_p is the location of the charging station p. The energy needed for a dead battery robot to reach the nearest charging station, from a location L, equals $E_r^{\text{recovery}}[L]$, while the time equals $T_r^{\text{recovery}}[L]$. $P_r^{\text{harvest}}[\tau]$ represents the harvested power of the robot r at time τ.

Each task $t \in \mathcal{T}_r$ has a starting time T_t^{start}, an end time T_t^{end}, a starting location L_t^{start}, and an end location L_t^{end}. The energy consumed while performing task t is defined as E_t and the task completion time is T_t.

C. Mathematical Model

In this section, the mathematical model of idle mode energy consumption, robot energy model, energy harvesting model, robot's recovery time, and recovery energy is formulated.

1) Idle mode energy model: The robot $r \in \mathcal{R}$ has an idle time $T_{r,j}^{\text{idle}}$ between executing tasks t_{j-1} and t_j, given by:

$$T_{r,j}^{\text{idle}} = T_{t_j}^{\text{start}} - T_{t_{j-1}}^{\text{end}} \tag{1}$$

The associated energy consumption over a time period $T_{r,j}^{\text{idle}}$ can then be calculated as:

$$E_{r,j}^{\text{idle}} = \int_{T_{t_j}^{\text{start}}}^{T_{t_{j-1}}^{\text{end}}} P_r^{\text{idle}}[t] \; dt \tag{2}$$

Where P_r^{idle} is the power consumption during the idle state.

2) Robot energy estimation model: Energy consumption estimation for mobile robots can generally be categorized into three main approaches. The first approach relies on a dynamic model of the robot and its motorized actuators. The second approach estimates power consumption from a physical perspective by modeling the energy required to move the robot and overcome traction resistance and related frictions. The third approach includes a broad range of data-driven and machine-learning models [11]. Moreover, the power consumption of a robot is a function of its subsystems, including sensing, computing, communication, and actuation instance powers. It can be shown that, in scenarios such as the one considered in this paper, the total sensing, computing, and communication powers remain constant with respect to the robot's task [12]. So the estimated energy consumption of the task $\hat{E}_t$, can be formulated as follows:

$$\hat{E}_t = \int \Bigg(\sum_{j=1}^{N} \left(\sigma_0 \dot{\theta}_{w_j}^2(t) + \sigma_1 \operatorname{sgn}\left(\dot{\theta}_{w_j}(t)\right) \dot{\theta}_{w_j}(t) \right) + \sigma_2 m + \sigma_3 \Bigg) dt \tag{3}$$

where N is the number of the robot's active wheels and $\dot{\theta}_{w_j}$ is each wheel speed, $\sigma_0, ..., \sigma_3$ are coefficients that can be estimated using the well-known regression methods and m is the total robot and payload mass. Additionally, to estimate the task time $\hat{T}_t$ using the robot's path planning information and obtaining the predicted route's waypoints (wp^k) the following equation can be used:

$$\hat{T}_t = \sum_{j=1}^{k} \left(\frac{d_l\left(wp^j, wp^{j-1}\right)}{v} + \frac{d_a\left(wp^j, wp^{j-1}\right)}{\omega} \right) \tag{4}$$

where $d_l(.), d_a(.)$ denotes the functions for calculating the linear and angular distance of two waypoints and v, ω are robot linear and angular speeds and k is the number of waypoints.

3) Battery capacity voltage curve model: To be able to have the robot energy level $E_r(t)$ based on the battery voltage level, one approach is to model the capacity voltage curve [13]. Normally in this method, the polynomial model of the battery capacity-voltage curve is modeled as the following equation:

$$E_r(V_r) = \sum_{n=0}^{n} a_n \cdot V_r^n \tag{5}$$

where n is the polynomial order number. Here, E_r is the robot energy and V_r is the robot battery voltage. By knowing the voltage, we can estimate the approximate energy level of the robot battery.

4) Energy harvesting model: The energy harvested by a robot r over a time period T_r^{harvest} considering a fixed P_r^{harvest} is given by:

$$E_{\text{r}}^{\text{harvest}} = P_r^{\text{harvest}} \cdot T_r^{\text{harvest}} \tag{6}$$

5) Recovery energy calculation: The energy needed to recover the robot and navigate the robot to the nearest charging station is:

$$E_r^{\text{recovery}}[\tau] = E_{t_p} - E_r[\tau] + E_r^{\text{startup}} \tag{7}$$

Where t_p is the task of navigating the robot from $L_r[\tau]$ to the nearest charging station L_p.

6) Recovery time calculation: The time needed to charge the robot to E_r^{recovery} using the energy harvesting rate P_r^{harvest} is:

$$T_r^{\text{recovery}}[\tau] = \frac{E_r^{\text{recovery}}}{P_r^{\text{harvest}}} + T_{t_p} + T_r^{\text{startup}} \tag{8}$$

IV. Scheduler Function

This section presents two pivotal algorithms designed for the EMM: the Recovery Calculation Algorithm (RCA) and the Idle Period Management Algorithm (IPMA).

A. Recovery Calculation Algorithm (RCA)

The RCA computes the energy E_r^{recovery} needed for a robot to reach a charging station and the time T_r^{recovery} required for the robot to reach the nearest charging station. It takes into account the current energy level, the distance to the nearest charging station, the navigation power required, and

Algorithm 1 Recovery Calculation Algorithm

Wake up the EMM
Input: Receive the V_r
Calculate E_r, P_r^{harvest} from Eq.6 and Eq.5
Calculate E_t from Eq.3
Calculate E_r^{recovery} from Eq.7
Calculate T_r^{recovery} from Eq.8
Return T_r^{recovery}, E_r^{recovery}

Algorithm 2 Idle Period Energy Management Algorithm

Input: Idle time $T_{r,j}^{\text{idle}}$, Startup time T_r^{startup}
Wake up the EMM
Procedure:

1) If $T_{r,j}^{\text{idle}} > T_r^{\text{startup}}$ and $E_{r,j}^{\text{idle}}[\tau, \tau + T_{r,j}^{\text{idle}}] > E_r^{\text{startup}}$:
   a) Switch off the robot to ”Off” mode.
   b) Wait until $\tau = T_{t_j}^{\text{start}} - T_r^{\text{startup}}$.
   c) Switch on the robot to ”Standstill” mode.
2) Else:
   a) Keep the robot in ”Standstill” mode.

the energy harvesting rate of the robot. E_r^{recovery} and T_r^{recovery} are two important components that the scheduler needs to plan for future tasks. Algorithm 1 shows the detailed steps for calculating these elements.

B. Idle Period Energy Management

The IPMA determines whether to keep the robot powered on or shut it down during idle times based on the duration of the idle period ($T_{r,j}^{\text{idle}}$) compared to a predefined startup time (T_r^{startup}). If the idle time exceeds T_r^{startup} and the $E_{r,j}^{\text{idle}}[\tau, \tau + T_{r,j}^{\text{idle}}]$ is bigger than the E_r^{startup}, the algorithm shuts down the robot to conserve energy and schedules it to start up shortly before the next task begins. Otherwise, it keeps the robot on to minimize startup delays. Algorithm 2 shows the idle state energy management procedure.

V. Exprimental Results

This chapter presents the experimental results of the energy management strategies implemented in a TurtleBot3 Burger (TB) robot that is depicted in Figure 3. The robot harvests ambient energy using a FIT0600 6V 1A solar panel with a size of $29 \times 15 \times 0.135cm$ and a maximum power of $5W$. The EMM is an ESP32 development board with a frequency of $2.4GH$ equipped with BLE communication. The EMM controls the application by turning the power on and off. There is no communication between the ESP32 and the robot. The robot requires a 12 V power supply which is managed by the step-up module on the EMM. Communication with the cloud is done through predefined commands. The cloud scheduler will turn off the robot for a fixed period by sending ”off, ’time’” to the robot. The robot will be powered off, and the EMM will enter sleep mode for that period and start a timer. To track power consumption and energy harvesting, the EMM will periodically (every $60s$) perform ADC measurements to

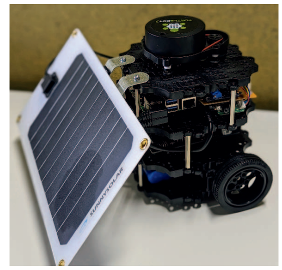

Fig. 3: TurtleBot3 Burger setup equipped with EMM and solar panel.

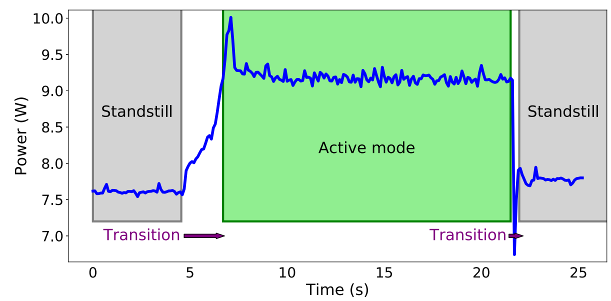

Fig. 4: TurtleBot3 power consumption during no-load Active mode and Standstill mode. This figure also shows the transition between the Standstill mode to Active mode and vice-versa.

monitor the battery level. To be able to use the solar panels we used a set of six 3.7V Lithium Polymer HW Li-18650 batteries instead of the on-market TB battery.

A. Energy Consumption During Standstill Mode

As demonstrated in Figure 4, we measured the energy consumption of the TB while it was in standstill mode. During this mode, the robot remains stationary, but the Raspberry Pi board, motor drive board, and its mounted sensors, such as LiDAR, remain active, leading to considerable energy consumption. The Standstill mode of the TB is quite energy-intensive, with prolonged periods in this mode causing significant energy use. The results show that the TB consumes an average of 7.5 W in Standstill mode, which is approximately 81% of its energy consumption in no-load Active mode (9.2 W) (cf., Figure 4).

B. EMM Energy Consumption

This section presents the energy consumption of the EMM board. Figures 5 and 6 show experimental results of the wake-up signal sent by the scheduler and the power consumption of the EMM during active mode respectively. The power consumption of the EMM board is on average 0.49 W. The average power consumption of the EMM in its sleep mode without any Analog-to-Digital Converters (ADC) measurement is 26.2 μW, and with an ADC measurement every 60 seconds (c.f, Figure5), it is 2.3 mW. Both values are negligible.

C. Startup Time from Off Mode to Standstill Mode

Start-up time is the time from sending the command from the scheduler until the robot is in standstill mode and ready

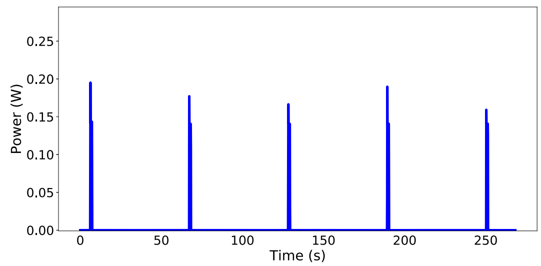

Fig. 5: Energy Management Module in sleep mode and performing an ADC measurement every minute to track the energy harvesting amount.

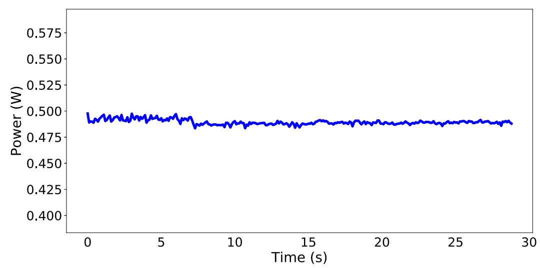

Fig. 6: Energy Management Module power consumption during active mode.

to do the tasks. Optimizing startup time is crucial for robots that spend extended periods in idle mode to minimize energy consumption. Figure 7 displays the density distribution of startup time and startup energy consumption. Although the peak probability density centers around a startup time of approximately 60 seconds, we focus on the worst-case scenario with a measured T_r^{startup} of 102 seconds and E_r^{startup} of 683 J for TB. Choosing the worst-case scenario enhances the reliability of idle mode energy management strategies.

D. TurtleBot3 Energy Model

To estimate the coefficients of the energy consumption model in Eq.3, the robot is tested under two different conditions of no-load and a 5 kg payload. The wheels' speed data is obtained from the odometry node in ROS, and the actual power consumption of the robot is measured using an additional power meter sensor. Additionally, σ_2 and σ_3 can be mathematically combined as $\hat{\sigma_2} = \sigma_2 + \frac{\sigma_3}{m}$. Using the linear regression method, the coefficients of Eq.3 are estimated ($\sigma_0 = -0.0381, \sigma_1 = 0.4736, \hat{\sigma_2} = 0.4543$). The results show a mean absolute percentage error of 3.5% with $R^2 score = 87.3$. The model performance validation is shown in Figure 8.

E. Idle Period Energy Consumption Experiment

This experiment compares the energy consumption of the robot during idle mode with and without the implementation of the EMM's idle management method. The scheduler assigns two delivery tasks t_1 and t_2 to the robot r. The idle time between these two tasks is $T_{r,j}^{\text{idle}} = 300$ (s), the scheduler manages the idle mode based on Algorithm 2. The results

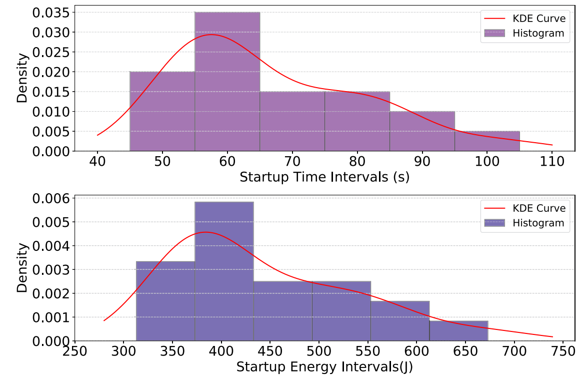

Fig. 7: TurtleBot3 start-up time measurement. This figure shows the distribution of the TB start-up time and energy consumption during 20 tests.

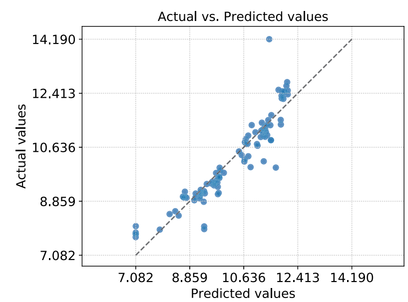

Fig. 8: Power consumption estimation validation with the test data under two different payloads (no-load and 5 kg payload) shows the bounded error for the performed estimation.

show that by using the EMM, the robot consumes $1205\ J$ of energy. Without the EMM, the consumption is $2250\ J$, which indicates a 54% reduction in energy consumption by using the EMM.

Figure 9 demonstrates that the proposed idle management method significantly reduces energy consumption during idle mode. This improvement highlights the effectiveness of the EMM in optimizing energy usage and extending the operational lifespan of the robot battery.

F. Robot Recovery Scenario

The robot recovery process involves recharging the robot using ambient energy sources when it is in a low-energy state. We compare the recovery time and energy required for the robot using indoor and outdoor energy harvesting.

The experiment assumes a 20×14 meter rectangular warehouse environment with one charging station at the coordinates of $L_p = (-10, 7)$ (meter) with the reference point (0,0) at the middle of the map (cf., Figure 10). The position of the robot is $L_r = (10, -7)$. Based on the robot model (cf., Eq. 3), the $E_{t_p}$and T_{t_p} needed for the robot to navigate from its current position to the nearest charging point are $1255\ J$ and $145.3\ s$, considering the obstacles and path planning method using the Gazebo simulation environment. The T_r^{startup} for the TB is considered as 102 seconds and $E_r^{\text{startup}} = 683 J$ (i.e.,

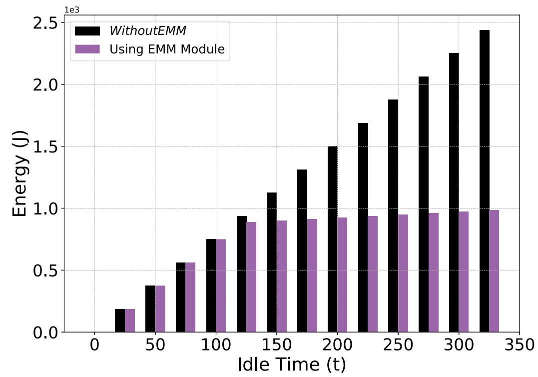

Fig. 9: Energy consumption of the TurtleBot3 robot in idle state using EMM and without EMM. For every 10 seconds, the idle time exceeds the start-up time, the EMM can reduce energy consumption by an average of 2.6% compared to the method without using the EMM.

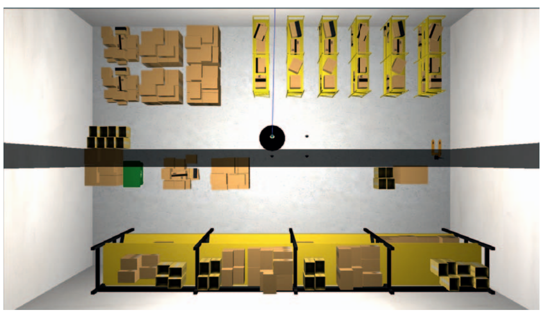

Fig. 10: Gazebo simulation environment map.

the worst-case from Figure 7). Also, the harvested power for outdoor and indoor is considered as $P_r^{\text{harvest}}{}_{\text{outdoor}}$=4.38 W and $P_r^{\text{harvest}}{}_{\text{indoor}}$=0.1 W based on the real experiments of the used solar panel under the sun and under the indoor light. The current battery of the robot is assumed $E_{r_1} = 150\ J$. The $T_{\text{outdoor}}^{\text{recovery}}$ = 645.2 seconds (10.75 minutes) and $T_{\text{indoor}}^{\text{recovery}}$ = 17667 seconds (4.9 hours). The results show the effectiveness of this method in recovering the robot, especially for outdoor scenarios.

VI. Conclusion

This paper introduces an innovative Energy Management Module (EMM) that utilizes a Bluetooth Low Energy (BLE) connection to enhance the control and efficiency of ambiently-powered robotic swarms. The EMM facilitates control over multiple modes—active, standstill, off, dead battery, and malfunction—enabling a more energy-aware and flexible scheduler. Key benefits include improved energy management during idle periods, precise wake-up timing, and the capability to revive dead batteries using a low-power connection. Furthermore, the EMM enhances robot reliability by performing hard resets during malfunctions. Predicting recovery times allows the scheduler and monitoring unit to adjust robot control based on reliability and energy optimization objectives, thereby ensuring efficient task scheduling and robot availability.

Integrating the EMM with a scheduler optimizes energy consumption during idle periods based on the duration of the idle time. In a case study, we found that for every 10 seconds the idle time exceeds the start-up time, our method can reduce energy consumption by an average of 2.6% compared to the method without using the EMM. Additionally, experiments demonstrate the EMM's efficacy in managing dead battery robot recovery using energy harvesting. Results indicate worst-case recovery times of 10.7 minutes for outdoor scenarios and 4.9 hours for indoor environments.

Acknowledgment

This work was supported in part by the European Union's Horizon Europe Framework under Grant 101093046 - project OpenSwarm, and in part by the Flemish Government under FWO Project G019722N - LOCUSTS. Views and opinions expressed are however those of the author(s) only and the European Commission is not responsible for any use that may be made of the information it contains.

References

[1] T. Verstraten, M. S. Hosen, M. Berecibar, and B. Vanderborght, "Selecting suitable battery technologies for untethered robot," *Energies*, vol. 16, no. 13, p. 4904, 2023.

[2] M. Soori, B. Arezoo, and R. Dastres, "Optimization of energy consumption in industrial robots, a review," *Cognitive Robotics*, vol. 3, pp. 142–157, 2023.

[3] M. Wesselhöft, J. Hinckeldeyn, and J. Kreutzfeldt, "Controlling fleets of autonomous mobile robots with reinforcement learning: A brief survey," *Robotics*, vol. 11, no. 5, p. 85, 2022.

[4] R. Garcia, A. Bittencourt, and E. Villani, "Relevant factors for the energy consumption of industrial robots," *J Braz. Soc. Mech. Sci. Eng.*, vol. 40, p. 464, 2018.

[5] O. M. Gul, "Energy harvesting and task-aware multi-robot task allocation in robotic wireless sensor networks," *Sensors*, vol. 23, no. 6, p. 3284, 2023.

[6] Z. Li, Q. Tang, and L. Zhang, "Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm," *Journal of Cleaner Production*, vol. 135, pp. 508–522, 2016.

[7] Paryanto, M. Brossog, M. Bornschlegl, and J. Franke, "Reducing the energy consumption of industrial robots in manufacturing systems," *The International Journal of Advanced Manufacturing Technology*, vol. 78, no. 5, pp. 1315–1328, 2015.

[8] J. M. Nilakantan, G. Q. Huang, and S. G. Ponnambalam, "An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems," *Journal of Cleaner Production*, vol. 90, pp. 311–325, 2015.

[9] O. Benedikt, B. Alikoc, P. Sucha, and Z. Hanzálek, "A polynomial-time scheduling approach to minimise idle energy consumption: An application to an industrial furnace," *Journal*, October 2019.

[10] J. Batista Abikarram, K. McConky, and R. Proano, "Energy cost minimization for unrelated parallel machine scheduling under real time and demand charge pricing," *Journal of Cleaner Production*, vol. 208, pp. 232–242, 2019.

[11] D. B. Licea, M. Ghogho, and M. Saska, "When robotics meets wireless communications: An introductory tutorial," *Proceedings of the IEEE*, 2024.

[12] L. Liu, R. Zhong, A. Willcock, N. Fisher, and W. Shi, "An open approach to energy-efficient autonomous mobile robots," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 11 569–11 575.

[13] X. Han, X. Feng, M. Ouyang, *et al.*, "A comparative study of charging voltage curve analysis and state of health estimation of lithium-ion batteries in electric vehicle," *Automotive Innovation*, vol. 2, pp. 263–275, 2019.