

An Open Approach to Energy-Efficient Autonomous Mobile Robots

Liangkai Liu¹, Ren Zhong¹, Aaron Willcock¹, Nathan Fisher¹, and Weisong Shi²

Abstract—Autonomous mobile robots (AMRs) have the capability to execute a wide range of tasks with minimal human intervention. However, one of the major limitations of AMRs is their limited battery life, which often results in interruptions to their task execution and the need to reach the nearest charging station. Optimizing energy consumption in AMRs has become a critical challenge in their deployment. Through empirical studies on real AMRs, we have identified a lack of coordination between computation and control as a major source of energy inefficiency. In this paper, we propose a comprehensive energy prediction model that provides real-time energy consumption for each component of the AMR. Additionally, we propose three path models to address the obstacle avoidance problem for AMRs. To evaluate the performance of our energy prediction and path models, we have developed a customized AMR called Donkey, which has the capability for fine-grained (millisecond-level) end-to-end power profiling. Our energy prediction model demonstrated an accuracy of over 90% in our evaluations. Finally, we applied our energy prediction model to obstacle avoidance and guided energy-efficient path selection, resulting in up to a 44.8% reduction in energy consumption compared to the baseline.

I. INTRODUCTION

With the vast improvements in computing technologies, e.g., sensors, computer vision, machine learning, and hardware accelerators, autonomous mobile robots (AMRs) have been widely deployed in several industry applications [1], [2], [3]. For example, an AMR can be programmed to patrol the fences of a private area so that the security team can be timely notified of intrusions [4]. Typically, the AMR is composed of locomotion (motor, control, etc.), sensors (camera, 2D LiDAR), computing boards, and batteries [5]. The motors translate the electricity into the movement of rotors to make AMRs move. Besides, intensive computer vision and deep neural networks are deployed to help the AMR understand the environment and drive autonomously. However, adding up these tasks increases the power dissipation of AMRs, and shrinks their working time per charge. Optimizing the energy consumption of AMRs has become a fundamental challenge for the broad deployment of AMRs.

Lots of efforts have been made to optimize the energy consumption of AMRs. These works could be divided into energy-efficient motion planning and energy-efficient computing systems. Motion planning is a critical task in the path planning of AMRs. Several works try to model the trajectories of different paths and dynamically select the most energy-efficient path [6], [5], [7], [8]. However, the

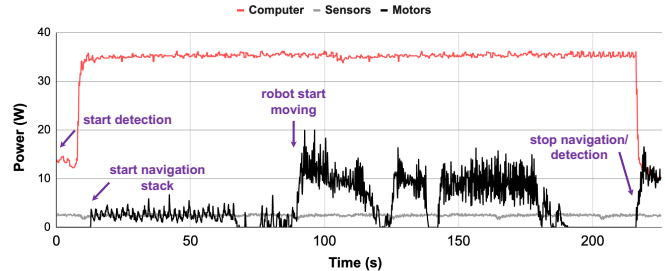


Fig. 1. The timeline for power dissipation for a typical AMR.

planning is based on motors' energy consumption. It wastes lots of energy in AMRs because they need to consider the energy consumption by the computing system for sensing, localization, and perception. To build an energy-efficient computing system for AMRs, previous research studies either explore the architecture constraints and accelerations in energy efficiency [9], [10], [11] or leverage software middleware to optimize the energy consumption of computer vision tasks [12]. None of the existing works have studied and optimized the end-to-end energy consumption of AMRs.

From the empirical studies we have done with real AMRs, one of the significant energy inefficiencies is the need for coordination between computation and controlling. Figure 1 shows a timeline for power dissipation for a typical AMR [13]. We can find that the sensor's power dissipation is very stable. When the detection starts, the computer's power dissipation increases quickly to the highest power, mainly because the detection causes GPUs to run at the maximum frequency. The navigation consists of the localization and path planning for the AMR. When the navigation starts, the motor's power dissipation is mainly related to the speed and acceleration of the AMR. Finally, the detection and navigation module stop when the AMR reaches its destination. Power dissipation to make the AMR stop is also non-negligible. As the timeline shows, the power dissipation pattern of the computer is different from that of the motor. There is no coordination between the perception and planning tasks running on the computing system with the motor control, which contributes to the energy waste for AMR. Furthermore, the energy consumption of the AMR adds another dimension of time to the power dissipation, which makes the coordination even harder.

To enable the coordination of computing tasks with motor control, we propose a complete energy model for AMR to describe the real-time energy consumption of different components, including the motors, computing board, and sensors. Next, we define a general obstacle avoidance prob-

¹Liangkai Liu, Ren Zhong, Aaron Willcock, and Nathan Fisher are with the Department of Computer Science, Wayne State University.

²Weisong Shi is with the Department of Computer and Information Sciences, University of Delaware.

lem for AMRs and propose a path model which models the control commands into a combination of moving straight and spinning. Next, we build a customized AMR called Donkey, which has the capability of end-to-end power profiling. On top of Donkey, we collect power and energy traces under different configurations of the navigation stack to fine-tune the energy model and the path model. Finally, we evaluate the performance of the proposed energy model on the Donkey. Based on evaluating different paths, the energy prediction model maintains an accuracy higher than 90%. Furthermore, we explore three types of path selection when an obstacle blocks the robot. We guide energy-efficient path selection under different configurations of obstacle sizes and control RPM. In summary, this paper makes the following three contributions:

- We propose a comprehensive energy prediction model for AMRs based on the motor’s control message and the status of runtime applications in the computing system.
- We design and implement the Donkey platform for end-to-end power profiling of AMRs, which is general for most battery-powered AMRs. We evaluate the performance of the energy model on Donkey. It achieves higher than 90% accuracy for all tested cases.
- We evaluate the energy prediction model into a general obstacle avoidance problem and provide guidance on energy-efficient path selection for different obstacle sizes and control RPM configurations. The best path shows up to 44.8% energy reduction than the baseline.

The rest of the paper is organized as follows. Section II studies the related work. Section III describes the end-to-end energy profiling and energy prediction model, respectively. Section IV evaluates the performance of the energy prediction model. Section V concludes the paper.

II. RELATED WORK

Various approaches have been proposed to increase the battery life of AMRs. They are divided into two directions: energy-efficient motion planning and energy-efficient computing systems.

A. AMR System Overview

Figure 2 shows an overview of a representative AMR system. In general, the sensor includes a 2D LiDAR and a camera. The localization module takes the real-time */scan* message with the pre-built map to get the real-time location of the AMR. Besides, object detection and tracking are based on the */image* from the camera. Next, the global planner generates a costmap with a route based on the AMR’s location and goal. The local planner leverages the detection and global planner results to generate the twist message for AMR’s control. Finally, the controller controls motors with a control loop that adjusts its command based on the desired speed and the motor’s feedback.

B. Energy Efficient Motion Planning

Previous studies have proposed to find the most energy-efficient path for robots to move to a specific location or to

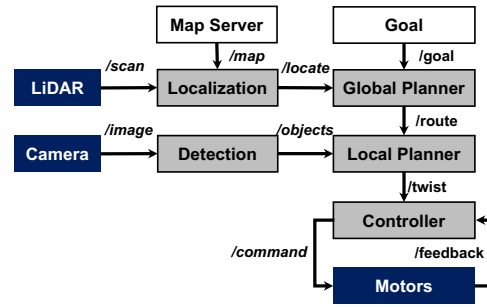


Fig. 2. A general design for AMR.

cover a large area [6], [5], [7], [8], [14], [15], [16]. Other solutions coordinate various AMRs to optimize their charging scheduling [17], [18], [19], [20], [21], [22], [23], [24], [25]. Most of the above studies focus on the energy consumption of the robots due to the mechanical parts. However, they ignored the energy consumption of the computing system, which becomes a significant part of autonomous mobile robots. Unlike all the above solutions, this paper focuses on the end-to-end profiling and optimization of AMRs, including motion planning and the computing system.

C. Energy Efficient Computing System

The computing system is the “brain” of the AMR to drive autonomously. A large portion of jobs executed by AMRs is related to computer vision, which exploits cameras and machine learning models to make the robots more intelligent and autonomous [26]. While some related studies focus mainly on performance [9], [10], [11] without considering the energy consumption, other solutions focus on pruning and compression techniques to make the trained model more energy efficient [27], [28]. However, none of these studies focus on how those applications are executed on the computing hardware of the robots. When multiple applications run concurrently in a shared environment, they are likely to slow down each other [29], [30], [31]. E2M addresses this challenge by optimizing computing system energy consumption and ensuring the desired applications’ performance while minimizing AMR’s computing resource’s energy consumption [12]. However, E2M does not optimize AMR’s path planning and control energy consumption. In this paper, we design the AMR with the capability of power profiling on both motion planning and computing system. To the best of our knowledge, this is the first work to study and optimize the end-to-end energy consumption of AMRs.

III. END-TO-END ENERGY MODEL FOR AMR

Based on the motivation, we found that the total power dissipation and breakdown varies with different controls and running applications in the computing system. Although we can obtain an accurate power prediction model for motors with different RPMs, the navigation’s control is unpredictable since the control always follows nearby waypoints and trajectories, making the end-to-end energy modeling of Donkey becomes challenging. To get the total energy

consumption of the AMR moving through a specific path, we introduce a path model which decouples a path into the combination of moving straight and spinning. Besides, we take a state machine-based energy model representing the energy consumption of the computing system.

A. Computing System Energy Model

Based on the computing system power profiling observation, new applications introduce increasing power dissipation, which will become stable in seconds. We introduce a state machine to represent the power stage of CPU, GPU, SOC, VDDRQ, and SYSSV for different applications. Therefore, for the computer's energy consumption, each component's energy is a combination of energy consumed under each state. The power dissipation under different running applications is recorded when stable. Since the runtime of the computer transits between four states, the total time of all the states equals T . Finally; there is a remaining energy consumption by the computing system, represented as $P_{other} * T$.

$$E_{sensor} = (P_{cam} + P_{lidar}) * T \quad (1)$$

$$E_{computer} = \sum P_{cpu}^i * t_i + \sum P_{gpu}^i * t_i + \sum P_{soc}^i * t_i + \sum P_{sys}^i * t_i + \sum P_{vdd}^i * t_i + P_{other} * T \quad (2)$$

$$\sum t_i = T \quad (3)$$

B. Path Model

In this paper, we model the path of the AMR moving from one point to another as a combination of moving straight and spinning controls. For crossing a wide-open area, a path includes two way-points, the starting, and ending point, and the straight trajectory connecting them is good enough. However, when dealing with obstacles, we should choose different strategies. To describe robot obstacle avoidance tasks, we consider a general task where a robot is trying to reach point B from point A while an obstacle is blocking the way. Figure 3 shows three typical obstacle avoidance paths: rectangle, triangle, and circle.

1) *Rectangle path*: The robot starts from point A and moves towards point B, L away. Suppose the size of the robot is $d * d$. The length of an obstacle is N while the width is M . Without loss of generality, the heading orientation of the robot is both towards the front. The linear speed and angular speed of the robot are constant in each model. Under the rectangle model, the robot first spins $\pi/2$ radians to head up; then, it moves forward by $(M + d)/2$. Then the robot spins $\pi/2$ radians to the right, moves $L + d/2$ past the obstacle, spins $\pi/2$ radians, and moves another $(M + d)/2$ to reach point B; then it spins $\pi/2$ radians at point B. The rectangle path moves forward for $L + M + 1.5d$ while spinning for 2π radians.

TABLE I
THE MOVEMENT OF THREE PATHS.

Path	Straight Moving (D_l)	Spinning (D_s)
1	$L + M + 1.5d$	2π
2	$L/\cos(\arctan(M/(L - N)))$	$4\arctan(M/(L - N))$
3	$\pi * L/2$	2π

2) *Triangle path*: The case for triangle and circle is much more complex than the rectangle case. Figure 3 illustrates the calculation. For the triangle model, the robot first spins θ radians and then moves straight. The θ is calculated by drawing a tangent from the robot's edge to the obstacle's edge. Then we can get that θ is $\arctan(M/(L - N))$. Then the robot moves forward for $L/(2\cos\theta)$. Next, it spins 2θ radians to change its head direction, followed by another $L/(2\cos\theta)$ moving straight to point B and spins another θ radian. The triangle path needs $L/\cos\theta$ moving straight and 4θ spinning, where θ equals $\arctan(M/(L - N))$.

3) *Circle path*: For the circle path, the robot first spins $\pi/2$ radians to head up, then follows a half circle whose radius is $L/2$ to bypass the obstacle and directly reach point B. In this circle, the linear movement is $\pi * L/2$, while the rotation is π radians. Then the robot spins another $\pi/2$ radians to the left to finish the task. The circle path needs $\pi * L/2$ linear movement and 2π radians spinning. Table I summarizes straight moving distances and spinning radians of three paths.

C. Motor Energy Model

A Brushless DC motor replaces the mechanical commutation function with electronic control, which makes it more accurate and reliable than a brushed DC motor. The variations in current values bring a big challenge in predicting the power dissipation of the brushless DC motor. Typically, the power prediction model is either a linear or polynomial model of the control commands to estimate the average power value. In [5], a six-degree polynomial model is introduced to predict the power dissipation of the DC motor.

$$\min \quad loss = \frac{1}{N} \sum_{i=1}^N (y^{t_i} - \hat{y}^{t_i})^2 \quad (4)$$

$$\text{where } \hat{y}^t = F(c_0^t, c_1^t, c_2^t, c_3^t, f_0^t, f_1^t, f_2^t, f_3^t) \quad (5)$$

Unlike previous works, which take seconds level average current value, we propose to predict the real-time current results at the millisecond level. The input to the power prediction model includes the motor's control command and feedback [32]. In Equation 5, $c_0^t, c_1^t, c_2^t, c_3^t$ and $f_0^t, f_1^t, f_2^t, f_3^t$ represents control commands and feedback of four motors while \hat{y}^{t_i} and y^{t_i} represent the predicted and actual power dissipation of a motor at time t_i . F represents the power dissipation prediction model, which can be linear, polynomial, deep neural network-based functions, etc. For the regression problem, mean squared error (MSE) is used to

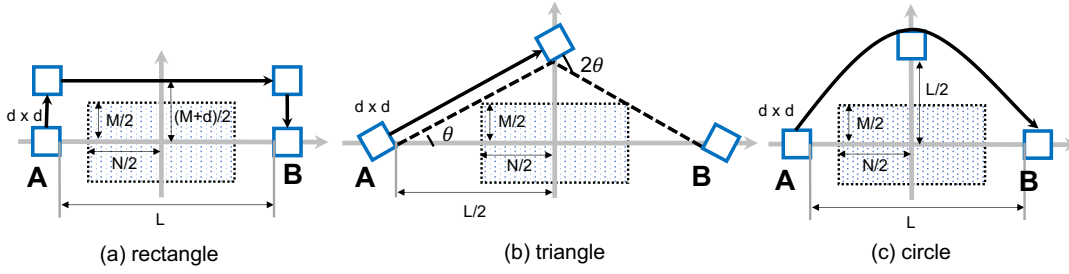


Fig. 3. Path model for rectangle, triangle, and circle for avoid obstacles.

define the loss of predicted fuel rate. Therefore, the objective of the regression problem is to get the minimum loss.

D. End-to-End Energy Model

The end-to-end energy model of Donkey can be decomposed into three parts: motor, sensor, and computing system, as is shown in Equation 7. The energy consumption for motors can be divided into the energy consumption of moving straight and spinning. $F_{st}(\omega_1)$ represents the motor's power dissipation when it's moving straight with RPM ω_1 , while $F_{sp}(\omega_2)$ represents the motor's power dissipation when it's spinning with RPM ω_2 . The straight moving distances D_l as well as spinning radians D_r for each path, are provided in Table I. Time for moving forward is calculated by using the moving distance D_l divided by the linear speed s_l at RPM ω_1 . Similarly, the time for spinning is represented as $D_r/s_r(\omega_2)$. Since the power dissipation of the sensor is stable during runtime, we calculate the energy consumption by the sensor's power multiples the total running time T for each path. We assume there is no waiting time for Donkey during path following. T can be calculated by summing the straight moving time and spinning time. The path with the minimum end-to-end energy consumption is selected for motion planning.

$$E_{total} = E_{motor} + E_{computer} + E_{sensor} \quad (6)$$

$$E_{motor} = F_{st}(\omega_1) \frac{D_l}{s_l(\omega_1)} + F_{sp}(\omega_2) \frac{D_r}{s_r(\omega_2)} \quad (7)$$

$$T = \frac{D_l}{s_l(\omega_1)} + \frac{D_r}{s_r(\omega_2)} \quad (8)$$

IV. EVALUATION

A. Experimental Setup

We design and build the Donkey platform following general rules to achieve representative power profiling. Figure 4 shows an overview of the Donkey platform. There are four Mecanum wheels installed on Donkey. Four brushless DC motors with a hall sensor and gearbox are used as the locomotion part. Each motor is controlled by a separate motor controller using Pulse-width modulation (PWM) signal through serial communications. The sensors on Donkey include an Intel RealSense L515 camera and an RPLiDAR A3. The computing board is an NVIDIA Jetson AGX Xavier.

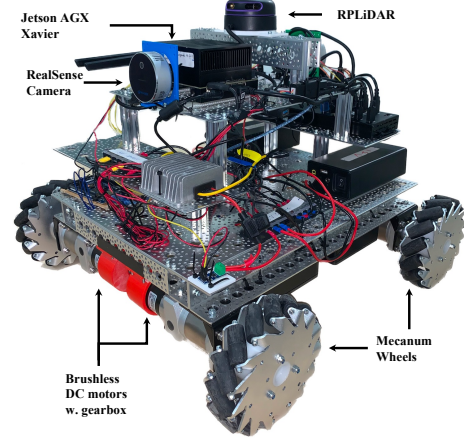


Fig. 4. Overview of Donkey platform.

The sensors and motor controllers are connected to the Jetson board through a 10-port USB hub.

The Jetson AGX Xavier board is installed with *JetPack 4.4-DP* [33] (L4T R32.4.2) and uses an *l4t-ml* docker image [34] as the base image for system setup. The *l4t-ml* image includes several libraries for machine learning-related applications: TensorFlow 1.15, PyTorch v1.5.0, torchvision v0.6.0, CUDA 10.2, cuDNN 8.0.0, OpenCV 4.1, etc. We install ROS melodic with a navigation stack and build Darknet as a ROS node [35].

In the power system of Donkey, two Lithium-ion batteries are installed to power the whole system. A 24V Lithium-ion battery with 42980mAh electricity is used to power motors and motor controllers. Another 12V Lithium-ion battery with 38400mAh electricity is used to power the remaining system, including the sensors, USB hub, Jetson AGX Xavier, etc. An Arduino is set up to acquire current results from current sensors from each branch and finally send them to the Jetson AGX through serial communications.

B. Motor Energy Prediction

In this paper, we implement three models to predict the power dissipation of the brushless DC motor: linear regression (LR), polynomial regression (PR), and multi-layer perceptron (MLP). For the evaluation of the power prediction model, we use the coefficient of determinism (R^2) to evaluate the power prediction model's performance. As shown in Equation 9, R^2 is defined as one minus the sum of predicted

errors' squares divided by errors' squares when using the mean value as the predicted value. y^{t_i} and \hat{y}^{t_i} represent the actual and predicted power, respectively, while \bar{y} represents the mean value of actual power dissipation during time t_i and t_N . We can find that R^2 represents the distance of the predicted value with the actual and average values. We define the power prediction accuracy to understand the regression performance from another perspective. From Equation 10, the accuracy is defined as one minus the absolute error's median value divided by the mean value.

$$R^2 = 1 - \frac{\sum_{i=1}^N (y^{t_i} - \hat{y}^{t_i})^2}{\sum_{i=1}^N (y^{t_i} - \bar{y})^2} \quad (9)$$

$$Accuracy = 1 - \frac{\text{median}\{|y^{t_1} - \hat{y}^{t_1}|, \dots, |y^{t_N} - \hat{y}^{t_N}|\}}{\bar{y}} \quad (10)$$

For both moving straight and spinning cases, we collected current data with control command and feedback data at different RPMs: 500, 1000, 1500, 2000, and 2500. For each RPM, we use LR, PR, and MLP to train the model to predict the power dissipation of motors. Before feeding into the model, we standardize the features by removing the mean and scaling to unit variance. By comparing different orders of the polynomial model, the six-degree polynomial model has the best prediction performance, the same as [5]. For MLP, we deploy a ten-layer MLP where the first seven layers have 100 neurons, and the other three have 50 neurons. The loss function for MLP is based on mean square error (MSE).

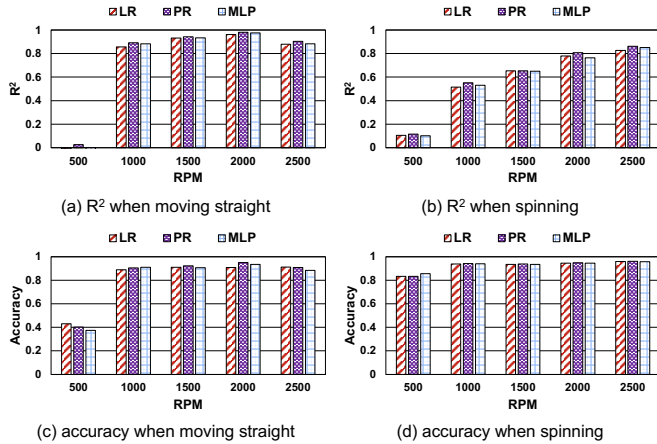


Fig. 5. Motor power prediction R^2 and accuracy for different RPMs when moving straight and spinning.

The results for LR, PR, and MLP under different RPMs are shown in Figure 5. R^2 reflects the regression performance, while accuracy reflects the median error of the prediction. All three models could achieve R^2 and accuracy higher than 0.8 for moving forward and spinning when the RPM is above 1000, reflecting good prediction performance. However, when the RPM is 500, the R^2 for moving straight and spinning is less than 0.2. It's because the feedback of the BLDC motor is not accurate at low RPM, which is common in a brushless three-phase hall sensor motor. The lower the

RPM is, the more error the position of the motor's rotor could be. The error at low RPM also brings challenges for the PID controller. This finding makes us avoid keeping the motor at low RPM.

C. End-to-End Energy Prediction Accuracy

To begin with, we control Donkey to move straight and spin at a constant RPM to collect all the current and power data. Given the RPM and range of movement, we calculate the total energy consumption based on the energy model. Besides, we collect the real-time energy consumption by moving the robot a certain distance or spinning it to a certain degree continuously. In the experiment setup, we set the moving straight distance as 10m while the spinning degree as 2π radians. The results of the predicted and measured energy consumption for moving straight and spinning are shown in Figure 6. The predicted values include the total energy consumption and the sensor, computer, and motor breakdown. The accuracy is calculated as the absolute error divided by the measured value. We can find that the prediction model has an accuracy higher than 90% for both moving straight and spinning cases, while the case where the spinning RPM is 1000 achieves 97.8%.

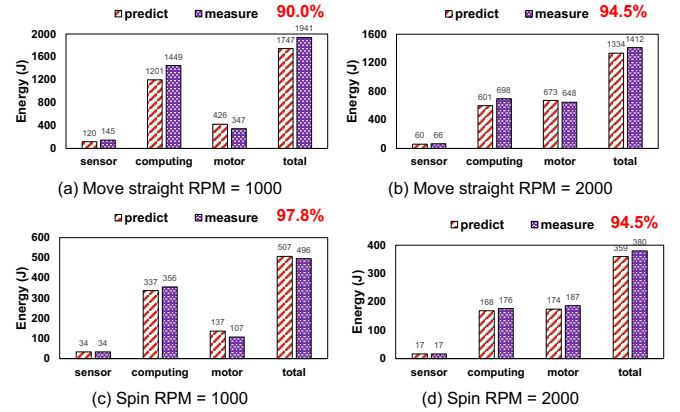


Fig. 6. The predicted and measured end-to-end energy consumption results.

D. Energy Efficient Path Selection

With an accurate end-to-end energy model, we can evaluate the selection of paths for an obstacle in total energy consumption. In general, several factors affect the calculation of the total energy consumption, including the obstacle's size (width M , length N), moving straight RPM, spinning RPM, robot size, etc. In this paper, we jointly consider the impact of the obstacle's size and moving straight/spinning RPM to select paths. By default, we consider only one obstacle and set the distance between A and B as 10 meters, while the robot size is set as 0.5m x 0.5m.

1) *Fixed obstacle size:* First, we fix the obstacle size to evaluate the impact of moving straight RPM and spinning RPM. The width M and length N of the obstacle are set as $L/4$, which is 2.5m. The straight moving RPM and spinning RPM are within the range of 1000 to 2000. We sample moving straight RPM and spinning RPM in this range for

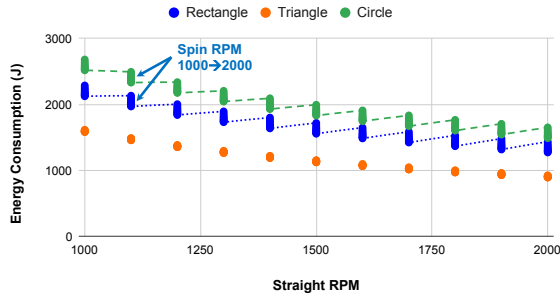


Fig. 7. The energy consumption of Donkey with different moving straight and spinning RPMs.

every 100. By taking the value of L , M , N , and RPM into Equation 7, we can get the relationship between end-to-end energy consumption and RPM of moving straight and spinning, as shown in Figure 7. The X-axis is the straight RPM, and Y-axis is the total energy consumption. For each straight RPM, the scatter points show the energy consumption for different spinning RPMs. We can find that the triangle path always has the lowest energy consumption regardless of the value of moving straight and spinning RPM. The increased straight RPM helps decrease the energy consumption for all three cases. Besides, the increase of spin RPM also helps to decrease the energy consumption in both rectangle and circle paths. When the straight RPM is 1000, the circle path has more energy consumption than the triangle path, and the difference decreases with the increase of the straight RPM. When straight RPM equals 2000, several points of rectangle and circle path overlap.

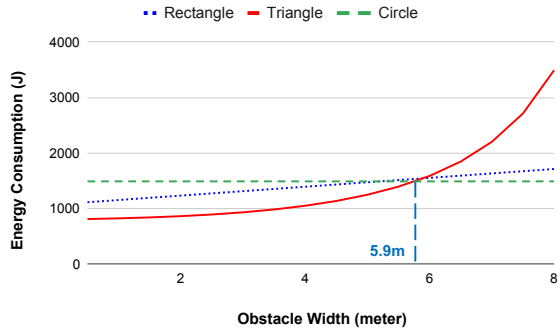


Fig. 8. The energy consumption of Donkey with different obstacle width.

2) *Fixed RPM*: To show how the obstacle's size affects the energy consumption of each path, we fix the move straight and spin RPM as 2000 and change the size of the obstacles. To simplify the problem, we assume the obstacle is a square while the size M changes from d to L . We sample M in this range by adding 0.05m for each step. The results of the energy consumption and obstacle width are shown in Figure 8. When the size of the obstacle is small, the triangle path shows the lowest energy consumption than the rectangle and circle paths. When the obstacle size is zero, the triangle path becomes the direct path between points A and B. However, when the obstacle's size increases, the energy consumption of the triangle path increases much

TABLE II

THE COMPARISON OF THE BEST AND BASELINE PATH UNDER DIFFERENT STRAIGHT RPMs.

Straight RPM	Best (J)	Baseline (J)	Reduction (%)
1000	1581	2517	37.2
1200	1351	2174	37.9
1400	1186	1929	38.5
1600	1063	1745	39.1
1800	967	1602	39.6
2000	890	1487	40.2

TABLE III

THE COMPARISON OF THE BEST AND BASELINE PATH UNDER DIFFERENT OBSTACLE WIDTHS.

Obstacle Width (m)	Best (J)	Baseline (J)	Reduction (%)
1.0	821	1487	44.8
2.0	860	1487	42.2
3.0	929	1487	37.5
4.0	1047	1487	29.6
5.0	1246	1487	16.2

faster than that of the circle and rectangle path. The triangle path consumes the most energy when obstacle width M becomes larger than 5.9m. The bigger the obstacle is, the more significant the energy consumption of the triangle path is different from other paths. In contrast, the circle path shows the lowest energy consumption when the obstacle width exceeds 5.9m.

3) *Energy Saving*: With the energy prediction model, the AMR can select the most energy-efficient path with corresponding system configurations. According to the path planning in the ROS Navigation stack [36], a local planner with a dynamic window approach is expected to choose the circle path [37], [38]. Therefore, we set the baseline as the circle path and compared it with the best path, which has the lowest energy consumption. We compare the best case with the baseline to show the energy-saving performance. Tables II and III compare the best and the baseline under different RPM and obstacle widths. With the energy prediction and path models, the best path shows up to 44.8% of energy reduction than the baseline.

V. CONCLUSIONS

Optimizing energy consumption has become a key challenge in the broad deployment of AMRs. Existing solutions fail to decrease the AMR's energy consumption without an end-to-end power profiling of the AMR. In this paper, we propose a complete energy prediction model for AMR. We evaluate the model on a customized AMR called Donkey for end-to-end power profiling. Based on the evaluation results, the prediction model achieves higher than 90% accuracy. Finally, we apply the energy prediction model to an obstacle avoidance problem and provide AMRs guidance in selecting the path that will consume less energy to achieve. The best path shows up to 44.8% of energy reduction compared with the baseline.

ACKNOWLEDGMENT

This work was partly supported by the U.S. National Science Foundation under Grant IIS-1724227.

REFERENCES

- [1] D. Avola, G. L. Foresti, L. Cinque, C. Massaroni, G. Vitale, and L. Lombardi, "A multipurpose autonomous robot for target recognition in unknown environments," in *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, July 2016, pp. 766–771.
- [2] H. Durmuş, E. O. Güneş, M. Kırıcı, and B. B. Üstündağ, "The design of general purpose autonomous agricultural mobile-robot: "AGROBOT";" in *2015 Fourth International Conference on Agro-Geoinformatics (Agro-geoinformatics)*, July 2015, pp. 49–53.
- [3] J. Lenchner, C. Isci, J. O. Kephart, C. Mansley, J. Connell, and S. McIntosh, "Towards data center self-diagnosis using a mobile robot," in *Proceedings of the 8th ACM International Conference on Autonomous Computing*, ser. ICAC '11. New York, NY, USA: ACM, 2011, pp. 81–90. [Online]. Available: <http://doi.acm.org/10.1145/1998582.1998597>
- [4] L. Liu, X. Zhang, Q. Zhang, A. Weinert, Y. Wang, and W. Shi, "AutoVAPS: an iot-enabled public safety service on vehicles," in *Proceedings of the Fourth Workshop on International Science of Smart City Operations and Platforms Engineering*. ACM, 2019, pp. 41–47.
- [5] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 5. IEEE, 2004, pp. 4344–4349.
- [6] S. Dogru and L. Marques, "Energy efficient coverage path planning for autonomous mobile robots on 3D terrain," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, April 2015, pp. 118–123.
- [7] J. A. Broderick, D. M. Tilbury, and E. M. Atkins, "Optimal coverage trajectories for a UGV with tradeoffs for energy and time," *Autonomous Robots*, vol. 36, no. 3, pp. 257–271, Mar 2014. [Online]. Available: <https://doi.org/10.1007/s10514-013-9348-x>
- [8] C. Henkel, A. Bubeck, and W. Xu, "Energy efficient dynamic window approach for local path planning in mobile service robotics**this work was conducted at the University of Auckland, Auckland, New Zealand," *IFAC-PapersOnLine*, vol. 49, no. 15, pp. 32 – 37, 2016, 9th IFAC Symposium on Intelligent Autonomous Vehicles IAV 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405896316308813>
- [9] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer *et al.*, "Autonomous driving in urban environments: Boss and the urban challenge," *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.
- [10] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 163–168.
- [11] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang, and J. Mars, "The architectural implications of autonomous driving: Constraints and acceleration," in *ACM SIGPLAN Notices*, vol. 53, no. 2. ACM, 2018, pp. 751–766.
- [12] L. Liu, J. Chen, M. Brocanelli, and W. Shi, "E2M: an energy-efficient middleware for computer vision applications on autonomous mobile robots," in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019, pp. 59–73.
- [13] Y. Wang, L. Liu, X. Zhang, and W. Shi, "HydraOne: an indoor experimental research and education platform for CAVs," in *Proceedings of the 2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19)*. RENTON, WA: USENIX Association, 2019.
- [14] S. Yu and C. S. G. Lee, "Lifetime maximization in mobile sensor networks with energy harvesting," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 5911–5916.
- [15] C. Anagnostopoulos, S. Hadjiefthymiades, and K. Kolomvatsos, "Accurate, dynamic, and distributed localization of phenomena for mobile sensor networks," *ACM Trans. Sen. Netw.*, vol. 12, no. 2, pp. 9:1–9:59, Apr. 2016. [Online]. Available: <http://doi.acm.org/10.1145/2882966>
- [16] Y. Wang, W. Peng, and Y. Tseng, "Energy-balanced dispatch of mobile sensors in a hybrid wireless sensor network," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 12, pp. 1836–1850, Dec 2010.
- [17] B. Kannan, V. Marmol, J. Bourne, and M. B. Dias, "The autonomous recharging problem: Formulation and a market-based solution," in *2013 IEEE International Conference on Robotics and Automation*, May 2013, pp. 3503–3510.
- [18] M. Rappaport and C. Bettstetter, "Coordinated recharging of mobile robots during exploration," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 6809–6816.
- [19] N. Kamra, T. K. S. Kumar, and N. Ayanian, "Combinatorial problems in multirobot battery exchange systems," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 852–862, April 2018.
- [20] F. Sempé, A. Muñoz, and A. Drogoul, "Autonomous robots sharing a charging station with no communication: a case study," in *Distributed Autonomous Robotic Systems 5*, H. Asama, T. Arai, T. Fukuda, and T. Hasegawa, Eds. Tokyo: Springer Japan, 2002, pp. 91–100.
- [21] F. Michaud and E. Robichaud, "Sharing charging stations for long-term activity of autonomous robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, Sep. 2002, pp. 2746–2751 vol.3.
- [22] D.-H. Lee, "Resource-based task allocation for multi-robot systems," *Robotics and Autonomous Systems*, vol. 103, pp. 151 – 161, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S092188901730310X>
- [23] D. Lee, S. A. Zaheer, and J. Kim, "A resource-oriented, decentralized auction algorithm for multirobot task allocation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1469–1481, Oct 2015.
- [24] T. F. Roos and M. R. Emami, "A framework for autonomous heterogeneous robot teams," in *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, Nov 2018, pp. 868–874.
- [25] F. de Lucca Siqueira, P. Della Mea Plentz, and E. R. De Pieri, "A fuzzy approach to the autonomous recharging problem for mobile robots," in *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, Aug 2016, pp. 1065–1070.
- [26] X. Zhang, Y. Wang, S. Lu, L. Liu, L. Xu, and W. Shi, "OpenEI: An open framework for edge intelligence," *arXiv preprint arXiv:1906.01864*, 2019.
- [27] T.-J. Yang, Y.-H. Chen, and V. Sze, "Designing energy-efficient convolutional neural networks using energy-aware pruning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5687–5695.
- [28] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," *Advances in neural information processing systems*, vol. 28, 2015.
- [29] A. Garcia-Garcia, J. C. Saez, and M. Prieto-Matias, "Contention-aware fair scheduling for asymmetric single-isa multicore systems," *IEEE Transactions on Computers*, vol. 67, no. 12, pp. 1703–1719, 2018.
- [30] J. Feliu, J. Sahuquillo, S. Petit, and J. Duato, "Perf&fair: A progress-aware scheduler to enhance performance and fairness in smt multi-cores," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 905–911, 2016.
- [31] R. Jejurikar and R. Gupta, "Optimized slowdown in real-time task systems," *IEEE Transactions on Computers*, vol. 55, no. 12, pp. 1588–1598, 2006.
- [32] "Method for sampling current of brushless direct current motor," 2011, <https://patents.google.com/patent/CN102820840A/en>.
- [33] "JetPack SDK 4.4 DP Archive," 2021, <https://developer.nvidia.com/jetpack-sdk-44-dp-archive>.
- [34] "NVIDIA L4T ML," 2021, <https://ngc.nvidia.com/catalog/containers/nvidia:l4t-ml>.
- [35] "YOLO ROS: Real-Time Object Detection for ROS," 2021, https://github.com/leggedrobotics/darknet_ros.
- [36] "move_base," http://wiki.ros.org/move_base.
- [37] "global_planner," http://wiki.ros.org/global_planner.
- [38] "dwa_local_planner," http://wiki.ros.org/dwa_local_planner.