Robotic Wireless Energy Transfer in Dynamic Environments: System Design and Experimental Validation

Shuai Wang, Ruihua Han, Yuncong Hong, Qi Hao, Miaowen Wen, Leila Musavian, Shahid Mumtaz, and Derrick Wing Kwan Ng

The authors proposes a hardware-in-the-loop joint optimization framework that offers three distinctive features: efficient model updates and re-optimization based on the last-round experimental data; iterative refinement of the anchor list for adaptation to different environments; and verification of algorithms in a high-fidelity Gazebo simulator and a multi-robot testbed.

Abstract

Wireless energy transfer (WET) is a ground-breaking technology for cutting the last wire between mobile sensors and power grids in smart cities. However, WET only offers effective transmission of energy over a short distance. Robotic WET is an emerging paradigm that mounts the energy transmitter on a mobile robot and navigates the robot through different regions in a large area to charge remote energy harvesters. However, it is challenging to determine the robotic charging strategy in an unknown and dynamic environment due to the uncertainty of obstacles. This article proposes a hardware-in-the-loop joint optimization framework that offers three distinctive features: efficient model updates and re-optimization based on the last-round experimental data; iterative refinement of the anchor list for adaptation to different environments; and verification of algorithms in a high-fidelity Gazebo simulator and a multi-robot testbed. Experimental results show that the proposed framework significantly saves WET mission completion time while satisfying energy harvesting and collision avoidance constraints.

Introduction

Powering massive Internet of Things (IoT) devices is a fundamental issue to realize intelligent monitoring, detection, manufacturing, and control in future smart cities. However, it has always been regarded as a great challenge due to the limited size, vast volume, and sporadic nature of IoT devices. Recently, wireless energy transfer (WET) has been considered as a viable solution that deploys energy harvesters (EHs) on IoT devices such that the received radio frequency (RF) signals can be converted into electrical energies [1]. The harvested energy can be used for subsequent uplink communication with wireless powered communication technology [2]. Compared to conventional energy-harvesting technologies such as solar, thermal, vibration, and magnetic resonance coupling, the advantages of WET are three-fold:

1. It involves no wire, no contact, and fewer batteries, and represents a controllable energy supply [1].
2. The same RF circuitry for wireless communications can be reutilized for WET [3].
3. RF signals facilitate one-to-many charging due to the broadcast nature of the wireless medium [3].

Current WET products (e.g., Powercast) only support short-range energy transmissions. Robotic WET [4–7] emerges as a promising solution, which mounts the energy transmitter on a mobile robot and navigates the robot through different regions in a large area in order to approach different EHs at different time slots. Compared to unmanned aerial vehicle (UAV) WET [8, 9], ground robots do not consume any propulsion energy to maintain stable hovering. For instance, the motion power of a Turtlebot is 9.3 W (i.e., 7 hours of operation time with a 14.8 V 4400 mAh battery), while the propulsion power of a UAV is above 100 W [8, 9]. However, ground robots need to face the complex collision avoidance problem on the ground, as illustrated in Fig. 1. Therefore, UAV WET is suitable for time-sensitive applications, while robotic WET is suitable for energy-sensitive applications. Existing algorithms for robotic WET can be categorized into two types: global planning algorithms [4–6] and local planning algorithms [10–12]. Global planning algorithms determine the anchor points, routes, and resources (e.g., charging time and beam directions at each anchor point), while local planning algorithms periodically adjust the route for collision avoidance in a dynamic environment. These global and local planning algorithms have been studied separately for robotic WET systems. Therefore, it is necessary to integrate both for more efficient robotic charging, i.e., achieving smaller mission completion time while satisfying the energy harvesting and collision avoidance constraints.

Generally, it is challenging to determine the robotic charging strategy in an unknown and dynamic environment due to the uncertainty of obstacles. First, effective global and local planning algorithms are based on accurate mathematical models (e.g., robot motion time model). But parameters (e.g., distance matrix) in these models could be inaccurate in dynamic environments. Second, joint optimization of anchors, routes, and resources is needed to adapt the planned

Shuai Wang is with the Shenzhen Institute of Advanced Technology, China; Ruihua Han, Yuncong Hong, and Qi Hao (corresponding author) are with the Southern University of Science and Technology, China; Miaowen Wen is with the South China University of Technology, China; Leila Musavian is with the University of Essex, UK; Shahid Mumtaz is with the Instituto de Telecomunicaçaões, Portugal; Derrick Wing Kwan Ng is with the University of New South Wales, Australia.

route to different environments, which can be computationally expensive. Finally, simulators should support close-to-reality features to verify the robustness of algorithms against practical uncertainties.

This article provides three main contributions to address the above challenges. Specifically, a hardware-in-the-loop (HIL) robotic WET system design is proposed, which allows efficient model updates and re-optimization based on the last-round experimental data. Furthermore, a joint optimization framework based on the K-Chebychev density-based spatial clustering of applications with noise (DBSCAN) and the successive local search is proposed to adjust the anchor list in different environments. Finally, all the algorithms are implemented and tested in a high-fidelity Gazebo simulator and a multi-robot testbed. The simulator and the testbed together form a digital twin cyber-physical platform. Extensive results based on the digital twin platform are provided to verify the effectiveness of the proposed HIL joint optimization framework. In the following, we first review the conventional robotic WET technologies before introducing our designs and implementations.

Conventional Robotic Wireless Energy Transfer

Robotic WET minimizes the total mission completion time (i.e., the sum of moving and charging time) by planning the anchor positions, routes, charging time, and energy beams, while satisfying the energy harvesting requirements at all IoT devices [4–6]. To achieve this goal, conventional robotic WET algorithms consist of four sequential steps:

- **Step 1:** Modeling. Measurements are usually stored as look-up tables, which cannot be directly used for subsequent optimization. We can transform the measurements into mathematical models, and the parameters in these models are obtained by curve fitting.
- **Step 2:** Anchor point generation. This divisde the EHs into multiple clusters and assigns an anchor for each cluster where the robot will stop for a while to charge the surrounding EHs.
- **Step 3:** Route planning. The robot needs to visit the anchor points generated from Step 2. Different routes (i.e., sequences of anchor points) result in different motion time.
- **Step 4:** Resource allocation. The robot needs to determine the amount of time spent at each anchor point based on the models, anchors, and routes in Steps 1–3. To fully exploit the degrees of freedom in the spatial domain, pencil-like energy-focusing beams can be shaped and steered toward the target devices.

In the following, we review the existing methods involved in each step.

Mathematical Models

To guarantee sufficient harvested energy at all nodes, the robot may visit and charge the IoT devices one by one. However, this requires an exceedingly long time spent in motion [4]. A more time-efficient strategy is to let the robot simultaneously charge multiple devices at only a few positions. These positions are called anchor points. The robot motion time model measures

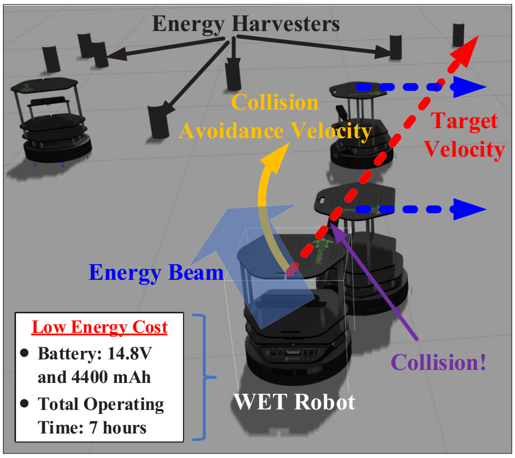

FIGURE 1. A robotic WET system with dynamic obstacles.

Robotic WET minimizes the total mission completion time by planning the anchor positions, routes, charging time, and energy beams, while satisfying the energy harvesting requirements at all IoT devices.

the moving time from any anchor to another, which is used for subsequent anchor generation and route planning later. This model is described by a directed graph, where the vertices represent the anchor points and the directed edges represent the feasible routes. The route length is represented by a distance matrix, with the element at the mth row and jth column representing the distance from anchor m to anchor j. The motion time is calculated as distance over velocity.

To implement resource allocation, it is necessary to model the energy loss of robotic WET. In general, wireless energies experience attenuations twice before being converted into electrical energies: signal propagation in the air and RF-to-current conversion in the EH.

- The wireless channel model is a function of the received RF power with respect to the transmitted RF power [9]. It can be categorized into statistical, deterministic, and quasi-deterministic methods. Statistical models are low-complexity but inaccurate parameterized models. Deterministic models (i.e., ray tracing) launch rays in various directions and traces the propagation, refection, scattering, and absorbtion using the electromagnetic field. However, it requires exceedingly long modeling time. Most industrial standards adopt quasi-deterministic channel models, which obtain hyper-parameters for different scenarios using ray tracing and generate wireless channels using some statistical method.
- The energy harvesting model is a function of the harvested power with respect to the received RF power [1]. This function is nonlinear, since an energy harvesting circuit consists of nonlinear elements such as diodes. According to [14], the energy harvesting model should satisfy the monotonicity, sensitivity, nonlinearity, and saturation properties. Energy harvesting models can be categorized into piecewise linear model, logistic model, and sensitivity-based logistic model [1]. The former two models are low-complexity but only support partial properties. The last model supports all properties but is a nonconvex function that may cause additional computational costs during the subsequent steps [14].

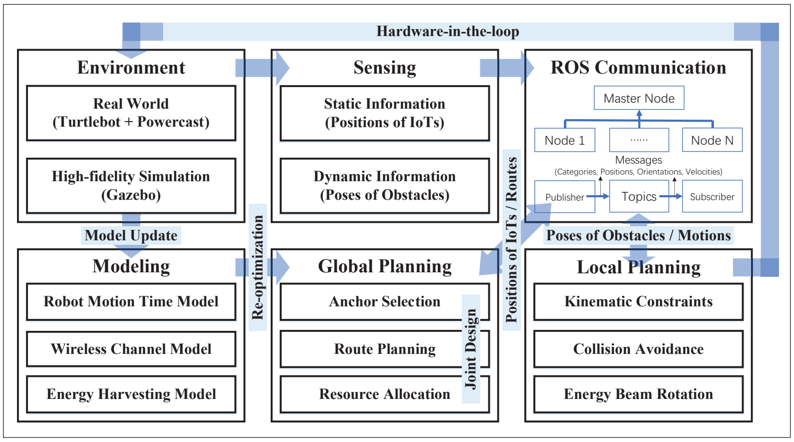

FIGURE 2. Hardware-in-the-loop joint optimization framework for robotic wireless energy transfer.

Anchor Point Generation

Anchor point generation can be viewed as a spatial clustering problem [6, 13]. In general, clustering for robotic WET can be categorized into distance-based and density-based methods. In distance-based methods (e.g., *k*-means clustering), a distance metric is used to determine the similarity between EHs. The method produces compact and spherical clusters around a set of centroids that are very sensitive to outliers. On the other hand, density-based methods (e.g., DBSCAN) adopt a density threshold to distinguish the important EHs from the outliers. As such, it can deal with unbalanced clusters and outliers pretty well. Different from *k*-means clustering, DBSCAN generates arbitary shapes, which provide higher flexibility than *k*-means for anchor point generation.

Route Planning

With the positions of anchor points, the next step is to determine the visiting sequence of these points via route planning. The route planning problem is a constrained discrete optimization problem [4, 5], where the constraints guarantee:

1. The robot returns to the starting point.
2. The robot visits the selected vertices.
3. The planned path is connected.

Conventionally, tree search algorithms such as branch-and-bound (B&B) can be adopted for systematically pruning ineffective solutions, leading to significant reduction of the computational complexity compared to exhaustive search while guaranteeing optimality. However, its complexity is still high since the solution space grows exponentially with the number of anchor points. Imitation learning emerges as a promising solution to solve the large-scale discrete optimization problem. The core idea is to treat route planning as a classification problem and adopt a deep neural network to mimic the behavior of tree search algorithms.

Resource Allocation

When the WET robot moves along the optimized route, the remaining factors that impact the system performance are the resources allocated at each anchor point. Common resources include the charging time (time-domain) and the energy beams (angle-domain) [14]. For charging time allocation, different anchor points along the route should be jointly considered. This is because the charging time at the current location might have a long-term impact on future locations, as the harvested energy at IoT devices can be stored in batteries. For beam allocation, there are two types: multi-antenna-based and directional-antenna-based. In particular, multi-antenna energy beamforming adjusts the power and phase at each antenna to form the desired beam directions. In this case, the harvested energy is the multiplication of the charging time and the harvested power, which introduces non-convexity to the resource allocation problem. Advanced optimization tools such as majorization minimization [14] (which iterates between solving and finding a surrogate problem of the primal problem) can be used to obtain suboptimal solutions. On the other hand, directional-antenna beamforming has a sector shape, and the robot needs to rotate itself to alter the beam direction. Using exhaustive search, the beam direction can be selected from a finite codebook with pre-designed beam patterns.

Proposed Hardware-in-the-Loop Joint Optimization Framework

System Description

Existing schemes assume that the WET robot has perfect knowledge of the environment, and the planned route can be directly adopted. In practice, however, the WET robot needs to perform periodic sensing and avoid collisions with the surrounding obstacles. This implies that the WET robot must adjust its route online, leading to a mismatch between the planned and actual routes. To mitigate the mismatch, the HIL joint optimization framework is proposed in Fig. 2, which supports interactions among experiments, simulations, modeling, and planning based on the robot operating system (ROS) communication. The goal is to allow efficient model updates and re-optimization based on the last-round experimental data. For example, the initial motion time between two anchors is estimated as straight line distance over velocity. But if the robot takes a detour due to the obstacles between the two anchors, the actual motion time would be longer than the estimated time. With feedback of the actual motion time obtained from experiments, the motion time model becomes closer to reality, thereby improving the system performance.

Specifically, the system in Fig. 2 operates in an iterative manner. Each iteration consists of the following operations:

- **Offline Planning Stage.** First, the models are fitted to the measurements obtained from the last-round experimental validation. Measurements may include the amount of motion time when the robot travels from one anchor point to another, the channel fading vs. the charging distance, and the harvested power vs. the incident power. Then the robot explores the environment to obtain the current positions of EHs and building a static map based on simultaneous localization and mapping (SLAM). Finally, with the fitted models and positions of EHs, the global planner generates the charging policy (including route, time, beam) offline.
- **Online Validation Stage.** First, the robot fol-

lows the target route produced in the offline stage. Then the robot performs object detection periodically using onboard sensors. The sensor outputs (i.e., ego-positions, ego-velocities, and poses of other obstacles) are shared to the industrial computer via ROS communication for local planning. Finally, the local planner adjusts the route constantly to avoid collisions. This is achieved by computing a velocity vector that is closest to a target route provided by the global planner, while taking into account the robot kinematics and the chances of collisions. The online stage is implemented in real environments or close-to-reality simulators, which would generate new measurements for next-round offline planning.

Below, we provide details of the new modules and features supported by the framework.

Joint Optimization for Global Planning

As shown in the lower middle of Fig. 2, the proposed framework adopts a joint optimization framework for global planning, which has two major differences compared to conventional algorithms. First, we adopt the K-Chebychev DBSCAN method instead of *k*-means clustering for anchor point generation. This is because *k*-means clustering can minimize the sum distance between the anchor point and its associated EHs. However, for WET systems, the key factor affecting the system performance is the distance from each anchor to the farthest EH in its cluster [13]. Therefore, after executing the DBSCAN algorithm, we re-compute the anchors' positions as Chebyshev centers via min-max optimization [13]. Furthermore, if the transmitter adopts a directional antenna, the coverage of an energy beam should also be considered (e.g., by adding an energy beamforming constraint) [1] to the min-max optimization problem [13]. Second, we add an anchor point selection module to prune ineffective positions and navigate the robots away from dense traffic when possible. This is realized via joint optimization based on the iterative local search framework [5] at the global planning layer. The algorithm iterates between selecting a subset of anchor points among all the candidates and executing the route planning and resource allocation algorithms. As such, the anchor positions are no longer fixed as in conventional robotic WET, but are adaptively optimized in different environments.

Local Planning for Collision Avoidance

Conventional robotic [4–6] or UAV-aided [8, 9] WET schemes ignore the local planning problem. In practice, however, the robot must avoid collisions with any object residing in the environment, while making progress toward the next anchor point for wireless charging. This is realized by the local planner shown in the lower right of Fig. 2. The local navigation algorithms depend on the mobility of obstacles. If the observed obstacles are static, the key is to compute a velocity vector for the robot while taking into account the robot kinematics and dynamics. If the obstacles are moving, the above approach can be applied by extrapolations of the observed velocities to estimate the future positions of obstacles. The problem of collision avoidance becomes challenging when the obstacles are not simply moving at a constant speed, but are also intelligent decision making agents that try to avoid collisions as well. This is because each robot can only estimate the positions and velocities of other agents but cannot know their reactions and intents. In this case, each robot needs to select a velocity outside the reciprocal velocity obstacle (RVO) region induced by other agents [12]. It has been proved that the RVO provides a sufficient and necessary condition for a robot to avoid collisions with an obstacle moving at a known velocity but unknown intention [10]. Note that the RVO scheme can be replaced by deep reinforcement learning (DRL), which can achieve higher efficiency under a properly tuned reward function and a specific environment. DRL adopts a policy network to map the environment state into the robot motions and is trained to converge to the actions with the maximum cumulative reward over several episodes. However, the DRL-based approaches are computationally expensive and sensitive to sensor noises.

The problem of collision avoidance becomes challenging when the obstacles are not simply moving at a constant speed, but are also intelligent decision making agents that try to avoid collisions as well. This is because each robot can only estimate the positions and velocities of other agents but cannot know their reactions and intents.

Hardware-in-the-Loop for Re-optimization

The key feature of HIL is that algorithms are tested in close-to-reality simulators or real-world environments in order to make re-optimization possible [15]. For simulation, we adopt Gazebo [10, 12], a high-fidelity robotic evaluation platform that adopts Open Dynamics Engine for motion generation of the robots and Open Graphics Library Engine for the visualization of the world. On top of the two engines, Gazebo realizes each physical object as a model that is composed of rigid bodies, joints, sensors, and interfaces for client programs to control the model. In our experiment, the WET robot is modeled as a combination of various bodies, where different bodies are assigned different mass, friction, and bounce features. The hinge joints among the bodies of a robot provide the physical mechanism to form kinematic and dynamic relationships such as rotations. The EHs are modeled as cylinders. The Gazebo implementation is shown in Fig. 1.

As for the experiment, we adopt the Turtlebot2 platform, equipped with an ultra-wideband (UWB) sensor that produces the echo-location of the robot. The onboard industrial computer controls the chassis as well as the direction of Powercast transmitter TX91503. A mobile battery powers the chassis, sensor, computer, and transmitter. Communication among these devices is implemented via ROS, which is a distributed communication framework supporting integrative and heterogenous systems [12]. In ROS, all processes that perform computations are implemented as nodes, where a master node controls the global system and slave nodes manage programs on each device. ROS offers a message passing interface that provides inter-process communication among these nodes via topics. A node sends a message by publishing it to a given topic, and any node interested in the message can subscribe to the associated topic.

Experimental Validation

We consider the task of 1 WET robot charging 20 EHs. The transmit power of Powercast TX91501 is 3 W at 915 MHz. The transmit antenna is direc-

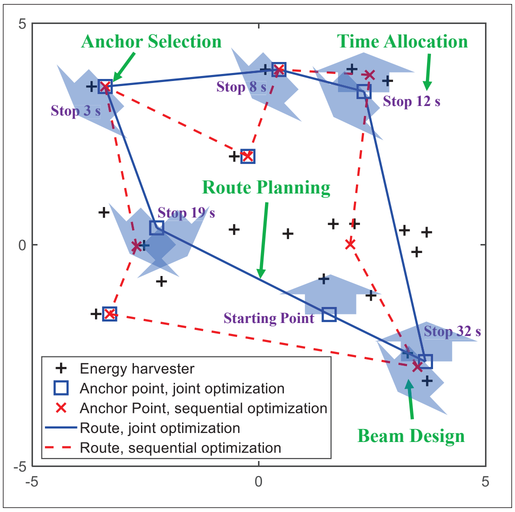

FIGURE 3. Comparison between the proposed joint optimization and conventional sequential optimization algorithms.

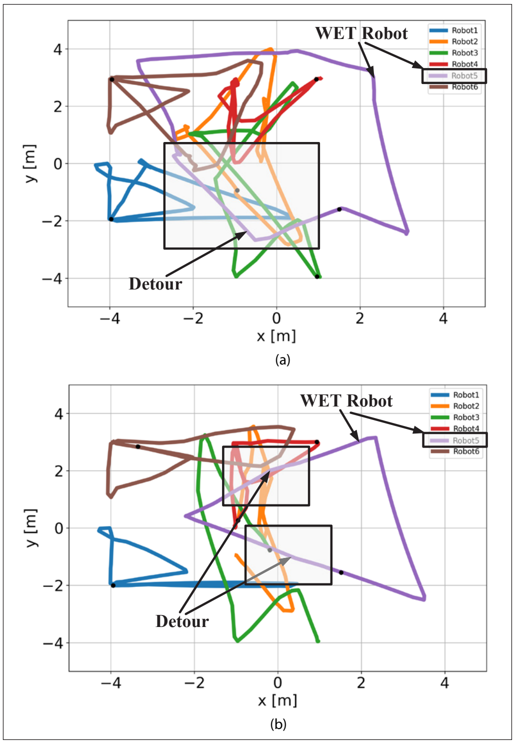

FIGURE 4. Routes in Gazebo when 1 WET robot charges 20 EHs while avoiding collision with 5 non-cooperative robots: a) before HIL re-optimization; b) after HIL re-optimization.

tional, and its beamwidth is 130°. The codebook contains 3 beam patterns (i.e., –65° to 65°, 55° to 185°, and 175° to 305°). The receiver gain at Powercast P2110 is 6 dBi. The maximum velocity of the WET robot is 0.2 m/s. The minimum energy to be harvested at each IoT device is 20 mJ. The path loss at 1 m is –31.6284 dB, and the path loss exponent is 1.73 according to the In-H channel model specified in 3rd Generation Partnership Project (3GPP) TR 38.901. The sensitivity-based energy harvesting model in [14] is adopted, with the parameters specified in Fig. 2 of [14].

First, Fig. 3 compares the results of joint optimization and sequential optimization in the perfect case. It can be seen that the anchor points (i.e., blue squares) generated by joint optimization are slightly shifted from the centroids (i.e., red crosses) in order to facilitate the beam design. Moreover, joint optimization only selects a subset of anchors from the candidates, allowing the robot to charge EHs in other clusters. This is in contrast to the conventional sequential method, where the robot needs to visit all the anchors, leading to excessive motion time. The above results corroborate discussions from earlier.

Then the task in Fig. 3 is implemented in Gazebo, with five noncooperative robots moving randomly in the same environment. The associated result is shown in Fig. 4a. It can be seen that the blue route in Fig. 3 and the purple route in Fig. 4a share high similarity. However, in Fig. 4a, the WET robot takes a detour when returning to the starting point, while the robot takes a straight way in Fig. 3. This is because the robot comes across other robots in the way in Gazebo and needs to find a trade-off between avoiding collision and reaching the next anchor point. Moreover, the robot spends additional time to rotate itself in Gazebo. Consequently, the motion time in Gazebo is 147 s, which is 50 percent longer than that in the perfect case (i.e., 110.369 s). The motion time in the real-world experiment is 150 s, meaning that the Gazebo simulator and the real-world testbed can together form a digital twin platform.

The route evolves into a triangle after HIL re-optimization. This implies that the re-optimized design automatically chooses to reduce the robot's movements and navigates the robot away from the dense traffic area. The associated result is shown in Fig. 4b. Compared to the result in Fig. 4a, the route evolves from a polygon to a triangle, whose route length is significantly smaller. This implies that the re-optimized design automatically chooses to reduce the robot's movements and navigates the robot away from the "dense traffic" area (i.e., the lower left of Fig. 4).

Finally, to verify the effectiveness of the proposed framework, we consider the same task and compare the amount of WET mission completion time for different schemes. The major findings are summarized below:

- The scheme with a fixed energy transmitter cannot accomplish the charging task within a reasonable amount of time, since the transmit power of TX91503 is only 3 W, and remote EHs can only receive μW-level power, which is below the sensitivity threshold.
- If the robot does not perform anchor point selection and reaches all the anchors, although the charging time can be reduced for each IoT, the total time consumption increases. This is because moving costs extra time, and there is a trade-off between spending time on moving vs. on charging in the robotic WET system.
- By jointly optimizing the anchor, route, time, and beam, the total amount of time is reduced to 184.3 s. Note that the task completion time (including flying and charging) for the UAV WET scheme is only 78.38 s. This is because the UAV does not need to avoid obstacles in the sky, and it adopts an air speed of 5 m/s. However, the UAV energy consumption is 78.38 × 100 = 7838 J, which is significantly larger than the robot energy consumption (i.e., 184.3 × 9.3 =

The route evolves into a triangle after HIL re-optimization. This implies that the re-optimized design automatically chooses to reduce the robot's movements and navigates the robot away from the dense traffic area.

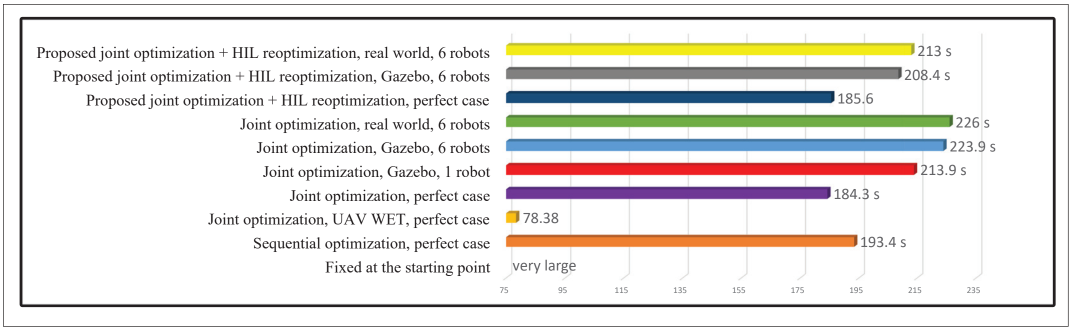

FIGURE 5. Comparison of WET mission completion time for different schemes in the perfect case, the Gazebo, and the real world.

When multiple robots serve as WET chargers, exploiting their interactions and collabo-rations can significantly enhance the system performance. Moreover, if the environment is absolutely unknown, integrated sensing and communication (ISAC) enables multiple robots to build and merge their local maps simultaneously.

1714 J).

- With the same joint optimization scheme, the time spent in Gazebo is longer than that in the perfect case. This is because in Gazebo, the robot needs to:
  1. Gradually rotate itself at transition points
  2. Take detours to avoid collision with EHs
  3. Accelerate/decelerate between two consecutive anchor points.
- Increasing the number of robots from 1 to 6 in Gazebo leads to an obvious increment in the amount of time. This implies that the traffic condition of the environment would have a non-negligible impact on the system performance.
- The task completion time in the real world is slightly longer than that in Gazebo. This is because Gazebo implementation is centralized, while the real-world implementation is distributed, which involves various uncertainties such as position inaccuracies and time asynchronization.
- Compared to joint optimization without HIL, the completion time of HIL joint optimization is slightly longer in the perfect case, but is significantly shorter in the real-world experiment (or Gazebo simulator). This implies that a good theoretical scheme may break down in practice due to the mismatch between models and environments and vice versa. This also demonstrates the significance of the HIL framework, which achieves robust performance in real-world dynamic environments.

Conclusion

This article reviews existing robotic WET algorithms and proposes a new HIL system design framework. By comparing the results in a perfect case, Gazebo, and real world, it is found that the robotic charging time may significantly increase from ideal to practical environments due to robotic dynamics and collision avoidance. It is shown that HIL re-optimization could improve the robustness of robotic WET against practical uncertainties.

When multiple robots serve as WET chargers, exploiting their interactions and collaborations can significantly enhance the system performance. Moreover, if the environment is absolutely unknown, the robot needs to create a global map of the environment, which can be time-consuming. Integrated sensing and communication (ISAC) is a promising technique to accelerate the map merging procedure, as ISAC allows each robot to build and share its local map simultaneously.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62001203, 61773197), the Guangdong Basic and Applied Basic Research Project (No. 2021B1515120067), the Shenzhen Science and Technology Program (No. JCYJ20200109141622964, RCB20200714114956153), the FCT project (Intelligent and Sustainable Aerial-Terrestrial IoT Networks-BATS) PTDC/EEI-TEL/1744/2021. This work was also supported by the UNSW Digital Grid Futures Institute under a cross-disciplinary fund scheme and the Australian Research Council's Discovery Project (DP210102169).

References

[1] B. Clerckx *et al.*, "Fundamentals of Wireless Information and Power Transfer: From RF Energy Harvester Models to Signal and System Designs," *IEEE JSAC*, vol. 37, no. 1, Jan. 2019, pp. 4–33.

[2] J. Hu, Q. Wang, and K. Yang, "Energy Self-Sustainability in Full-Spectrum 6G," *IEEE Wireless Commun.*, vol. 28, no. 1, Feb. 2021, pp. 104–11.

[3] O. L. A. López *et al.*, "Massive Wireless Energy Transfer: Enabling Sustainable IoT Toward 6G Era," *IEEE IoT J.*, vol. 8, no. 11, June 2021, pp. 8816–35.

[4] S. Guo, C. Wang, and Y. Yang, "Joint Mobile Data Gathering and Energy Provisioning in Wireless Rechargeable Sensor Networks," *IEEE Trans. Mobile Comp.*, vol. 13, no. 12, Dec. 2014, pp. 2836–52.

[5] S. Wang, M. Xia, and Y.-C. Wu, "Backscatter Data Collection with Unmanned Ground Vehicle: Mobility Management and Power Allocation," *IEEE Trans. Wireless Commun.*, vol. 18, no. 4, Apr. 2019, pp. 2314–28.

[6] T. Liu *et al.*, "An Effective Multi-Node Charging Scheme for Wireless Rechargeable Sensor Networks," *Proc. IEEE INFOCOM*, Toronto, Canada, July 2020.

[7] A. Tomar, L. Muduli, and P. K. Jana, "A Fuzzy Logic-Based On-Demand Charging Algorithm for Wireless Rechargeable Sensor Networks with Multiple Chargers," *IEEE Trans. Mobile Comp.*, vol. 20, no. 9, Sept. 2021, pp. 2715–27.

[8] J. Xu, Y. Zeng, and R. Zhang, "UAV-Enabled Wireless Power Transfer: Trajectory Design and Energy Optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, Aug. 2018, pp. 5092–5106.

[9] Q. Wu *et al.*, "A Comprehensive Overview on 5G-and-Beyond Networks with UAVs: From Communications to Sensing and Intelligence," *IEEE JSAC*, vol. 39, no. 10, Oct. 2021, pp. 2912–45.

[10] M. G. Mohanan and A. Salgoankar, "A Survey of Robotic Motion Planning in Dynamic Environments," *Robot Auton. Sys.*, vol. 100, Feb. 2018, pp. 171–85.

[11] D. H. Lee *et al.*, "Finite Distribution Estimation-Based Dynamic Window Approach to Reliable Obstacle Avoidance of Mobile Robot," *IEEE Trans. Ind. Electron.*, vol. 68, no. 10, Oct. 2021, pp. 9998–10,006.

[12] R. Han, S. Chen, and Q. Hao, "A Distributed Range-Only Collision Avoidance Approach for Low-Cost Large-Scale Multi-Robot Systems," *Proc. IEEE Int'l. Conf. Intell. Robots Sys.*, Las Vegas, NV, 2020.

[13] O. M. Rosaba *et al.*, "Massive Wireless Energy Transfer with Multiple Power Beacons for Very Large Internet of Things," 2021; https://arxiv.org/pdf/2106.12968.pdf.

[14] S. Wang *et al.*, "Wirelessly Powered Two-Way Communication with Nonlinear Energy Harvesting Model: Rate Regions

Under Fixed and Mobile Relay," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, Dec. 2017, pp. 8190–8204.
[15] M. D. Stefano *et al.*, "A Relative Dynamics Formulation for Hardware-in-the-Loop Simulation of On-Orbit Robotic Missions," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, Apr. 2021, pp. 3569–76.

Biographies

SHUAI WANG (s.wang@siat.ac.cn) received his Ph.D. degree from the University of Hong Kong. He is currently an associate professor at the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences.

RUIHUA HAN (hanruihuaff@gmail.com) is a Ph.D. student at the Southern University of Science and Technology.

YUNCONG HONG (hongyc@mail.sustech.edu.cn) is a Ph.D. student at the Southern University of Science and Technology.

QI HAO (hao.q@sustech.edu.cn) received his Ph.D. degree from Duke University. He is currently an associate professor with the Department of Computer Science and Engineering, Southern University of Science and Technology.

MIAOWEN WEN (eemwwen@scut.edu.cn) received his Ph.D. degree from Peking University. He is currently a full professor with the School of Electronic and Information Engineering, South China University of Technology.

LEILA MUSAVIAN (leila.musavian@essex.ac.uk) received her Ph.D. degree in telecommunications from Kings College London, United Kingdom. She is currently a full professor with the School of Computer Science and Electronic Engineering, University of Essex.

SHAHID MUMTAZ (smumtaz@av.it.pt) received his Ph.D. degree from the University of Aveiro, Portugal. He is currently a senior research scientist at the Instituto de Telecomunicações.

DERRICK WING KWAN NG [F] (w.k.ng@unsw.edu.au) received his Ph.D. degree from the University of British Columbia. He is currently an associate professor with the School of Electrical Engineering and Telecommunications, University of New South Wales.