On designing cyber-physical-social systems with energy-neutrality and real-time capabilities

Mohamed Irfanulla Mohamed Abdulla, Maryline Chetto, Audrey Queudet, Lamia Belouaer

On designing cyber-physical-social systems with energy-neutrality and real-time capabilities

1st Mohamed Irfanulla MOHAMED ABDULLA
E-Cobot and LS2N Laboratory-UMR CNRS 6004
University of Nantes
Nantes, France
mohamed-irfanulla.mohamed-abdulla@ls2n.fr

2nd Maryline CHETTO
LS2N Laboratory-UMR CNRS 6004
University of Nantes
Nantes, France
maryline.chetto@ls2n.fr

3rd Audrey QUEUDET
LS2N Laboratory-UMR CNRS 6004
University of Nantes
Nantes, France
audrey.queudet@ls2n.fr

4th Lamia BELOUAER
E-Cobot
Nantes, France
l.belouaer@e-cobot.com

Abstract—In this paper, Cyber-Physical-Social Systems (CPSS) explicitly refers to the collaborative robots for industry. Battery autonomy is a predominant factor that limits the operating time of a cobot. As a consequence, it becomes essential for it to minimize energy consumption while satisfying all its constraints including real-time ones. This paper reviews the current state-of-the-art relating to approaches for energy efficiency specifically dedicated to cobots. The analysis indicates that energy harvesting technology combined to energy-aware real-time schedulers and energy-efficient path planners could permit to build energy-neutral CPSS.

Index Terms—Cyber-physical-social system, cobot, energy, real-time computing, energy-efficient path planner, energy harvesting, energy-neutral.

I. INTRODUCTION

Industry 4.0 takes Cyber-Physical-Social Systems (CPSS) paradigm to reduce the human complexity challenges [1]. In particular, collaborative robots [2] are designed to cooperate with humans. *Mobile cobots* can drive autonomously and assist people in moving heavy loads and work on unergonomic tasks. In this paper, cobots refer to mobile cobots.

The cobot HUSKY in Fig. 1, developed by E-cobot[1] is considered in our work. The cobot operates in three different modes: (i) teleoperation mode or manual mode, (ii) follower mode, and (iii) autonomous mode.

It can carry loads up to 150 kg and tows up to 1 ton. Such a cobot works on industrial applications including logistic evacuations, order preparations, logistics flow optimizations, supply chains, order deliveries, and disinfectants.

Currently, the cobot operation time is insufficient for industrial working hours as the energy of the cobot is unexplored. Energy-efficiency is still an open challenge for cobots. As the demand for cobots in industries evolves, their working time and performance with respect to the available energy must be maximized. This work concentrates

[1]https://e-cobot.com/en/husky-smart-mobile-cobot-2/

Fig. 1. Cobot HUSKY

in formalizing the energy utilization of cobots and maximizing their operating time with energy-efficient techniques.

This paper provides a review of the state-of-the-art on energy-efficiency in mobile robots as to our knowledge, there is no survey on energy-efficiency for cobots. The paper discusses the limitations of the current techniques and provides the research trend to be implemented on cobots so as to make them energy-efficient systems. By nature, any cobot is a hard real-time system due to multiple computations that should be performed within strict deadlines. Therefore, we introduce real-time scheduling mechanisms which are energy-aware [25],[26],[27] and specifically adapted to energy harvesting systems (i.e., systems able to draw energy from ambient sources). Jointly used with energy-efficient path planners, a cobot may now adopt a deterministic and energy-neutral operation mode with a quasi-perpetual autonomy.

The paper is organized as follows: Section II explains the cobot architecture. Section III investigates the current

energy-efficient techniques in mobile robots. Section IV discusses the limitations of the reviewed techniques with cobotic constraints and the energy model required to produce an energy-efficient system. Section V is associated with the innovative ideas that must be implemented on cobots to comply with both hard real-time and energy constraints so as to satisfy industrial requirements. Finally, the conclusion is given in Section VI.
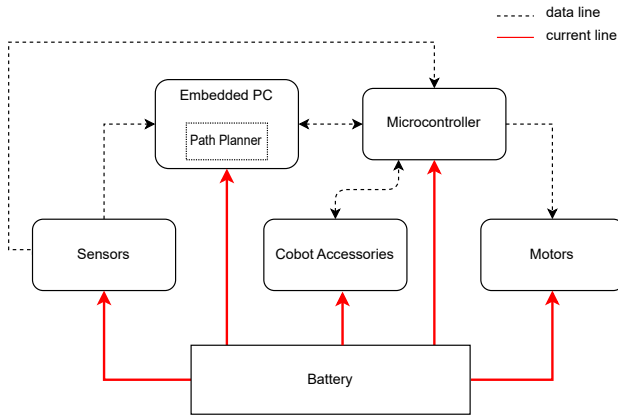
II. TYPICAL COBOT ARCHITECTURE

Fig. 2. Typical architecture of a cobot

Fig. 2 depicts the typical architecture of a cobot. It includes:
1) The *motion unit* which relies on the motors and motor drivers. Usually, the actuator of a mobile robot is a DC motor [13], but cobots can use either AC or DC motors. The cobot considered in this work is a differential drive actuated by brushless AC motors.
2) The *control unit* which is a micro-controller unit working with an embedded operating system to control the motor drivers, to operate cobot accessories, and to read sensor data.
3) The *sensor system* which observes the dynamic environment and determines the orientation of the cobot; the most common sensors are laser scanners and cameras to detect potential obstacles.
4) The *embedded PC* which computes the velocity for the projected trajectory by path planning algorithms. One standard operating system utilized is Linux supporting robotics middlewares like Robot Operating System (ROS)[2], Orca[3], Player Project[4], etc.
5) The *rechargeable battery* which is a vital source to power the cobotic system. The battery State of Charge (SoC) allows to retrieve the energy level of the cobot, which is an essential factor that impacts on operation time.

Battery recharging is a time-consuming task, during which the cobot completely stays at rest. Fast charging system decreases the charging time, but the efficiency and

performance remain unchanged. The most reliable practice is to reduce the recharging cycle by maximizing energy utilization.

III. STATE-OF-THE-ART ON ENERGY-EFFICIENCY

To our knowledge, there is no survey of the recent literature on energy optimization for cobots. It is of practical interest to examine how the energy of mobile robot may vary and may be available, as the rechargeable battery is usually the only energy source. The state-of-the-art on energy-efficient mobile robot systems can be classified into three main techniques:
1) Energy-efficient path planning
2) Component-level power saving
3) Power-aware real-time task scheduling

A. Energy-efficient path planning

Path planning or Motion planning is the process of finding the route and determining the velocity for the robot to travel from start to goal states, while avoiding obstacles. Many studies and approaches consider motion optimization to solve the energy-efficiency problem as motion systems are considered high energy consumers. The method in [4] for energy optimization evaluates the best minimal energy cost for different routes at different velocities. The author implements an energy-efficient motion planning technique on a three-wheeled omni-directional robot. The energy efficiency is calibrated depending on the work area size and the energy required to cover it. The robot energy is calculated with a six-degree polynomial approximation power model for different motions. The proposed simulator validates the energy-efficient path planning and velocity profile for different scenarios saving up to 51% of energy.

Energy optimization is essential for the mobile robot operating in unexplored areas. In [5], the proposed method saves the energy and reduces the travel distance for robot exploration. It involves a robot orientation based target selection and the desirable energy-efficient route by motion planning algorithm. Orientation-based target selection selects the next target depending on the direction of the robot (from the robot's left in a clockwise direction), thereby avoiding the exploration of the repeated area. The motion planning algorithm applies a modified Dijkstra's algorithm that includes location and direction information. It avoids stops and turns to reduce the acceleration and deceleration mechanism of the motor to reduce energy consumption significantly. The orientation-based simulation result shows a robot explores the complete workspace with 38.9% shorter traveling distance and 38.5% less energy consumption than utility-based method.

The optimal motion planning strategy proposed in [6],[7],[8] focuses on energy minimization by treating energy cost function in the A* path planner heuristic function. Robots travel through waypoints of the generated path, and this method optimizes the arrival time and velocity at each waypoint to smoothen the trajectory, which results in optimizing the energy and travel time. The proposed algorithm is validated by simulation and tested on Pioneer 3DX

differential wheel robot shown in Fig. 3. It chooses to move in longest path that consumes less energy than the shortest path with friction zones. In [9], the algorithm outperforms the previous A* path planning approach in both energy and travel time. It produces a smooth trajectory through optimal spline (seventh order polynomial curves) parameters and optimizes the velocity profile. The experiment is done on Robotino three-wheeled omni-directional robot that shows 7.5% again in reducing the energy consumption.
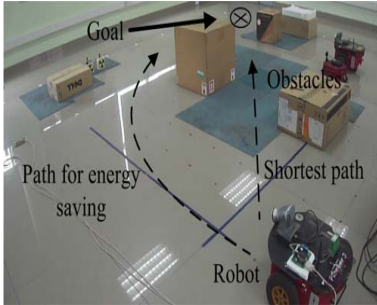

Fig. 3. Pioneer-3DX robot choose energy-efficient path [8]

Offline local path planning is a conventional path planning algorithm for most mobile service robots. Methods proposed in [10] and [11] implement an extended dynamic window local path planning approach that incorporates energy consumption function in the cost function. In contrast, to all the above proposed methods (dealing only with the static environment), this method interprets the dynamic environment. It defines an energy cost function based on the route length and travels with a constant velocity to reach the goal state. Experiments on a four-wheeled omni-directional robot show that the proposed method reduces energy consumption by 9.79% compared to the conventional dynamic window approach (DWA) algorithm. In [11], total energy consumption of the robot performing different motions at high speed is recorded. The results show that idle energy consumption is much higher than the motional energy due to high idle power. Here the trajectory with less idle energy is obtained by considering both high speed and optimized motional power achieved with the special ability of omni-directional robot holonomic mobility.

Simulation and experimental based results of energy-aware path planning (EAPP) for a mobile robot using ROS navigation is formalized in [12]. Experiments are done with an NVIDIA Jetson TX2 embedded PC, which has a built-in power sensor and a Turtlebot 2. EAPP uses Planning Domain Definition Language (PDDL) to formalize the environmental model (i.e., clearance zone to avoid obstacles and high energy zones) and action model (i.e., movement actions and energy-related actions). Fig. 4 (a), shows a map with clearance zone (red area) and high energy zone (blue area); when EAPP chooses the energy-efficient path, the robot traverses in the green path, and in the red path when energy metrics are not concerned. Fig. 4 (b) illustrates the battery discharge when running the standalone ROS navigation stack and the proposed EAPP. The proposed method outperforms with an extension up to 42.8%

of battery life. Although the approach is energy-efficient, the main limitations are computation time, dynamic obstacles avoidance, and energy model of the robot.
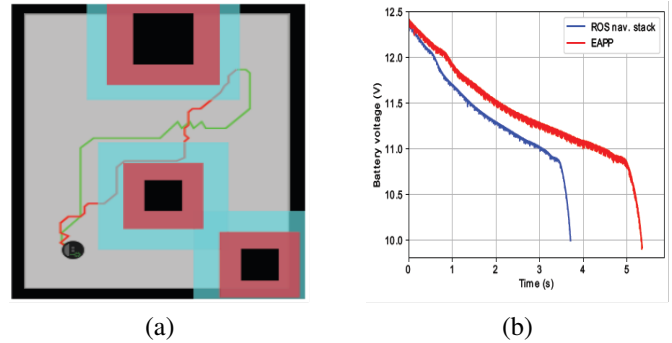

Fig. 4. (a) Turtlebot 2 in Stage robot simulator. (b) Battery discharge [12]

In [13], the paper analyses the significance of the energy cost models for different steering mechanisms and environmental model. It also shows comparative results of optimality, completeness, and computational time for different energy-efficient motion planning methods.

To summarize, an energy-efficient system by path planning is accomplished by reducing the energy consumption of the system with the following measures: (i) considering path planner heuristic function with the energy cost function, (ii) avoiding frequent acceleration and deceleration in the account of the velocity profile, (iii) traveling by avoiding high energy cost area estimated in the trajectory, and (iv) reducing the idle current consumption.

B. Component-level power saving

All the components require energy to function properly. So component-level energy saving techniques are also related to the energy-efficiency of the system. In [3], the author introduces the dynamic power management (DPM) technique to save static power. This method shuts down the component in idle states. A typical example is to shutdown some sensors when the robot is at a stop state in order to reduce power consumption. This method is prediction-based and requires high competence to turn on the component when required; failing will result in a system failure.

Another famous power management technique is the dynamic voltage frequency scaling (DVFS) approach which consists in reducing the supply voltage and clock frequency in order to reduce the power consumption of the processor. Nevertheless, these techniques are effective and practiced only with real-time systems. In [14], the author employs DVFS and controls the motor speed simultaneously to save energy. They conclude from the realization of four benchmarks with the processor and motor scaling schedule that by accelerating the frequency of the processor and decelerating the speed of the motor, energy consumption is reduced.

C. Power-aware real-time task scheduling

Low-power consumption and effective task scheduling are evident techniques to reduce overall energy consumption.

Very few studies are done to implement real-time scheduling algorithms to save energy in mobile robots. In [15], a power-aware EDF algorithm with a PID feedback control mechanism is studied to reduce power consumption. The scheduler changes the frequency profile of the processor to meet the power variation. Simulation results of power-aware EDF is compared with standalone EDF shows 47.28% and with DVS-EDF shows 3.58% increase in battery autonomy. Research work in [16] uses an incremental approach to produce a time-valid schedule based on the graph. The power spikes on the time-valid schedule are removed by applying max power constraints. The schedule is improved by min power utilization for mission-critical embedded systems. The time and power valid schedule is studied for the Mars rover system-level power-aware design.

IV. DISCUSSION

This paper reviews the current state-of-the-art on energy-efficient approaches for mobile robots with cobotic constrains which is summarized in Table I. Existing studies on energy-efficient motion planning consider only a small mobile robot system with a significantly undersized power system compared to a cobot. The works in [10], [11] are suitable in terms of the efficient path planning algorithms, but the steering mechanism of the robots considered is different from that of cobots. However, the experimental results show a distinguished reduction in power consumption. Nevertheless, the system is not deterministic and cannot satisfy the industrial constraints (e.g. dynamic environment, utilization time, and payload). Real-time system techniques are of minimal scope practiced on mobile robots to reduce energy consumption. The classified results of a power-aware real-time system are only simulation-based and are not validated on real robots. In [3], the author provides a proposal to consider both a motion planning technique and a real-time system for implementing an energy-efficient system, but to our knowledge no work is done considering both. It is a quintessential practice to consider real-time systems as cobot tasks have to meet hard deadlines.

We conclude that it is important to develop an energy model of the cobot to estimate energy consumption from all the energy-efficient motion planning approaches. It is essential to identify power consumption, as energy consumption is the multiplication of instantaneous power consumption of the component by the active time of the component; in simple expressions, energy is the integral of power over time. The power consumed by each component must be recorded to verify its compliance with the energy model. The power breakdown of the Pioneer-3DX robot in [3] shows the percentage of power consumed by each component. In [17], the author proposed a separate power model for the robot and laptop as the laptop power source is different from the robot. They decomposed motor and sensor as dynamic parts and other components as static parts. DC motor power consumption was realized using a polynomial approximation. An online model-based energy prediction system in [18] uses the same polynomial model to estimate the energy consumption of the motion system while additionally considering the acceleration. For including loss constrains in the energy model, it is essential to determine the kinematics and dynamics model of the robot, taking into account the steering mechanism [13] [19]. The energy model in [20] provides a well-formulated model for a differential drive system driven by DC motors. In [11], the energy consumption model of a four-wheeled Mecanum mobile robot with brushless AC motor is given by the sum of ideal energy and motion energy. The energy losses are not studied, but it is essential to consider the losses and provide a well-formulated energy model in our work. Power consumption of the component is measured using power profiling tools; power monitor [21], data acquisition card [3], and current sensors [11],[17].

The works summarized in Table I is limited for small mobile robots which have less energy source and different components compared to cobot which is considered in our work. These limitation lead to inventory ideas that have to be considered and improved for designing an energy-efficient cobot.

V. RESEARCH TREND

A. Real-Time Systems

"Real-time does not mean fast computing. The correctness of the system depends not only on the logical result of the computation but also on the time at which the results are produced [28]". Fig. 5 illustrates the comparison of timing constraints in real-time systems. A real-time constraint is associated with a deadline. For *hard* real-time system constraints, a failure to meet even a single deadline may lead to catastrophic system failures (e.g. autonomous robots, ABS). In *soft* real-time systems, the system does not fail even if deadlines are missed (e.g data acquisition system). Each missed deadline can degrade the overall performance of the system that is often associated to a metric called *Quality of Service (QoS)* [23]. Both hard and soft real-time systems must be predictable and deterministic.
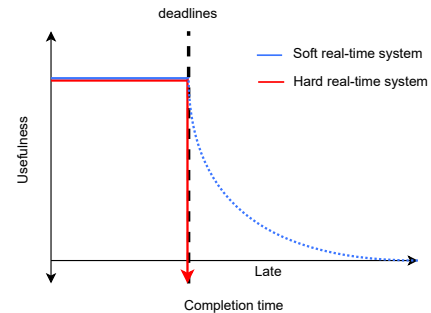

Fig. 5. Real-time system constraints

The cobot exhibits hard real-time constraints; For example, if the motion or sensor control misses its deadline, the cobot will collide with the obstacles causing catastrophic incident. Additionally, cobots also embeds functionalities with soft

TABLE I
SUMMARY OF STATE-OF-THE-ART ON ENERGY-EFFICIENCY FOR MOBILE ROBOTS WITH COBOTIC CONSTRAINTS

Techniques	Most relevant works	Advantages	Limitations
Energy-efficient path planning	[4] [5]	energy-efficient exploration	undersized power supply, static environment, nondeterministic, DC motors
	[6] [7] [8] [9]	optimal motion planning	
	[12]	PDDL domain modeling, ROS Navigation stack	
	[10] [11]	dynamic environment, brushless AC motors	validated on four wheel omni-directional robot, nondeterministic
Real-time task scheduling	[14]	DVFS	not validated on real robots
	[16]	mission-critical	
	[15]	power-aware PID feedback control EDF	

real-time constrains; For instance, if the communication task misses its deadline, the cobot does not fail but it merely degrades in QoS.

Robotic middlewares like ROS, perform navigation in autonomous mode but they do not guarantee any deterministic behavior to the system. ROS does not have any real-time compatibility, which conversely leads to the development of ROS2[5]. It is essential to make the cobot deterministic in order to comply with hard real-time constraints.

Optimal real-time schedulers including Rate Monotonic (RM) or Earliest Deadline First (EDF) [29] must be implemented to make the cobot achieve a deterministic behavior. Table II shows simple example of two cobotic tasks: a motion control task and a laser sensor task. T and C parameters respectively refers to the period of activation and the worst-case execution time of the tasks. The total processor utilization factor U is defined as the fraction of processor time spent in the execution of the taskset and is calculated as follows: $U = \sum_{i=1}^{n} \frac{C_i}{T_i}$. A necessary condition for a set of periodic tasks to be feasibly scheduled on one processor with RM scheduling is based on Liu and Layland's condition [24] given by:

$$U \leq n(2^{\frac{1}{n}} - 1) \quad (1)$$

where n represents the number of tasks in the taskset.

In our example, the processor utilization of the given task set is $\frac{4}{10} + \frac{8}{30} = 0.666$. The schedulability bound in (1) for two tasks is equal to 0.83. As the processor utilization is less than the schedulablility bound, tasks are schedulable (i.e., they will meet all their deadlines). According to RM the task with the lowest period has always the highest priority. Fig. 6 illustrates the Gantt chart of the cobotic taskset scheduled by Rate Monotonic scheduling algorithm on the first hyperperiod.

It is also important to consider some interesting real-time scheduling algorithms that integrate energy constraints [25],[26],[27]. These energy-aware real-time scheduling are capable of maximizing energy autonomy by decreasing power consumption of the cobots. As special ability of this

[5]https://index.ros.org/doc/ros2/Tutorials/Real-Time-Programming/

TABLE II
A SIMPLE COBOTIC TASK SET

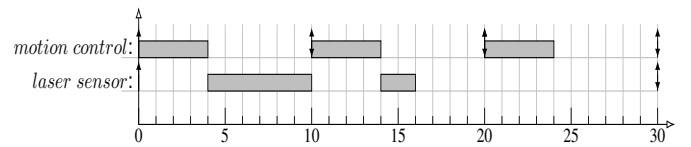| Tasks	T (ms)	C (ms)	Utilization
motion control task	10	4	0.400
laser sensor task	30	8	0.266

Fig. 6. Gantt chart of a cobot taskset scheduled by Rate Monotonic

scheduling approach, energy harvesting techniques will be studied and implemented from the observed energy losses and ambient source to potentially make the cobot energy-neutral. Implementation of these scheduling algorithms on a standalone general-purpose Linux system would not be performant from a real-time perspective. Therefore, we plan to use the dual-kernel approach Xenomai (i.e., a real-time patch for the Linux kernel), as this is the best option for guaranteeing hard real-time constraints [22].

B. Future Works

The novelty of this work is to implement real-time system scheduling algorithms with strict timing and energy constraints on a cobotic framework. This work will result in the utmost utilization of the energy resource available. On the other hand, to reduce energy consumption, our review is apparent to consider an energy-efficient path planning function with the energy model. Fig. 7 shows the proposed architecture for the cobot. The embedded PC will run dual kernel RTOS with a scheduling algorithm considering strict deadlines and energy needed for the process. The energy-efficient path planner with the energy cost function computes an energy-efficient trajectory to follow. In addition to this system, as an advantage of the scheduling algorithm, the energy harvesting from ambient sources will be implemented to harvest energy during the cobot operation. The difficulty of this approach is to realize
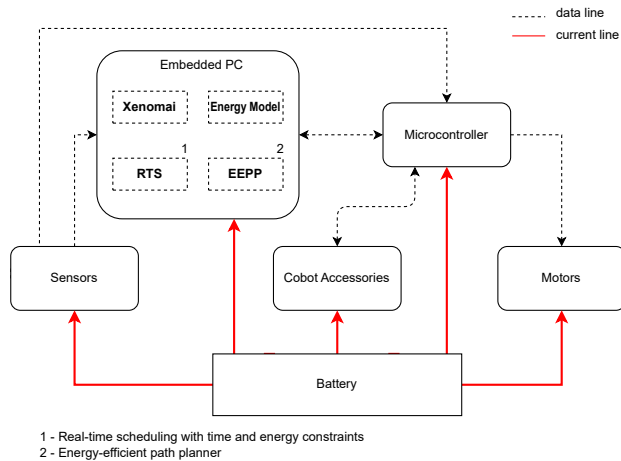
Fig. 7. Deterministic and Energy-efficient architecture of the Cobot

the energy utilization of each process to schedule. Finally, this work will adapt a CPSS with energy-neutrality and real-time capabilities.

VI. CONCLUSION

Industrial attention towards cobot has expanded. It is vital to consider a cobot with an energy-neutral design to meet the energy demand entirely by a renewable source. This paper has presented the current state-of-the-art on energy-efficient mobile robots. Most approaches have focused only on motion planning to reduce energy consumption. To our knowledge, real-time systems with energy constraints have not been explored. It is essential to consider cobot as a hard real-time system to exercise strict deadlines. Eventually, scheduling approaches with timing and energy constraints will provide a deterministic and energy-efficient design. These approaches can also practice energy harvesting from renewables. With this practice, the maximum energy demand for the cobot will have coincided with renewables. This approach intends to expand the CPSS application in the following ways: (i) improve the processor utilization, (ii) increase the operating time with reduced energy consumption, and (iii) importantly satisfy the energy demand with renewables.

REFERENCES

[1] B. Yilma, Y. Naudet, H. Panetto. "A new paradigm and meta-model for cyber-physical-social systems". 21st IFAC World Congress, Berlin, Germany, 2020.
[2] M. A. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. A. Moore, R. B. Gillespie and P. Akella, "Cobot architecture," IEEE Transactions on Robotics and Automation, 2001.
[3] Y. Mei, Y. Lu, Y. C. Hu and C. S. G. Lee, "A case study of mobile robot's energy consumption and conservation techniques," ICAR '05. Proceedings., 12th International Conference on Advanced Robotics, Seattle, WA, 2005.
[4] Y. Mei, Y. Lu, Y. C. Hu and C. S. G. Lee, "Energy-efficient motion planning for mobile robots," IEEE International Conference on Robotics and Automation, ICRA '04 Proceedings., New Orleans, LA, USA, 2004.
[5] Y. Mei, Y. Lu, C. S. G. Lee and Y. C. Hu, "Energy-efficient mobile robot exploration," ICRA '06. Proceedings., IEEE International Conference on Robotics and Automation, Orlando, FL, 2006.
[6] S. Liu and D. Sun, "Optimal motion planning of a mobile robot with minimum energy consumption," IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Budapest, 2011.
[7] S. Liu and D. Sun, "Modeling and experimental study for minimization of energy consumption of a mobile robot," IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Kachsiung, 2012.
[8] S. Liu and D. Sun, "Minimizing Energy Consumption of Wheeled Mobile Robots via Optimal Motion Planning," IEEE/ASME Transactions on Mechatronics, 2014.
[9] R. Datouo, F. B. Motto, B. E. Zobo, A. Melingui, I. Bensekrane and R. Merzouki, "Optimal motion planning for minimizing energy consumption of wheeled mobile robots," IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, 2017.
[10] C. Henkel, A. Bubeck, W. Xu, "Energy Efficient Dynamic Window Approach for Local Path Planning in Mobile Service Robotics" Auckland, New Zealand, IFAC-PapersOnLine, 2016.
[11] L. Xie, C. Henkel, K. Stol, P. Xu. "Power-minimization and energy-reduction autonomous navigation of an omnidirectional Mecanum robot via the dynamic window approach local trajectory planning". International Journal of Advanced Robotic Systems, 2018.
[12] R. Maidana, R. Granada, D. Jurak, M. Magnaguagno, F. Meneguzzi, A. Amory. "Energy-Aware Path Planning for Autonomous Mobile Robot Navigation". Florida Artificial Intelligence Research Society Conference, North America, 2020.
[13] H. Zhang, Y. Zhang, and T.Yang , "A survey of energy-efficient motion planning for wheeled mobile robots", Industrial Robot, 2020.
[14] J. Brateman, C. Xian and Y. Lu, "Energy-Effcient Scheduling for Autonomous Mobile Robots," IFIP International Conference on Very Large Scale Integration, Nice, 2006.
[15] W. Bin, W. Tianmiao, W. Hongxing, W. Meng and S. Zili, "Power-aware real-time task scheduling with feedback control for mobile robots," 3rd IEEE Conference on Industrial Electronics and Applications, Singapore, 2008.
[16] L. Jinfeng, P. H. Chou, N. Bagherzadeh and F. Kurdahi, "Power-aware scheduling under timing constraints for mission-critical embedded systems," Proceedings of the 38th Design Automation Conference, Las Vegas, NV, USA, 2001.
[17] L. Jaiem, S. Druon, L. Lapierre, D. Crestani, "A Step Toward Mobile Robots Autonomy: Energy Estimation Models", Towards Autonomous Robotic Systems. TAROS. Lecture Notes in Computer Science, vol 9716. Springer, Cham, 2016.
[18] R. Parasuraman, K. Kershaw, P. Pagala and M. Ferre, "Model Based On-Line Energy Prediction System for Semi-autonomous Mobile Robots," 5th International Conference on Intelligent Systems, Modelling and Simulation, Langkawi, 2014.
[19] M. Jaramillo, F. Mauricio, D. Sedat, J. B. Gómez-Mendoza and M. Lino. "Energy estimation for differential drive mobile robots on straight and rotational trajectories." International Journal of Advanced Robotic Systems, 2020.
[20] M. Wahab, F. Rios-Gutierrez and A. El Shahat, "Energy modeling of differential drive robots," SoutheastCon, Fort Lauderdale, FL, 2015.
[21] H. Linfei, Z. Liang, K. Jongwon. "Energy Modeling and Power Measurement for Mobile Robots" Energies., 2018.
[22] Brown, "How fast is fast enough ? Choosing between Xenomai and Linux for real-time applications.", 2010.
[23] A. Marchand, M. Chetto. "Quality of Service Scheduling in Real-Time Systems". International Journal of Computers, Communications and Control, Agora University of Oradea, 2008.
[24] Liu, C. Laung, and James W. Layland. "Scheduling algorithms for multiprogramming in a hard-real-time environment." Journal of the ACM (JACM), 1973.
[25] M. Chetto, A. Queudet. "A Note on EDF Scheduling for Real-Time Energy Harvesting Systems". IEEE Transactions on Computers, Institute of Electrical and Electronics Engineers, 2014.
[26] M. Abdallah. "Ordonnancement temps réel pour l'optimisation de la Qualité de Service dans les systèmes autonomes en énergie". Systèmes embarqués. University of Nantes, 2014.
[27] R. Osta. "Contributions to Real Time Scheduling for Energy Autonomous Systems". Embedded Systems. University of Nantes, 2017.
[28] J. A. Stankovic, "Misconceptions about real-time computing: a serious problem for next-generation systems," in Computer, 1988.
[29] Jane W S Liu, "Real-Time systems," 2000.