Robotica: (2021) volume 39, pp. 1202–1215. © The Author(s), 2020. Published by Cambridge University Press.
doi:10.1017/S0263574720001010

Wireless Robotic Components for Autonomous Vehicles

Juan Francisco Villa-Medina, Miguel Ángel Porta-Gándara and Joaquín Gutiérrez*

Engineering Group, Centro de Investigaciones Biológicas del Noroeste S.C., La Paz 23000, BCS, México
E-mails: jfvilla@cibnor.mx, maporta@cibnor.mx

(Accepted October 08, 2020. First published online: November 6, 2020)

SUMMARY
A versatile architecture is presented to implement autonomous vehicles. The focus idea consists of a set of standalone modules, called wireless robotic components wireless robotic components (WRCs). Each component performs a particular function by means of a radio modem interface, a processing unit, and a sensor/actuator. The components interact through a coordinator that redirects asynchronous requests to the appropriate WRCs, configuring a built-in network. The WRC architecture has been tested in marine and terrestrial platforms to perform tasks of waypoint and wall following. Results show that the tested system complies with adaptability and portability that allow conforming a variety of autonomous vehicles.

KEYWORDS: Wireless robotic components; Autonomous vehicles; Distributed architecture; Wireless network.

1. Introduction

Robots are composed of an appropriate configuration of diverse components, which provide the ability to perform an assigned task in a certain environment. Based on the concept of modular components, the configuration of manipulators was proposed with the combination of predefined link and joint modules.[1,2] Other works have used a diverse component repertoire: platforms, actuators, sensors, and wheels, to find the appropriate robot configuration;[3,4] including the integration of mechatronic blocks constructed from basic components,[5–8] open frameworks for multi-use purposes in the broader community,[9] and commercial/didactic kits[10] have been used to implement self-developed robotic prototypes. Similarly, the use of homogeneous components has been explored to construct manipulative and locomotive configurations of modular robots, including the development of self-reconfigurable and synchronization algorithms.[11–13]

Additionally, several modular robot approaches have defined algorithms and functions as modular units for the comprehension and validation of software components, such as navigation, mapping, learning, and planning.[14,15] Other strategies have considered each module as an agent to the robot configuration, which are based on the multi-agent concept to collaboration and coordination of multi robots.[16,17] In this context, a variety of software frameworks have been proposed to manage the heterogeneity and communication of the components or robots, providing an abstraction of hardware, such as sensors and actuators.[18] For instance, the open source framework named Robotic Operating System (ROS) is a robotic library for software standardization and the Hardware-ROS (H-ROS) for the hardware.[19,20] The automotive open system architecture is a software framework for the integration of modules from multiple Original Equipment Manufacturer (OEM) suppliers in the automotive sector.[21] The advanced explosive ordnance disposal robotic system is a common open architecture

* Corresponding author. E-mail: joaquing04@cibnor.mx

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

in military robots, which are formed with modules that maintain common physical, electrical, and logical standard interfaces.[22] Another approach to standardize electrical, mechanical, and interface of military vehicles is the generic vehicle architecture.[23]

The development of wireless networks has improved the communication methods to support real-time data transmission for a wide range of applications, together with the emerged paradigms, such as Internet of Things.[24,25] Several wireless technologies provide connectivity, security, and scalability through standard protocols such as IEEE 802.11 (WiFi), IEEE 802.15.1 (Bluetooth), IEEE 802.15.3 (UWB), IEEE 802.15.4 (ZigBee, Z-Wave), IEEE 802.16 (WiMAX), and IEEE 1609.3 (WAVE). These wireless technologies in robotic field are used to sense, communicate, and coordinate modules or robots,[26,27] forming robot networks for distributed and mobile applications. For example, a simulator to evaluate control algorithms and wireless propagation on fleet of unmanned aerial vehicle networks has been developed for robotic research;[28] other aerial robot network was modeled to provide wireless communication to clients on the ground with optimal area coverage, fault-tolerance, and redundancy.[29] An aerial monitoring network based on quadcopters has been implemented for research and rescue application, using WiFi communication to transmit images to a ground station and coordinate the flight routes of the aerial vehicles.[30]

Modular robots proposed as a network composed of modules, in which the messages are transmitted among them, rely on either centralized or decentralized control to perform the assigned task.[31] In a centralized control, there is a central module (a computing module) that receives all messages to specify actions for each module, whereas each module in a decentralized strategy performs sensing, computing, and actuating to control the function in module level, exchanging only relevant messages among modules. In several heterogeneous modular architectures, the control has been achieved through decentralization approaches from mechanisms based on biological concepts (distributed electrical or chemical signaling), autonomous decentralized computer control systems to swarm intelligence.[32–36] For instance, hormone-based distributed control strategy, in which a message transmitted by a module triggers functions at different modules, which decide locally the execution of their predefined actions. Each message is broadcasted in the distributed modular system, without specific destination.[37]

This article presents the practical application of an architecture based on wireless robotic components (WRCs) to implement autonomous vehicle prototypes, marine and terrestrial. Each WRC involves a sensor/actuator, a processing unit, and a communication interface to perform a specific function. The interoperability among WRCs uses asynchronous communication based on component requests. The distributed relation among WRCs allows the robotic system to accomplish an assigned task. The goal of this novel architecture is to implement a practical assembly system for autonomous vehicles, based on the interchangeability of wireless components with no standardized logical or physical interfaces as such, neither compatible with software frameworks (e.g., ROS). This modular architecture not only integrates the functionality in the software but also include the operationality of the appropriate sensor/actuator, conforming a heterogeneous component network with a decentralized control.

2. Wireless Robotic Component Concept

A robotic system consists of several components integrated to accomplish a particular task. The WRC approach requires that each sensor/actuator be implemented in a standalone array with capabilities of communication and computation to execute a specific function through a wireless link unit and a processing unit (embedded device or computer), respectively (Fig. 1). Each WRC is developed from the technical specifications (i.e., operation range, resolution, transmission speed, bandwidth, and voltage, among others) to operate/control the particular sensor/actuator, including the logical and physical interfaces, such as accessing protocols and power requirements. A power common bus provides all the necessary voltages.

All the WRCs are programmed to perform their assigned functions and conform a heterogeneous WRC built-in network for the configuration of an autonomous vehicle. Each WRC intercommunicates by its wireless link unit to request certain command or to exchange data through a byte-oriented protocol. In this proposed architecture, some WRCs are designed to generate requests based on their specific functions and others attending such requests. One special WRC is the coordinator (WRC-C), which disperses the requests among the WRCs. When a WRC is powered, a registration request is

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

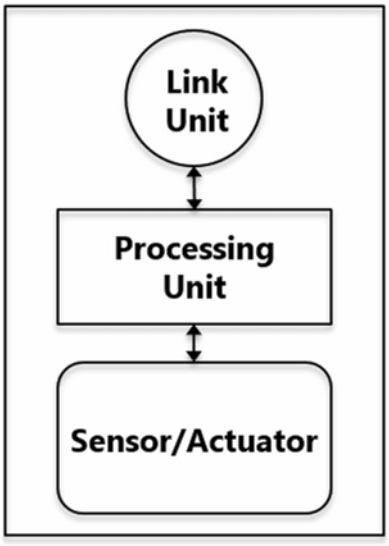


Fig. 1. Wireless robotic component: A link unit, a processing unit, and a sensor/actuator.

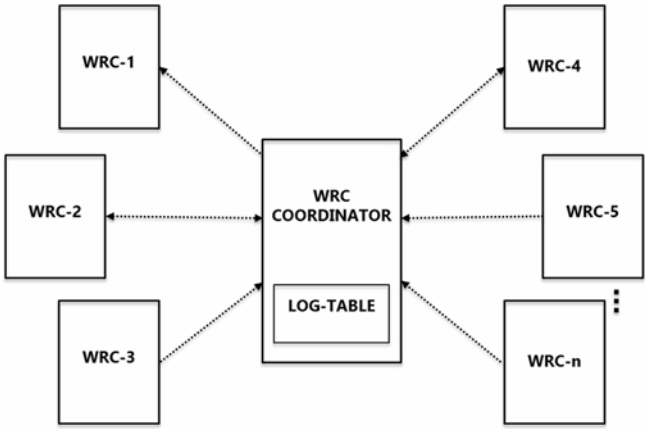


Fig. 2. Wireless robotic component concept.

sent to the WRC-C and inserted in a log-table. Registering a WRC involves giving an identifier and listing the requests that can generate/attend.

This form of coordination offers an open-loop control mechanism to perform the task, based on a communication minimal to deliver commands with priority levels or relevant data to the WRCs, which is an adaptation of hormone-inspired approaches. The requests are routed by the WRC-C on the basis of the request matching, reducing the network traffic. Each WRC controls the content and priority level of the command/data for each request that is sent or it processes the request received using an action-selection mechanism based on identifier and priority, allowing the reactions on the changes of the environment or internal.

The WRC approach is a network of distributed functions, which allows the robotic system to be scalable and implemented with diverse components. The concept deploys interchangeable WRCs, and others that are for specific functions, which facilitates the assembly process and diminishes the time of autonomous vehicle configuration (Fig. 2).

3. Wireless Robotic Component Design

The WRC system implementation is based on the ZigBee star topology, which provides an efficient and scalable communication scheme, allowing the creation of wireless personal area networks, with a relatively low powered consumption, low data rate, and wide range up to 1600 m outdoors in line sight. It is based on the IEEE standard 802.15.4 at 2.4 GHz bandwidth.[38]

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

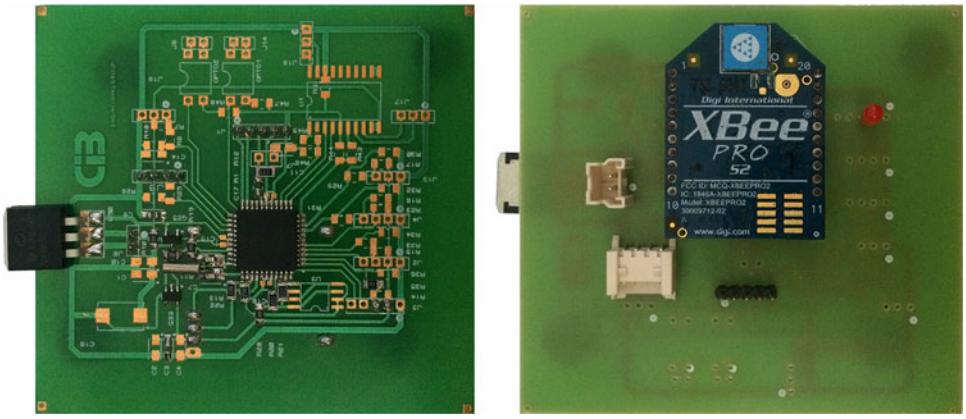


Fig. 3. Generic card (GC) to implement the WRCs, based on a microcontroller as the processing unit and a ZigBee transceiver module as the link unit.

The WRCs were assembled using an own development generic card (GC) designed for multipurpose applications (Fig. 3), by means of a 16-bit microcontroller PIC24FJ64GB004 (Microchip Technologies, Chandler, AZ) as the processing unit. This nanoWatt XLP microcontroller operates in a range 2.0–3.6 V at 8 MHz with internal oscillator, 64 KB of program memory, 8 KB of SRAM, and hardware real-time clock/calendar (RTCC). It has 44-pins that include up to 25 digital input/output lines, 13 and 10-bit analog to digital converters, two serial peripheral interface modules, two I2C, two UART, and five 16-bit timers. The GC also includes the link unit, which is a radio modem XBee-PRO S2 (Digi International, Eden Prairie, MN), programmed at 100 mW of transmission power, data rate of 250 kbps, and interfaced through one of the microcontroller serial ports. The processing unit is programmed with the algorithm to implement its appropriate function using the C compiler 4.12 (Custom Computer Services).

Since the WRC concept can be realized with different microprocessor circuits, two kinds of WRC-Cs were built, one using the outlined GC and other using the WRCM4110 microprocessor card (Digi International, Eden Prairie, MN), which operates in 3.3 V and 90 mA at 29 MHz with 512 K Flash Memory, 256 K data SRAM, six serial ports, 40 digital input/output lines, and backup battery to support RTCC and data SRAM. It is programmed with Rabbit Dynamic C compiler 10 (Digi International, Hopkins, MN).

3.1. Wireless robotic components interaction

All WRCs send a registration request to the WRC-C to be enrolled in the log-table with a unique identifier. This request includes the 64-bit device ID, the 16-bit network address, and a list that represent all the different requests, which can be performed or attended by the WRC (i.e., forward, turn, stop). According to the log-table information, the WRC-C transmits to those WRCs capable to execute that request.

The communication among the WRCs is performed by a 5-byte request protocol. The first byte is an identifier of the action or command requested. The second byte is an optional parameter with the percentage information about the type of request. The third states the priority of the request. The fourth is the WRC registered identifier assigned by the WRC-C. The last one is the protocol sequence terminator used to validate the request completeness in case of a byte loss, not only by failure but also by the hot swapping or the removal of a WRC. For instance, the 5-byte protocol can be [03, 70, 80, 12, 255], where the number 3 corresponds to a right turn (a number 4 may represent a left turn), a 70% of the angular span, with an 80% of priority, the 12 is the WRC number in the log-table, and 255 is the end of the chain.

The attending request decision (Fig. 4) is realized by those pertinent WRCs, comparing the received priority and the identifier to execute the command or action solicited. For equal priority requests, the first one received is performed. The interaction among the WRCs is heterarchical through requests emerging from the expected behavior of the robotic system.

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

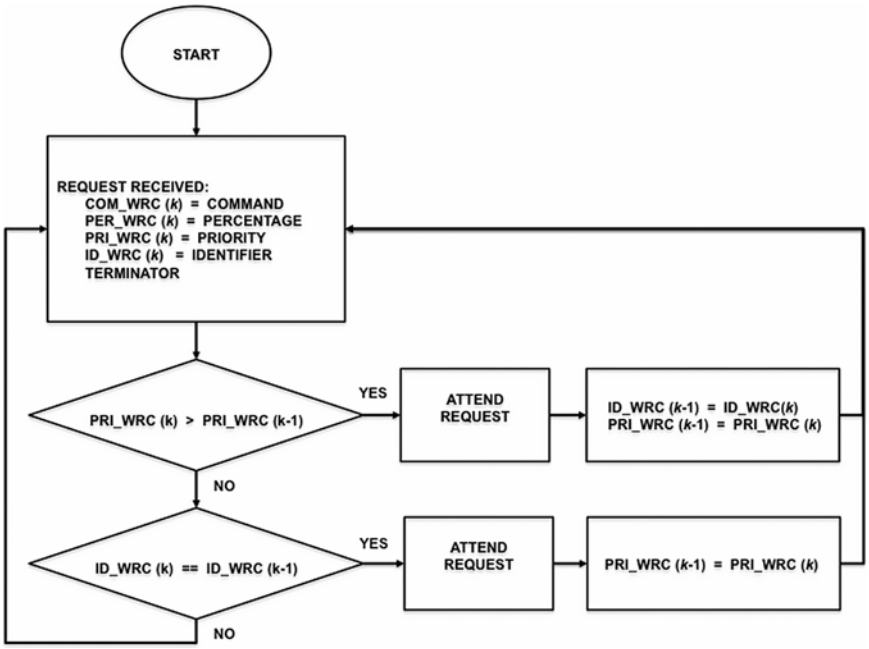


Fig. 4. Decision-attending WRC algorithm based on priority.

3.2. Wireless robotic components developed

A set of WRCs has been assembled to perform a specific function and can be interchanged to realize diverse tasks in a marine vessel, a terrestrial Ackerman vehicle, and a terrestrial differential vehicle:

1. *WRC coordinator*
   This WRC registers all the WRCs in the log-table and broadcasts the requests. It consists of a GC for the terrestrial vehicles and the WRCM4110 microprocessor card for the marine vessel.
2. *WRC traction*
   This WRC embeds the traction function to implement the locomotion for the Ackermann vehicle, employing the rear wheels coupled mechanically to a 12 V DC motor, which is controlled by its processing unit through a MC33887 H-bridge motor driver (NXP Semiconductors, Eindhoven, Netherlands) switched by four 4N32 optocouplers (Vishay Intertechnology, Malvern, PA) to maintain isolation between power and driving circuits. The requests forward, reverse, and stop generate a pulse width modulation output signal to control the average power and direction of the motor.
3. *WRC propeller*
   This WRC drives the propulsion of the marine vessel with two thrusters used in the port and starboard sides. A programmed GC attends the requests headway, astern, and stop, through two electronic speed control cards ESCON Module 50/5 (Maxon precision motors, Foster City, CA) to drive the brushless motors EC-i 40 40 mm (Maxon motor, Sachseln, Switzerland) with planetary gearhead GP 32 C 32 mm coupled to right- and left-hand 0.07 m four blade propellers (Koch Metalworks, ZG Nieuwpoort, Netherlands), respectively.
4. *WRC caterpillar traction*
   The differential locomotion uses two-caterpillar track WRCs, for left and right. Each one is equipped with a 12 V DC gear motor 120 RPM, driven through a programmed GC with serial output signals and an H-bridge power driver Motor Mind Be (Solutions Cubed, Chico, CA), to perform forward, reverse, turn, and stop actions.
5. *WRC direction driver*
   The steering of the Ackermann vehicle is controlled driving a TP-S3003 servomotor with 4.1 kg-cm at 6V (Futaba Corporation, Huntsville, AL) through the GC, programmed to generate the pulse width control and attend the right, left, and center requests.

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

6. *WRC rudder driver*
   A rudder, driven with a high torque −36.0 kg at 6.0V-SW1212SG Waterproof servomotor (SAVOX, Salt Lake City, UT), performs the steering of the marine platform. A GC programmed permits to control the rudder angle up to 180° to attend the right, left, and center requests.
7. *WRC position tracker*
   The tracker is built by means of an ASHTECH MB100 L1/GLONASS+SBASGPS OEM receiver (Trimble Navigation Limited, Sunnyvale, CA), with a high sensitivity Trimble AV34 GNSS antenna. The NMEA0183 message output is connected to a serial interface (Rx and Tx signals with baud rates at 9600 bps) of the GC, programmed to calculate the distance and heading between the actual position and a destination point at a frequency of 1 Hz. This WRC sends the stop and forward requests and the heading data.
8. *WRC pointer*
   This WRC is an electronic compass based on the miniature surface mount high-performance inertial measurement unit VN-100 rugged (VectorNav Technologies, Dallas, TX), which is connected to the GC via serial communication at 9600 bps with a sampling rate of 2 Hz. The desired orientation is calculated from the WRC tracker data and the compass to send right, left, or center requests.
9. *WRC obstacle avoider*
   This component is an array of two 40 KHz PING ultrasonic distance meters (Parallax Inc., Rocklin, CA), connected each one to a bidirectional pulse pin of the GC and located in the front left and right of the vehicle. A fuzzy controller generates the right, left, and center requests based on the distance measurements from about 0.02 m to 3 m at a sampling frequency of 4 Hz.
10. *WRC wall-follower*
    This component is also developed using two 40 KHz PING ultrasonic distance meters. The transducers are located in a lateral side of the vehicle at 90° and 45° angles to track the wall. A fuzzy controller generates the right, left, and center requests at rate of 4 Hz.
11. *WRC bathymetric*
    This component is developed using a DST Micron Mk2 Echosounder OEM (Tritech International Limited, Scotland, UK) for depth measurement. This is 500 KHz monolithic transducer with 490–520 KHz chirped pulse, beam width of 6° conical at 510 KHz, and operating range from 0.5 to 50 m with a resolution ± 1 mm and accuracy ± 5 mm. It operates at 12 V DC at 225 mA. The serial output is connected to an UART port of the GC for a sampling rate of 1 Hz. This WRC generates a stop request for shallow water, preventing the propeller of the thruster been trapped.
12. *WRC communicator*
    This WRC permits point-to-point communication between a laptop and the robotic system, by means of two paired long XTend 900 MHz OEM RF modules (Digi International, Hopkins, MN) with half-duplex link, with selectable power output between 1 mW and 1 W and data rate 9.6 to 115.2 Kbps. The user sends a request to receive the general status of the system. The computer runs a control/register program to set the manual or autonomous mode and to record the activity.

4. Tests and Results

The architecture of the proposed WRC was conceived to build a variety of mobile robot configurations. The goal is to interchange for diverse robotic platforms, common WRCs, and the specific ones to integrate autonomous vehicles. Those configurations were implemented for different environments and through development of specific tasks; the performance of the overall robotic system was evaluated.

The WRC system was tested in the three different autonomous platforms: marine, Ackermann, and differential. Each one was built with WRCs for the common functions and specific WRCs according to the platform type (Table I). A waypoint-following task was performed using the marine vessel and the Ackermann; meanwhile, the terrestrial vehicles developed a wall-following task. To analyze the vehicle performance, the requests of each WRC were registered by means of the WRC communicator, recording the 5-byte communication protocol.

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

Table I. Configuration of Vehicles Based on WRCs

Platform	WRC											
	1	2	3	4	5	6	7	8	9	10	11	12
						Waypoint-following Task						
Marine	×		×			×	×	×			×	×
Ackermann	×	×			×		×	×				×
						Wall-following Task						
Ackermann	×	×				×	×	×	×	×		×
Differential	×			×			×	×	×	×		×

WRC-1: Coordinator	WRC-2: Traction	WRC-3: Propeller
WRC-4: Caterpillar	WRC-5: Direction	WRC-6: Rudder
WRC-7: Tracker	WRC-8: Pointer	WRC 9: Avoider
WRC-10: Follower	WRC-11: Bathy	WRC 12: Comm



Fig. 5. Autonomous marine vessel and its deck with the batteries and the WRC generic cards enclosed in a portable box. Includes the common WRC tracker and pointer to perform the waypoint-following task (Table I).

4.1. Marine vessel performance

The marine surface vehicle (Fig. 5) is built with PVC. The vessel is a custom-made twin hull torpedo of 1.2 m length overall and 0.6 m beam. The bridge is a rectangular prism container that carries all the electronic components, including two 19.8 V LiFe rechargeable batteries. The autonomy is about 4 h at the cruise speed of 1 m/s.

The vehicle performed a four-waypoint-following task into a seawater pond of an area about 9000 m^2. Each point was separated by 40 m forming a quadrilateral shape. The points are considered circular areas with a radius proportional to the characteristic length of the vehicle, in this case, a 1.5 m of radius. When the vehicle intercepts the point, it is considered that the point has been reached.

The WRCs interaction is as follows:

- The coordinates of each point are send by the user through the WRC communicator link to the tracking WRC and then to the WRC pointer and the requests are generated to reach the first target point.
- Through the WRC-C, the requests from WRC tracker and WRC pointer are send to the propeller and rudder driver, respectively.
- According to the priority, the actions are executed with a given magnitude for the different WRCs (for instance, forward at a certain percentage of the propulsion power and at certain angle of the rudder position).
- This process is cyclic until the first point is reached. A stop request by the WRC tracker is then sent to the WRCs,
- The next waypoints are then processed in the same manner.

The marine vessel departs from the start waypoint S toward the first waypoint 1 until the closed trajectory [S, 1, 2, 3, S] is completed (Fig. 6). The data received from the WRC tracker outline the

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

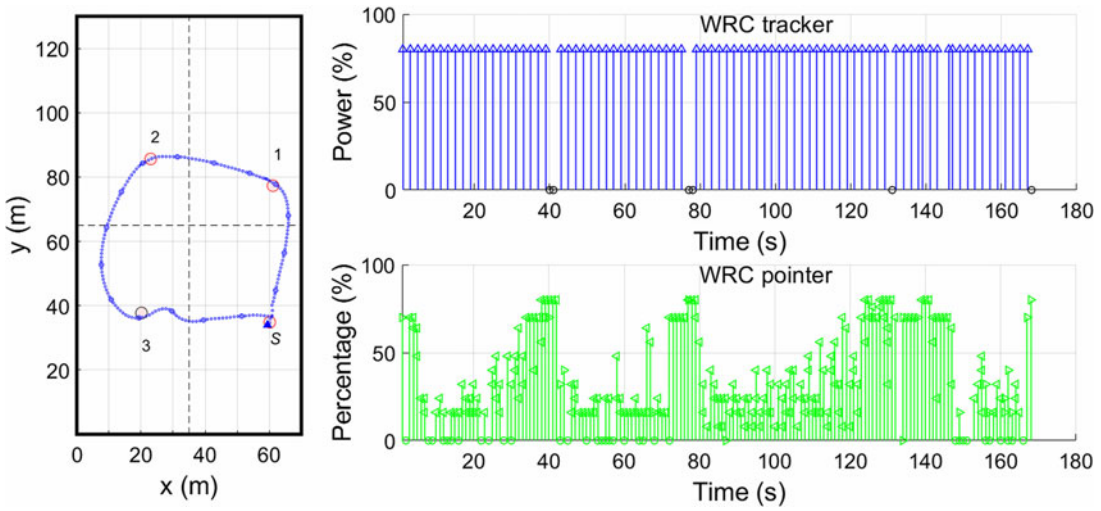


Fig. 6. Marine vessel traveled waypoint trajectory and the requests generated by the WRC tracker and WRC pointer, respectively.

position of the vessel, plotting each 10 data with a bigger marker size in the left graph. The vessel reaches each waypoint attending the forward (△) and stop (○) requests of the WRC tracker and the left (◁) and right (▷) requests of the WRC pointer. The task lasted for 168 s. The two right graphs show the tracker and pointer requests generated. In the WRC tracker graph, the forward requests are shown with an 80% propulsion power, which is the vessel cruise speed. The black-circled markers are the stop requests that correspond to a point reached in the trajectory. The WRC pointer graph shows the heading requests. The percentage is proportional to the orientation difference between the target and heading angles. At the top of each request, the direction of a marker triangle shows the right or left turn. When the vessel is correctly oriented, the zero % is represented with a green circle.

A trajectory detail is shown in Fig. 7. In the 131st s, a stop request generated by the WRC tracker is shown as a circle in the power graph, corresponding to the unique vessel position inside point 3. In the 134th s, a change in the heading was sent by the WRC pointer to redirect the vessel toward the end point S and is represented by the right-pointing triangle in the percentage graph.

4.2. Ackermann performance

The Ackermann platform (Fig. 8) is made of aluminum at 1:10 scale, 0.5 m long, and 0.4 m wide, with four independent traction wheels of 0.16 m diameter. The vehicle is powered by two 9.9 V at 4500 mAh rechargeable LiFe batteries connected to the power bus card.

The terrestrial vehicle was set to perform the waypoint-following task to prove the common WRCs used in the marine vessel. Only three waypoints in a paved court of an area of 28 m × 14 m were selected. The waypoint trajectory is a right triangle with 7 and 14 m of side length and about 16 m of hypotenuse. The circular area of each reached point is the same defined in the marine vessel performance. The WRCs communicator, tracker, and pointer were placed in the Ackermann platform, without any software change. The logic interaction procedure of the WRCs is similar to the previously described procedure for the marine vessel. The waypoint trajectory started at point S and lasted for 109 s (Fig. 9). The vehicle was oriented opposite toward the first point. To acquire the correct orientation, the WRC pointer generates right-turn requests.

To test other WRCs, a wall-following task was performed, using a rectangular perimetral wall with obstacles (Fig. 9b). One constraint was to preserve a 0.5 m of separation between the vehicle and the wall. For this task, the follower and avoider WRCs were set into the vehicle. The closed circuit reference path is drawn with a dotted line. The start point S was selected anywhere in the path, the obstacle A was detected by the avoider, and several left-turn requests were generated to bypass the obstacle. Meanwhile, the WRC follower generated right-turn requests trying to preserve the separation distance from the wall. The programmed strategy for evasion consists in back and right movements to avoid near detected objects as can be seen for obstacle B.

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

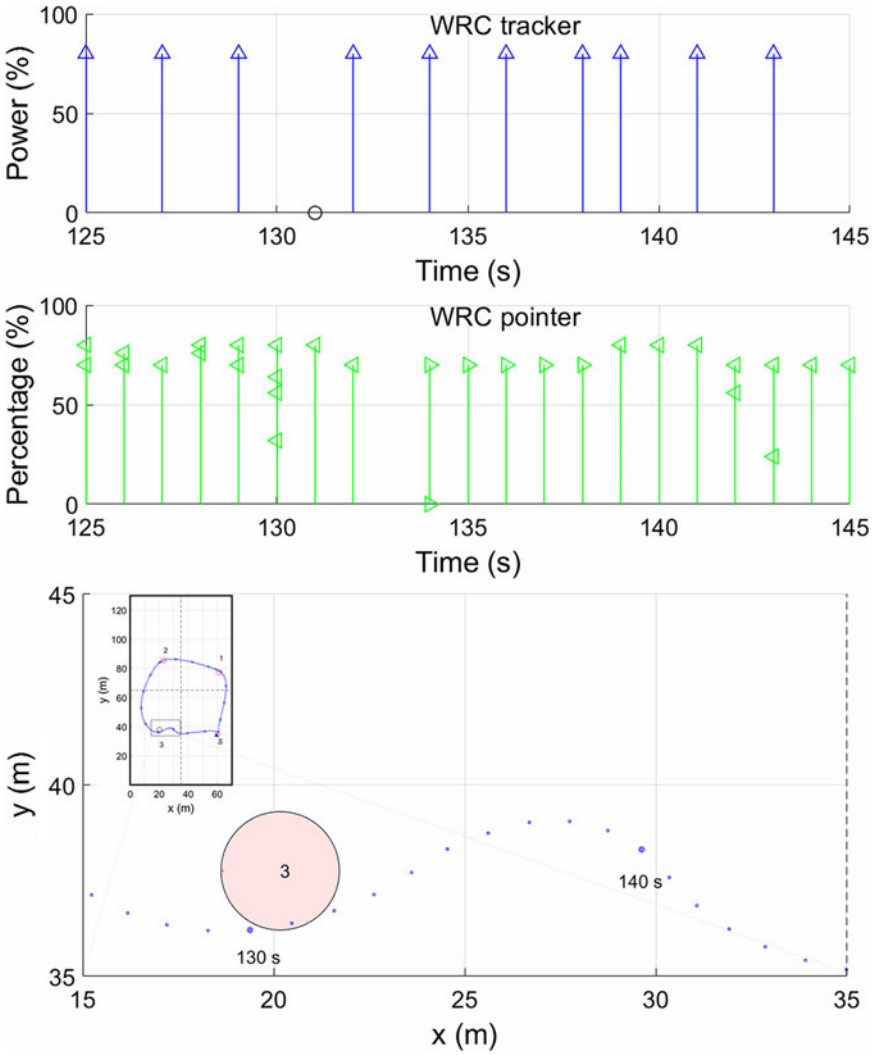


Fig. 7. Feature of the marine vessel trajectory and the tracker and pointer requests during the 125–145 s interval.

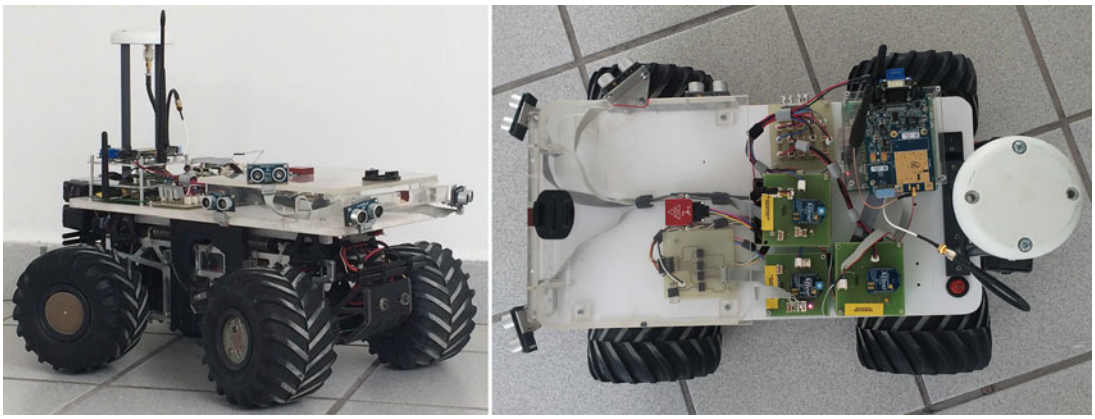


Fig. 8. Autonomous Ackermann platform with common WRCs to the marine vessel: tracker and pointer to perform the waypoint-following task and WRCs commons to the differential vehicle to perform the wall-following task: follower and avoider (Table I).

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

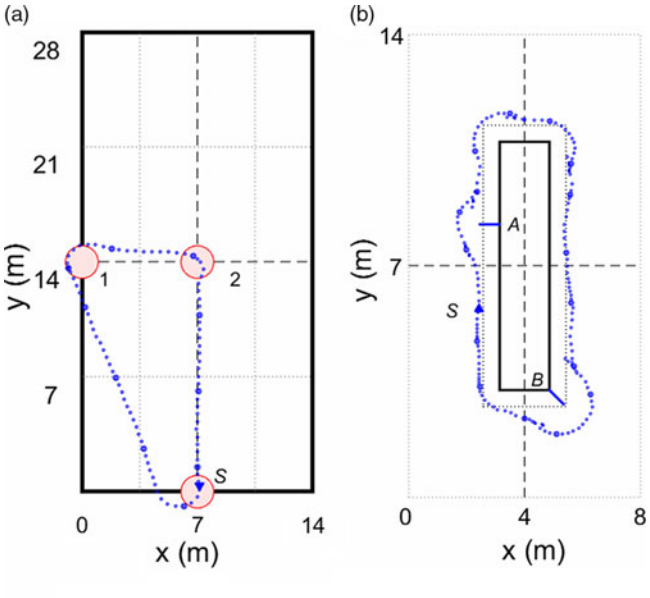


Fig. 9. Ackermann vehicle traveled trajectories: (a) waypoint and (b) wall-following.

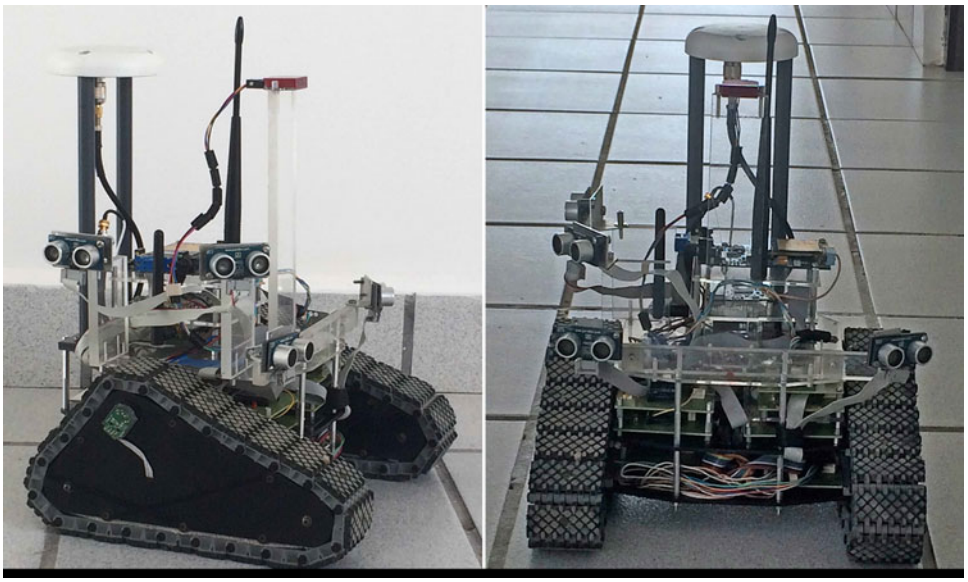


Fig. 10. Autonomous differential platform with common WRCs to the Ackermann vehicle to perform the wall-following task: follower and avoider (Table I).

4.3. Differential performance

The settled vehicle is a caterpillar type, 0.25 m × 0.25 m × 0.30 m, with a mass of 2 kg. It is made of a polypropylene and aluminum chassis combined with ABS plastic (Fig. 10). The tracked vehicle is powered by two 9.9 V at 4500 mAh rechargeable LiFe batteries to provide up to 4 h of continuous operation, with a mean speed of 0.7 m/s. The wall-following task with obstacles was realized to compare with the Ackermann performance.

The avoider and follower WRCs were taken and set directly in the vehicle, without any software change. In this case, the caterpillar traction WRCs were used to perform differential locomotion. Figure 11 shows the resultant 136 s trajectory and the requests in the interval between the 60th and the 70th s. In this period, the vehicle detected and avoided obstacle A, according to the avoider and follower WRC requests attended by the caterpillar WRCs. In the 66th s, the WRC avoider generated two left-turn requests with an incremental magnitude from 90% to 100%. Meanwhile, the WRC

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

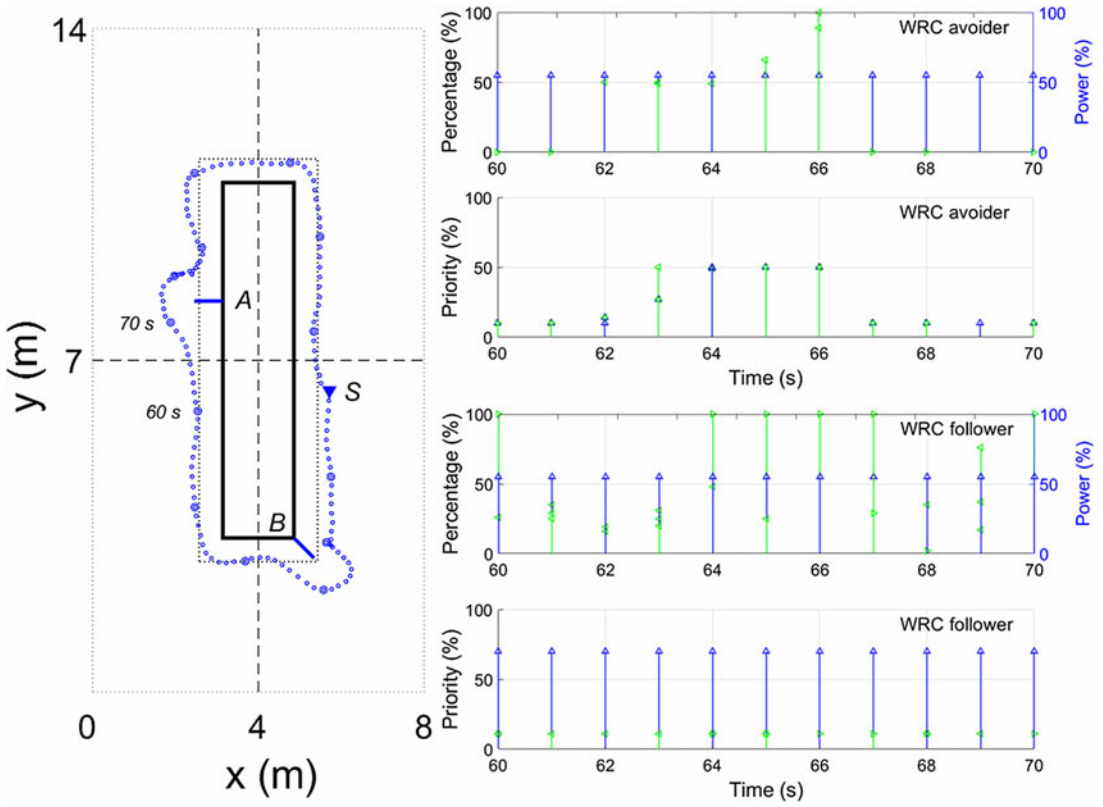


Fig. 11. Trajectory of the differential vehicle and the avoider and follower requests for the 60 to 70 s interval.

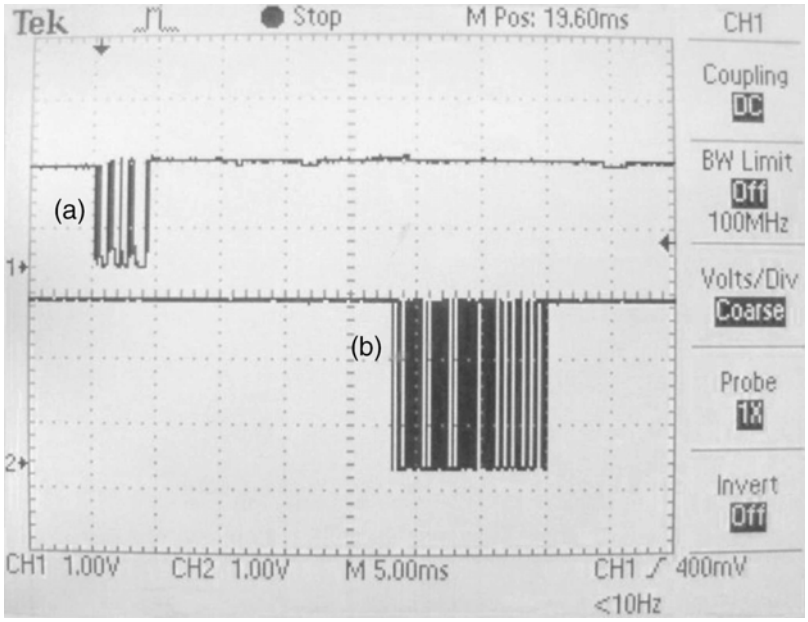


Fig. 12. WRC-C request latency distributed among the WRCs: (a) request signal by the processing unit (AT mode) and (b) link unit WRC-C output signal (API mode).

follower generated a right-turn request with a 100% of magnitude, opposed to the first request. The left turn was attended by the caterpillar traction WRCs because of their greater priority. At the same second, both WRCs generated non-opposite forward requests; the attended one came from the WRC follower, which had the greater priority.

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

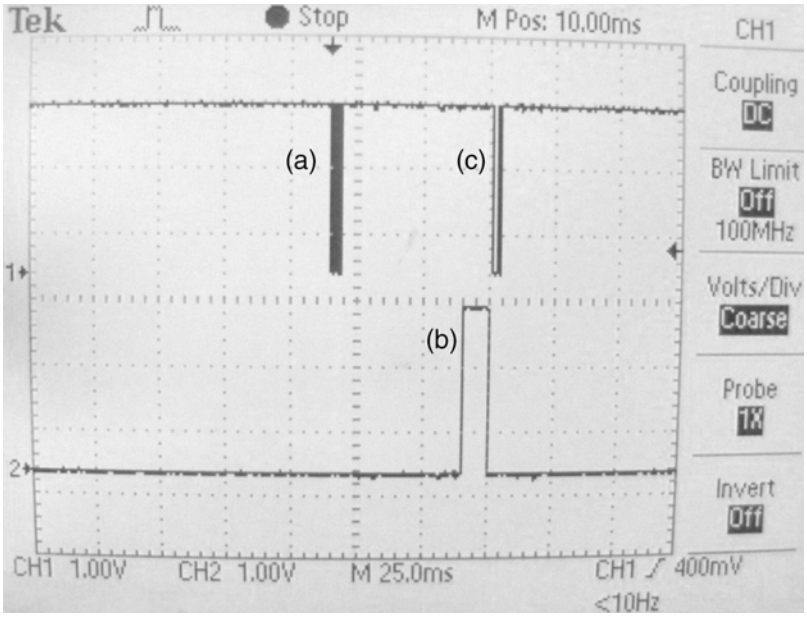


Fig. 13. WRCs latency between (a) sending. (b) attending requests, (c) new request.

4.4. WRC performance

The wireless communication provides a practical interconnection among the system components than a wired one. However, an intrinsic limitation of the wireless networks is the Rx/Tx latency of the WRCs.

For the marine and the terrestrial platforms, the WRC network operated at 19,200 bps to transmit 5-byte packets in a star topology. The WRC-C latency to redirect each request sent from a source WRC to the sink WRC is near 23 ms (Fig. 12), whereas the latency between a sending and attending request is about 50 ms through the WRC-C (Fig. 13).

5. Conclusion

The WRC concept was conceived for the design and implementation of autonomous vehicles to execute a task through distributed sensing, computing, communicating, and actuating in a networked environment of robotics. The robotic components can be integrated to configure different robotic systems.

The WRC system has been assembled in marine and terrestrial platforms to test and prove the concept. Results show that it is capable of interchanging the same WRCs to implement different robotic vehicles and executing the functions according to the desired task.

The WRC is programmed with an algorithm that needs no change for executing a prescribed function, but can be adapted to a particular task requirement or a specific environment.

The WRCs are capable of sending/receiving data/commands and executing actions/decisions according to the embedded software, acting as nodes in a wireless sensor/actuator network. The wireless communication provides more versatile and simpler interconnection than a wired one. A limitation of the wireless network is the number of WRCs that may configure the robotic system due to the latency of the WRC-C. In applications that require a greater speed, the wired version could be implemented, but more hardware is needed.

There are research open topics and other disciplines converging in robotics that may be covered for the further improvement of the WRC system.

Acknowledgments

The authors are very grateful to the CIBNOR technicians Jorge Cobos and Alfonso Álvarez.

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

References

1. R. Harrison, R. Western, P. Moore and T. Thatcher, "A study of application areas for modular robots," *Robotica* **5**(3), 217–221 (1987).
2. D. Tesar and M. S. Butler, "Generalized modular architecture for robot structures," *ASME Trans. Manuf. Rev.* **2**(2), 91–118 (1989).
3. S. Farritor, *On Modular Design and Planning for Field Robotic Systems* (Massachusetts Institute of Technology, Cambridge, MA, 1998).
4. P. C. Leger, *Automated Synthesis and Optimization of Robot Configurations: An Evolutionary Approach* (The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, 1999).
5. P. Ben-Tzvi, A. A. Goldenberg and J. W. Zu, "Articulated hybrid mobile robot mechanism with compounded mobility and manipulation and on-board wireless sensor/actuator control interfaces," *Mechatronics* **20**(6), 627–639 (2010).
6. H. Rajaie, O. Zweigle, K. Häussermann, U.-P. Käppeler, A. Tamke and P. Levi, "Hardware design and distributed embedded control architecture of a mobile soccer robot," *Mechatronics* **21**(2), 455–468 (2011).
7. Z. Shiller, "A bottom-up approach to teaching robotics and mechatronics to mechanical engineers," *IEEE Trans. Educ.* **56**(1), 103–109 (2013).
8. M. Garduño-Aparicio, J. Rodríguez-Résendiz, G. Macias-Bobadilla and S. Thenozhi, "A multidisciplinary industrial robot approach for teaching mechatronics-related courses," *IEEE Trans. Educ.* **61**(1), 55–62 (2018).
9. M. Vona and N. H. Shekar, "Teaching robotics software with the open hardware mobile manipulator," *IEEE Trans. Educ.* **56**(1), 42–47 (2013).
10. A. Takacs, G. Eigner, L. Kovacs, I. J. Rudas and T. Haidegger, "Teacher's kit: Development, usability, and communities of modular robotic kits for classroom education," *IEEE Robot. Autom. Mag.* **23**(2), 30–39 (2016).
11. K. Gilpin and D. Rus, "Modular robot systems," *IEEE Robot. Autom. Mag.* **17**(3), 38–55 (2010).
12. H. Ahmadzadeh, E. Masehian and M. Asadpour, "Modular robotic systems: Characteristics and applications," *J. Intell. Robot. Syst.* **81**(3), 317–357 (2016).
13. W. Saab, P. Racioppo and P. Ben-Tzvi, "A review of coupling mechanism designs for modular reconfigurable robots," *Robotica* **37**(2), 378–403 (2018).
14. D. Brugali and P. Scandurra, "Component-based robotic engineering (Part I) [Tutorial]," *IEEE Robot. Autom. Mag.* **16**(4), 84–96 (2009).
15. A. Bonarini, M. Matteucci, M. Migliavacca and D. Rizzi, "R2P: An open source hardware and software modular approach to robot prototyping," *Robot. Auton. Syst.* **62**(7), 1073–1084 (2014).
16. B.M. Innocenti Badano, A Multi-Agent Architecture with Distributed Coordination for an Autonomous Robot, *PhD Dissertation* (Universitat de Girona, Spain, 2009).
17. K. Hwang, C. Lo and W. Liu, "A modular agent architecture for an autonomous robot," *IEEE Trans. Instrum. Meas.* **58**(8), 2797–2806 (2009).
18. N. Mohamed, J. Al-Jaroodi and I. Jawhar, "Middleware for Robotics: A Survey," *Proceeding of the IEEE Conference on Robotics, Automation and Mechatronics* (2008) pp. 736–742.
19. P. Corke, "Integrating ROS and MATLAB [ROS topics]," *IEEE Robot. Autom. Mag.* **22**(2), 18–20 (2015).
20. V. Mayoral, A. Hernández, R. Kojcev, I. Muguruza, I. Zamalloa, L. Usategi and A. Bilbao, "The Shift in the Robotics Paradigm–The Hardware Robot Operating System (H-ROS); An Infrastructure to Create Interoperable Robot Components," *NASA/ESA Conference on Adaptive Hardware and Systems* (2017) pp. 229–236.
21. S. Fürst, "AUTOSAR - An Open Standardized Software Architecture for the Automotive Industry," *1st AUTOSAR Open Conference and 8th AUTOSAR Premium Member Conference* (2008).
22. M. A. Hinton, M. J. Zeher, M. V. Kozlowski and M. S. Johannes, "Advanced Explosive Ordnance Disposal Robotic System (AEODRS): A common architecture revolution," *Johns Hopkins APL Tech. Dig.* **30**(3), 256–266 (2011).
23. UK Defence Standardization, *NATO Defence Standard 23–09: Generic Vehicle Architecture*, (UK Ministry of Defence, 2010).
24. L. Atzori, A. Iera and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.* **54**(15), 2787–2805 (2010).
25. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Commun. Surv. Tuts.* **17**(4), 2347–2376 (2015).
26. S. Hayat, E. Yanmaz and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.* **18**(4), 2624–2661 (2016).
27. D. I. Curiac, "Towards wireless sensor, actuator and robot networks: Conceptual framework, challenges and perspectives," *J. Netw. Comput. Appl.* **63**, 14–23 (2016).
28. C. Ghedini, C. H. C. Ribeiro and L. Sabattini, "Toward efficient adaptive adhoc multi-robot network topologies," *Ad Hoc Netw.* **74**, 57–70 (2018).
29. D.G. Reina, H. Tawk and S.L. Toral, "Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks," *Ad Hoc Netw.* **68**, 16–32 (2018) .
30. E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner and C. Bettstetter, "Drone networks: Communications, coordination, and sensing," *Ad Hoc Netw.* **68**, 1–15 (2018).

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press

31. H. Ahmadzadeh and E. Masehian, "Modular robotic systems: Methods and algorithms for abstraction, planning, control, and synchronization," *Artif. Intell.* **223**, 27–64 (2015).
32. H. Ihara and K. Mori, "Autonomous decentralized computer control systems," *Comput.* **17**(8), 57–66 (1984).
33. A. G. O. Mutambara and H. E. Durrant-Whyte, "Estimation and control for a modular wheeled mobile robot," *IEEE Trans. Cont. Syst. Tech.* **8**(1), 35–46 (2000).
34. H. Hamann, J. Stradner, T. Schmickl and K. Crailsheim, "A Hormone-Based Controller for Evolutionary Multi-Modular Robotics: From Single Modules to Gait Learning," *IEEE Congress on Evolutionary Computation* (2010) pp. 1–8.
35. C. Yang, V. Vyatkin and C. Pang, "Model-driven development of control software for distributed automation: A survey and an approach," *IEEE Trans. Sys. Man Cybern. Syst.* **44**(3), 292–305 (2014).
36. R. C. Foust, E. S. Lupu, Y. K. Nakka, S.-J. Chung and F. Y. Hadaegh, "Autonomous in-orbit satellite assembly from a modular heterogeneous swarm," *Acta Astronaut.* **169**, 191–205 (2020).
37. W.-M. Shen, B. Salemi and P. Will, "Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots," *IEEE Trans. Robot. Auto.* **18**(5), 700–712 (2002).
38. M. Di Francesco, G. Anastasi, M. Conti, S. K. Das and V. Neri, "Reliability and energy-efficiency in IEEE 802.15.4/ZigBee sensor networks: An adaptive and cross-layer approach," *IEEE J. Sel. Areas Comm.* **29**(8), 1508–1524 (2011).

https://doi.org/10.1017/S0263574720001010 Published online by Cambridge University Press