

Design and Operation of a Novel Hexapod Robot for Surveillance Tasks

Fernando Gomez-Bravo¹, Pablo Villadoniga¹,
and Giuseppe Carbone²(✉)

¹ Electronic, Computing and Automation Engineering
Department – DIESIA-Engineering School, University of Huelva,
Carretera Palos de La Frontera s/n, 21819 Palos de La Frontera, Huelva, Spain
{fernando.gomez, pablo.villadoniga}@diesia.uhu.es

² DICEM, LARM: Laboratory of Robotics and Mechatronics,
University of Cassino and South Latium,
Via Di Biasio 43, 03043 Cassino, FR, Italy
carbone@unicas.it

Abstract. In this paper, a novel hybrid hexapod legged robot R3HC is proposed as based on a collaboration between Universities of Huelva and University of Cassino. This robot has been designed and built for patrolling areas where flat surfaces are mixed with obstacles and/or small unevenness. Its main design features, which are based on previous experiences with the Cassino Hexapod robot series, are also discussed. Both hardware and software are aiming at low-cost and user-friendly features as also described in the preliminary tests.

Keywords: Robot design · Surveillance · Hybrid wheel-legged vehicles · Path planning · Obstacle avoidance

1 Introduction

Hybrid walking robots are wheeled-legged mechanisms that move autonomously by walking on legs and also by rolling on wheels [1–6]. They are attracting the interest of the scientific community due to their significant advantages when trekking on small obstacles and uneven terrains. These features can be very useful for example while patrolling large indoor areas in industrial plants or nuclear power plants. In fact, hybrid hexapods robots present a stable structure that allow a suitable rolling locomotion over flat ground surfaces in combination with the ability to overcome small obstacles such as ditches, crests, and steps, whose size is comparable with half the robot own height as also detailed in [7–12]. In the frame special mention should be given to the hybrid hexapods developed at Laboratory of Robotics and Mechatronics (LARM) in Cassino aiming specifically at exploration of cultural heritage sites [10].

This work focuses attention on the design of a new hybrid hexapod, named R3HC (3 Revolute Huelva-Cassino Robot), that has been developed at University of Huelva as inspired by the Cassino Hexapod robots family. The novelty of the design is identified in the kinematic architecture, which incorporates an additional revolute joint for the rotation of each leg. This feature improves the maneuverability, while increases

the control complexity with a total of six additional motors to be controlled. Accordingly, control hardware and software have been fully redesigned to keep a suitable user-friendliness of the robot operation as a key for patrolling and surveillance tasks.

This paper presents the key aspects of R3HC robot in terms of mechanical design as well as hardware and software for the robot operation. Software has been particularly designed to allow the robot to overcome an obstacle by lifting the legs while the robot is rolling on wheels. For this purpose, a motion planning procedure is proposed also as based on the methodology the authors have proposed in [11]. The proposed approach calculates optimum path planning trajectories also by considering obstacle sizes and power consumption efficiency. Preliminary experimental tests have been reported to validate the proposed design of R3HC as well as its operation strategies.

The paper is organized as follow. In Sect. 2 the description of the mechanical design is presented. Section 3 is devoted to the introduction of the hardware architecture. In Sect. 4 the software architecture is described. Section 5 is dedicated to present the experimental results. The paper closes with the conclusions.

2 Wheeled Hexapods

2.1 Cassino Hexapod Robots

Cassino Hexapod is a series of hybrid legged-wheeled mobile robots that has been conceived and built at LARM laboratory in Cassino starting from year 2000. Solutions have been ranging from Cassino Hexapod version I, Fig. 1(a), to Cassino Hexapod version II, Fig. 1(b), to Cassino Hexapod version III, Fig. 1(c). Main characteristic of the Cassino Hexapod series is its low-cost and user-friendliness as based on a combination of legs and wheels for the inspection and operation in non-accessible sites, such as Montecassino Abbey, [10]. Cassino Hexapod consists of six legs and proper walking motion is possible by synchronizing the motion of all the leg joints. In particular, each leg has on board three DC motors. Thus, a total of 18 DC motors are controlled for its operation. Namely, each leg has two revolute joints (with a motor each) and an additional motor actuates the wheel at the leg extremity.

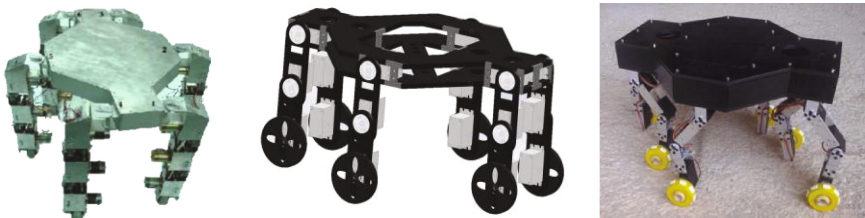


Fig. 1. The Cassino hexapod robots

The evolution of Cassino Hexapod from version I to III has consisted in significant changes in the robot structure as well as in its overall size, which is $0.6 \text{ m} \times 0.5 \text{ m} \times 0.5 \text{ m}$ for version I, $0.3 \text{ m} \times 0.25 \text{ m} \times 0.25 \text{ m}$ for version II, and

0.4 m × 0.3 m × 0.3 m for version III. The latest version is also equipped with omni-wheels that allow the robot to turn and move in any planar direction.

2.2 The Robot R3HC

R3HC is a wheeled-legged robot as shown in Fig. 2. Each of the leg incorporates a twisting joint that connects the leg with the robot’s body, (see Fig. 3). By controlling these articulations and the wheels speed, R3HC is capable of describing smooth trajectories and rotating about itself.

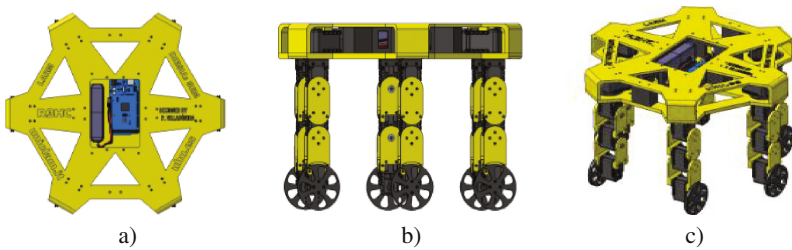


Fig. 2. A 3D CAD model of R3HC robot: (a) top view; (b) side view; (c) 3D view.

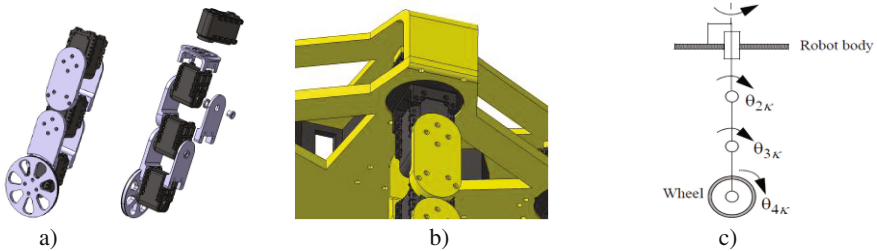


Fig. 3. Leg schemes: (a) leg assembly; (b) detail of the twisting joint; (c) a kinematic scheme.

Legs are numbered from 1 to 6, Fig. 3(c) shows the articular diagram of leg k and its 4 joints ($\theta_{ik}, i = 1..4$). When robot navigates on wheels, motion is controlled by changing θ_{1k} and $\omega_{k4} = \dot{\theta}_{k4}$, being k the indexes of the wheels on ground. On the other hand, if leg k is involved in an obstacle stepping maneuver, leg motion is controlled by changing θ_{2k} and θ_{3k} . Each leg has a length of 0.21 m, and the robot fits into a cylinder of 0.26 m high and a diameter of 0.40 m, and has an overall weight of 300 N.

Each joint is driven by a digital servomotor, thus, the robot has a total of 24 servomotors. Each one has its own digital controller responsible of moving the joint to the reference received from the main controller. Coordination of motions is implemented in an Arduino type board, which is carried on top of the structure. Next sections are devoted to present the main characteristic of this control board and the features of the software developed for robot motion control.

3 R3HC Hardware Architecture

Hardware architecture of R3HC is based on an Arduino Mega 2560 board [12] connected to 24 Dynamixel AX-12A servomotors [13]. The block diagram representing this architecture is shown in Fig. 4. This Arduino board supports a microcontroller Atmel ATmega2560 [14], with 256 Kb of FLASH memory and 4 Kb of RAM memory.

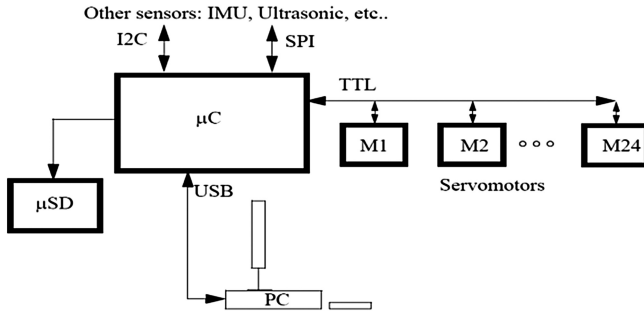


Fig. 4. Hardware architecture: block diagram.

The board is provided with an USB port for connecting a Personal Computer (PC) and an auxiliary shield that contains a μSD for saving data after the experiment. The board is also prepared with an I2C and a SPI bus for including other types of sensor: ultrasonic, IMU, temperature etc. Communications with Dynamixel servomotors are supported by TTL levels, following a half-duplex protocol that uses the same wire for transmitting and receiving data. All servos are connected to a linear bus. The microcontroller sends packets with instructions and the address of the recipient servomotor (see Fig. 5). In this application, instructions can be: an angular reference, a velocity reference or a request for reading the encoders. Servomotors reply with the required value and the address that identifies them, see Figs. 5 and 6.

Transmission and reception are implemented in two different ports of the microcontroller.

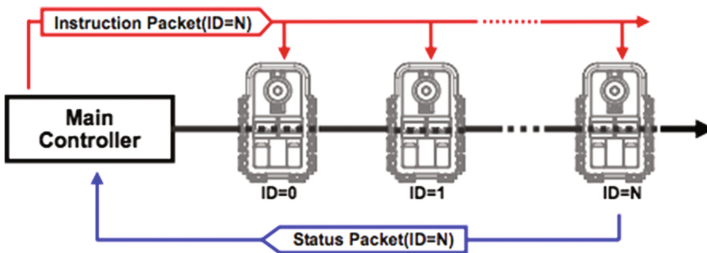


Fig. 5. Communication protocol between microcontroller and servomotors.

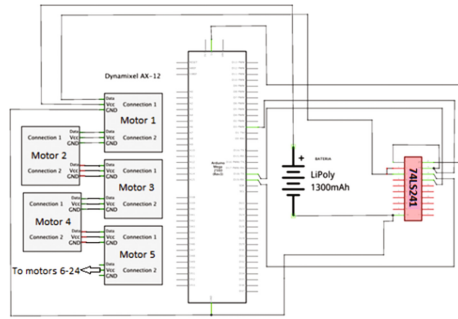


Fig. 6. Motors connection: electric scheme.

In order to make compatible this feature with the one wire communication protocol, the buffer driver with 3-State Outputs 74LS241 [15] is employed as an interface between the microcontroller and the motors. The electric scheme of the circuit is represented in Fig. 6. Servomotors are connected in a cascade of three wires connection, implementing a linear bus for communications and power supply.

4 R3HC Software Architecture

Robot motion control is based on a main program that allows establishing the target of the robot navigation, determines the control references, and manages communication with the servomotors. The flowchart of this module is represented in Fig. 7. This software has been specially designed for implementing a navigation strategy that avoids collisions by considering obstacle stepping. This design is inspired in the methodology reported in [11], where the steps were computed in such a way that robot does not stop during the leg lifting movements. Thus, in accordance with that technique, an optimal step motion has been previously determined, so that the robot can execute leg movements, at the moment it is required and overcome the obstacle while the navigation velocity keeps constant.

As it is shown in Fig. 7, the program starts by defining the navigation target. This target can be, either a specific pose to be achieved, a path to be followed, or a place where the robot should be placed. Before that, the program runs a continuous loop that evaluates the pose of the robot and establishes if the objective has been achieved or not. Within the loop, the program is prepared to detect the existence of an obstacle to be overcome. On the one hand, if no obstacle is detected, the program determines the modulus of the linear and angular velocities of the body of the robot (V and Ω), by taking into account the current pose, and the navigation target. These values are established so that the robot moves toward the target. Then, from V and Ω , the correspondent values of θ_1 and ω_4 of each leg are computed and sent to the servomotors. Before that, the values of the encoders are read, and the pose of the robot is estimated.

On the other hand, if an obstacle is detected, the program evaluates if the distance to the robot is appropriate to start the leg lifting motion. In the case it is, the program selects

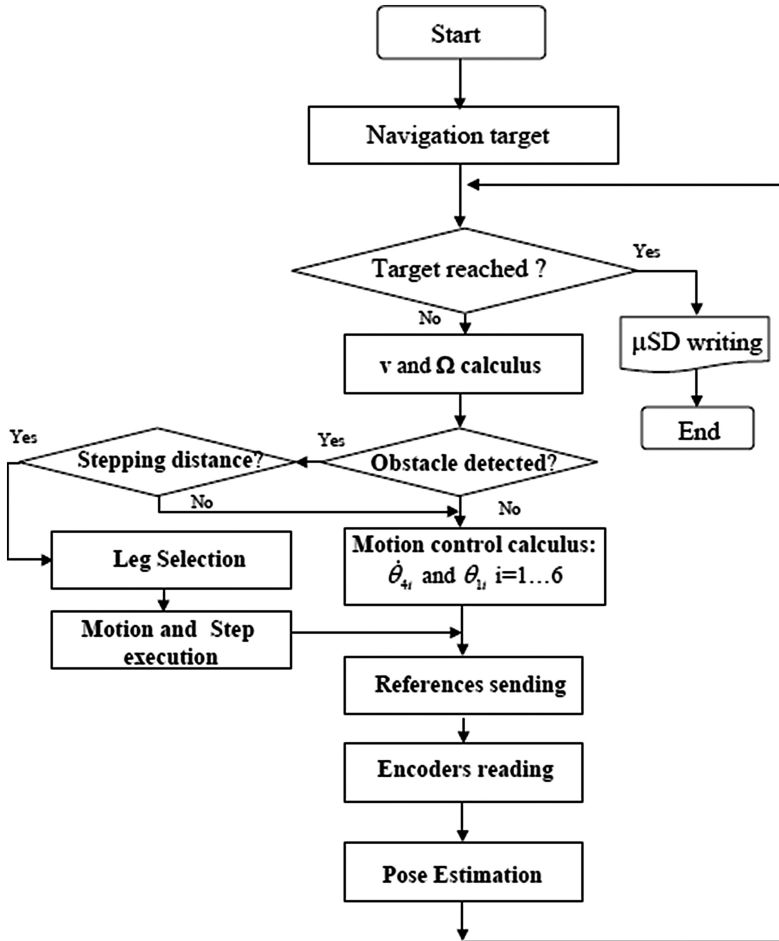


Fig. 7. Main program flowchart.

the leg to lift and the module ‘motion and step execution’ coordinates the execution of both: continuous navigation and leg lifting. The flowchart of this module is illustrated in Fig. 8(a). This software, is the responsible of determining the value of the joint 2 and 3 of the selected leg k (i.e., θ_{2k} , θ_{3k}) and the values θ_{1i} and ω_{4i} of the legs i that keep on ground. The flowchart of the module for computing the values of the joint of the leg raising is represented in Fig. 8(b). It takes into account the current time, and the joints reference trajectory, which has been calculated off-line. Then, by means of a linear interpolation the module provides the corresponding joint values for each moment. That way, while the robot is overcoming the obstacle, it keeps moving forwards to the target as referring to one leg in the flowchart. However, this procedure is easily extended to a step involving two or three legs, according to the methodology reported in [11]. In the following section, several experiments will be shown, one of them involves a two leg obstacle stepping maneuver.

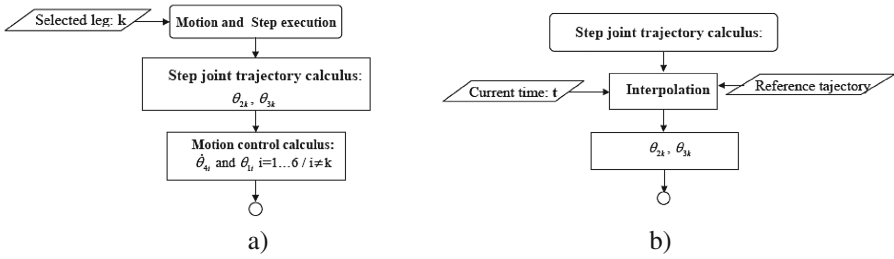


Fig. 8. Flowcharts: (a) Motion and step execution; (b) Step joint trajectory calculus

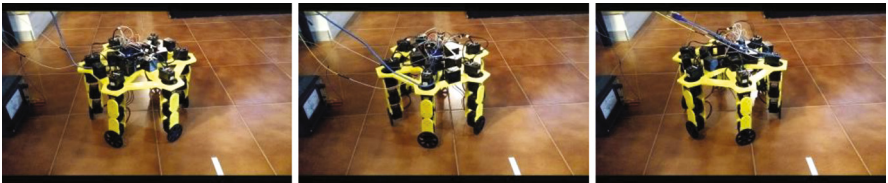


Fig. 9. Snapshots of R3HC experimenting a turning about itself.

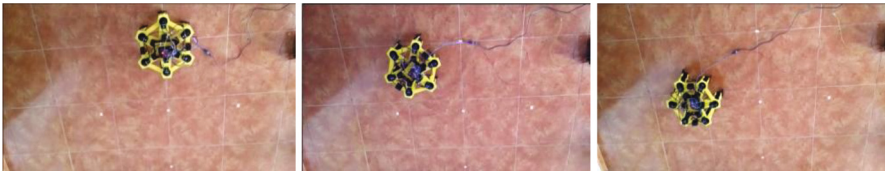


Fig. 10. Snapshots of R3HC following a circumference path.

5 Experimental Result

Many different experiments have been carried out. In Fig. 9, the velocity of the central point of the robot is $V = 0$ and the angular velocity is $\Omega = 0.2$ rad/s. No obstacles are present and the robot move by rolling on the six wheels. The twisting joints are oriented in such a way that the wheels describe a circumference centered in the robot’s center, so it turns around itself. Figure 10 illustrates an experiment where $V = 0.5$ m/s and $\Omega = 0.2$ rad/s. In this case, the twisting joints are oriented so that the robot describes a circumference of radius equal to 0.5 m. This experiment demonstrated the robot capability for negotiating a curved trajectory. In the case of a particular curved path, V and Ω change continuously, then the main program will calculates each moment, the correspondent values for the twisting joint and the wheels velocities (Fig. 11).

The last experiment illustrates an obstacle stepping maneuver. The robot is moving forward, and an obstacle appears opposite to the front and rear right legs. Once the front leg is at the correct distance, the robot starts lifting it, without stopping the

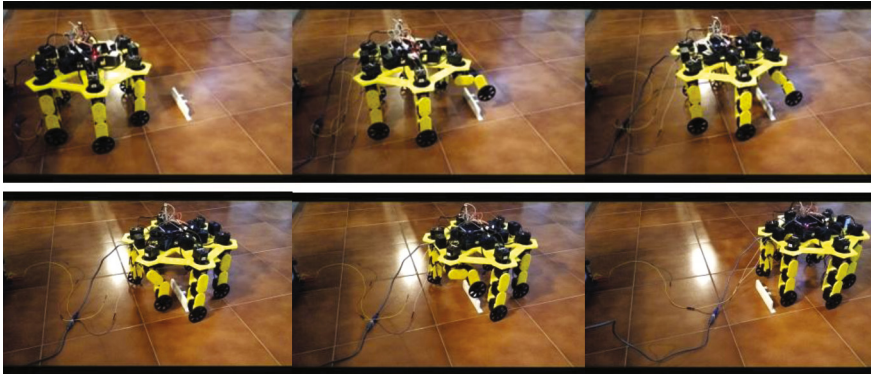


Fig. 11. Snapshots of R3HC in an obstacle overcoming maneuver.

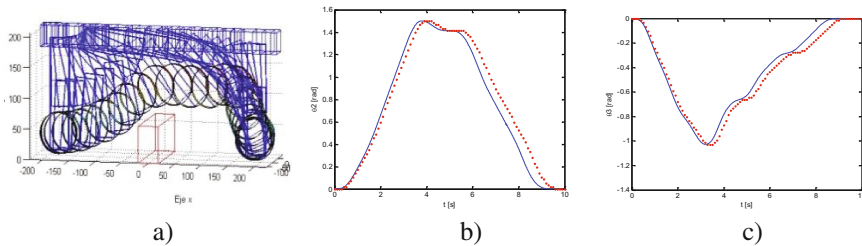


Fig. 12. Step motion: (a) leg motion; (b) trajectory of joint θ_2 ; (c) trajectory of joint θ_3 .

forward motion. The leg executed the step motion shown in Fig. 12(a), by following the joint trajectories illustrated in Fig. 12(b) and (c). The reference values are represented with continuous line. The values provided by the motor encoders are shown with a dotted line. The motion of the rear leg follows a similar procedure. Notice that, in order to maintain forward, both wheels velocities and twisting joint of the other legs keep receiving the correspondent references, as is described in Sect. 4.

6 Conclusions

This paper presents a new robotic hybrid hexapod named as R3HC, based on the Cassino hexapod family with an additional twisting joint at each leg. The mechanical design has been described and detailed description of hardware and software architecture has been provided. This control strategy, combined with the new mechanical design, makes the robot capable of turning about itself, and following trajectories with arbitrary curvature. Software has been particularly designed to avoid obstacles by performing optimal obstacle stepping maneuvers without stopping the robot. Preliminary experiment results show the efficiency of the proposed design.

References

1. Burkus E, Odry P (2007) Autonomous hexapod walker robot Szabad(ka). In: SISY 2007, 5th international symposium on intelligent systems and informatics, Subotica, Serbia, pp 103–106
2. Carbone G, Ceccarelli M (2005) Legged robotic systems. In: Cutting edge robotics ARS scientific book, Wien, pp 553–576
3. Tedeschi F, Carbone G (2014) Design issues for hexapod walking robots. *Int J Robot* 3(2):181–206
4. Grand C, Benamar F, Plumet F, Bidaud P (2004) Stability and traction optimization of a reconfigurable wheel-legged robot. *Int J Robot Res* 23(10–11):1041–1058
5. Smith TB, Barreiro J, Smith DE, SunSpiral V, Chavez-Clemente D (2008) ATHLETE's feet: multi-resolution planning for a hexapod robot
6. Aras M, Shahriyel M, Anuar MK, Annisa J (2012) Development of hexapod robot with manoeuvrable wheel. *Int J Adv Sci Technol* 49:119–136
7. Carbone G, Shrot A, Ceccarelli M (2007) Operation strategy for a low-cost easy operation cassino hexapod. *Appl Bion Biomech* 4(4):149–156
8. Carbone G, Ceccarelli M (2008) A low-cost easy-operation hexapod walking machine. *Int J Adv Robot Syst* 5(2):161–166
9. Carbone G, Gomez-Bravo F, Selvi O (2012) An experimental validation of collision free trajectories for parallel manipulators. *Mech Based Des Struct Mach Int J* 40(4):414–433
10. Cigola M, Pelliccio A, Salotto O, Carbone G, Ottaviano E, Ceccarelli M (2005) Application of robots for inspection and restoration of historical sites. In: Proceedings of 22nd international symposium on automation and robotics in construction, ISARC 2005, Ferrara, Paper 37
11. Gómez-Bravo F, Aznar MJ, Carbone G (2014) Planning an optimal step motion for a hybrid wheeled-legged hexapod. In: New advances in mechanisms, transmissions and applications, pp 207–214
12. Arduino Webpage: Arduino Mega2560. <https://www.arduino.cc/>. Accessed 28 Jan 2017
13. Dynamixel Webpage: AX-12A. <http://www.robotis.us/dynamixel>. Accessed 28 Jan 2017
14. Atmel Webpage: ATmega2560. <http://www.atmel.com/devices/atmega2560.aspx>
15. <http://www.alldatasheet.es/datasheet-pdf/pdf/171558/TI/74LS241.html>