International Journal of Engineering & Technology, 7(4.16) (2018) 71-75

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Implementation of a Three-Legged Omnidirectional Hopping Robot

Elmer R. Magsino[1]*, Gerald P. Arada[2], Martin Klarence V. Manalo[3], Maxine Claryze C. Micu[4], Jona Liza M. Montrias[5], and Giecel T. Palo[6]

[12345]De La Salle University, Manila, Philippines
**Corresponding author E-mail:elmer.magsino@dlsu.edu.ph*

Abstract

In this work, we present the design and implementation of a three-legged robot that employs hopping as a means of locomotion while maintaining stability throughout the motion process. The developed three-legged hopping robot follows a tripod structure to house all the electronics and actuators. It is also radio-controlled, thereby, allowing flexibility and range to its users to control the direction and movement. A commercially off-the-shelf Arduino-based microcontroller is used to implement sensing, control and actuation of the tri-pod. The hopping mechanism is dependent on a heuristic approach by knowing beforehand the maximum height the three-legged robot can clear during its hop. During locomotion via hopping, a chosen leg is set to provide a different output force from the two other legs in order to move the robot to a specified direction. We tested the platform on even and uneven surfaces to determine its performance while maintaining stability.

Keywords: *Arduino-based Microcontroller; Hopping Algorithm; Omnidirectionality; Three-legged Hopping Robot*

1. Introduction

Mobile robotic locomotion is characterized depending in its means to move. Generally, mobile robots are classified as wheeled and legged robots. Wheeled robots operate on wheels to navigate its surroundings but are limited to an almost flat terrain. They move fast and are very stable. In environments where wheels are not possible for transport, legged robots are the choice. However, stability and speed in movement are concerns during design and implementation [1]. Legged robots have already garnered a vast research attention and breakthroughs. Intuitively, the number of legs is derived from nature, such as humans, dogs, spiders, centipedes, octopus, etc. [2]-[5].

However, one type of legged machines that is not based on nature is a robot with three legs, though making it move is based on nature, i.e., hopping [6]. Hopping or rhythmical jump is categorized under jumping, which is a bio-inspired robot movement [7]. Jumping in nature is a means of avoiding enemies or capturing food or traversing a path. The jumping patterns are categorized as (1) pause and leap and (2) continuous hopping [8].

In [9], miniature robots can jump based on storing and releasing elastic energy. This is inspired by observing fleas. Another miniature jumping robot is developed in [10]. Its total mass is seven (7) grams, has a height of 5 cm, and overcomes a maximum obstacle height of 1.4 meters. It also details how its jump energy is computed and how the hardware is designed. The jumping mechanism, alongside a vision system, is used to autonomously climb a stair in [11].

Most published research deals with miniature hopping robot and uses either single or dual legs for jumping, except the one found in [7], thus we developed our three-legged platform based on this. In [7], a Central Pattern Generator (CPG) is used for providing a periodic pattern for each leg. This pattern dictates the movement and posture of the robot during hopping.

In this research, we propose a heuristic approach of implementing a hopping mechanism for a three-legged robot. Initially, a predetermined maximum height is assumed to be given, therefore, allowing us to pre-define the necessary force needed by each leg to move in a specific direction. The algorithm is characterized by using a finite state machine.

The paper is structured as follows. Section 2 presents the hardware design of our three-legged robot. Section 3 discusses how we implemented the heuristic hopping approach to the developed platform. We present results in Section 4. Finally, we conclude our work in Section 5. We also present future prospective investigation in Section 5.

2. Design and Construction of the Three-Legged Robotic Platform

Figure 1 shows the general framework of the three-legged robot. The entire robot consists of the following major components: (1) radio controller, (2) receiver, (3) controller module, (4) hopping mechanism, and (5) feedback. We discuss each block below.

The *Radio Controller* is used to input commands wirelessly from the user to the robot. We used an off-the-shelf remote-controlled car controller and customized it according to our desired application. The *Receiver* located at the platform receives the commands and sends them to the controller module. The *Controller Module* converts user commands and the data from the sensors to be processed in the hopping algorithm. These enable the operation of the hopping mechanism. The *Hopping Mechanism* consists of three legs that have springs which continuously compress and decompress in a quick manner in order to lift the robot from the ground, thus performing a hopping movement.

The spring of each leg compresses at varying speed with respect to the other legs to enable hopping at the desired directions. Finally, *Feedback* constitutes the gyroscope from the IMU (Inertial Measurement Unit) module to be used for gathering direction data.

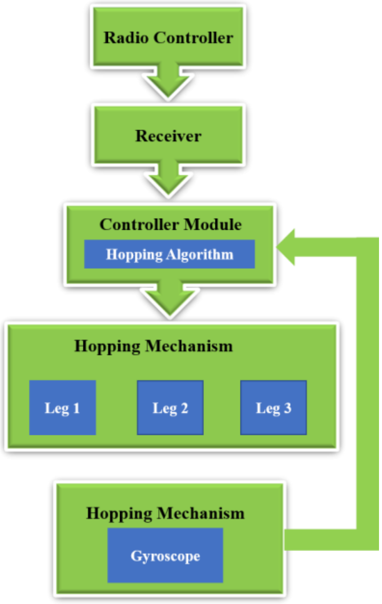

Figure 1: Block Diagram of the Three-Legged Hopping Robot

Figure 2 illustrates the isometric view of the hopping robot design. We discuss and enumerate each robotic part below.

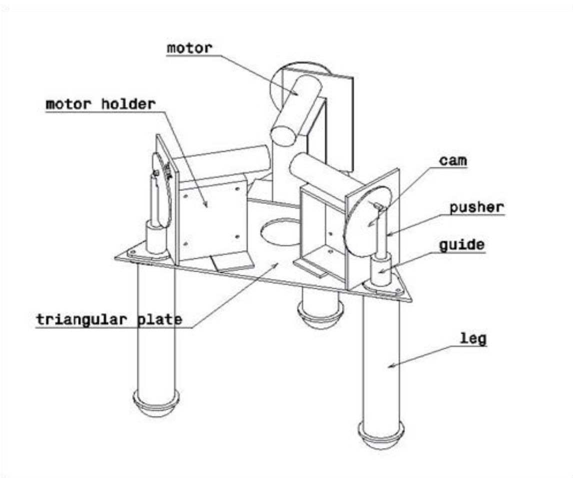

Figure 2: Isometric View of the Three-Legged Robotic Platform

The *Triangular Plate* is the main frame of the robot that holds all electronics and mechanical parts. The *Motor* is the actuator employed to achieve the entire hopping mechanism. It rotates the cam continuously and aids the quick release mechanism for hopping. The *Motor Holder* stably holds the motor and prevents the motor from any type of movement. The *Cam* pushes the pusher downward as it rotates counter-clockwise. The pointed tip quickly releases the spring. The *Pusher* has a slotted upper part that glides through the sides of the cam as it is being pushed while its flat and wider lower part pushes the spring downward. The *Guide* keeps the pusher in place. Lastly, the *Leg* keeps the spring in place. Its bottom part acts as an object that is being pushed and is lifted from the ground.
The constructed three-legged robotic platform is shown in Figure 3 and the specifications are given in Table 1. The triangular platform is made from acrylic plastic while the legs are from carbon steel.

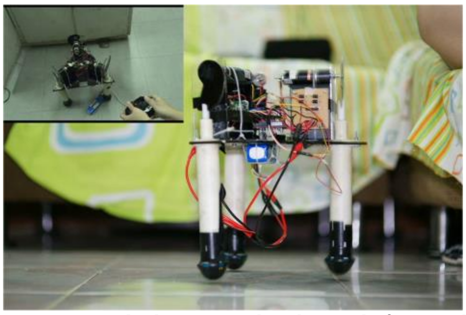

Figure 3: Constructed Three-Legged Robotic Platform with a wireless controller (inset)

Table 1: Robot Specifications

Parameter	Value
Triangular dimension (each side)	304.8 mm
Total Height	338.5 mm
Motors	24 VDC Maxon 11873
Power Supply	14.8V 1800 mAH 4S
Controller Range	4 m
Controller Frequency	49 MHz
Controller Type	2-channel Wireless Radio Controller
Spring	185 mm, 25-coil
Spring Constant	0.251 N/mm

Figure 4 presents the four hopping states undergone by the robot for the proposed heuristic hopping approach during the forward movement. In each state, the compression value of the spring inside each leg varies as well as the vertical direction movement of the robot. We determine these values through experiments. In our work, these values allow the robotic platform to move in any direction at a stable manner. The values in *Leg* variables represent the percent displacement (compression value) of the spring from its original length while the values in *Vdir* variable represent the vertical direction of the robot body.

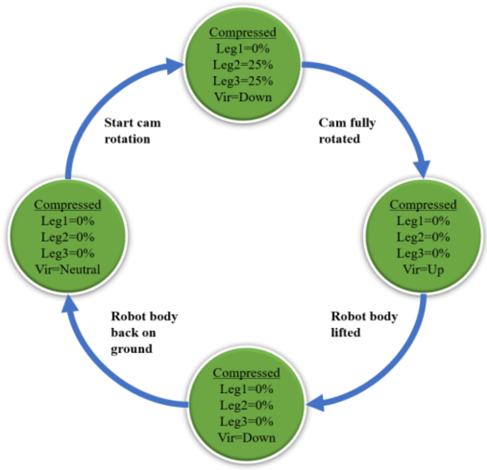

Figure 4: Finite state model for the hopping algorithm for the forward direction

3. Heuristic Hopping Mechanism

In the *Original State*, all springs (inside the legs) are in their uncompressed state, signifying a compression value of 0%. Furthermore, the vertical direction is neutral since there is no change yet in the compression values.
The *Compressed State* begins once the cam started rotating. In this state, two legs have the same compression values and larger than the remaining leg to enable hopping in certain directions. We denote that Leg1 is the front leg while the other two are at the hind legs. In order to move forward, Leg2 and Leg3 must be more compressed than Leg1 since the more compressed the spring, the

higher the stored jump energy, thus, the higher the upward force when released.

In the *Released State*, which begins when the cam has fully rotated, the springs quickly decompress and return to their original lengths by moving upward, thereby, pushing the body also in an upward direction. Since the legs have different compression values, their release or decompression forces also vary. The legs with larger and quicker forces on them tend to lift that part of the body higher than the other one which drives the robot to the direction of the leg with lower compression value.

The *Land State* begins after the robot body has been lifted. In this state, the compression values of the legs remain but the vertical direction of the robot changes from upward to downward, bringing the robot body back to the ground. As the robot continuously hop, it returns to its original state, becomes compressed, then undergoes the same cycle of states again.

Figure 5 demonstrates the hopping mechanism of the robot. In this figure, the condition of the leg is presented, particularly the compression of spring and rotation of cam in each state.

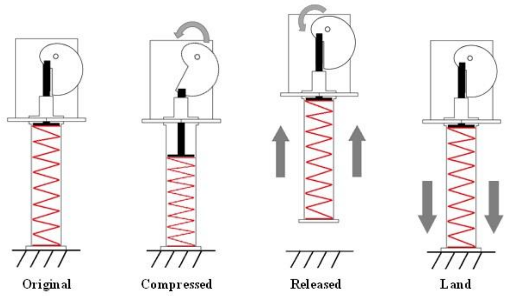

Figure 5: Illustration of the hopping mechanism

The flowchart shown in Figure 6 illustrates the program flow of the project. The main program uses the switch-case style in calling other function.

Five cases are used in the main program: four cases for the direction of the hopping robot, and one default case that will be executed when no input is detected from the remote controller. The PWM for each motor in a direction does not necessarily have to be the same. This is a case by case basis and has a lot of factors affecting it, such as the current state of the mechanical components of the robot

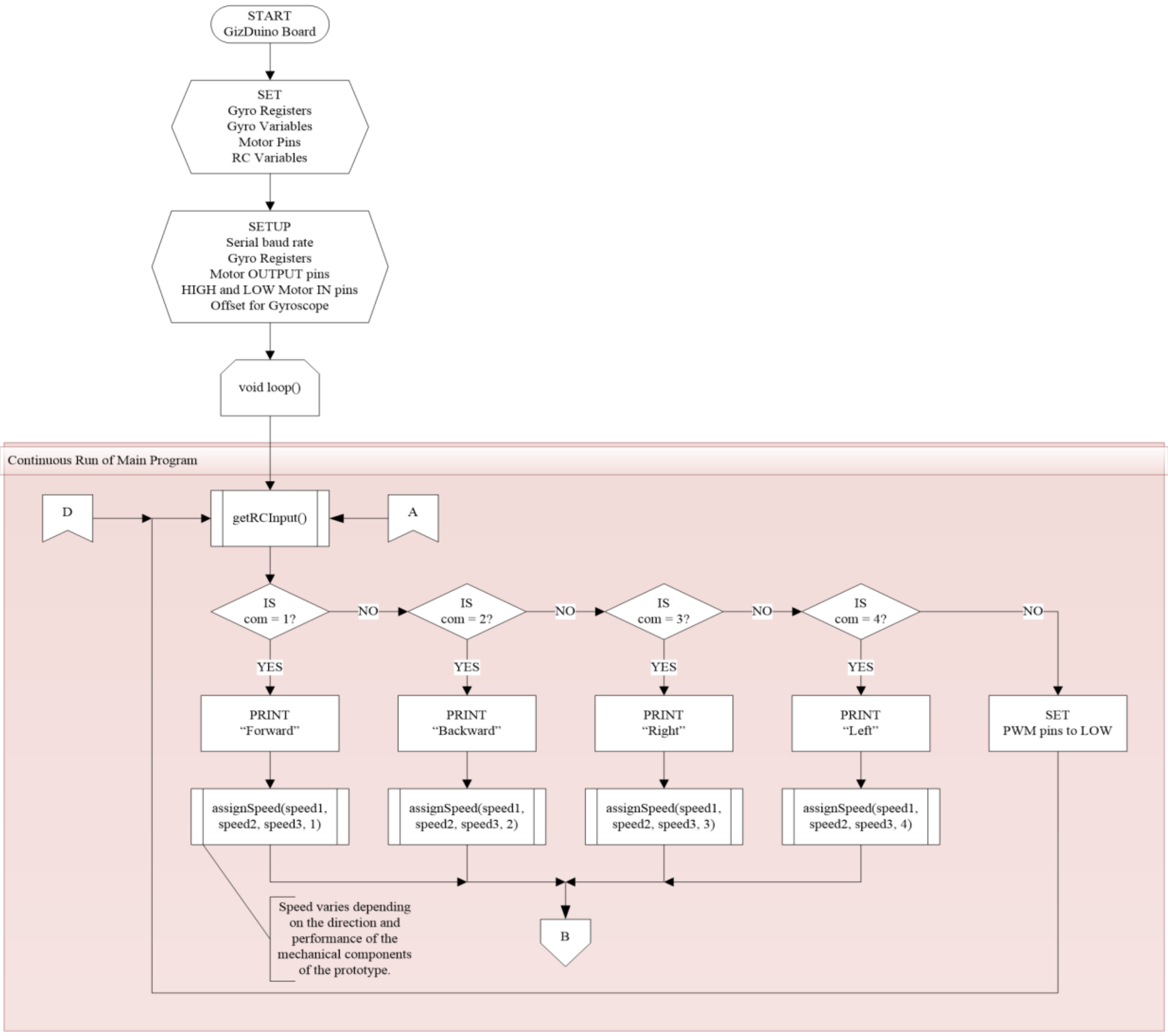

Figure 6: Implementation of the Hopping Algorithm

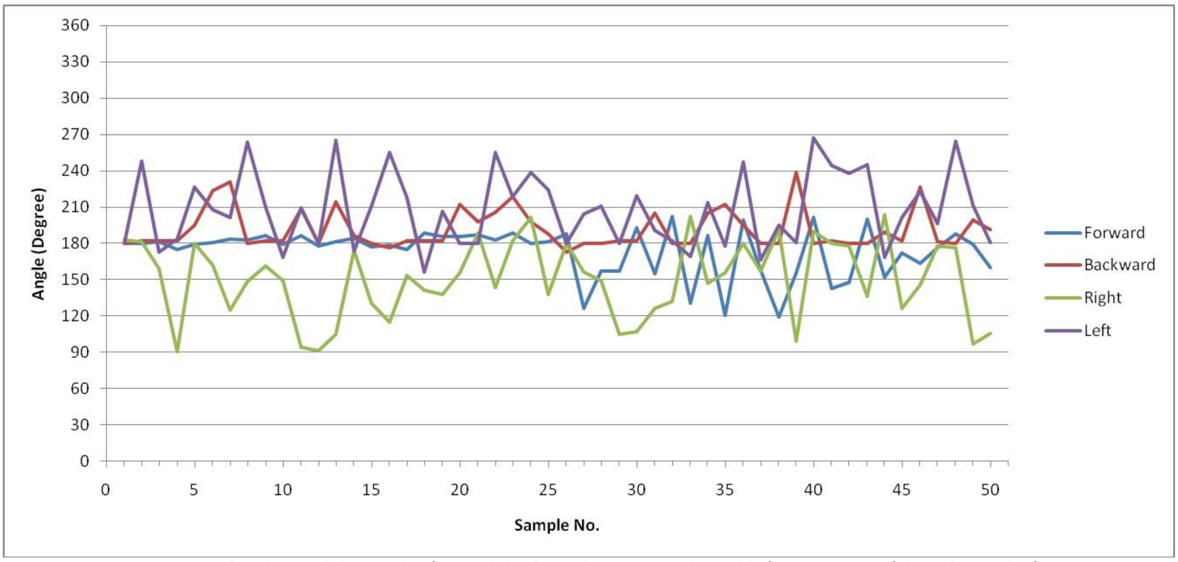

Figure 7: Results obtained during the forward, backward, turning right and left movement of the robotic platform.

There are three main functions in the program: getRCInput(), assignSpeed(), and computeZangle().

The getRCInput() function detects the data sent by the radio controller and converts it into a command that the Arduino program will understand. There are four commands that can be executed: Forward, Backward, Right, and Left.

The assignSpeed() function contains the code that writes the PWM to the three motors, printing its current values, and calling another function, the computeZangle().

The computeZangle() uses the raw value from the gyroscope of the IMU and converts it to angle with the use of some math equations. A starting angle of 180° and offset is applied here so that each start of the program has a common basis. To compute for the angle, the rate (deg/s) is multiplied to the time that passed between each reading. After converting it to an angle, the values are limited to 0° – 360°, and the final value is printed.

4. Experimental Results

The three-legged robotic platform was tested to determine how it moves towards a given direction. We tested four directions, namely: (1) Forward, (2) Backward, (3) Turn Right and (4) Turn Left. For each direction, we performed 10 trials and obtained the first 50 samples from the gyroscope. The data collected are based from the readings of the gyroscope while the robot is in hopping motion in a specific direction. All trials are performed under the following conditions: (1) The initial angle reading at the gyroscope is 180°. (2) The readings are sampled every 100ms. (3) The Z-Angle (Yaw) for gyroscope is used to determine where the robot is facing.

Figure 7 depicts the gyroscope response. We observe that by getting the average (plus a certain threshold value) of each movement, we can then determine the state where the robot is currently performing.

Figure 8 and Figure 9 demonstrate the hopping sequences of the robot on an even and uneven surface respectively. In these trials, the robot is controlled to move around the area.

It can be observed that the leg with the moving cam induces hopping on that particular leg and the high-bouncing ball underneath the legs assist in slightly increasing the hopping height. Furthermore, the direction of rotation of the cam reinforces the movement of the robot to a particular direction.

Figure 8 demonstrates the actual performance of the robot when hopping on an uneven surface. The uneven surface looks like a stair having six steps where each step is made of ¼ inch-thick plywood. As seen in Figure 8, it can be observed that the robot can hop down without losing balance given that the height of each step is small. Furthermore, since the robot is omni-directional, the robot had no problem moving around it.

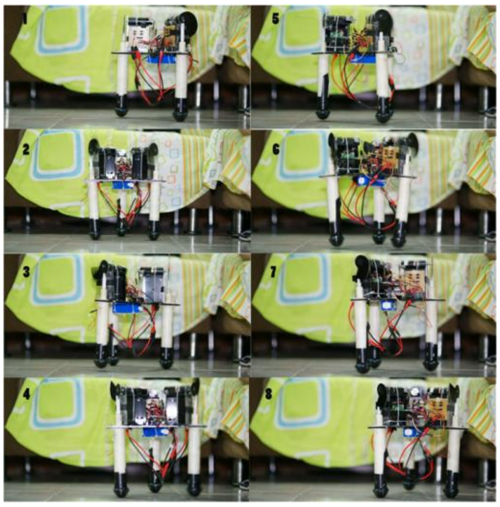

Figure 8: Snapshots of the hopping robot on an even surface

Figure 9: Snapshots of the hopping robot on an uneven surface

5. Conclusion

In this research work, the design and construction of a three-legged platform has been presented. We have developed the robotic platform based on components and parts that are readily available off-the shelf and are found in most typical hardware and electronic shops. The three-legged platform can move in any direction via remote wireless control and on an even and uneven surface. The simple method of quick-release cam and follower mechanism proved to be an effective and reliable hopping mechanism.

We introduced a simple heuristic approach to achieve hopping. We have done this by manually tuning the compression of the springs based on gathered experimental results. In the controller, we have hard-wired the different compression values of the springs to implement the omni-directionality of the robot. While in motion, the controller obtains the instantaneous average of the gyroscope reading to determine if indeed the desired motion is attained.

For future work, we intend to improve the heuristic approach of performing the hopping algorithm. Currently, the hopping robot can only achieve a very small jumping height of 1cm. We want to achieve a hopping/jumping capability of the robot to be at least 5 cm.

References

[1] E. R. Magsino, "Implementation of a Walking Bipedal Robot using Position Control Algorithm," 2006.

[2] H. Kimura, T. Yamashita and S. Kobayashi, "Reinforcement learning of walking behavior for a four-legged robot," *IEEJ Transactions on Electronics, Information and Systems*, (2002), vol. 122, no. 3, pp. 330-337, 2002.

[3] A. M. Hoover, S. Burden, X.-Y. Fu, S. S. Sastry and R. S. Fearing, "Bio-inspired design and dynamic maneuverability of a minimally actuated six-legged robot," *3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics,* (2010), pp. 869-876.

[4] K. Karydis, I. Poulakakis and H. G. Tanner, "Probabilistic validation of a stochastic kinematic model for an eight-legged robot," *IEEE International Conference on Robotics and Automation)*, (2013), pp. 2562-2567.

[5] S. Inagaki, H. Yuasa, T. Suzuki and T. Arai, "Wave CPG model for autonomous decentralized multi-legged robot: Gait generation and walking speed control," *Robotics and Autonomous Systems*, (2006), vol. 54, no. 2, pp. 118-126.

[6] S.-H. Hyon and T. Mita, "Development of a biologically inspired hopping robot-" Kenken," *IEEE International Conference on Robotics and Automation Proceedings*, (2002), pp. 3984-3991.

[7] A. Kassim, M. Miskon, N. Rahim and T. Yasuno, "Moving motion control system on developed tripod hopping robot," *International Conference on Electrical, Control and Computer Engineering,* (2011), pp. 284-287.

[8] R. Armour, K. Paskins, A. Bowyer, J. Vincent and W. Megill, "Jumping robots: a biomimetic solution to locomotion across rough terrain," *Bioinspiration & Biomimetics*, (2007), vol. 2, no. 3, p. S65-S82.

[9] M. Noh, S.-W. Kim, S. An, J.-S. Koh and K.-J. Cho, "Flea-inspired catapult mechanism for miniature jumping robots," *IEEE Transactions on Robotics*, (2012), vol. 28, no. 5, pp. 1007-1018.

[10] M. Kovac, M. Fuchs, A. Guignard, J.-C. Zufferey and D. Floreano, "A miniature 7g jumping robot," IEEE International Conference on Robotics and Automation, (2008).

[11] S. Stoeter and N. Papanikolopoulos, "Autonomous stair-climbing with miniature jumping robots," IEEE Transactions on Systems, Man, and Cybernetics, Part B, (2005), vol. 35, no. 2, pp. 313-325.