

Research Article

A Wireless Sensor Network System with a Jumping Node for Unfriendly Environments

Jun Zhang, Guangming Song, Guifang Qiao, Zhen Li, and Aimin Wang

School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

Correspondence should be addressed to Guangming Song, mikesong@seu.edu.cn

Received 24 December 2011; Accepted 16 May 2012

Academic Editor: Jiming Chen

Copyright © 2012 Jun Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile robots have been adopted to repair failed wireless sensor network systems for node damage, battery exhaustion, or obstacles. But most of the robots use wheeled locomotion manner, which does not work well or even fails when confronted with obstacles in uneven terrains. To solve this problem, this paper presents the design of a jumping robot to serve as a robotic node for wireless sensor networks. The robot can jump up to or over obstacles to repair the broken network connections. The robot senses its posture angle by using an acceleration sensor and self-rights automatically by using a pole leg after falling down on the ground. The robot also can steer and adjust its take-off angle by the pole leg. A network monitoring system with the proposed robot is built to test its basic locomotion capabilities and the network repair function. Experimental results show that the robot can jump about 90 cm in height and traverse 50 cm far at a take-off angle of 75 degrees. The robot can repair the network by jumping up to a 10 cm high platform. The proposed system with a jumping node can provide powerful support for applications in unfriendly environments.

1. Introduction

With the development of microelectronics and sensor technology, wireless sensor networks technology is widely used in many fields such as environment monitoring [1], home automation [2], industrial control [3], and military application [4]. But network deployment and network repair are difficult tasks when the working environments are dangerous or even cannot be reached by humans. This problem can be solved by mobile sensor networks technology.

Mobile sensor network systems usually use mobile robots as sensor nodes to execute network deployment and monitoring tasks [5]. But mobile sensor networks need lots of mobile robots. The costs increase with the increase of the number of mobile sensor nodes. So it is a feasible way to combine a large number of cheap static sensor nodes with some mobile sensor nodes. It is efficient to deploy the static sensor nodes and mobile nodes cooperatively. The mobile nodes are able to enlarge coverage range and enhance connectivity of the network. They also can repair the network when there are disabled static sensor nodes.

Traditional mobile robots use wheeled locomotion manner. In [6, 7] wheeled robots are used in WSN systems to detect intruders in indoor environments. In [8], the authors use a wheeled mobile robot as sink node to cooperate with other static sensor nodes and form a mesh network. Wheeled robots have always been confronted with the problem of locomotion in uneven terrains with many obstacles. If obstacles are higher than the radius of their wheels, wheeled robots cannot overcome obstacles efficiently.

Researchers are beginning to study creatures and their locomotion manners to design robots with more efficient locomotion capabilities. For example, the bipedal robot in [9] based on studying human locomotion, the quadruped walking robot Little Dog in [10] based on studying four-legged animals, and the hexapod robot RHex in [11] based on studying six-legged insects. In [12], a flying robot is used to deploy and repair a sensor network.

In the nature, many creatures have the capabilities of jumping locomotion such as frogs, locusts, and kangaroos. These inspire researchers to design some robots with jumping capabilities such as the Jollbot in [13], the JPL Hopper

TABLE 1: The three-axis acceleration changes according to state transition of the jumping robot when self-righting.

Direction	x	y	z
Left side	$0 \rightarrow 0$	$\text{Max} \rightarrow 0$	$0 \rightarrow \text{Max}$
Right side	$0 \rightarrow 0$	$\text{Min} \rightarrow 0$	$0 \rightarrow \text{Max}$
Front side	$\text{Max} \rightarrow 0$	$0 \rightarrow \text{Max} \rightarrow 0$	$0 \rightarrow \text{Max}$

down on its left, right, or front side after landing on the ground. The automatic self-righting principle is as follows: when the jumping robot falls down on one of the three sides, the acceleration sensor can provide posture angle information for the robot. The pole leg rotates and the robot body will be propped up. The robot detects its posture angle periodically. When standing up, the robot will stop rotating the pole leg and start rotating it in the opposite direction to fold the pole leg up. The infrared sensor in the front part of the body frame will detect passing by of the pole leg. When the sensor finds the pole leg, the microprogrammed control unit (MCU) on the control board will control the pole leg to stop rotating.

The acceleration sensor is mounted on the foot of the robot. The surface of the sensor is parallel with the XY plane as shown in Figure 2. In theory, the three-axis acceleration values change with variations of the posture angle when the robot is self-righting as shown in Table 1. When the robot falls down on its left side the accelerations of the three axes are $x = 0$, $y = \text{Max}$, and $z = 0$. When the robot falls down on its right side, $x = 0$, $y = \text{Min}$, and $z = 0$. When the robot falls down on its front side, $x = \text{Max}$, $y = 0$, and $z = 0$. These characteristics are used by the robot to judge its initial states when it is falling down. After self-righting, $x = 0$, $y = 0$, and $z = \text{Max}$, this can be used by the robot to judge its righting state.

During self-righting from left side, y decreases from Max to 0 , z increases from 0 to Max , and x is 0 all the time. During self-righting from right side, y increases from Min to 0 , z increases from 0 to Max , and x is also 0 all the while. During self-righting from front side, x decreases from Max to 0 , y increases from 0 to Max and then decreases to 0 again, and z increases from 0 to Max .

Changes of the robot posture angle during self-righting are used to design self-righting control algorithm. The automatic self-righting algorithm is shown in Figure 3. After parameter initialization, the MCU samples sensor data periodically and judges the acceleration z (AZ) to see if it is larger than the threshold T_{Max} . If this is true, the robot will judge the absolute value of y $\text{abs}(AY)$. If it is larger than the threshold T_{Min} , the robot will know it falls down on its left or right side. If $AY > 0$, the robot will consider that it has fallen down on its left side and start to rotate adjusting motor clockwise. If $AY < 0$, the robot will consider that it has fallen down on its right side and start to rotate adjusting motor anticlockwise. If $\text{abs}(AY) < T_{\text{Min}}$, or the robot finishes one step of self-righting, the robot will judge acceleration x (AX) to see if it is larger than T_{Min} . If it is true, the robot considers that it falls down on its front side. Then the jumping motor starts to rotate until the infrared sensor in the left part of

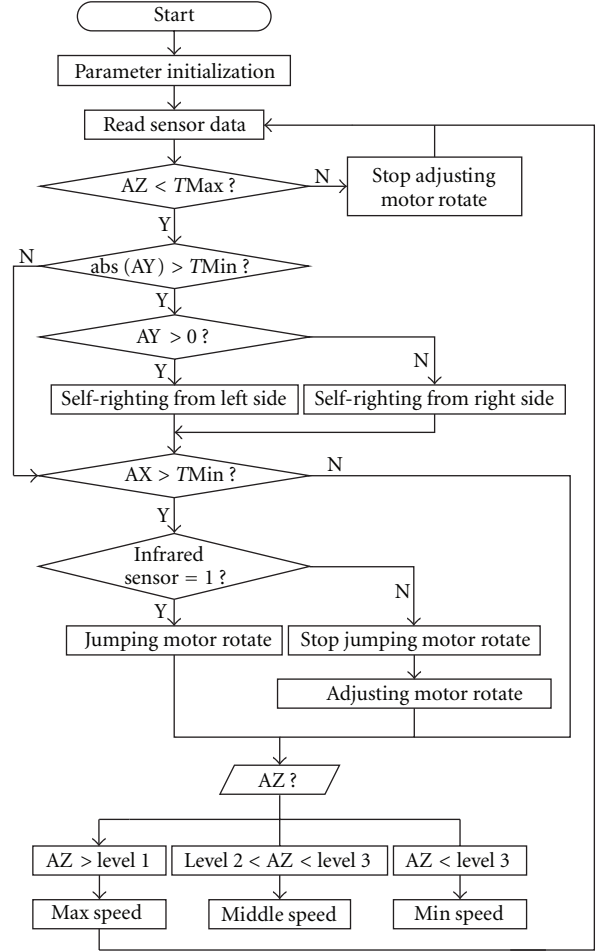


FIGURE 3: The automatic self-righting algorithm of the jumping robot.

the body frame has detected the critical position of the cam. Then the adjusting motor rotates counterclockwise to make the robot shift to the state that its right side contacts the ground. Then the robot will stand up using the same method as it falls down on its right side. The robot needs higher speed when self-righting in the beginning than in the end. The robot adjusts the speed of adjusting motor according to AZ . The three levels in Figure 3 are decided by posture angles of the robot.

The adjusting mechanism also can adjust jumping direction and take-off angle. The length of the pole leg is larger than the distance between the axis of the adjusting motor and the ground. The steering principle is that the robot can rotate the pole leg to steer after self-righting. The pole leg rotates to contact and leave the ground. The front part of the robot is propped up and moves a step in the ground while the foot of the robot is still in the same position. After this motion process, the robot can steer an angle in the ground. This angle also can be adjusted by adjusting the rotation of the jumping motor. The stable and efficient steering needs cooperation between the adjusting motor and the jumping motor.

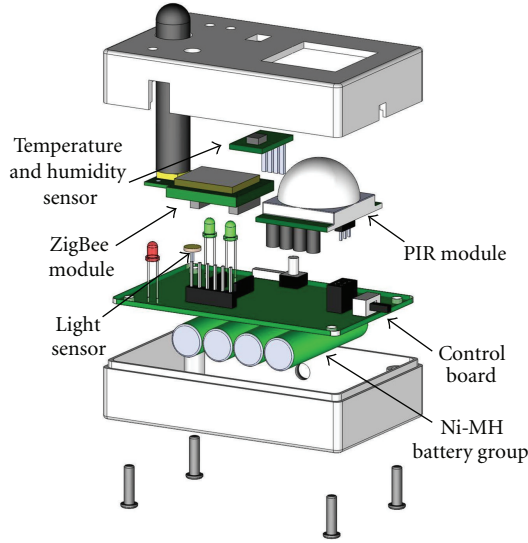


FIGURE 4: CAD model of the static sensor node.



FIGURE 5: Prototype of the sensor node.

An additional weight (AW) is mounted at the end of the pole leg to be used for take-off angle adjusting. After self-righting and steering, the robot adjusts the pole leg in different heights to get different center of mass (COM) positions of the robot. Different COM heights correspond to different take-off angles. So the take-off angle can be adjusted by rotating the pole leg in different positions.

3.2. Static Sensor Node Design. The CAD model of the static sensor node is shown in Figure 4. It is composed of a temperature & humidity sensor module, a light sensor, a pyroelectric infrared (PIR) sensor module, a ZigBee wireless communication module, a control board, and a Ni-MH battery group. The prototype of the static sensor node is shown in Figure 5. The static sensor nodes are used to detect whether there is a person passing by them and sense the temperature, humidity, and light of their working environment. These sensor data will be transmitted to the gateway by the ZigBee module. The nodes also periodically sample the voltage themselves. When finding the voltage lower than the preset threshold, the ZigBee module will send

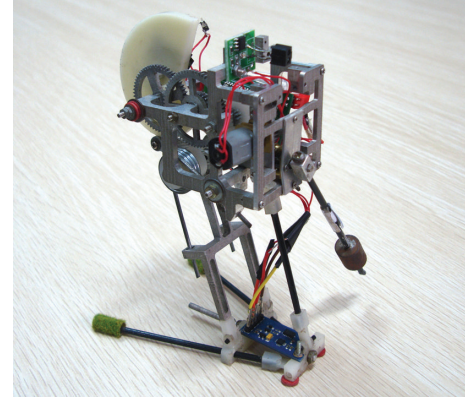


FIGURE 6: Prototype of the proposed jumping robot.

this message to the gateway. The gateway will control the jumping robot to take the place of the static node which will fail to work in the near future.

3.3. Network Repair Principle. The ground and other objects can reflect and absorb signal of wireless communication especially when sensor nodes with small size are deployed in uneven terrains of outdoor environments. The signal intensity of wireless communication is influenced by the deployment height of sensor nodes. Jumping robot can jump up to a relative high position to route sensor data. This will enhance communication quality of networks.

The network repair mentioned here supposes that the positions of the static sensor nodes and the initial position of the robot are known. When one sensor node is disabled or sheltered by obstacles, this may affect communication between the node and the gateway, and even result in interruption of the network. When the gateway finds that one sensor node does not send environment data continually to it, the gateway will send the inquiring command to the sensor node. If the sensor node will not acknowledge the command, the gateway will consider that the sensor node is disabled. When this happens, a jumping robot can be controlled by the gateway to move to the position of the disabled sensor node. The jumping robot then replaces the sensor node to monitor this area and route data of other sensor nodes to the gateway.

4. Experiment

A testbed is built to test the jumping and automatic self-righting capabilities of the proposed jumping robot. The steering and take-off angle adjusting functions are controlled by the operator through the graphic user interface running on the server. Two network interruption scenarios are set by the operator. The network repair functions of the WSN system in the two scenarios are tested.

4.1. Jumping Test. The prototype of the proposed jumping robot is shown in Figure 6. The jumping height of this prototype can be adjusted by installing two or four torsion springs. In this test four springs are used to get a maximum

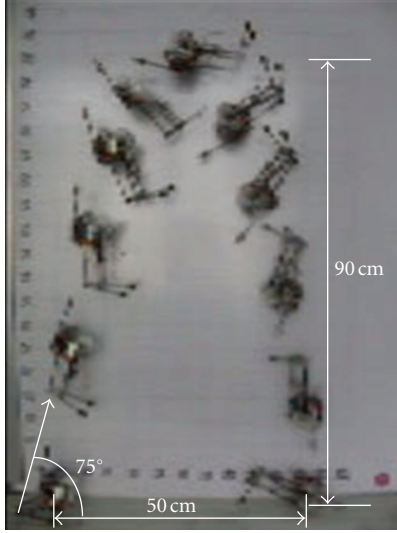


FIGURE 7: Jumping trajectory of the prototype robot.

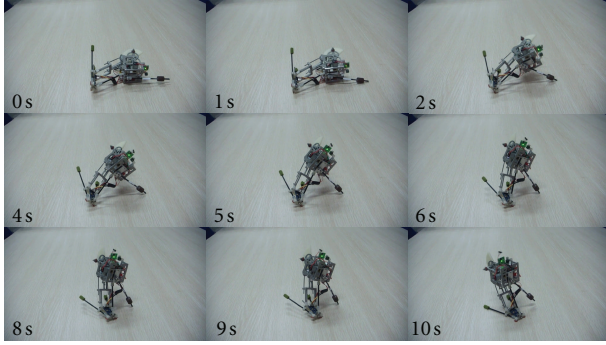


FIGURE 8: The automatic self-righting sequences of the jumping robot from left side.

jumping height. The torque of the motor is 93 N·mm. The total mass of the robot is 154 g. The jumping trajectories of the prototype robot have been recorded as shown in Figure 7. It can jump about 90 cm in height and traverse 50 cm far at a take-off angle of 75°. This test verifies that the jumping robot has powerful obstacle overcoming capabilities.

4.2. Automatic Self-Righting Test. In this test, the robot uses the control algorithm presented in Section 3.1 to stand up automatically after falling down. The result of self-righting sequences from left side is shown in Figure 8. The robot self-rights step by step. The changes of three-axis acceleration values during the self-righting process are shown in Figure 9. In the beginning, the falling on the left side is judged by the condition that acceleration z is smaller than its threshold 900 mm/s² and acceleration y is bigger than its threshold 300 mm/s². During the 10 s self-righting, y is decreasing, z is increasing, and x is not changing evidently. In the end, y is smaller than its threshold, and z is bigger than its threshold. The robot stands up successfully.

The self-righting sequences from right side are shown in Figure 10. The accelerations changes during the self-righting

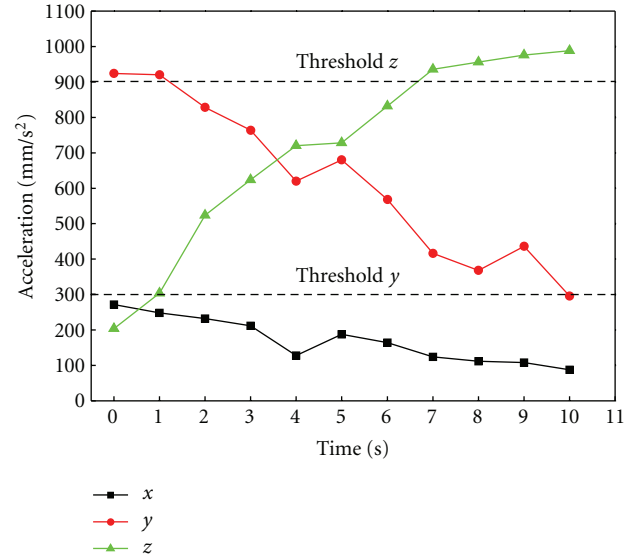


FIGURE 9: Acceleration changes during the self-righting process from left side.

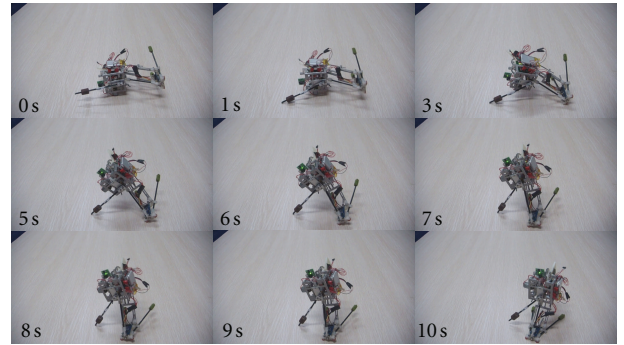


FIGURE 10: The automatic self-righting sequences of the jumping robot from right side.

process are shown in Figure 11. In the beginning, the falling on the right side is judged by the condition that acceleration z is smaller than its threshold 900 mm/s² and acceleration y is smaller than its threshold -300 mm/s². During self-righting, y and z are increasing and x is not changing evidently. In the end, y and z are bigger than their thresholds. The robot stands up successfully.

The self-righting sequences from front side are shown in Figure 12. The acceleration changes are shown in Figure 13. In the beginning, the falling on the front side is judged by the condition that acceleration z is smaller than its threshold 900 mm/s² and acceleration x is bigger than its threshold 300 mm/s². Firstly, the jumping motor rotates to make the robot shift from frontside to rightside. In 20 s, the robot shifts to right side. The acceleration of x changes from about 800 mm/s² to 200 mm/s². The acceleration of y changes from about -200 mm/s² to -1000 mm/s². The acceleration z is still smaller than its threshold. Then, the robot judges this shift costing about 20 s and starts to self-right from right side

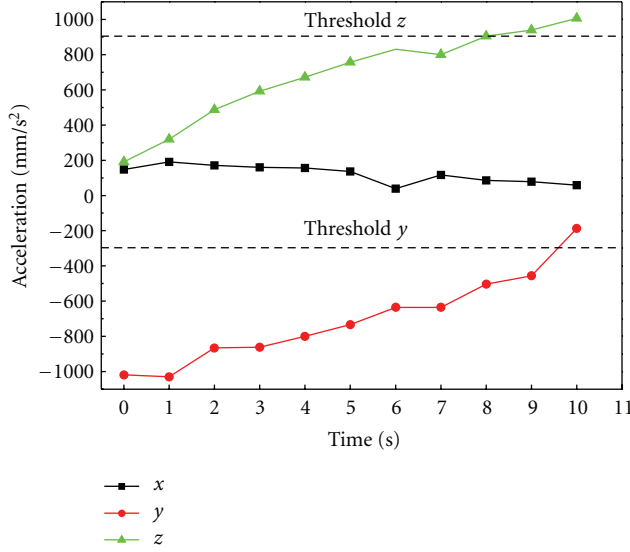


FIGURE 11: Accelerations changes during the self-righting process from right side.

in 40 s. In the end, y and z are bigger than their thresholds. The robot stands up successfully.

Self-righting of all the three sides are tested 40 times. The self-righting success rates are 82.50% from left side, 97.50% from right side, and 87.50% from front side. The reason of the difference between left and right sides is that the COM of the robot is not symmetrical in left and right sides. This can be solved by adjusting the position of the lithium battery. The success rate of the front side is also low for the strong vibration of the shift from front side to right side. This problem can be solved by optimizing the control algorithm.

4.3. Network Repair Test. The testbed setup of network repair experiment is shown in Figure 14. The network interruption is caused by obstacle. Because the distance between every two sensor nodes is in one-hop range, we set the max children node number of the gateway and sensor nodes to 1 and takeoff the 3 dB antenna of sensor nodes to implement mandatory multi-hop transmission. The gateway is powered up first and the sensor nodes are powered up one by one to ensure that a multihop chain network topology can be established.

We can see sensor data from every sensor node in the graphic user interface running on the server. The link quality indicator (LQI) of every node is recorded in the database. The results are shown in Figure 15. A 10 cm high obstacle is deployed between node 4 and nodes 5 and 6 in 30 s. The sensor data of nodes 5 and 6 cannot be received because they are sheltered by the obstacle. Then the robot is controlled by the gateway to jump up to the obstacle to repair the network. After about 50 s, sensor data of nodes 5 and 6 are seen in the graphic user interface again.

The testbed setup of network repair experiment is shown in Figure 16. The network interruption is caused by turning off one routing node. The sensor nodes are deployed in a



FIGURE 12: The automatic self-righting sequences of the jumping robot from front side.

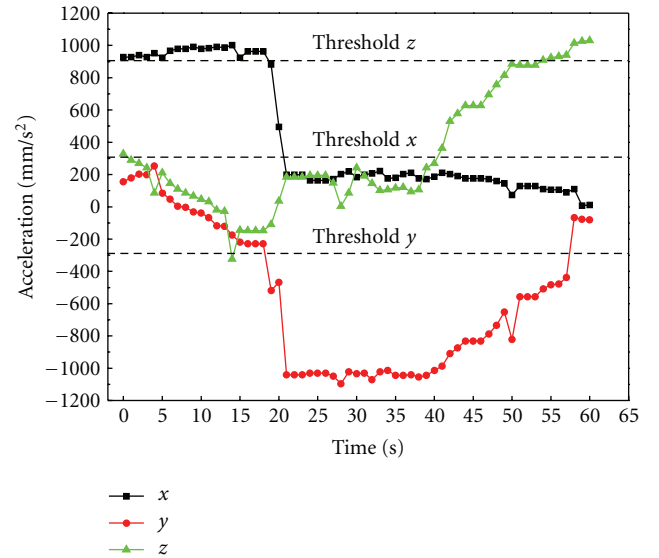


FIGURE 13: Accelerations changes during the self-righting process from front side.

line. Two desk lamps are used to make light sensor data of every sensor node different. The light sensor data during network repair process are recorded. The results are shown in Figure 17. Node 4 was turned off by the operator in 90 s. Then we can see in the graphic user interface that nodes 4, 5, and 6 do not send data to the gateway. The robot is powered up and controlled by the gateway to move to the position of node 4. After joining into the network, the robot sends its sensor data to the gateway. The gateway records the LQI value of the robot. The robot jumped to the position of node 4 to repair the network. When the robot reached the position of

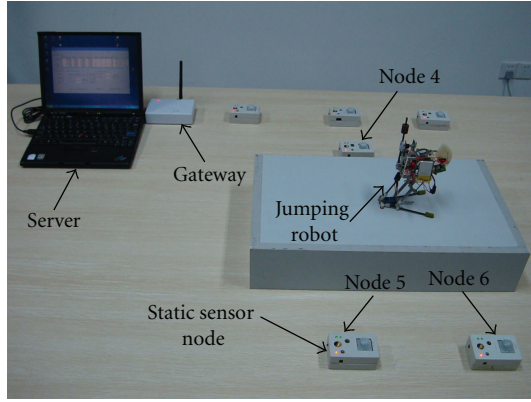


FIGURE 14: Testbed setup of the network repair experiment. The network interruption is caused by a cubic obstacle.

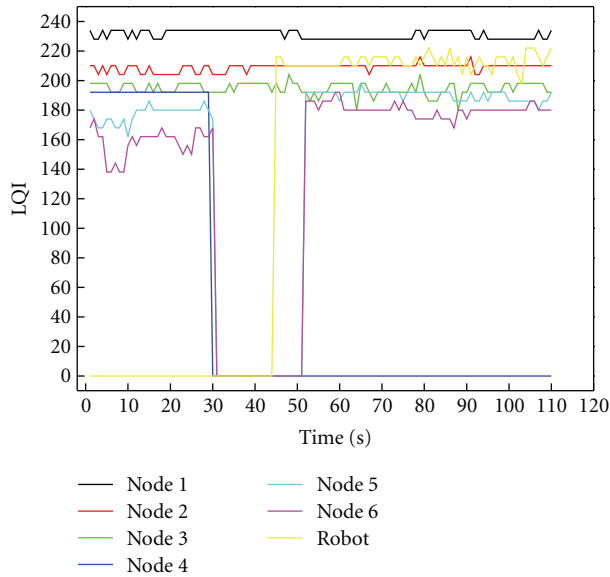


FIGURE 15: Result of network repair test. The network interruption is caused by obstacle sheltered.

node 4, then nodes 5 and 6 restarted to send data to gateway by routing through the robot.

5. Conclusions

We have presented a WSN system with a jumping robot as the mobile node. The robot uses the cam with quick-return characteristic to save and release energy. A pole leg is used to self-right after falling down. The pole leg also can be used to adjust the jumping direction and take-off angle of the robot. Based on the jumping robot design and some static sensor node design, a WSN monitoring system is built to test the functions of the robot and the function of the network repair. Results show that the jumping robot can jump 90 cm in height and traverse 50 cm far. The robot is able to automatically self-right after falling down on the ground. The steering and take-off angle adjusting of the robot can be

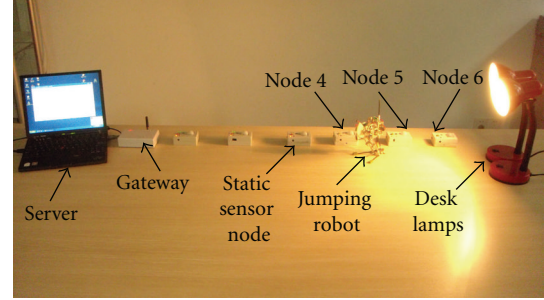


FIGURE 16: Testbed setup of network repair experiment. The network interruption is caused by turning off one routing node.

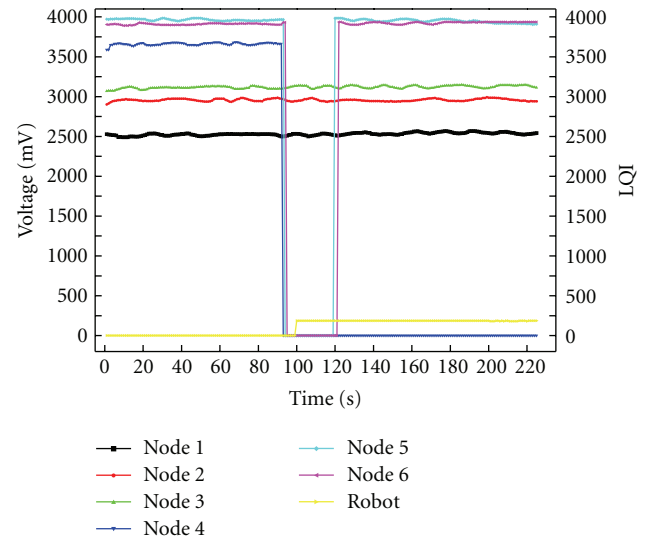


FIGURE 17: Result of network repair test. The network interruption is caused by turning off one routing sensor node.

controlled by the operator through the graphic user interface running on the server. The robot can be controlled to move to the position of the disabled static sensor node to repair the network and route sensor data.

The self-righting capability of the jumping robot is not very robust. It only can work in relatively smooth surfaces. It is difficult for this generation of the robot prototype to keep standing in uneven terrains. The pole leg of the robot will be designed to provide powerful support for the robot in the next generation. The total energy saved in the torsion springs is fixed during every jumping. The take-off angle adjusting method in this generation also has limitations. Changeable energy storage solutions and wide range of take-off angles adjusting methods will be investigated in the future to make the robot jump according to the height of obstacles for the purpose of energy saving.

Future work will focus on the following three aspects. (1) We plan to add a compass to sense the current direction of the robot in order to achieve the function of automatic steering. (2) A code wheel will be pasted on the surface of the cam to make steering and take-off angle adjusting more precise. (3) The solutions of localization and trajectory

planning for continuous jumping of our robot will be investigated. Therefore the prospective multijumping robots will be widely used in WSN systems.

Acknowledgments

The research reported in this paper was carried out at the Robotic Sensor and Control Lab, School of Instrument Science and Engineering, Southeast University, Nanjing, Jiangsu, China. This work was supported in part by the Scientific Research Foundation of Graduate School of Southeast University under Grant YBJJ-1221, the Research & Innovation Program for Graduate Student in Universities of Jiangsu Province under Grant CXLX-0102, Program for New Century Excellent Talents in University under Grant NCET-10-0330, and Natural Science Foundation of Jiangsu Province under Grant BK2011254.

References

- [1] P. Jiang, H. Xia, Z. He, and Z. Wang, "Design of a water environment monitoring system based on wireless sensor networks," *Sensors*, vol. 9, no. 8, pp. 6411–6434, 2009.
- [2] G. Song, Y. Zhou, W. Zhang, and A. Song, "A multi-interface gateway architecture for home automation networks," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1110–1113, 2008.
- [3] C. A. Boano, N. Tsiftes, T. Voigt, J. Brown, and U. Roedig, "The impact of temperature on outdoor industrial sensor network applications," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 3, pp. 451–459, 2010.
- [4] M. A. Hussain, P. Khan, and K. K. Sup, "WSN research activities for military application," in *Proceedings of 11th International Conference on Advanced Communication Technology (ICACT '09)*, pp. 271–274, Phoenix Park, Republic of Korea, February 2009.
- [5] G. Song, Y. Zhou, Z. Wei, and A. Song, "A smart node architecture for adding mobility to wireless sensor networks," *Sensors and Actuators A*, vol. 147, no. 1, pp. 216–221, 2008.
- [6] C. H. Lin, S. H. Yang, H. T. Chen, and K. T. Song, "Mobile robot intruder detection based on a zigbee sensor network," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC '08)*, pp. 2786–2791, Singapore, October 2008.
- [7] Y. Li and L. E. Parker, "Detecting and monitoring time-related abnormal events using a wireless sensor network and mobile robot," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, pp. 3292–3298, Nice, France, September 2008.
- [8] J. Takahashi, T. Yamaguchi, K. Sekiyama, and T. Fukuda, "Communication timing control and topology reconfiguration of a sink-free meshed sensor network with mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 187–197, 2009.
- [9] S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.
- [10] F. Doshi, E. Brunskill, A. Shkolnik et al., "Collision detection in legged locomotion using supervised learning," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '07)*, pp. 317–322, San Diego, Calif, USA, November 2007.
- [11] E. Sayginer, T. Akbey, Y. Yazicioglu, and A. Saranli, "Task oriented kinematic analysis for a legged robot with half-circular leg morphology," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '09)*, pp. 4088–4093, May 2009.
- [12] P. Corke, S. Hrabar, R. Peterson, D. Rus, S. Saripalli, and G. Sukhatme, "Autonomous deployment and repair of a sensor network using an unmanned aerial vehicle," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 3602–3608, New Orleans, LA, USA, May 2004.
- [13] R. Armour, K. Paskins, A. Bowyer, J. Vincent, and W. Megill, "Jumping robots: A biomimetic solution to locomotion across rough terrain," *Bioinspiration and Biomimetics*, vol. 2, no. 3, article S01, pp. S65–S82, 2007.
- [14] J. Burdick and P. Fiorini, "Minimalist jumping robots for celestial exploration," *International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 653–674, 2003.
- [15] B. G. A. Lambrecht, A. D. Horschler, and R. D. Quinn, "A small, insect-inspired robot that runs and jumps," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1240–1245, Barcelona, Spain, April 2005.
- [16] U. Scarfogliero, C. Stefanini, and P. Dario, "The use of compliant joints and elastic energy storage in bio-inspired legged robots," *Mechanism and Machine Theory*, vol. 44, no. 3, pp. 580–590, 2009.
- [17] H. Tsukagoshi, M. Sasaki, A. Kitagawa, and T. Tanaka, "Design of a higher jumping rescue robot with the optimized pneumatic drive," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1276–1283, Barcelona, Spain, April 2005.
- [18] J. Zhao, N. Xi, B. Gao, M. Mutka, and L. Xiao, "Development of a controllable and continuous jumping robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 4614–4619, Shanghai, China, 2011.
- [19] S. A. Stoeter and N. Papanikolopoulos, "Kinematic motion model for jumping scout robots," *IEEE Transactions on Robotics*, vol. 22, no. 2, pp. 398–403, 2006.
- [20] M. Kovač, M. Schlegel, J. C. Zufferey, and D. Floreano, "Steerable miniature jumping robot," *Autonomous Robots*, vol. 28, no. 3, pp. 295–306, 2010.
- [21] E. Kadioglu and N. Papanikolopoulos, "A Method for transporting a team of miniature robots," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2297–2302, Las Vegas, Nev, USA, October 2003.
- [22] J. Zhang, Y. Zhu, H. Wang, and J. Zhang, "Design of a bio-inspired jumping robot for rough terrain," in *Proceedings of the 2nd Conference on Environmental Science and Information Application Technology (ESIAT '10)*, pp. 40–43, Wu Han, China, July 2010.

