Cooperation of Autonomous NXT Robots Using Bluetooth Wireless Technology

By: Jesse Bell, Skyline High School, Dallas, TX and Elizabeth Freeman – Frisco High School, Frisco, TX

RET: Research Experiences for Teachers on Sensor Networks Summer 2013

University of North Texas, Denton Texas
www.teo.unt.edu/ret/

Abstract: *This paper highlights the exploration of multi-agent NXT Robotics systems using a Bluetooth communication channel. The project consisted of using Bluetooth technology to coordinate movements between two agents. The benefits of creating a swarm of robots with individual capabilities include a more controllable system as opposed to a single and more complicated machine. The lead robot was programmed to follow a specified path using a light sensor, then send, via Bluetooth, a message indicating follower instructions. The sensitivity of the light sensor and the Lego Mindstorms software limitations created inconsistencies in the follower program. Hypotheses regarding the lack of success in the following capabilities of the robots involve limitations of the NXT-G software. One conclusion is that inability to adjust the Bluetooth settings is the source of miscommunication between the robots. Specifically, the ability to adjust the rate of messages sent/received may improve overall communication.*

Keywords – Autonomous; Robot; Bluetooth; NXT; Mindstorm; Lego

I. INTRODUCTION

It may be difficult to imagine that just 40 years ago robots were primarily thought to be only in one's imagination or in cartoons on television on Saturday mornings. But today, robots are commonplace and their presence is illustrated in almost every industry across the globe. Robots are a staple among military and police operations where human involvement is unsafe. Most recently, there has been much interest in deploying multiple robots for studying asteroids as well as other planets.

Current research at NASA, for instance, involves the deployment of robots on Mars that can communicate with each other and at the same time perform various different tasks. This is an improvement on a system where a single robot carries out multiple functions. Details and barriers involving such a complex machine are avoided with the use of several, more simple, machines. The most critical of the tasks needed for success would be navigation and communication between the robots. Navigation is paramount because the robots must be able to travel in a given area. Communication is important because the robots must maintain the ability to transmit information between each other in order to achieve a common goal.

Communication between autonomous robots is critical in any circumstance, regardless of purpose. Two ways to accomplish communication may be directly from robot to robot, or indirectly, via a base station. Rooker and Birk show success in having a swarm of robots performing random exploration as a pack while maintaining wireless communication with a base station [10]. Work by Zermas shows that this type of involvement, often referred to as an absolute reference coordinating system, causes an increase in error, and is not preferable when actual trajectory, or desired path is in order [7]. When the base station was not involved, robots inclined to find themselves in a deadlock. A solution to this was the assignment of roles to each robot. A lead robot could coordinate the movement and direction of robots toward a specific location in order to resolve or avoid such a problem. This scenario could continue under many circumstances [10]. Without a lead robot to coordinate movements, the use of wireless communication, like Bluetooth, along with sensors to maintain a constant distance between members of the swarm was found successful [8].

Coordinating multiple robots using wireless communication protocols is relatively commonplace as of today. However, the use of the recently developed Bluetooth protocol for these types of tasks is original. "An accurate communication system is crucial to solve the leader/follower task" [6]. The robots used in this project consisted of two NXT LEGO Mindstorm Robot Kits from LEGO Mindstorms. The LEGO

Mindstorms system was used because of its price, flexibility, and ability to communicate with other robots via Bluetooth. Mindstorms software is user friendly to beginning programmers with its icon based platform.

Bluetooth is a short-range Radio Frequency (RF) technology, capable of point-to-point or point-to-multipoint connections at speeds up to 1Mbps [12]. This type of communication is suitable for our application because its signals do not require line-of-sight and may even connect through most physical barriers with a range of up to 10 meters. In addition, Bluetooth is an attractive choice due to its low cost, low power consumption, small size and high versatility [3]. Shephard and Mansoor focused on the use of Bluetooth versus other wireless communication methods. They believe that Bluetooth is a promising technology for use in lightweight mobile robotic systems because of its low power consumption [12].

Of the many tasks imposed on a system of robots two goals must be specifically met. The first involves ability of a lead robots exploration of its environment in order to discover its target. The second focuses on the following swarm to receive information about this environment through some type of communication. One study refers to this behavior as the “Honeybee” task. The “Honeybee” task because it is based upon the behavior of the common honeybee where each honeybee worker has the ability to seek out pollen-producing flowers in close proximity of the hive then return to the hive to perform a dance in order to communicate its location [6].

Paztor et al. used a simulation with four robots. In their work, the master could communicate only with one slave at a time and the slaves could not communicate with each other. Their network is called a piconet and can consist of up to seven slaves [5]. The Lego NXT system specifically can coordinate one master and up to three slaves. Similar work using a master-slave scheme involving NXT robots and Bluetooth communication differs in that the slaves are also capable of communication with each other [9]. Software other than the Lego Mindstorms is required for such two way communication, such as RobotC or MatLab.

A required task in the sensing robot field is that the robots must be able to move from the start to the end point autonomously without human interaction [11]. In this study the lead robot used a light sensor to follow a strip of black electrical tape on the floor in order to direct itself from one place to another. Specific visually oriented programing accomplished this task, which brought its own set of challenges.

The LEGO Mindstorm NXT robots used in this study consist of 2 servo motors, one light sensor, and one ultrasonic sensor. NXT-G software was used in all programming and is similar to the National Instruments software, “LabView ™” system that is common in many secondary science classrooms. In addition, besides using the official visually-oriented programming, the robots can also be programmed in several other languages such as Java, C, or Pyton to name a few [4].

The Mindstorm NXT Kit has a box-shaped processor that resembles a brick (fig. 1). The device is configured with three output motor ports and four sensor ports. In this study the light sensor was used to detect and report intensity of light and the ultrasonic sensor for measuring distance and detecting objects within a specified range [4].

Fig. 1: Lego NXT Brick

II. METHODS

A. Robot Set-up with Sensors

Having an exact structure and wheel-base was important when comparing the movement of the two robots given the same program. Both the lead and follower robots were built to maintain similarity between the body structure, wheels, motors, and sensors. The programs used for movement powered the left and right wheels separately.

The NXT light sensors detected the intensity dark versus light and allowing the lead robot to detect color change along the floor. The sensor was mounted at the front of the robots and low with about three millimeters clearance above the floor (fig 1a). Due to the nature of the non-white floors of the lab, calibration of the sensor was set to read the lighter tan-colored tiles as the lightest shade available (measured as 100) and the black of the electrical tape as the darkest shade available (measured as 1). Ninety degree angles were avoided so as to allow smooth left and right movements. Fig. 2b.

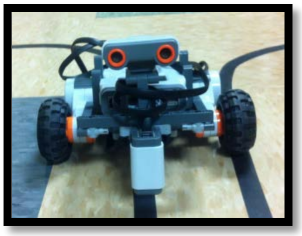


Fig. 2a: constructed robot showing placement of the ultrasonic and light sensors

Fig. 2b: example of path traced by electrical tape to be followed by the light sensor of the lead robot.

Ultrasonic sensors were used in both robots to prevent the robots from running into other objects. The ultrasonic sensors were set to react at five inches and upon approaching another object the program would stop.

B. *Line Follower Program I*

The light sensor in this program (Fig. 3) responds to differences in light on a two-point scale. In this case the robot responds when light detected is greater than 50% brightness or when light detected is less than 50% brightness. The motors B and C turn the left and right wheels, respectively at different powers to achieve a left or right turn.

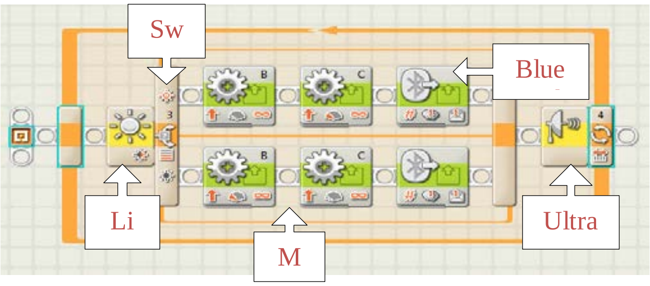


Fig. 3: Line follow program I (key: Li = light sensor, Sw = switch for light and dark readings, Blue = send Bluetooth message, Ultra = ultrasonic sensor command to stop robot upon contact with any object.)

The line follower program works as follows: first the light sensor takes a reading. As with the black line on the floor, if the sensor is situated on top of the black tape it reads “dark”, or <50%. If the sensor is to the side of the black tape, the lighter color of the floor causes it to read “light”, or >50%. Within the switch block there is top row and a bottom row. When >50% is detected, motor B is activated at 10% and motor C is activated at 30%. This facilitates a smooth left turn. When the darkness of the black tape is detected the opposite occurs and the power of each motor is 30% and 10% allowing for a smooth right turn. The robot proceeds with the left-right pattern of movement as it detects the edge of the black tape on the floor. This program works only if the robot is situated along the right-most edge of the black tape, or color border. In the case of this experiment the robot followed the outer edge of the taped loop in a counter-clockwise direction, or the inner edge of the loop in a clockwise direction.

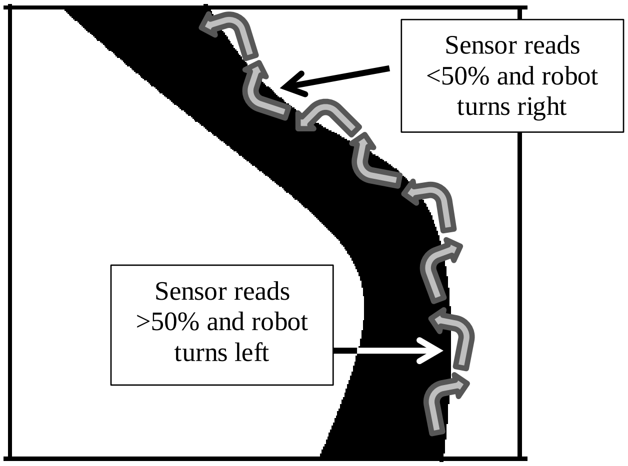


Fig. 4: Sketch to diagram movement of robot under line follow program I.

The arrows in Fig. 4 are small and relative to the speed of the robot. When the power to the motors is set above a certain threshold, the turning and velocity of the robot causes the wheels to move the sensor past the opposite edge of the tape, thus finding the lighter color of the floor and turning left as instructed.

C. *Line Follow Program II*

The program uses the light sensor’s intensity readings to convert a 1 – 100 scale of black – white into a smaller and more manageable scale. By dividing all numbers detected by the light sensor by 20, the resulting new set of numbers 1 – 5 is a reasonable range, which makes it easier to work with. For example, if the light sensor detected a value of 80 it would send the number 4 out to the motor portion of the program. The number block from Fig. 5 is how the conversion is accomplished. Effectively, this is an improvement on the simpler program that simply commands a left turn when it detects black and right turn when it detects white as described previously as Light Follow Program I.

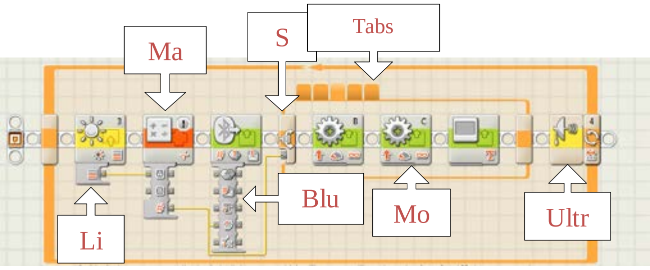
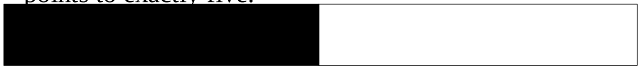


Fig. 5: Line follow program II (key: Li = light sensor, Ma = math block, S = Switch, Blu = Bluetooth Send message, Mo = Motor blocks, Tabs = organize motor action based on light sensor output, Ultr = Ultrasonic sensor)

Table 1: Light Sensor Intensity Output Ranges

a) Light sensor reads 100 gradients of light/ dark

b) Mindstorms program converts 100 potential data points to exactly five.



c) In a simple "left-right" program, the output comment from the sensor is less specific.

The initial program was modified to fit the current configuration on the robot built [2]. Changes included orientation of the motors and incorporation of the Bluetooth communication factor to serve the greater purpose of attaining a follow scenario.

Table 2: Motor Actions Based on Light Intensity Readings

Light Sensor Calculated Reading	**Action of Motors**
0	Hard Right turn by greatly reducing power to motor C
1	Gradual Right turn by slightly reducing power to motor C
2	Straight Ahead with motors equal
3	Gradual Left turn by slightly reducing power to motor B
4	Hard Left turn by greatly reducing power to motor B

The tabs seen on the inside loop in Fig. 5 are each assigned the differing scenario 0 – 4, and the motor blocks within each tabbed window drive the robot in a different direction as described in Table 1.

D. Bluetooth Message Sent and Received

Each of the programs I and II involved the lead robot sending a numerical message to the following robot. This was chosen over logic or text, the only other option available for the Lego NXT robots. The light sensor reads the intensity of brightness and sends this information as a number. It was easier to keep this output information as a number rather than convert it to text or logic when designing the Bluetooth component of the send and receive message.

Program I: The send message blocks of program I are located within each switch tab. Whether light or dark was sensed depends on which block was activated. Randomly assigned numbers facilitated this; when the sensor detected the dark of the tape, a "5" was sent via Bluetooth to the follow robot. When the sensor detected the lighter color of the floor, a "2" was sent via Bluetooth to the follow robot. For example, if a message of 2-5-2-5-2-5-2-5 is sent from the lead robot to the follower robot, both robots were capable of following the line.

In the receive program, Fig. 6, two possible scenarios are simultaneously occurring. Either the follow robot is receiving a "2" or it is receiving a "5". Whichever number is received dictates what the wheels will do, or how much the robot will turn. A 0.25 second delay is in place to accommodate for the distance between the two robots, and all commands are placed in a loop until the ultrasonic sensor detects a potential obstacle.

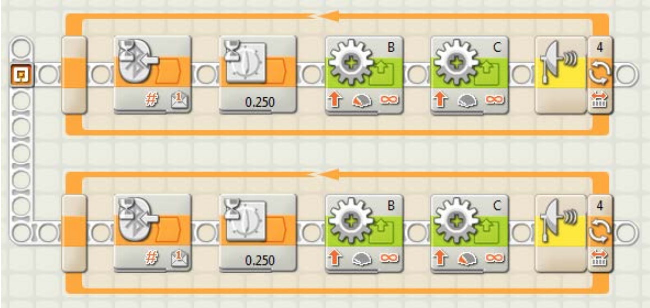


Fig. 6: Program I for following robot

Program II: In program II the Bluetooth module in the lead robot receives its numerical input directly from the math block. Intensity, read numerically, from the light sensor is converted to the previously discussed scale of 0-4, and the Bluetooth command block sends exactly that number to the follower robot.

Fig. 7 shows how the math block feeds the calculated intensity reading to both the send message block, and to the switch block containing the motors.

The receive message block is placed into a loop where it continuously receives and acts upon numerical messages from the lead robot. A wire feeds into this loop as a logic statement and serves to facilitate the continuous receipt of messages. A second wire from the receive message block extends to the switch block. A 0.25 second pause is in place in this program as well to accommodate for the distance between the robots. As numbers are reported to the switch block, the follower robot is capable of making movement based on the specific number received.

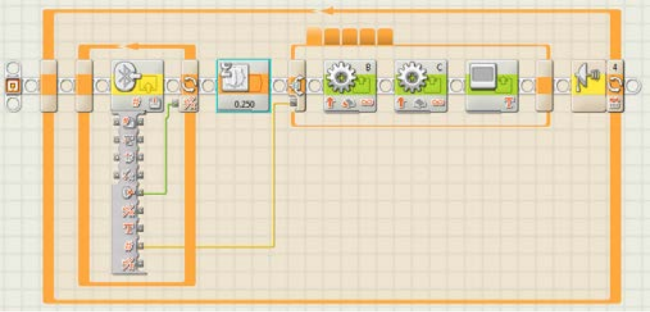


Fig. 7: Program II for following robot.

III. RESULTS AND DISCUSSION

Findings and trials with the lead and follow robots showed a variety of results and performances. Troubleshooting and unexpected accommodations became necessary for both robots in multiple areas before a predictable and reliable program was found.

A. *Line Follow Troubleshooting*

Without alteration of the program or the robot itself, the lead robot was inconsistent with its ability to follow the black line on the floor. The arrows in Fig 8 are small and relative to the speed of the robot. When the power to the motors is set above a certain threshold, the turning and velocity of the robot causes the wheels to move the sensor past the opposite edge of the tape, thus finding the lighter color of the floor and turning left as instructed. The larger arrows (Fig 8) represent a greater velocity and show the problem of overrunning the width of the black tape and finding the opposite edge before the black of the tape is detected by the light sensor. Although exaggerated, the arrows showing the movement and path of the robot in each case are true to the specific scenarios encountered. Until the power settings on both motors were decreased significantly, the cause of erratic turning behavior was not obvious.

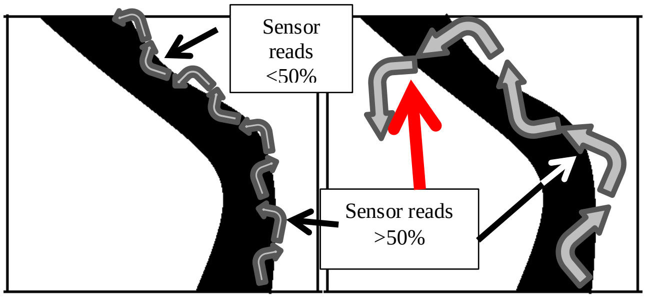


Fig. 8: Left: line follow behavior with slower motor settings; Right: Line follow behavior with quicker motor settings.

Another unexpected issue occurred with the electrical tape itself. Often times, the light sensor would not respond to the black tape at all. After multiple trials with different shades, colors, materials, and lighting scenarios, it was discovered that the light sensor is more sensitive than originally thought to the reflectivity of the surface it reads. The sensor was consistent when reading the waxed surface and lighter colored linoleum floor, but not the fresh black electrical tape. Treatment with sand paper to dull the electrical tape eliminated all issues regarding the light sensor's ability to read the dark line.

B. *Lead-Follow Relationship*

The two robots were randomly unsuccessful at tracing a similar path as the lead robot follows the black line. Different complexities in the path drawn by the electrical tape do not dictate the level of success and margin of error between the exact paths between the lead and follow robots. Regardless of turning radius or the variation of turns, the margin of error and unpredictability was all equal in magnitude.

Possible solutions to the following problem included the issue of battery power. It is suspected that the lead robot's job of both, following the black tape line and sending the Bluetooth message, may have an effect on its performance. Also, differences in battery charge between the lead and follow robots may lead to their unequal velocity and/or turning radius as the follow robot attempts to receive and act upon the numerical messages. If a difference in charge between the two robots was truly the case, one would hope they still turn in the same direction.

Each of the trials shown in all of the pictures were completed back to back without any changes to the robots settings or the programs uploaded to lead or follow robots. When moving between rooms I did recalibrate the light sensor in hopes the line follow lead robot would produce and send reliable information.

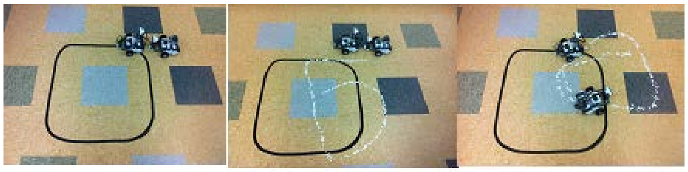


Fig. 9: Left: Shows where the two robots started and each picture involves a complete circle by the lead robot. The lead robot faithfully followed the box in a counterclockwise direction while the follow robot received the Bluetooth information. The lead robot traced the black line while the white line represents the movement of the follow robot.

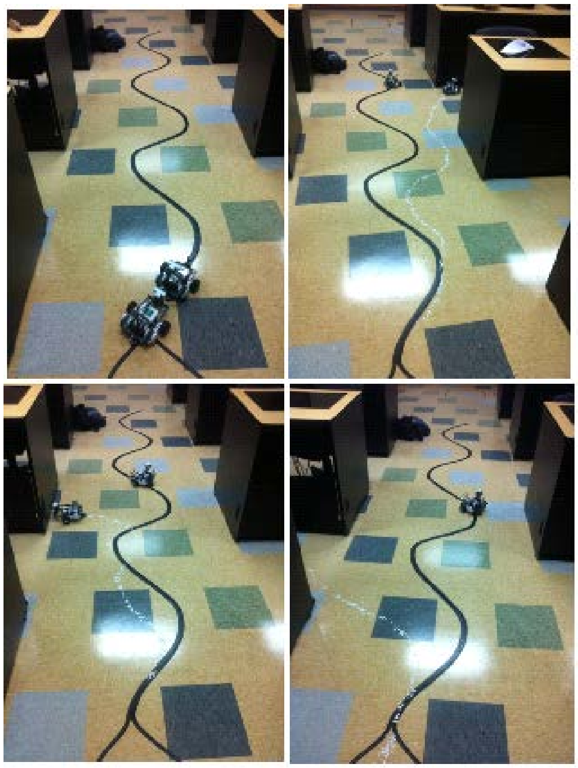


Fig. 10: In the picture shown, a slightly different track involving both left and right turns. UL: Shows the starting positions for the lead and follow robots for each trial. The lead robot traced the black line while the white line represents the movement of the follower robot.

Fig. 11: Red and yellow dots represent starting locations for the lead and follow robots, respectively. Lead robot followed the black line, while the follower robot path is represented by the white line.

One hypothesis to explain the lack of following abilities in the follower robot is within the details of Bluetooth communication itself. As with Program I, the messages sent are two's and five's. It is assumed that the rate at which the numbers are sent is extremely rapid. If this rate is too quick, the receiving (follower) robot may not be able to process these numbers and act upon the message in a timely manner. Using the Lego software, it is not possible to know how often messages are being sent by the lead robot. It is also not possible to modify the rate at which these messages are sent. For example, if the lead robot sends 10 seconds worth of numbers directing the follow robot left and right in a specific sequence, it is possible that the follow robot is receiving only pieces of the message. Perhaps, only every third or fourth message or less is read. A sequence of messages describing a straight line would be that of left-right-left-right-left… and so on, where the number of left turns equals the number of right turns. Action that depicts what is received may be only left-left-left-right-left-right-right, at first appearing to be random, may simply be an incomplete picture.

IV. CONCLUSION

As our interests in space exploration increase and technology improves, the presence of robots in our society will continue to increase. Research in Robotics and need for autonomous robots is increasingly necessary. Our work shows that the Lego Mindstorms robots can bridge basic programming skills to those needed to accomplish complicated robotics tasks. We conclude that limitations identified with the Lego NXT-G software hinder the ability to accomplish a truly autonomous lead-follower relationship. While sending a single message and demanding a single action is possible, a continuous stream of messages involving specific actions requires better coordination between robots. Bluetooth synchronization between leader and follower robots may be necessary in order to accomplish any continuous coordinated movements, our work shows that the NXT-G software cannot accomplish this task. Future research using any other third party firmware and/or programming languages such as Java or RobotC may yield more reliable results.

ACKNOWLEDGMENTS

RET (Research Experiences for Teachers) Site on Sensor Networks, Electrical Engineering Department, and Institute of Applied Sciences, UNT, Denton, Texas. This material is based upon work supported by the National Science Foundation (NSF) under Grant No. 1132585 and the IEEE Control Systems Society (CSS) Outreach Fund. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or the IEEE.

Dr. Miguel Acevedo and Dr. Rudi Thompson of the University of North Texas for their support and opportunity to work with them this summer.

Dr. Kamesh Namuduri and Omar Costilla Reyes for your guidance and knowledge throughout our learning process.

REFERENCES

[1] “Building and Programming”, Tufts University, http://legoengineering.com.

[2] “Projects for NXT 2.0”, Lego Mindstorms, http://nxtprograms.com.

[3] A. H. Ismail, et al, "Investigation of homogeneous multi robots communication via Bluetooth," *Computer Applications and Industrial Electronics (ISCAIE), 2012 IEEE Symposium on* , vol., no., pp.124,129, 3-4 Dec. 2012, doi: 10.1109/ISCAIE.2012.6482082

[4] A. Koller, G.Kruijff, “Talking robots with lego Mindstorms,” in *Proceedings of the 20th Coling, Geneva, 2004.*

[5] A. Pasztor, T. Kovacs, Z. Istenes, “Compass and Odometry Based Navigation of a Mobile Robot Swarm Equipped by Bluetooth Communication”, *Computational Cybernetics and Tech. Informatics,* pp. 565-570, May 2010. DOI: 10.1109/ICCCYB.2010.5491208

[6] D. H. Barnhard, et al, “Using Bluetooth Communication for Coordinated Robot Research.”, Artificial Intelligence Center, University of Georgia, 2001.

[7] D. Zermas, “Control of a Leader-Follower Mobile Robotic Swarm Based on the NXT Educational LEGO Platform.”, IEEE Conf. Pub, pp. 1381-1386, 2011, DOI: 10.1109/ISIE.2011.5984361.

[8] E. J. Chung, et al, “Development of a Multiple Mobile Robotic System for Team Work”, Int. Joint Conf.SICE-ICASE, 2006, pp. 4291 - 4296 DOI: 10.1109/SICE.2006.314877.

[9] J. Steshenko, E. Kagan, I. Ben-Gal, “A simple protocol for a society of NXT robots communicating via Bluetooth.”, IEEE Conf. Pub., pp. 381-384, 2011.

[10] M. Rooker, A. Birk, ”Multi-robot exploration under the constraints of wireless networking.”, *Science Direct, Control Engineering Practice,* vol. 15, pp. 435-445, Nov. 2006.

[11] O. Wongwirat, T. Hanidthikul, N. Vuthikulvanich, “A formal approach in robot development process using a UML model.”, 10th Intl Conf. on Control, Automoation, Robotics, and Vision, pp. 1888-1893, Dec. 2008.

[12] R. Shephard, S. Mansoor, ”Bluetooth Based Proximity Sensing for Reactive Mobile Robots”, IEEE TENCON 2005, pp. 1-6, DOI: 10.1109/TENCON.2005.301049 .