Proceedings of the 2008 IEEE
International Conference on Robotics and Biomimetics
Bangkok, Thailand, February 21 - 26, 2009

Autonomous Staircase Detection and Stair Climbing for a Tracked Mobile Robot using Fuzzy Controller

E. Mihankhah[†], A. Kalantari[†], E. Aboosaeedan[†], H.D. Taghirad[‡], and S.Ali.A. Moosavian[§],
Department of Mechanical Engineering, K.N. Toosi University of Technology, Tehran

***Abstract*— Theoretical analysis and implementation of autonomous staircase detection and stair climbing algorithms on a novel rescue mobile robot are presented in this paper. The main goals are to find the staircase during navigation and to implement a fast, safe and smooth autonomous stair climbing algorithm. Silver is used here as the experimental platform. This tracked mobile robot is a tele-operative rescue mobile robot with great capabilities in climbing obstacles in destructed areas. Its performance has been demonstrated in rescue robot league of international RoboCup competitions. A fuzzy controller is applied to direct the robot during stair climbing. Controller inputs are generated by processing the range data from two LASER range finders which scan the environment one horizontally and the other vertically. The experimental results of stair detection algorithm and stair climbing controller are demonstrated at the end.**

***Key words*— Autonomous stair case detection, Autonomous stair climbing, Tracked rescue mobile robot, Fuzzy controller.**

I. Introduction

STAIRCASE is a common terrain for a mobile robot which is supposed to work in real environment. Climbing the stairs is very important for search and rescue mobile robots as for a rescue mission inside a building, almost in all cases, it is necessary to search number of levels. It is possible for a human operator to drive Silver[1] (see Fig.1) upstairs but this task, as it is also mentioned is influenced by the operator's skill to a great extent. A human mistake during the stair climbing, caused by insufficient visual information or radio drop-off for instance, can end to the turn over of the robot and an unsuccessful rescue mission. Besides removing human mistakes, adding autonomous stair climbing skill to the rescue mobile robot, can speed up the mission. It also let the human operator focus on the rescue mission to find the objectives of the mission instead of paying extra attention to driving the robot.

Several researches on autonomous stair climbing have been introduced in recent papers on different platforms.

[†] - Student of masters in Mechatronics, member of Resquake Robotics Team, email: {mihankhah, kalantari, aboosaeedan} @ee.kntu.ac.ir.

[‡] - Associate professor, Department of Electrical Engineering, member of ARAS Research Group, email: taghirad@kntu.ac.ir

[§] - Associate professor, , Department of Electrical Engineering, member of ARAS Research Group, email: moosavian@kntu.ac.ir

[1] - Rescue Mobile Robots manufactured by Resquake Robotics Team were generally named ResQuake in previous publications.

In [1], [2], [3] and [4], different stair climbing algorithms on tracked vehicles have been the discussed. In [1], fusing sensor data on an urbie type robot has been presented where the authors used extended Kalman filter to fuse gyro and visual information to detect the position of robot. In [2], LASER range finder (LRF) data is also added. A state space model of the robot dynamics was presented in both and a linear controller was applied to control it.

Fig. 1. Silver, the experimental platform of this research, with two LASER range finders installed on its passive robotic arm

A hexapod has been used as the platform in [5], where the stair climbing is mainly based on six independently actuated legs of the robot. The stair climbing in this paper is open loop and the success of climbing depends on the starting conditions.

A stair climbing method is implemented on a biped robot in [6]. Stereo vision and LASER range finder were used to estimate stairs dimensions. A table of trajectory parameters was provided for different stairs.

Also some special mechanisms (e.g. snake-like [7]) capable of autonomous stair climbing have been introduced in recent papers. In these researches the mechanism itself is the topic and stair climbing is presented as the capability of the platform. Speed and efficiency of the climb are not considered in these researches.

In [8], a fuzzy controller has been implemented that uses heading error and yaw rate to calculate the turn rate signals. In [9], a fuzzy controller is applied to a simulated robot to climb the stairs in a simulation environment.

Stair detection has not been the matter of discussion in above papers and the robot was assumed to have been directly heading the staircase. As it is several times experienced in tests of operation in real world models of destructed environments, similar to Rescue Robot Red arena of international RoboCup competitions, well positioning in front of the stairs and directing properly before starting the climb is just as important as the climb itself; Therefore autonomously positioning towards the staircase can help the operator save more time. A stair detection algorithm is presented in [10] where the author detects the up-going staircase with a horizontally scanning LRF and the down-going stair case with infra red sensors mounted at the bottom of the wheelchair which is the platform of that research.

In this paper it is assumed that the robot is navigating the environment autonomously. The objective is to (i) generate the "Stair case found" signal. (ii) Move towards a suitable area in front of the staircase and (iii) Climb the stairs.

Innovative compact experimental platform of this research and configuration of sensor installed on it are described in section II. This platform has been awarded several times in international RoboCup Competitions. Perception is presented in section III. In section IV the focus has been on fuzzy controller which is applied for the stair climbing purpose. The results will be demonstrated in section V. Section VI introduces the future work and the research is concluded in section VII.

II. Experimental Platform and Sensor Configuration

A. Locomotion System

Silver is a tracked mobile robot which was designed in 2004 by Resquake Robotics team at K.N. Toosi University of Technology. This robot was first introduced in RoboCup2005 -Osaka, Japan- in rescue robot league and was awarded for the "2nd Place Best Design" of rescue robot. Both the hardware and software of the robot have been improved for RoboCup2006, 2007 and 2008 world championship competitions. The robot was awarded for the "Best operator interface in RoboCup2006 -Bremen, Germany- and Resquake won the "3rd Place" in RoboCup2008 -Suzhou, China- in rescue robot league main competitions and "2nd Place" in "Best in Class Mobility" Challenge. This robot has four independently actuated arms that help it climb obstacles up to 40cm high. Tracks and arms are driven by DC motors. The position of arms and the speed of tracks are measured. Physical specifications of Silver are summarized in table I and an illustrative description of its main characteristics has been presented in [11] and [12].

A robotic arm is mounted on Silver which in the current application holds two LRFs at the height of 45cm in front of the body. The arm is not active in current application. Silver with two LRFs mounted on the robotic arm is shown in Fig.1.

TABLE I
Physical Specifications of Silver

Overall weight	25 Kg
Length with fully extended arms	80 cm
Length with fully folded arms	41 cm
Minimum height	26 cm
Width	40 cm
Maximum velocity	32 cm/sec
Arms maximum angular velocity	4 rpm

A Sony VGN-UX380 laptop is installed on the robot which is used as the high level processor of the robot. The high level programming language is Microsoft C#.Net 2008.

B. Sensors

As depicted in Fig.1, two Hokuyo URG-04LX LRF are installed on the robotic arm. These sensors can measure distances up to 4 meters with 1mm resolution and 5mm tolerance. They provide 769 distance data from 270 degrees sweep angle with the sampling rate of 10Hz. The LRF installed in front (VLRF) scans vertically and the one at the back (HLRF) scans horizontally.

III. Perception

Fig.2 shows the overall autonomous stair climbing process diagram.

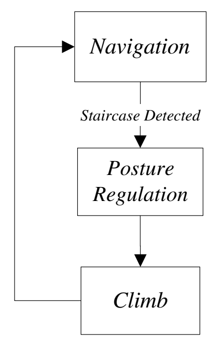

Fig. 2. Overall Process Diagram

Before starting the climb, positions of robot arms are set to 110 degrees. This arm configuration is needed for approaching the staircase. Details will be cleared in the section III.C.

Please note that the autonomous navigation is not the objective of this paper.

A. Finding the staircase

In order to generate "Staircase is found" signal the input array from VLRF is processed. The navigation algorithm receives the interrupt following the success of this procedure to find the staircase, and enters the next state to align the robot in front of the staircase. A sample scan of the up going staircase of Fig. 3(b), is illustrated in Fig. 3(a).

The short lines that can be extracted (we applied splitmerge line fitting algorithm [13]) from the range of 0 to 135 degrees of the sensor viewport represent the stair faces.

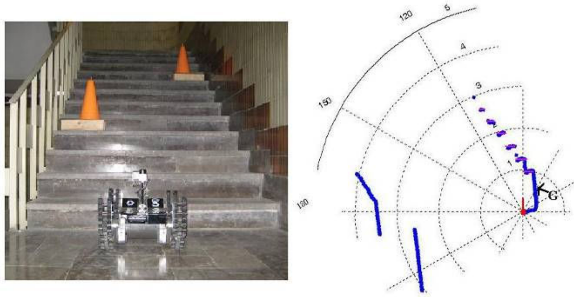

Fig. 3(a). A sample scan of an up-going staircase on the left, (b). The sample staircase on the right

The candidate lines for describing the staircase have the following characters:

1) They are vertical with a length of $\mathbf{L_s}$:
   $12cm < L_s < 25cm$
   A maximum length of 25cm is typical for standard stairs, but the minimum is taken 12cm because the upper stairs are not completely in the field of view of VLRF and the short lines represent a portion of the faces of the upper stairs.
2) The distance between two consecutive lines that represent two consecutive stairs is in range of 25cm to 35cm providing that the robot faces a standard staircase. This is not the same with lines extracted from scanning a book shelf, as in case of a bookshelf the vertical lines are not distant horizontally.
3) At least one pair of above lines is to be found.
4) When the lines have all the 3 above characters, then the HLRF should be able to see a line of at least 150cm in range of 45 to 135 degrees of scanned points.

If all above 4 characters are found the “staircase is found” signal interrupts the navigation algorithm and robot enters the next state.

B. Posture Correction

Staircase can be detected when the robot is faced with the staircase in any distance and angle that VLRF could get an informative scan. But in order to make stair climbing safer and easier, it is needed that robot starts from a good posture. A good posture can be described as being positioned in front of the first step, as near as possible to the center point and facing as directly as possible towards the stairway. In order to reach the proper posture, two separate procedures are considered. Firstly, the robot should be sent to some approximately proper posture by an open loop control, and then the posture should be regulated by a closed loop control.

1) Approximate Posture correction

Both position and heading of the robot need to be corrected before starting the climb. First, robot should turn for 90-Θ degrees, so that it becomes approximately parallel to the line which is extracted from distance points measured by HLRF when the robot found the staircase. Θ is calculated from (1). Then the robot should travel along this way for D millimeters:

$$\theta = tg^{-1}\left(\frac{Y1 - Y2}{X1 - X2}\right) \quad (1)$$

$$D = X_m \cos(\theta) + (Y_m - Y_0)\sin(\theta) \quad (2)$$

$$Y_0 = \frac{(Y1 - X1)(Y2 - Y1)}{X2 - X1} \quad (3)$$

Where Θ, X1, Y1, X2, Y2, D, X_m and Y_m are shown in Fig. 4.

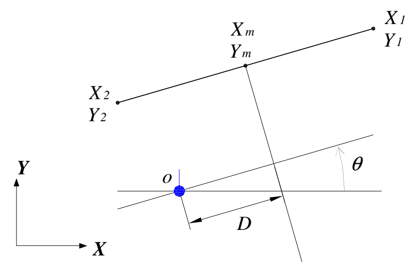

Fig. 4. Parameters of the line that represents a step in the staircase. These parameters are used to calculate D and Θ for approximate posture correction. These are also the inputs of the fuzzy controller for final posture correction.

At last the robot should turn 90 degrees towards the stair case. The robot is planned to move to a primary proper posture based on our knowledge of its linear and rotational velocities and by means of an open loop control. It will then be sent as close as possible to the stairs and its posture is regulated by means of a closed loop controller simultaneously.

1) Final Posture correction

Up to this point, the robot is in a posture that the stair climbing state can be started. But, for a safer and easier climb it is better that robot takes advantage of the distance remained to get to the first step by correcting the posture using a closed loop control. Fuzzy controller will be applied for stair climbing purpose, so it is also possible for the robot to use the same logic to tune the posture. The only difference would be the input information. In the case of posture correction, the first input is slope of the line extracted from the distance array produced by HLRF that represents the heading misalignment and the second is the misalignment of robot position to the center point of the mentioned line. See Fig.4.

Details of the controller implementation are presented in section IV. Final posture correction finishes when robot gets close enough to the first step. This can be monitored by checking VLRF data. In Fig.3(b) the vertical line which is tagged “G” has a very easy to find character. This line starts from robot body and ends to the first step. The distinguishable character is that the distance of all points on this line to the robot position (the position where the VLRF in installed) is always equal to the installation height of VLRF. For this reason, the disappearance or its decrease in size to a small length means the robot is close enough to the first step.

C. *Climbing*

Climbing the stairs can be divided to three main tasks. (1) Approaching, (2) Climbing and (3) Landing. The details are presented here after:

1) Approaching

When the robot is close enough to the stairs, well positioned and well oriented, it is the time to change arm configurations so that robot stays along the stairway slope. Because of the current arm configurations (all in position of about 110 degrees), robot starts pitching when trying to go ahead. When the robot is well-aligned with the stairway slope the front beams of HLRF are not anymore interfered by stairs. Therefore, pitching should continue until the HLRF can see no more near objects in scanning range of 80 to 100 degrees. Then, the arm configuration should change to fully extended configuration. Fig.5 shows the robot position and configuration when approaching the stairs.

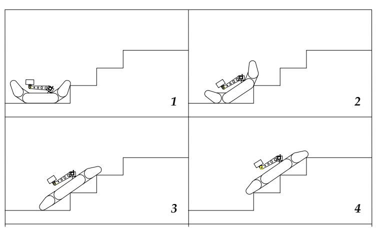

Fig. 5. The robot approaches the stairs. Note the arms configurations

2) Climbing

All it was done up to this point was to prepare the robot for the climb by tuning its start posture. There are two objectives for a fast, safe and efficient stair climbing. (i) Staying parallel to stairway walls and (ii) Centered between the stairway sides. It should be noted that at the moment we assume there are either walls or banisters supporting the stairway sides. To control the two above parameters, we use the same Fuzzy logic control that we used for final posture correction. But this time the first input is the difference between the distances of the nearest object on the right and left. The nearest object on the right is found by finding the minimum distance among the -45 to 45 degrees of scanning range of HLRF and the nearest object on the left is found by searching 135 to 225 degrees. The second control input is the misalignment of the robot orientation with the staircase walls. In Fig. 6 the two mentioned errors are shown. Details will be provided in section IV of this paper.

3) Landing

Climbing accomplishes when a long line can be extracted from VLRF inputs starting at y=0. This line is not parallel to Y axis (See line "G" in Fig.7) because VLRF and the ground plane are not parallel. It is enough for the length of this line to be more than twice the depth a standard stair (more than 70cm) to indicate the end of staircase. When this line is found, which means the robot has reached the top, the arm configuration should change to 200 degrees to guarantee a soft landing. The robot continues driving forward until the line "G" becomes vertical (along Y axis). At this point the position of robot arms are set back to 110 degrees and robot navigates on the upper floor.

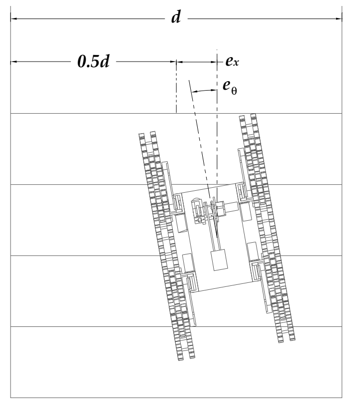

Fig. 6. X and Θ errors for stair climbing. Fuzzy controller minimizes these two error values

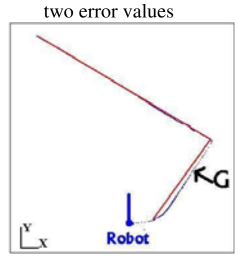

Fig. 7. Climbing finishes when a long line can be extracted from Vertical LASER range finder inputs starting at Y = 0. This line is not vertical because Vertical LASER range finder and the ground plane are not parallel when the robot has not completely landed.

IV. CONTROL

The fuzzy controller both for final posture correction and for stair climbing are described in this section.

To correct the errors in X misalignment and Θ misalignment the speeds of right and left tracks should be controlled. To keep the robot moving ahead it should be guaranteed that both tracks are not stopped. So, it is assumed that both tracks run with maximum speeds and the controller sets the amount of decrease in the speeds of right and left tracks.

Triangular membership functions for fuzzy inputs and fuzzy outputs, Mamdani MIN - MAX inference and middle of maxima defuzzification method are used. Membership functions are shown in Fig.8. Rule bases for different X and Θ conditions are listed in table II(A) and (B).

TABLE II

A) RULE BASE FOR Θ

Θ Value	Neg.	Zero	Pos.
Decrease in Left Track Speed	Small	Small	Large
Decrease in Right Track Speed	Large	Small	Small

B) RULE BASE FOR X

X Value	Neg.	Zero	Pos.
Decrease in Left Track Speed	Large	Small	Small
Decrease in Right Track Speed	Small	Small	Large

Please note that the type and limits of the input and output membership functions of the fuzzy controller are usually determined by try and error. It has been the same for this controller.

The effects of two inputs on the outputs are separately considered. The reason is that sometimes X and Θ conditions force contradictory decisions on the right and left tracks. For example when both Θ and X are negative, to correct error in Θ, robot should turn counterclockwise and to correct the error in X, robot should turn clockwise. To solve similar contradictions in other applications, there are several control configurations like cascade and hierarchical [14] configurations. The latter approach is used here that means decrease in Right and left speeds are twice calculated by separate Θ controller and X-controller. Then the final result is determined by weighting the two above results in (4) and (5).

$$DLS = \frac{w1 \times DLSX + w2 \times DLS\theta}{w1 + w2} \quad (4)$$

$$DRS = \frac{w1 \times DRSX + w2 \times DRS\theta}{w1 + w2} \quad (5)$$

Where w1 is amount of the non-zeroness of X-misalignment, w2 is the non-zeroness of Θ-misalignment, DLSX and DRSX are the amount of decrease in left and right speeds calculated by X-controller, DLSΘ and DRSΘ are the amount of decrease in left and right speeds calculated by Θ-controller and DLS and DRS are final amounts of decrease in left and right speeds.

V. RESULTS

The above algorithms were applied to Silver on different staircases and by applying different disturbances to the robot. The robot experienced successful climb on all available standard test staircases. Here we focus on the results from a 12-steps test staircase. Dimensions of the test staircase are provided in table III.

The staircase is bounded by a wall on the right and banisters on the left. Some informative photos from the first test are shown in Fig.9. The graphs in Fig.10 depict X errors and Θ errors of the stair climbing. Please note that in this test, some obstacles are added to stair sides on the left and right.

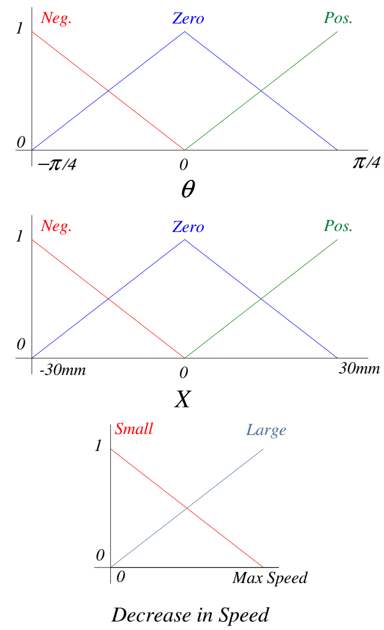

Fig. 8. Input membership functions for X and Θ, and output membership functions for decrease in the amount of speeds. The shape of output membership functions is the same for left and right track speeds.

TABLE III

DIMENSIONS OF TEST STAIRCASE

Stair height	18cm
Stair Depth	32cm
Stair Width	150cm

Fig.9. Informative photos from the first test

It could be seen in Fig.10 that the controller manages to keep the robot in a safe distance of the stair sides and is successful in decreasing the error of heading misalignment that means to increase the speed and efficiency of the climb. When (See Fig. 10) robot gets near the first obstacle (7s), a positive error in X (about +14cm) is generated. The controller tries to cancel this error. The cost is worsening errors in Θ. When the next obstacle is detected on the right (13s), error value in X becomes negative (about -11cm). The controller again tries to compensate the error and finally robot gets to the end of staircase with compensated X and low Θ errors. Please note that the noise in graphs are due to track slippages, intrinsic measurement errors of the sensors and errors of line fitting algorithm.

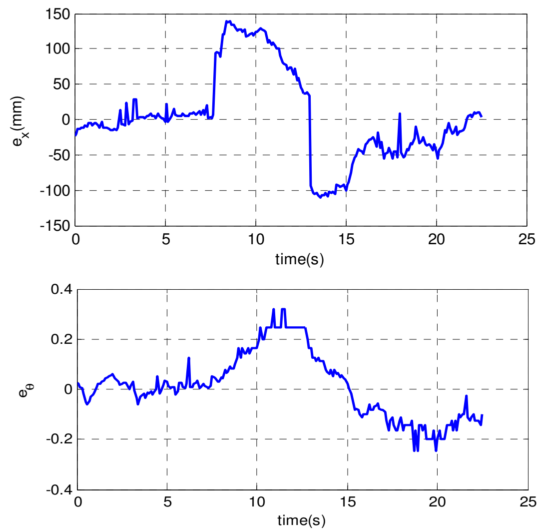

Fig. 10. Errors in X at the top and errors in Θ at the bottom

In another test, one of the front arms were forced to stay fully folded when the robot was climbing. This means to add a non linear and non measurable disturbance to the climb process. Because the traction forces on both sides are not anymore equal the tendency of the robot to increase both position and orientation errors are added to the stair climbing process. Although it was more difficult for the controller to compensate the errors (because of the unequal slippage of the tracks in this test), robot passed the test successfully.

VI. Future Work

Assuming that there exists either a wall or banisters at the stair sides is not always correct; especially in case of destructed environments. To include the staircases with no bounding elements to autonomously surmountable obstacles, feedback from stair sides are to be removed. In that case, it is needed to get information from stair faces. The control will be similar to what explained in "final posture correction" part in section III of this paper. In order to do that, a stabilizing mechanism, to compensate roll and pitch angles of the robot on the stairs, is required to keep the LRF sensors parallel to the ground plane. We will add it in the future work.

We will apply other control methods and compare controller results in future works.

Steps for declining the staircase are very similar to the steps for climbing. We will consider this in future work as well.

VII. Conclusion

An autonomous stair climbing from generating "staircase found" signal to landing on top of the staircase was presented in this paper. Fuzzy control system was applied to keep the robot away from stairway sides in order to guarantee a safe climb and to keep the robot body parallel to stair sides in order to increase climbing speed and efficiency. Experimental results were presented when different disturbances were applied to the robot during the climb. The method showed success when it was tested on different standard staircases.

References

[1] A. I. Mourikis, N. Trawny, Stergios I. Roumeliotis, D. M. Helmick, and L. Matthies, "Autonomous Stair Climbing for Tracked Vehicles," *International Journal of Robotics Research*, vol. 26(7), Jul. 2007, pp. 737-758.

[2] D. M. Helmick , S. I. Roumeliotis, M. C. McHenry, and L. Matthies, "Multi-sensor, high speed autonomous stair climbing," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, Lausanne, Switzerland, vol. 1, Sep/Oct. 2002, pp. 733-742.

[3] J. Liu, Y. Wang, S. Ma, and B. Li, "Analysis of stairs-climbing ability for a tracked reconfigurable modular robot," in *Proc. IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR)*, Kobe, Jun. 2005, Japan, pp. 53–38.

[4] S. Steplight, G. Egnal, S. H. Jung, D. Walker, C. Taylor, and J. Ostrowski, "A mode-based sensor fusion approach to robotic stairclimbing," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, Takamatsu, Japan, Nov. 2000, pp. 1113-1118.

[5] E. Z. Moore, D. Campbell, F. Grimminger, and M. Buehler, "Reliable stair climbing in the simple hexapod 'RHex'," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 3, Washington, DC, May 2002, pp. 2222–2227

[6] M. Yim, S. Homans, and K. Roufas, "Climbing with Snake-like Robots," in *Proc. IFAC Workshop on Mobile Robot Technology*, Jejudo, Korea, May 2001

[7] M. Fair, and D.P. Miller, "Automated Staircase Detection, Alignment, & Traversal," in *Proc. IASTED Int'l Conf. on Robotics and Manufacturing*, Cancun, Mexico, May 2001, pp 218-222.

[8] M. Fair, "Autonomous Stair Climbing with a Mobile Robot," M.S. Thesis, School of Aerospace & Mechanical Eng., University of Oklahoma, Sep. 2000.

[9] W. Chonnaparamutt, and A. Birk, "A Fuzzy Controller for Autonomous Negotiation of Stairs by a Mobile Robot with Adjustable Tracks," RoboCup 2007: Robot WorldCup XI, Lecture Notes in Artificial Intelligence (LNAI), Springer, 2007, pp. 196-207.

[10] J. D. Martens, and W. S. Newman. "Stabilization of a Mobile Robot Climbing Stairs," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, San Diego, CA, USA, vol. 3, 1994, pp. 2501-2507.

[11] S. Ali A. Moosavian, H. Semsarilar, and A. Kalantari, "Design and Manufacturing of a Mobile Rescue Robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS)*, Beijing, China, Oct. 2006, pp. 3982-3987

[12] S. Ali.A. Moosavian, and A. Kalantari, "Experimental Slip Estimation for Exact Kinematics Modelling and Control of a Tracked Mobile Robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2008, Nice, France.

[13] D. H. Douglas, and T. K. Peucker, "Algorithms for the Reduction of the Number of Points Required to Represent a Line or its Caricature," in *The Canadian Cartographer*, vol. 10(2), Dec.1973, pp. 112–122.

[14] J. Vascak, "Fuzzy Control of a Physical Double Inverted Pendulum Model," in *Proc. Computational Intelligence (Theory and Applications) International Conference 6-th Fuzzy Days*, Dortmund, Germany, May 1999, pp. 482-494.