Preprints of the 18th IFAC World Congress
Milano (Italy) August 28 - September 2, 2011

Autonomous stair climbing with multisensor feedback

Przemysław Łabęcki, Krzysztof Walas, Andrzej Kasinski

Poznan University of Technology, Poznan, Poland (e-mail: {przemyslaw.labecki,krzysztof.walas,andrzej.kasinski}@cie.put.poznan.pl).

Abstract: This article presents an autonomous stair climbing procedure with multisensor feedback. The task is performed by the six-legged robot Messor. The control procedure is performed in a closed-loop, and requires appropriate measurements. The sensory system of the robot consists of a structural light system and a laser range finder. The first one provides information about the dimensions of stairs and the second one measures current pose of the robot on stairs. The execution of the procedure is fully automatic. Once the stairs are found no other external input to the system is required till the top of the stairs. The climbing strategy was developed in simulation and then transferred to the real robot controller. The Messor robot is able to climb stairs of different sizes, and thanks to the sensory feedback copes well with the imperfectness of the environment.

Keywords: mobile robots, walking, vision, perception, multisensor

1. INTRODUCTION

Mobile robots are supposed to work in a man-made environment. But in most cases they are unable to cope with common obstacles in this surroundings, for example stairs. This paper presents an automatic procedure for real size stairs climbing with a six-legged walking robot. The input of the procedure are dimensions of the stairs and position of the robot on the stairs. This data are provided by the multisensor system mounted on the robot. The system built by the authors synthesises and progresses the work done by other scientists in recent years.

In the field of mobile robotics much research has been done to enable robots to negotiate stairs. The stair climbing algorithms for wheeled robots have been investigated. In such case wheels should be lifted in vertical axis to traverse such an obstacle, see Choi et al. (2007). Some research has been done also w.r.t. tracked robots. The robots presented by Matthies et al. (2002) and by Mihankhah et al. (2009) are able to climb stairs.

These robots have however one main disadvantage, they are non-holonomic and are not easy to move horizontally on the staircase. In contrary, walking robots are holonomic systems, but are not so well backed with the sensors as tracked robots. For example, the experiment reported by Moore et al. (2002) refers to the open-loop control strategy involving the available environment model with defined parameters. The paper presented by Kalakrishnan et al. (2010) demonstrates the ability of a four-legged robot to perform complex movements, for example to climb stairs. The biped robots are also capable of negotiating stairs but in this case the problem is the poor balance of the robot. The exemplary experiments with biped robots are presented by Albert et al. (2001); Gutmann et al. (2004). The mentioned biped robots use computer vision system to navigate on stairs. Albert et al. (2001) use a structural light system, and Gutmann et al. (2004) use stereo vision. The tracked robot presented by Matthies et al. (2002) relies on a vision system while climbing stairs, and the robot presented by Mihankhah et al. (2009) uses a laser range finder for this purpose.

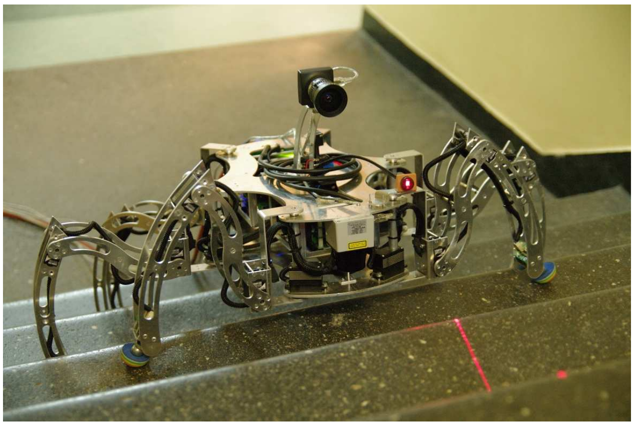

Fig. 1. Messor Robot

In our approach the six-legged robot “Messor” was used to perform the task. The choice of kinematic architecture of the robot is based on its augmented static balance capability. Keeping balance enables the robot to climb stairs without falling down. The presented control procedure is performed in a closed-loop, thus the execution of the task is fully automatic. Once the stairs are found no other external input is required till the top of the stairs. The sensory system comprises of a structured light system and a Laser Range Finder (LRF) and exploits both sensors at one time.

The article consist of five parts. At the beginning the robot Messor and its sensory system are described. Next, data acquisition by the sensory system is presented. Then the

Copyright by the International Federation of Automatic Control (IFAC)

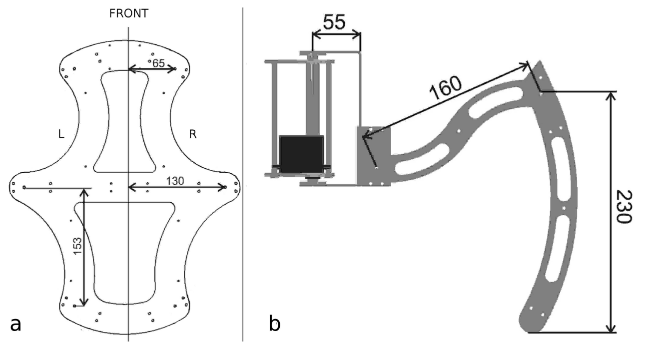

Fig. 2. Messor Robot: the trunk (a), the leg (b)

development of the climbing strategy is discussed. At the end the validation of the proposed strategy with the real robot is given.

2. ROBOT "MESSOR"

"Messor" is a six-legged walking robot with three degrees of freedom in each leg. Its trunk is 26 cm wide and 30.6 cm long. The segments of the leg are: coxa 5.5 cm, femur 16 cm, tibia 23 cm. It weights 4.3 kg. The general view of Messor is shown in Fig. 1. The mechanical structure of the trunk and the leg is shown in Fig. 2. The dimensions of the robot trunk and its legs are related to the dimensions of standard stairs. The highest possible stair in the contemporary buildings are up to 20 cm high, the depth of the step is obtained from the inequality $60 < 2*h+d < 65\ cm$, where h – height of the step, and d – depth of the step (Fig. 3 a). The distal part of the leg is 23 cm long, so it is 3 cm longer then the maximal height of the step. The length of the trunk is designed in a way that two pairs of the legs could be placed on the same step. The ability of the robot to climb stairs was one of the guidelines for the mechanical design. The sensory system of the robot was also conceived with the stair climbing task as a target. The detailed description of the system is given in the following section.

3. SENSORY SYSTEM OF THE ROBOT

The measurements necessary to support the stair climbing algorithm can be divided into two groups (Fig. 3). The first group consists of measurements used to determine stair dimensions such as stair height h and depth d. The second group comprises of measurements used to determine the position and orientation of the robot w.r.t. the stairs such as the distance to the nearest stair d_F, robot orientation θ and the relative position d_{LR} of the robot w.r.t. the stair borders (computed as $(d_L + d_R)/d_L$).

Acquisition of the stair height h and depth d data is performed by using the vision system based on the laser light beam projected vertically and a video camera uEye (UI-1225LE-C). The schema of the system is shown in Fig. 4.

The second group of measurements is related to the current position and orientation of the robot. Data is obtained by using the Hokuyo laser scanner (URG-04LX). Because of the limitations of the URG-04LX, it is not advisable to rely on the explicit range measurements

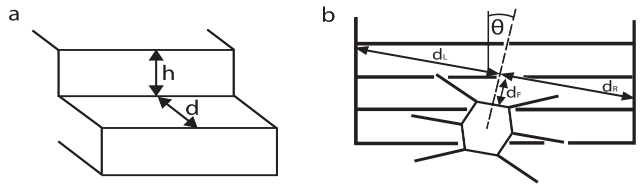

Fig. 3. Measurements required for stair climbing

obtained from the scan. However, for a given distance to the stair, the measurement error is similar for all scan points. This allows to use the LRF to determine the orientation θ and the relative position w.r.t. stair borders d_{LR}. The accuracy of range measurement could be improved by using remission values, and the technique described by Yoshitaka et al. (2006). Finally the remaining parameter d_F has to be acquired by the structured light system. All the measurements acquired by the robot require additional conditioning, which is described in the next section.

4. ACQUISITION AND CONDITIONING OF THE MEASUREMENT RESULTS

4.1 Measurements using the URG-04LX laser rangefinder

The URG-04LX LRF provides a set of points in a 2D Cartesian coordinate system (Fig. 5 a). To acquire the desired measurements, linear segments are fitted to the points. First, the points are divided into connected groups (Fig. 5 b). All groups are then filtered using an averaging filter, to achieve smoothness of the curve, and each smoothened curve is simplified to a few key points using the Douglas-Peucker algorithm (Fig. 5 c). These key points are used to determine the ends of linear segments. Afterwards, lines are fitted to points between these key points, to produce linear segments (Fig. 5 d). The line segment laying in front of the robot is used to determine the orientation w.r.t. the stairs and the extreme points of all segments are used to determine the relative position on the stairs. A detailed description of the process can be found in an article by Łabęcki and Walas (2010).

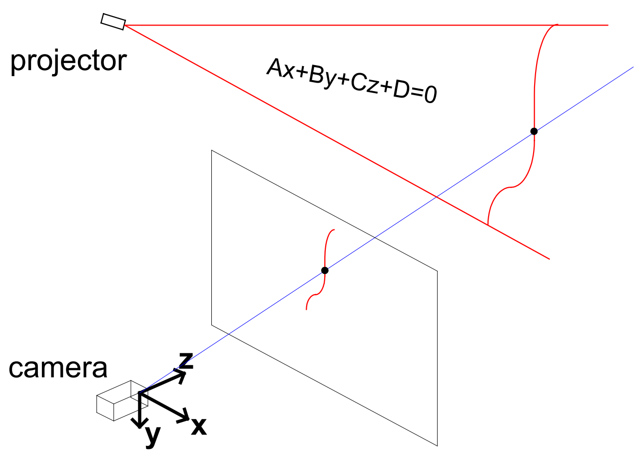

Fig. 4. General schema of the structured light system

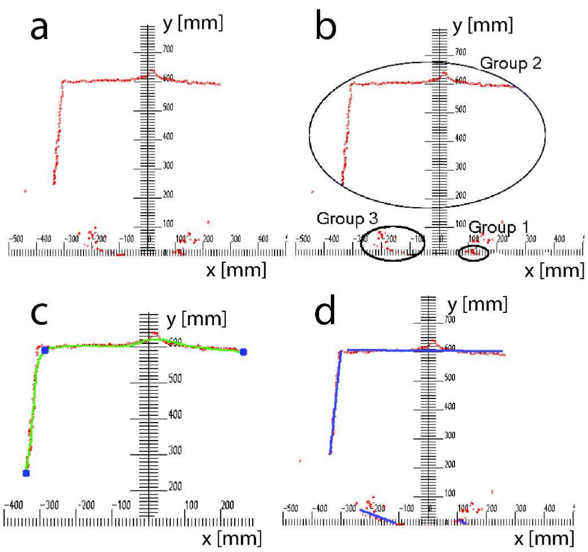

Fig. 5. Processing of the LRF measurements. Raw LRF measurements (a), points divided to groups (b), smoothened line with key points (c), fitted linear segments (d)

4.2 Measurements using the structured light system

The structured light system is used to acquire the width and the height of the step as well as the distance to the nearest step. These parameters are calculated using the 3D coordinates of the stair contour found by the structured light system. To find the linear segments laying on the stair contour, a differential image is taken (Fig. 6 c). This image is then thresholded to find the red stripe. Then connected components (blobs) are found in the binary image (Fig. 6 d). Blobs are thinned (by averaging the pixel position in each row), producing curves, to which linear segments can be fitted. The fitting method is similar to the one used in LRF data processing (smoothing, Douglas-Peucker and line fitting), as shown in Fig. 6 d and e. The found linear segments are then classified to two groups: vertical and horizontal. Only the vertical segments are used. The top points of two nearest vertical stripe segments are sufficient to determine all desired parameters, providing that the optical axis of the camera is perpendicular to the vertical faces of the stairs. The algorithm for finding stair parameters using a structured light system has been described in detail in the article by Łabęcki and Walas (2010). In current paper the calculation of 3D points coordinates is changed, comparing to the algorithm presented in Łabęcki and Walas (2010). In the original algorithm, a perfectly aligned location of the laser stripe plane in the camera coordinates was assumed. In practice, such location is difficult to achieve and any misalignment produces measurement errors. Therefore a more general method of calculating 3D points based on the position of the stripe in the image has been used. This method is presented in the next subsection, together with an algorithm for calibration of a corresponding structured light system.

4.3 Range calculation using the structured light system

The elements needed to calculate the 3D position of a point that corresponds to a laser stripe pixel are: intrinsic camera parameters (acquired via camera calibration), pixel position in the image and the parameters of the laser stripe plane in the camera coordinate system. The process of calculating laser stripe plane parameters (i.e. the process of calibrating the structured light system) has been described at the end of this subsection.

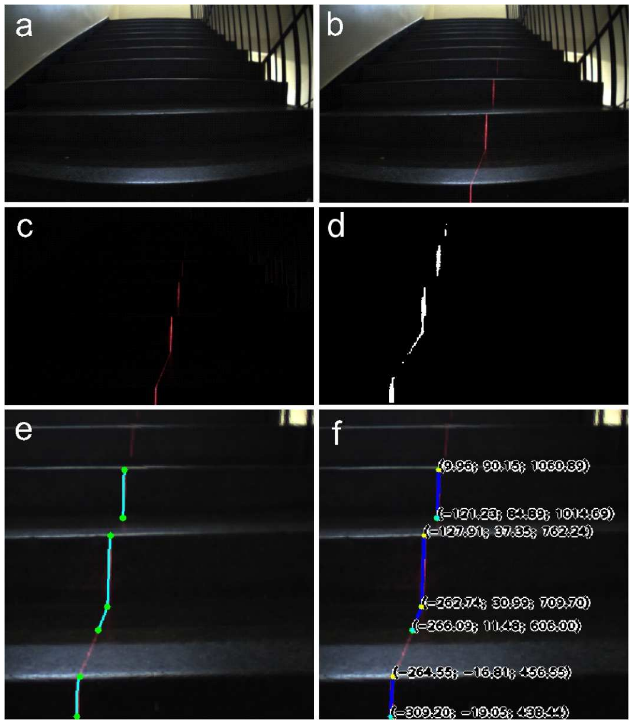

Fig. 6. Processing of the data acquired by the structured light system. Images for the differential image (a, b), the differential image (c), found laser stripe (d), processed laser stripe (e), found linear segments (f)

Every stripe pixel detected in the image is a projection of a 3D point from the stripe plane. Therefore the 3D coordinates of this point can be found by calculating the intersection of the projection line corresponding to this pixel with the laser stripe plane. This concept has been illustrated in Fig. 4.

After correcting the distortion caused by the lens (and using intrinsic camera parameters for an undistorted image), the parameters of the line corresponding to a given pixel can be calculated using two points: the coordinates of the pixel in the camera coordinate system and the origin of this system. The coordinates of the pixel are defined as follows:

$$\mathbf{P}_K = \begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} u_p - cc_{u1} \\ v_p - cc_{u2} \\ f \end{bmatrix}, \tag{1}$$

where f is the focal length of the camera (in pixels), $[u_p \; v_p]^T$ are the image coordinates of the pixel, $[cc_{u1} \; cc_{u2}]^T$ are the coordinates of the principal point of the camera (in image coordinates) and $\mathbf{P}_K = [x_p \; y_p \; z_p]^T$ are the coordinates of the pixel in the camera reference frame, scaled in pixels.

The line between the point $\mathbf{P}_K$ and the origin of the coordinate system can be expressed using parametric equations:

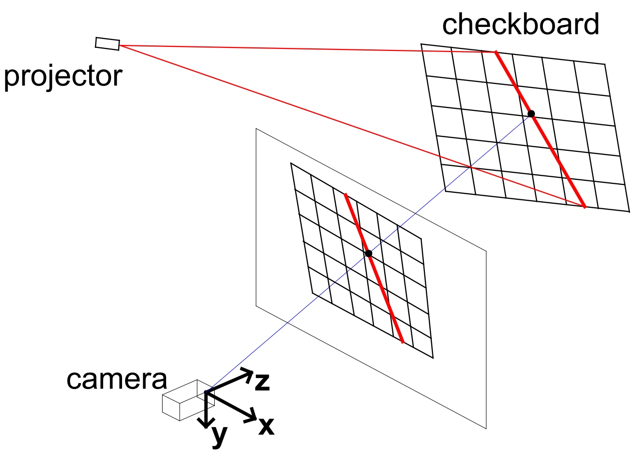

Fig. 7. The intersection of the line corresponding to a selected stripe pixel with the checkboard plane

$$\begin{cases} x = x_p t \\ y = y_p t \ , \\ z = z_p t \end{cases} \tag{2}$$

assuming $[x_0\ y_0\ z_0]^T = [0\ 0\ 0]^T$ and $[a\ b\ c]^T = [x_p\ y_p\ z_p]^T$. If the laser stripe plane is given in the form of:

$$Ax + By + Cz + D = 0, \tag{3}$$

then the parameter t_i for which the line corresponding to a pixel intersects with the plane can be calculated as:

$$t_i = \frac{D}{Aa + Bb + Cc}. \tag{4}$$

The coordinates of the intersection are therefore:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} at_i \\ bt_i \\ ct_i \end{bmatrix}. \tag{5}$$

The process of finding parameters of the laser stripe plane in the camera coordinate system is performed in two stages. The first stage is the calibration of the camera using a planar checkboard. Intrinsic camera parameters are found in this process. Extrinsic parameters are also calculated, therefore the position of the checkboard visible in every calibration image is known. It is essential for the second stage, that the laser stripe is visible within the checkboard in each calibration image.

The second stage is the calibration of the structured light system. From every calibration image, two pixels from the stripe are selected. As stated before, the parameters of the checkboard plane in the camera coordinates are known for every calibration image. Therefore the position of the point corresponding to the selected stripe pixel can be calculated in the same way the position of pixel in 3D space is computed, using equations 1–5. An illustration of this process is shown in Fig. 7. For every calibration image, two pixels laying on the laser stripe are selected to calculate the corresponding points. Each of these points lays in the stripe plane. Therefore a group of calibration images with laser stripe visible within the checkboard provides a cloud of points laying in one plane – the laser stripe plane. The parameters of this plane were calculated using least squares fitting, although any robust plane fitting algorithms would be appropriate.

5. CLIMBING STRATEGY

The preliminary results of the climbing strategy development were presented in Walas (2010). The strategy was implemented in a realistic simulator of a six legged robot. The simulator is based on the PhysX™ technology provided by NVIDIA®. The tests on the simulator resulted with the strategy which block diagram is presented in Fig. 8. The description of each phase of the climbing is given in the following paragraphs.

Prepare for climbing – the robot stands at some distance from the stairs. To perform initial measurements of the dimensions of the stairs, first the orientation of the robot w.r.t. stairs is established. Then the proper height and depth of the step is obtained. The robot moves closer to the stairs and repeats the measurements to obtain more accurate stair dimensions. Then the robot walks until it reaches the edge of the stairs. Here the robot is aligned one more time. The array *dist[]* is set. Each variable of the array holds information on the distance of the pair of legs to its nearest step. While the robot is at the edge of the stairs, the distance of each pair to the nearest step is established by using geometrical dimensions of the robot. The *dist[]* is the key structure in the presented climbing strategy. This stage of the strategy is shown in Fig. 10 a.

Place foot on step – the robot is at the edge of the stairs. It can place its first pair of legs on the step. The *dist[]* variable for the appropriate pair of legs is set and is equal to the depth of the stair. This stage of the strategy is shown in Fig. 10 b.

Find pair with least distance to the nearest step – to perform the walking motion on the stairs the appropriate distance for the walking phase has to be established. The function searches the variables from *dist[]* with the least value and stores the index of the variable in *move*.

Walk – the walking procedure refers to the variable *move* to index *dist[]* array and to perform a walk in this cycle of the procedure. The walking procedure works in closed-loop. Due to the slippages the robot is unable to reach the proper position. Using the visual feedback, the difference between the reference value and the actual value is obtained. The robot walks the missing distance. This stage of the strategy is shown in Fig. 10 c.

Move pair back – while putting the legs pair on the step the legs are moved forward. After walking some distance according to the *walk* the robot is able to move the feet back in order to gain some distance on the step. In narrow steps the adaptive part of this function is switched on. The problem posed by the step narrowness is the following. As the distance walked by the robot is smaller than backward movement of the legs, the only possibility to move the legs back is by the distance walked by the robot. The remaining part of the backward movement is then performed in the next cycle. This stage of the strategy is shown in Fig. 10 d.

End of stairs – the strategy is performed until the top of the stairs is reached. When the end of the stairs is reached the remote operator or some other algorithm from the higher control layer should stop climbing. The robot performs the landing procedure.

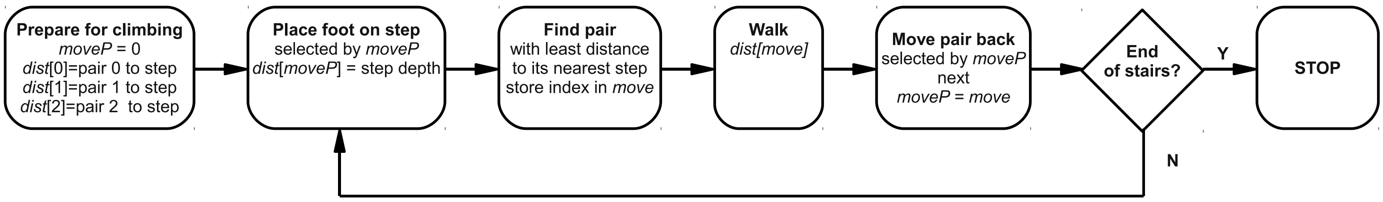

Fig. 8. Block diagram of climbing strategy

Each *place foot on step* is preceded by alignment. Another adaptive function is the horizontal positioning on the step. Before each walk procedure, the robot is able to change its horizontal position in order to avoid some obstacle or to walk at the equal distance from stairs' side-borders.

The obtained strategy has been tested on the real robot and validated. The results of the tests are reported in the following section.

6. STRATEGY VALIDATION ON THE REAL ROBOT

The strategy implemented in the simulator was then transferred to the real robot. The controller of the robot enables setting the pose of the robot as well as locating tip of each leg independently. The control is based on robot kinematics. The climbing strategy worked well in the open loop mode, where the information normally obtained from the sensors were passed as the parameters by the user. In this mode the robot was able to climb 2 steps and then falls down due to the lack of the feedback.

The output of the test in the real envirinment with the closed loop control is shown in Fig. 9. The top figure shows the trajetory of the robot Center of Mass (CM) during the stair climbing in xz-plane, and the bottom one shows the same trajectory seen in the xy-plane. The small letter indicates pairs of legs which are lifted in each phase of climbing: front(f), middle(m), rear(r). As it could be seen the CM is always above the level of the step, and the sequence of lifting pairs after passing the initial phase is stable namely: front, rear, middle. The sequential character of the strategy is due to the sequential character of the obstacle.

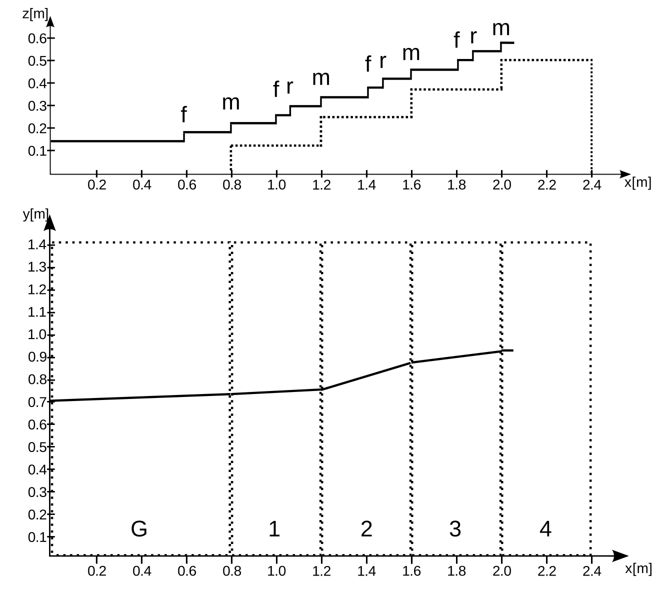

Fig. 9. Robot Center of Mass trajecotry recorded during the real world experiment: top – the trajectory in xz-plane, bottom – the trajectory in xy-plane. The letter G indicates ground and the numbers indicate steps. The small letters indicate the pair of legs lifted in each phase: f – front, m – middle and r – rear. Dashed line depicts the geometry of stairs.

As it is shown in Fig. 9 bottom, the robot path was skewed. It is due to the power suply cable which was draged by the robot. The high influence of this perturbation is observed when the robot is on the second step. It is due to the fact that the whole cable was lying on the ground. The closed-loop algorithm is robust to this kind of noise but only when both sides of the staircase are visible. The robot finds its relative horizontal position as a proportion of distances to the borders.

During the integration of the data from the sensors into the strategy, more problems arised. One of the problems was caused by the field of view of the camera. In some cases the step nearest to the robot was not visible for the computer vision system, so the appropriate subtraction of step depth from the measurement was necesary in order to obtain appropriate distance to this nearest step. The subtraction had to be also adaptable. In situation when the nearest step was in the field of view of the camera before performing the movement, but after performing the movement the step was invisible we didn't have to perform the subtraction beforehand but we had to do this afterwards.

One more observation was made. The structured light system behave well in the in-door environment but there was a deficite of power of the laser stripe while performing the experiments out-door in the daylight.

The evidence of validation of the stair climbing strategy is presented in Fig. 10 and also could be seen in the movie: http://lrm.cie.put.poznan.pl/climbing_stairs_IFAC.wmv.

7. CONCLUSIONS

Autonomous stair climbing strategy with multisensor feedback was proposed. The strategy was first developed in the robot simulator environment Walas (2010) and then deployed to the real robot controller. The use of the sensory data in the strategy posed some problems, but most of them were successfully solved. The robot is able to climb stairs with the height of 12 cm and depth of 37 cm, as well as of more challenging 14 cm high and 29 cm long. The array with the distance of each leg-pair to its proximal step is a crucial data structure to the strategy. It holds the record of movements of the legs during the stair climbing. The sensory feedback provided by the vision system and the LRF allows us to cope with the imperfectness of the en-

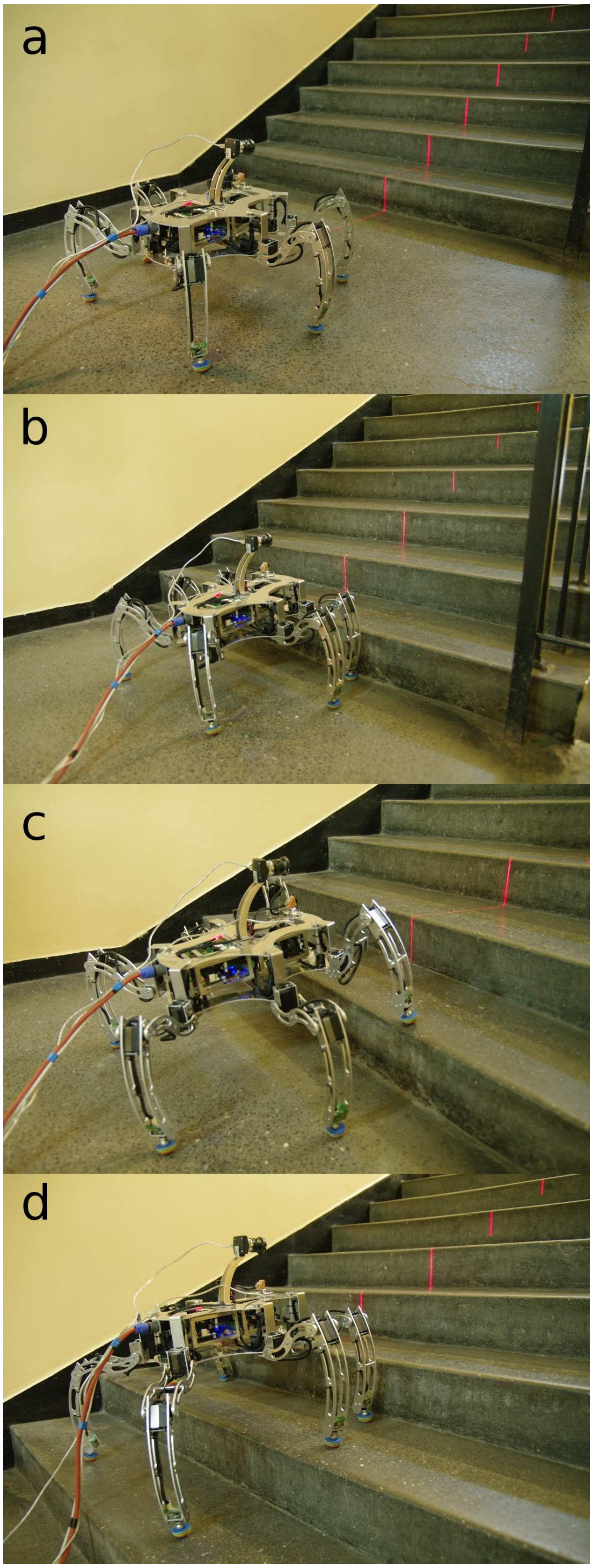

Fig. 10. Phases of climbing procedure. Prepare for climbing (a), at the edge of the stairs (b), on the first step (c), on the second step (d)

vironment model and of the robot itself. The contribution of the authors is the successful closed-loop stair climbing strategy for the real size stairs performed on the six-legged robot.

In the future the laser stripe with more power could be used to allow the robot to work also in the outdoor environment. The authors are currently performing some preliminary tests with other vision based perception system i.e. stereo vision. Another approach for the perception system under development is an active haptic perception concept. It involves the use of the legs' force sensors to sense the surroundings. It could be used as an auxiliary source of data to support vision system, which is disabled by occlusions, or could work as the main sensor on its own.

REFERENCES

Albert, A., Suppa, M., and Gerth, W. (2001). Detection of stair dimensions for the path planning of a bipedal robot. In *Proc. IEEE Int. Conf. on Adv. Intell. Mechatron.*, 1291–1296. Como, Italy.

Łabęcki, P. and Walas, K. (2010). Multisensor perception for autonomous stair climbing with a six legged robot. In *Proc. Int. Conf. on Climbing and Walk. Robots and the Support Technol. for Mob. Mach. (CLAWAR'10)*, 1013 – 1020. Nagoya, Japan.

Choi, H.D., Woo, C.K., Kim, S., Kwak, Y.K., and Yoon, S. (2007). Independent traction control for uneven terrain using stick-slip phenomenon: application to a stair climbing robot. *Auton. Robots*, 23(1), 3–18.

Gutmann, J.S., Fukuchi, M., and Fujita, M. (2004). Stair climbing for humanoid robots using stereo vision. In *Proc. IEEE Int. Conf. on Intell. Robots and Syst. (IROS'04)*, 1407–1413. Sendai, Japan.

Kalakrishnan, M., Buchli, J., Pastor, P., Mistry, M., and Schaal, S. (2010). Fast, robust quadruped locomotion over challenging terrain. In *Proc. IEEE Int. Conf. on Robotics and Autom. (ICRA'10)*, 2665 – 2670. Anchorage, AK.

Matthies, L., Xiong, Y., Hogg, R., Zhu, D., Rankin, A., Kennedy, B., Hebert, M., Maclachlan, R., Won, C., Frost, T., Sukhatme, G., McHenry, M., and Goldberg, S. (2002). A portable, autonomous, urban reconnaissance robot. *Robotics and Auton. Syst.*, 40(2-3), 163 – 172.

Mihankhah, E., Kalantari, A., Aboosaeedan, E., Taghirad, H., Ali, S., and Moosavian, A. (2009). Autonomous staircase detection and stair climbing for a tracked mobile robot using fuzzy controller. In *Proc. IEEE Int. Conf. on Robotics and Biomim. (ROBIO'08)*, 1980–1985. Bangkok, Thailand.

Moore, E., Campbell, D., Grimminger, F., and Buehler, M. (2002). Reliable stair climbing in the simple hexapod rhex. In *Proc. IEEE Int. Conf. on Robotics and Autom. (ICRA'02)*, 2222–2227. Washington, DC.

Walas, K. (2010). Fully parametrized stair climbing strategy for a six-legged walking robot. In *Proc. Int. Conf. on Climbing and Walk. Robots and the Support Technol. for Mob. Mach. (CLAWAR'10)*, 777 – 784. Nagoya, Japan.

Yoshitaka, H., Hirohiko, K., Akihisa, O., and Shinichi, Y. (2006). Mobile robot localization and mapping by scan matching using laser reflection intensity of the sokuiki sensor. In *Proc. IEEE 32nd Conf. Ind. Electron. (IECON'06)*, 3018 – 3023. Paris, France.