

THE ANALYSIS OF INFLUENCE OF SENSORS' FAILURE ON THE PERFORMANCE OF MOBILE ROBOT AUTONOMY

Submitted: 4th September 2014; accepted 2nd October 2014

Piotr Bigaj, Jakub Bartoszek, Maciej Trojnacki

DOI: 10.14313/JAMRIS_4-2014/35

Abstract:

This work is concerned on sensitivity analysis of semiautonomy algorithm of a mobile robot to environmental sensors' failures. Construction of the robot, semiautonomy algorithm and used sensors have been described. The algorithm bases on a reactive hybrid approach that merges data from different types of sensors and calculates resulting velocities. This algorithm also takes into account environmental sensors' damage by modifying the behavior of robot in accordance to actual sensors' set state of health. Simulation research using Matlab/Simulink package and experimental tests' results of semiautonomy algorithm were presented. The experimental tests were carried out in outdoor conditions. The research and tests are performed for normal environmental sensors' operation and for selected sensors' damage. On that basis, sensitivity of semiautonomy algorithm to selected environmental sensors damage was tested.

Keywords: *mobile robot, semiautonomy algorithm, sensors' failure, matlab/Simulink, simulation research, algorithm sensitivity analysis, hybrid reactive approach*

1. Introduction

One of the most discussed issues in the field of mobile robotics is the problem of autonomous movement in highly diverse and unknown environment [1]. In most cases, the problem of robot movement is described as a problem of finding the shortest and collision-free movement trajectory from current robot position to its goal. Algorithms used to ensure proper realization of robots assignments highly depends on data gathered from wide range of sensors. Depending on type of used sensors, different type of sensor failure may occur. The nature of those failures is often possible to be determined only after they have occurred. High diversity of possible failures, and problem of detecting them cause the need of taking into account possibility of incorrect data presence in the system at first stage of algorithm design.

As these failures are difficult to predict, their appearance may cause serious robot's and its surroundings damages. Diagnosing failures and the fastest possible reaction to them is the key issue. In scientific elaborations two main approaches to the described problem are mentioned [2]. The first of them is based on creation of

sensors model and comparing its prediction to real data [3]. Some of proposed solutions require interference into whole system structure [4]. Another, different approach has been presented in [5]. The goal of that elaboration was to develop a method that will enable detection of a previously unnoticed sensor failure and an adequate reaction to it. Described method is based on determining the level of credibility of the data sent by the sensors. The complex solution based on merging two mentioned approaches was presented in [6]. Algorithm described in that elaboration is model based but also uses artificial neural network to detect and react on sensors' failures.

The best remedy for sensors failures is sensor redundancy but it requires an increase of financial costs of whole robot construction.

This article presents the methods of reacting to the sensors' failures used in the control algorithm of the IBIS robot.

2. The IBIS Mobile Robot

The commercial version of IBIS is designed for pyrotechnical and combat missions to operate in diverse terrain like sand, snow or rocky bulk. The basic technical features of the robot can be found in [7].

For research activity, a new version of IBIS robot without manipulator has been designed (Fig. 1a). The IBIS robot can perform in one of two modes: teleoperation in which the movement of the robot is controlled by the operator, and semiautonomy in which the robot follows a path omitting obstacles. The robot has been equipped with a frame with sensors forming the modular structure. Sensor's frame is a module that allows operation in semiautonomy mode.

The frame has been equipped with four types of obstacle detecting sensors: 2D laser scanner, laser rangefinders, true-presence radar sensors and tactile sensors. Their position – shown in Fig. 1b – has been settled in the way that sensors cover the whole area around the robot and the main information concerns the area in front of IBIS.

3. Semiautonomy Method

The robot is driven by six direct-drive motors, three on each side and the robot is set on a track based. In our approach, we hybridize two different approaches: Braitenberg algorithm [8] (subbehavior 0) and modified Vector Field Histogram [9] (subbehavior 1) that lead to two omitting obstacles subbehaviors. Two following subbehaviors are azimuth setting (subbehavior 3) and linear velocity set-

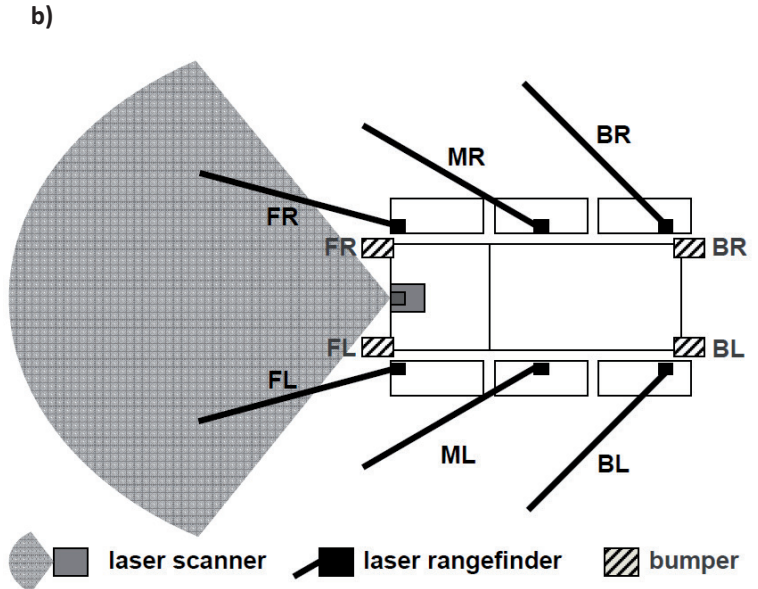


Fig. 1. IBIS mobile robot version for autonomy method development (a), distribution of the sensors (b)

ting (subbehavior 4). Therefore there are four partial control signals u_i that are combined as follows:

$$u(t) = \begin{bmatrix} u_R(t) \\ u_L(t) \end{bmatrix} = f_a \left(\sum_{l=0}^3 \mu_l \alpha_l \begin{bmatrix} u_{lR}(t) \\ u_{lL}(t) \end{bmatrix}, \begin{bmatrix} u_R(t-1) \\ u_L(t-1) \end{bmatrix} \right), \quad (1)$$

where:

l – subbehavior number,

t – iteration number, μ_l – subbehavior credibility factor,

α_l – subbehavior basic weighting factor,

$u_i = [u_{iR}(t), u_{iL}(t)]^T$ – partial control for particular subbehaviors,

f_a – acceleration limiting function,

$[u_R(t), u_L(t)]^T, [u_R(t-1), u_L(t-1)]^T$ – current and previous control vectors.

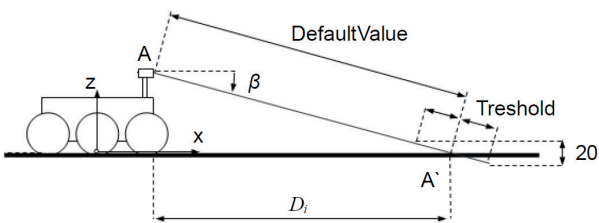


Fig. 2. Illustration of DefaultValue and Treshold

Current control vector is sent to the motor driver, which executes the robot movement. Vectors u_0, u_1, u_2 determine angular velocity partial control of the robot whilst u_3 corresponds to its linear velocity. This is because two subbehaviours: Braitenberg subbehavior, VFH subbehavior and azimuth setting subbehavior are calculated in a way that $u_{iL} = -u_{iR}$ by their nature (see (4),(11),(12)) and azimuth setting output is angular velocity proportionate to current and desired azimuth.

Classic Braitenberg algorithm is a reactive approach where direct data from sensors is multiplied

by weighting factors that produces direct control for the robot. Six laser rangefinders are placed on the top of the robot's frame, therefore Braitenberg matrix is 2x6. Braitenberg control signals are calculated as follows (% is modulo operation):

$$u_0(t) = \begin{bmatrix} u_{0R}(t) \\ u_{0L}(t) \end{bmatrix} = \begin{bmatrix} W_{1,1} \dots W_{1,6} \\ W_{2,1} \dots W_{2,6} \end{bmatrix} [s_{RF,1} \dots s_{RF,6}]^T, \quad (2)$$

$$s_{RF,i} = 1 - \left(|D_{idef} - D_i| \% D_{idef} \right) / D_{idef}, \quad (3)$$

where:

i – laser rangefinder's number,

$s_{RF,i}$ – laser rangefinder's reading,

D_p, D_{idef} – a length of segment of vertical projection of segment $|AA'|$ (Fig. 2) for current robot's position and if a robot is placed on flat surface,

$W_{w,i}$ – Braitenberg weighting factor that have following properties:

$$\forall i = 1 \dots 6, W_{1,i} = -W_{2,i}$$

and

$$\forall i = 1 \dots 3, \forall w = 1, 2, W_{w,i} = -W_{w,7-i}, \quad (4)$$

where: w – wheel's index, equals 1 for right and 2 for left wheel.

These properties lead to a fact that if we have m (m is even number) sensors placed symmetrically along main movement axis, we need to define only $m/2$ weighting factors W .

The s_i values take into consideration both convex and concave obstacles (refer to absolute value of D_{idef} and D_i subtraction in (3)) and the D_i can exceed D_{idef} several times leading to $s_{RF,i} > 1$ so that modulo operator has to be used and concave 0. For indoor conditions, s_i can be simplified to:

$$s_i = 1 - D_i / D_{idef}. \quad (5)$$

A scanner mounted in the front part of the robot is treated as 101 laser beams with angular resolution of 1° . As in the case of laser rangefinders D_i projection values are calculated for each beam leading to angular histogram of form:

$$\mathbf{H}' = [s'_{s,0} \dots s'_{s,100}]$$

$$s'_{s,i} = 1 - \left(\left| D_{idef} - (D_i - W_{safe}) \right| \% D_{idef} \right) / D_{idef}, \quad (6)$$

where $s'_{s,i}$ is a laser scanner's reading and \mathbf{a} angle width vector is then created:

$$\begin{aligned} \mathbf{a} &= [a_0 \dots a_{100}], a_i \\ &= \text{atan} \left\{ W_{safe} / \left(1 - \frac{\left| D_{idef} - (D_i - W_{safe}) \right| \% D_{idef}}{D_{idef}} \right) \right\}, \quad (7) \end{aligned}$$

where W_{safe} is a radius of safety region around the robot. Elements of \mathbf{a} vector correspond directly with elements of \mathbf{H}' vector. The \mathbf{a} vector's elements correspond with the distance from each obstacle placed on i 'th scanner beam in meaning of angle. It can be seen as dilation or "fattening" of obstacles so that robot can be treated as a point robot. $s_{s,i}$ values are calculated in following steps:

Step 1. A histogram vector \mathbf{H}' is then widened according to robot's dimensions W_{safe} as following to create true angular histogram \mathbf{H} :

$$\begin{aligned} \mathbf{H} &= [s_{s,0} \dots s_{s,100}], s_{s,i} = \max_{i=0..100} (S_i), S_i : s'_{s,j} \in S_i \\ &\Leftrightarrow (j - a_j \leq i) \vee (j + a_j \geq i). \quad (8) \end{aligned}$$

Vector \mathbf{H} is produced based on vectors \mathbf{H}' (containing distances over angle) and \mathbf{a} vector representing angle range for particular angles.

Step 2. Vector \mathbf{H} is then binarised with D_{th} threshold (i.e. where obstacles are further then $D_{idef} - D_{th}$) to vector $\mathbf{B}_H = [b_0, \dots, b_{100}]$, and openings (valleys) are found (see Fig. 3). The openings (valleys) are consecutive substrings of zeros in the binrised \mathbf{B}_H vector of which set is describes as follows:

$$\begin{aligned} \mathbf{P} &= \{ [p_{b0}, p_{e0}], [p_{b1}, p_{e1}], \dots, [p_{bn}, p_{en}] \}, \\ \forall p_{bj} &= i1 : (b_{i1} = 1 \wedge b_{i1+1} = 0), \\ \forall p_{ej} &= i2 = \min(K), K : (b_{i2} = 0 \wedge b_{i2+1} = 1 \wedge i2 > i1), \quad (9) \end{aligned}$$

where p_{bj}, p_{ej} are begin and end points of valley.

Step 3 $[p_{bk}, p_{ek}]$ - valley number k is $(p_{ek} - p_{bk}) / 2 - A$, then chosen for which the smallest of all other valleys is i.e. a valley closest to the goal point in meaning of angle is chosen. A is a desired bearing value calculated as follows:

$$A = \text{atan} \left(\frac{\sin(L_2 - L_1) \cos(B_2)}{\cos(B_1) \sin(B_2) - \sin(B_1) \cos(B_2) \cos(L_2 - L_1)} \right), \quad (10)$$

where (B_1, L_1) and (B_2, L_2) are current robot's position and the destination point in WGS-84 coordinate system.

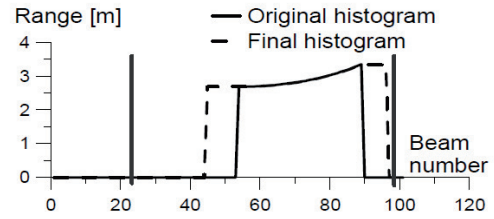


Fig. 3. Illustration of VFH method

Step 4. For this best opening (valley), next control signal is equal to:

$$\mathbf{u}_1(t) = \begin{bmatrix} u_{1R}(t) \\ u_{1L}(t) \end{bmatrix} = \begin{bmatrix} \beta_1 \left(\frac{p_{ek} - p_{bk}}{2} \right), -\beta_1 \left(\frac{p_{ek} - p_{bk}}{2} \right) \end{bmatrix}^T, \quad (11)$$

$\beta_1 = 0.02$ because control signals has to be of range $<0,1>$ and maximum number of consecutive zeros is 100 as this correspond with no obstacle situation.

A goal orientation control signal is controlled by the difference of current azimuth and desired bearing:

$$\mathbf{u}_2(t) = \begin{bmatrix} u_{2R}(t) \\ u_{2L}(t) \end{bmatrix} = \begin{bmatrix} -\beta_2 (A_c - A) \\ \beta_2 (A_c - A) \end{bmatrix}, \quad (12)$$

where A_c it is current azimuth value. $\beta_2 = 1/180^\circ$ because maximum deviation from desired azimuth is 180° and control are normalized to 1.

The linear velocity control is calculated as follows:

$$\mathbf{u}_3(t) = \begin{bmatrix} 1 - u_{3\max}(t) \\ 1 - u_{3\max}(t) \end{bmatrix}, u_{3\max} = \max \left(\sum_{i=0}^2 u_{iR}(t), \sum_{i=0}^2 u_{iL}(t) \right). \quad (13)$$

And is equal to maximum of all previous control signals. This means that if a robot is close to obstacles it decreases its linear velocity as $u_{3\max}$ has high value because robot has to maneuver amongst obstacles. On the other hand if no obstacle is sensed $u_{3\max}$ equals zero and therefore robot's linear steering maximum value.

Sensors' failures are represented as vector $\mathbf{F} = [\varphi_1, \dots, \varphi_6, \varphi_s]$, where $\varphi_{1..6}$ refer to rangefinders' failures (value 1 stands for failure, 0 - sensor is working properly), φ_s refers to scanner's failure. They are an important factor for construction site applications as knowing that the device has malfunction increases the safety. Vector \mathbf{F} contains information whether particular sensor is or is not providing reliable data. Based on that information a credibility factor for each subbehaviour is calculated. The value of which corresponds to how do we treat the data from the sensor. Four basic situation for sensor's failure can be considered:

1. No response from the sensor. The measurement system has been broken switched off or connections cables failed
2. Too low measurements values frequency. This situ-

ation can cause by cover of optical subsystem of the sensor by dirt or splash. This situation can be often on construction sites where transporting robot maneuvers through the dust or dirt that is especially important case for construction site.

3. Too high frequency of measurements values frequency. This situation can be caused by not proper mounting of the sensor or electronic part of measuring system failure.
4. Internal sensor failure. In this situation, the sensor returns an internal malfunction code corresponding to overheat, not accomplished selftest or others.

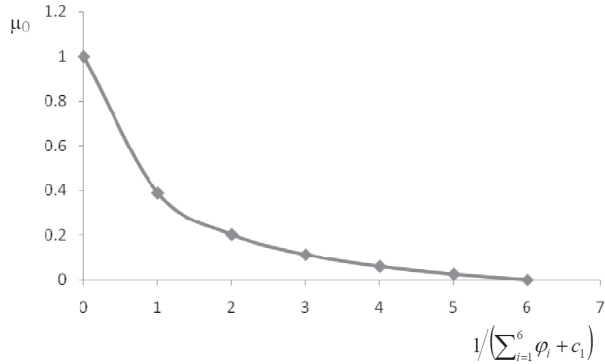


Fig. 4. Illustration μ_0 credibility factor change for changing the number of laser rangefinder's failures

If all sensors work properly, all subbehaviors credibility factors are equal to 1 and they are decreased if F becomes non-zero. The value of credibility factor is changed as follows (see also Fig. 4):

$$\mu_0 = 1 / \left(\sum_{i=1}^6 \varphi_i + c_1 \right) + c_2, \tag{14}$$

where: c_1, c_2 have particular values, so that no sensor's failure lead to $\mu_0=1$ and all rangefinders have failure equal $\mu_0=0$, i.e. $c_1 = -3 + \sqrt{15}, c_2 = -1 / (3 + \sqrt{15})$. The μ_0

credibility factor change function is set in a way that first failures that occur decreases μ_0 significantly and this decrease is gradually smaller for next failures.

$$\mu_1 = \begin{cases} 0 & \text{if } \varphi_S = 1 \\ 1 & \text{if } \varphi_S = 0 \end{cases}, \mu_3 = \mu_2 = \begin{cases} 0 & \text{if } \varphi_S \prod_{i=1}^6 \varphi_i = 1 \\ 1 & \text{if } \varphi_S \prod_{i=1}^6 \varphi_i = 0 \end{cases}, \tag{15}$$

where: f_a function limits the maximum acceleration. f_a is necessary to produce smooth movement and prevent robot to execute oscillatory movement when passing between obstacles. Limiting function has a following form:

$$f_a(u(t), u(t-1)) = \begin{cases} u_W(t) \cdot [1, 1]^T & \text{for } \frac{u_W(t) - u_W(t-1)}{\Delta t} \leq a_{th} \\ (a_{th} \Delta t + u_W(t-1)) \cdot [1, 1]^T & \text{for others} \end{cases}, \tag{16}$$

where: $W=\{L,R\}$, Δt - step time, a_{th} - threshold acceleration corresponding to 3 [m/s²] when accelerating, and 6 [m/s²] when decelerating. Weighting factors in our algorithm are not constant values because different subbehaviors need to have different importance in particular situations e.g. when no obstacle occurs the second and the third subbehavior need to have significant dominance over omitting obstacles subbehaviors as there is nothing to omit. For the simplicity we have decided to use linear segment function to calculate α_i which value depends on the distance to the nearest obstacle. The values of α_2 and α_3 are increasing while the distance to obstacle is expanding. On the contrary the values of α_0 and α_1 which are decreasing - see Fig. 5 for reference.

Two situations are distinguished basing on the projection of D_1 (see Fig. 2) on Y-axis (see Fig. 1b). This projection represents a lateral distance to obstacle when robot performs its movement. If the projection produces distance smaller than W_{safe} more emphasis is put on omitting obstacles subbehaviour 0 and 1

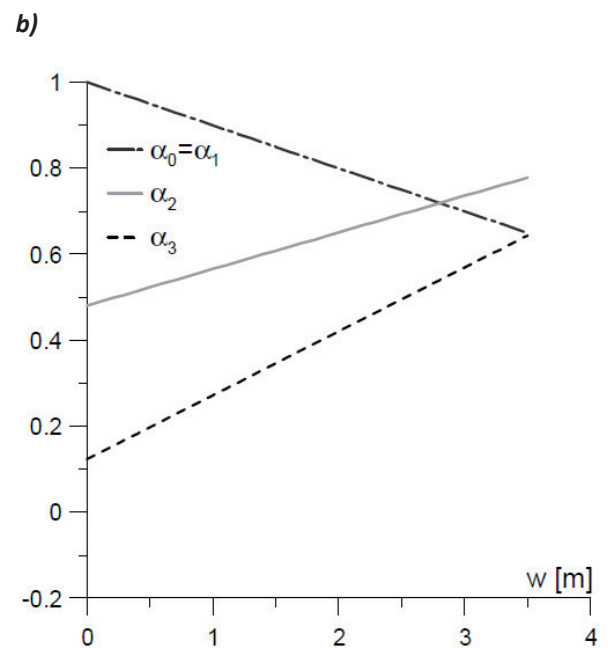
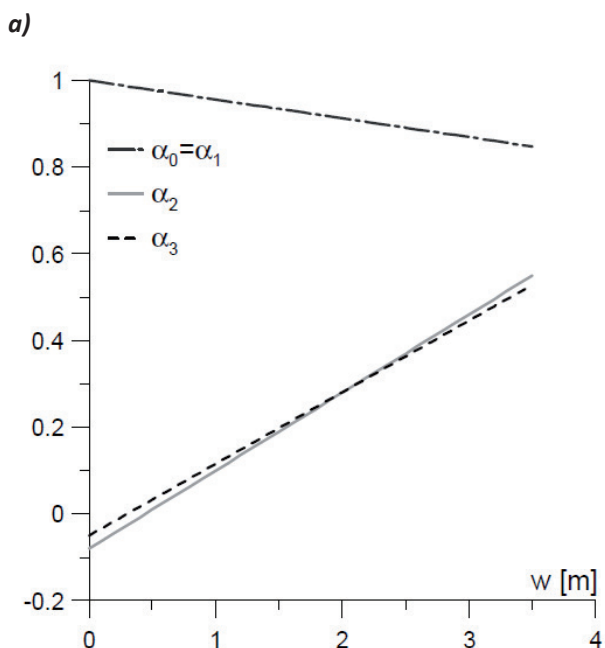


Fig. 5. Illustration of α_i functions a) the nearest obstacle is closer than W_{safe} distance, b) the nearest obstacle is further than W_{safe} distance

(Fig. 5a). On the other hand (Fig. 5b), going towards goal subbehaviours 2 and 3 are preferred. Basically situation from Fig. 5b could be neglected because lateral distance of W_{safe} means that robot will pass by the obstacle for safety reasons, which are a clue on construction site we would like to maximize the distance to obstacle whilst omitting it and therefore α_0 and α_1 are non-zero values. If obstacle is further than 3 m – i.e. approximately $4W_{safe\ going}$ towards the goal subbehaviours start to dominate over omitting obstacles.

4. Simulation Research

A dedicated software environment has been created for autonomy method testing and evaluating. It is based on Matlab/Simulink package. The software responsible for surroundings simulation and virtual sensors indications has been separated from semiautonomy method.

This work presents the simulation research results, in which obstacles were placed on the robot's route to the target. The mobile robot's semiautonomy algorithm has been evaluated in terms of sensitivity to the failure of selected environmental sensors.

It is important to perform by the robot "go to the target omitting obstacles" behavior possibly in the shortest time, with the shortest route and with constant velocity (which results from energy consumption). Therefore in order to receive a conclusive assessment of the results, the following quality rates have been proposed:

a) the sum of squares of the robot's distance to the target

$$E = \sum_{i=0}^N e^2(i \cdot \Delta t), \quad (17)$$

where:

i – iteration number,

e – the robot's distance to the target,

N – the number of iterations till the robot reaches its target or the simulation ends before the target is reached,

b) standard deviation of the robot's speed

$$S = \sqrt{\sum_{i=0}^N (v_R(i \cdot \Delta t) - v_M)^2 / (N - 1)}, \quad (18)$$

where:

v_R – robot's linear speed,

v_M – robot's average speed,

c) length of the route from the starting position to the target

$$s = \sum_{i=0}^N v(i \cdot \Delta t) \Delta t, \quad (19)$$

d) the time T it takes for the robot to reach the target, assuming the target is achieved for $e \leq 0.5$ [m],

e) robot's medium speed v_M (within a time from 0 to T).

It should be noted that quality rates (a) – (d) should be minimized whilst (e) should be maximized. In the case the robot cannot reach the target within the assumed time $T_{max} = 100$ [s], the quality rates s and T reach the value of $+\infty$. Remaining rates reach the values calculated for T_{max} .

Simulation 1 – all environmental sensors are working properly

The first presented simulation has been performed with all of the robot's sensors working properly. The results of this simulation are presented in Figs. 6–7 and Table 1. They are used as reference for the next simulations, in which influence of selected sensors' failure is presented.

In Fig. 6 and Table 1 motion trajectories of the robot and quality rates for all simulations. Fig. 7 illustrates robot's linear speeds v_R , its medium value v_M (a) as well as robot's distance e and angle $d\gamma$ to the target (b).

The relation (weighting) between robot's subbehaviors remains constant during the robot's movement. It changes only in case of detection environmental sensors' failure.

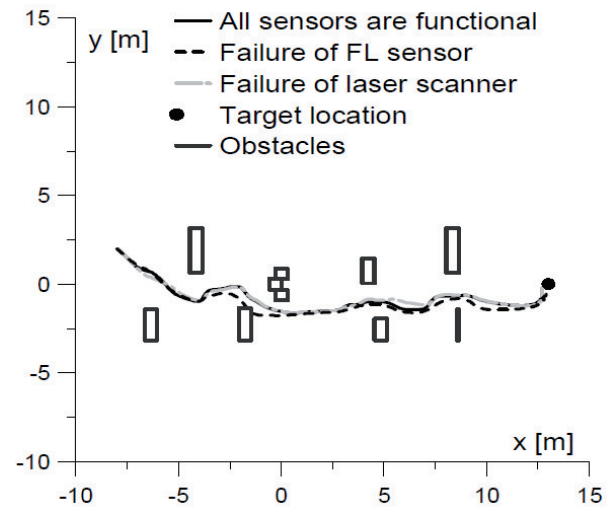


Fig. 6. Motion trajectories of robot for Simulations 1–3

Table 1. Values of quality rates for Simulations 1–3

	Sim1	Sim2	Sim3
E	7317	7681	8538
S	0.131	0.126	0.074
s	23.8	23.6	23.5
T	47.0	50.3	54.6
v_M	0.51	0.47	0.43

The relation (weighting) between robot's subbehaviors remains constant during the robot's movement. It changes only in case of detection environmental sensors' failure. Performed numerous simulations show that elaborated method allows omitting obstacles and getting to defined destination point. Robot follows a path with variable speed: the speed is decreased, as the robot gets closer to obstacle (Fig. 7a). During the whole movement, the distance to the target is decreasing (Fig. 7b).

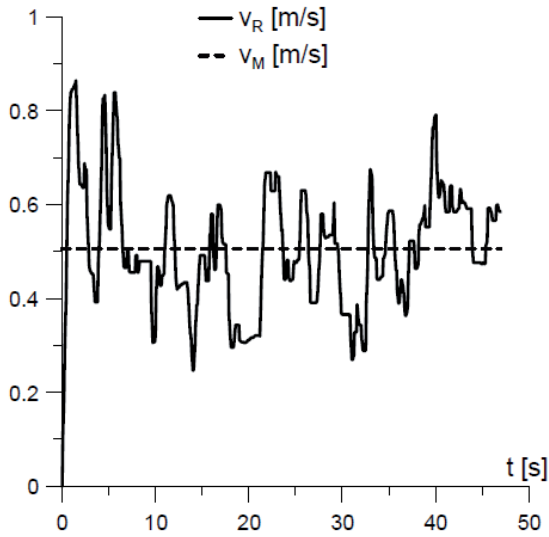


Fig. 7. Results of Simulation 1

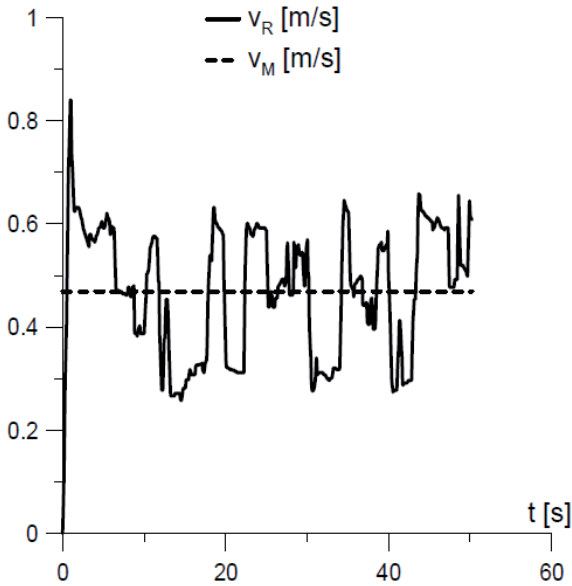
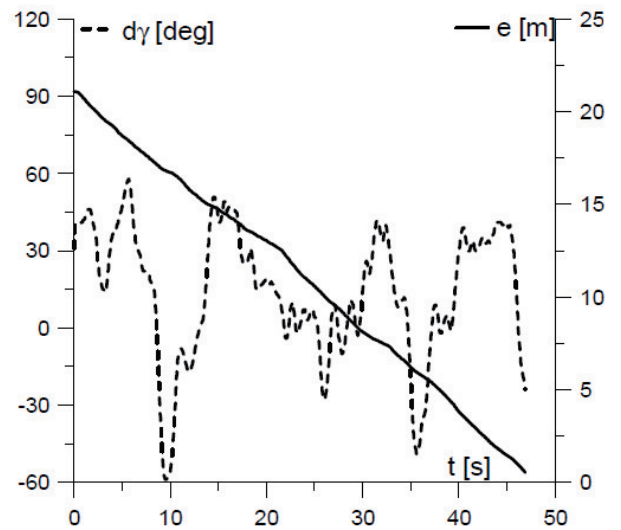


Fig. 8. Results of Simulation 2

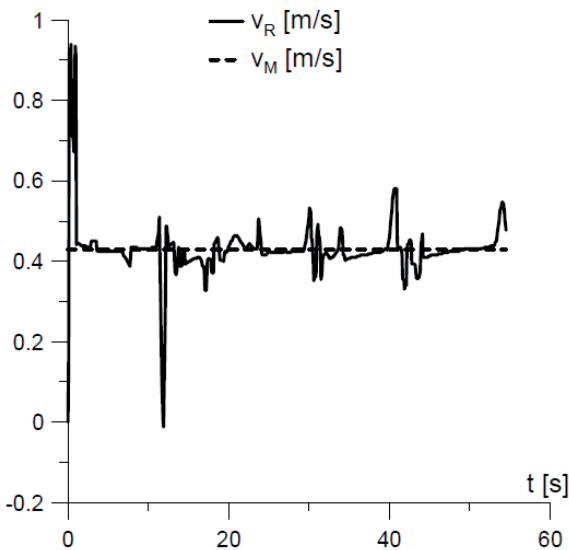
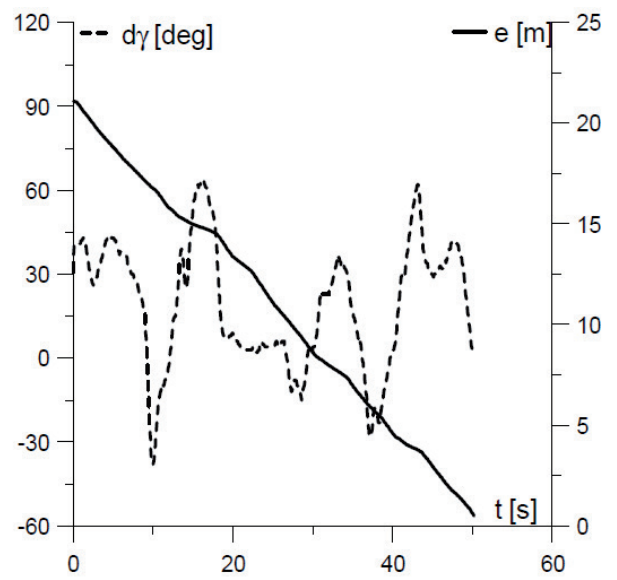
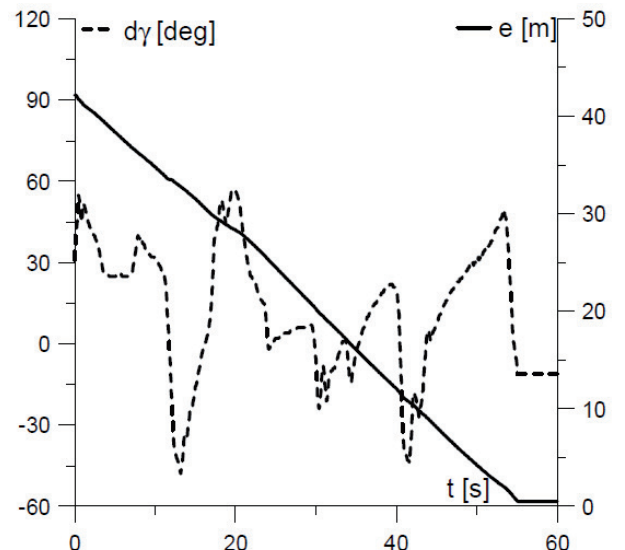


Fig. 9. Results of Simulation 3



Simulation 2 – failure of the laser rangefinder marked as FL

The second simulation has been performed in the case of a failure of the laser rangefinder FL (see Fig. 1b). The results of this simulation have been illustrated in Fig. 6, Fig. 8 and Table 1. After the laser rangefinder’s failure has been detected by the semi-autonomy algorithm the weighting was modified for the subbehavior related to the modified Braitenberg algorithm. This weighting has been reduced after failure detection.

Comparing the results of simulations 1 and 2, it can be concluded that in the case of a failure of the FL laser rangefinder, the robot achieved its target after longer time. Quality rates E , T and v_M achieved worst values whilst the value of S and s are slightly better.

Simulation 3 – failure of the laser scanner

In the last, third simulation, the failure of laser scanner was examined. The results of the simulation are shown in Fig. 6, Fig. 9 and Table 1.

Presented results indicate that semiautonomy algorithm has high resistance to selected sensors’ damage. Robot with damaged sensors moves to the target relatively fast, which is related to the fact that it has less information about obstacles. Therefore it moves more freely but less safely.

Within this work simulation research has been also conducted assuming the damage of remaining sensors in the case of a single sensor failure. Based on all performed simulations, it can be concluded that, in most cases, the developed semiautonomy algorithm is capable of dealing with the damage of a single sensor.

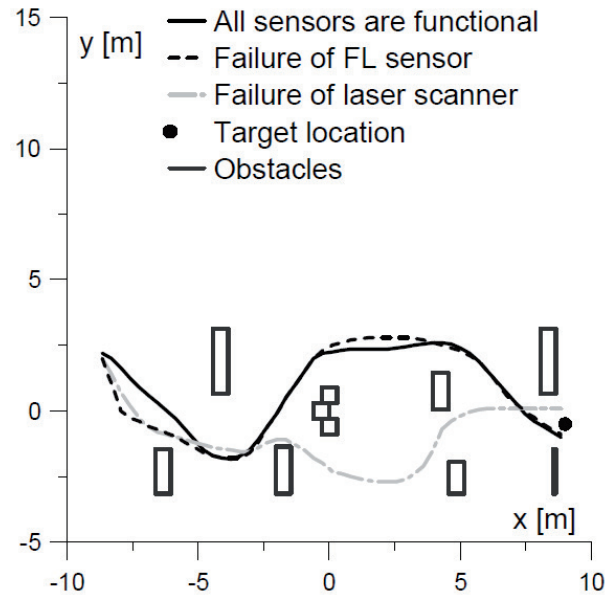


Fig. 10. Motion trajectory of robot for Experiments 1 – 3

Table 2. Values of quality rates for Experiments 1 – 3

	Exp1	Exp2	Exp3
E	9851	11626	37372
S	0.19	0.12	0.09
s	22.5	22.8	20.8
T	100	101	367
v_M	0.22	0.23	0.06

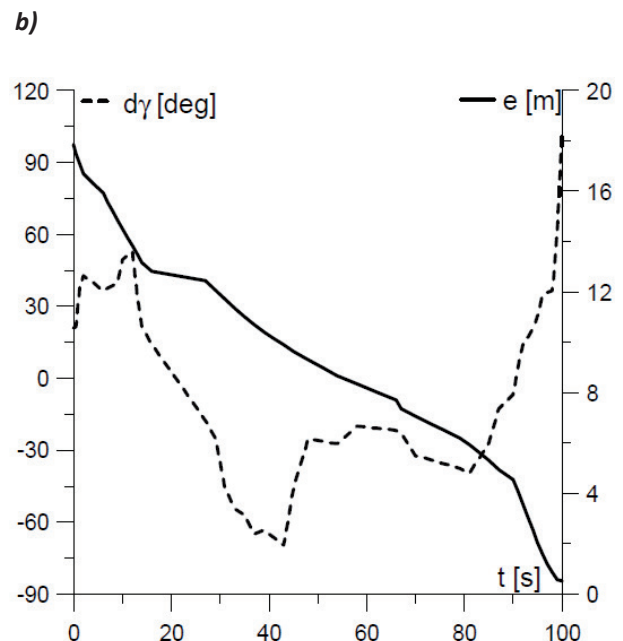
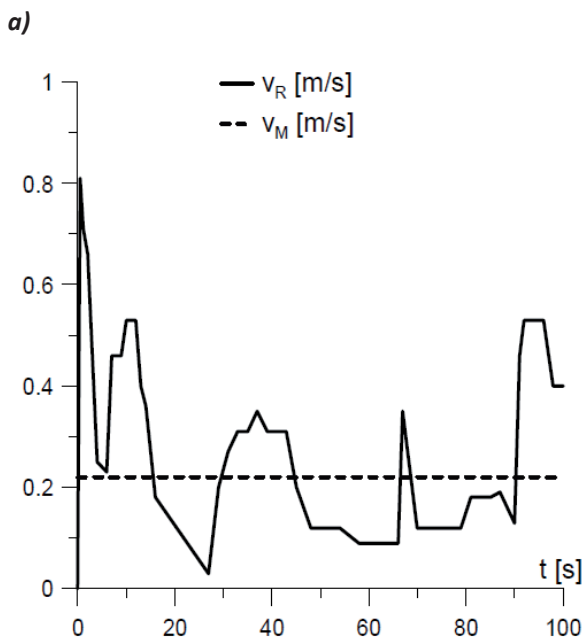


Fig. 11. Results of Experiment 1

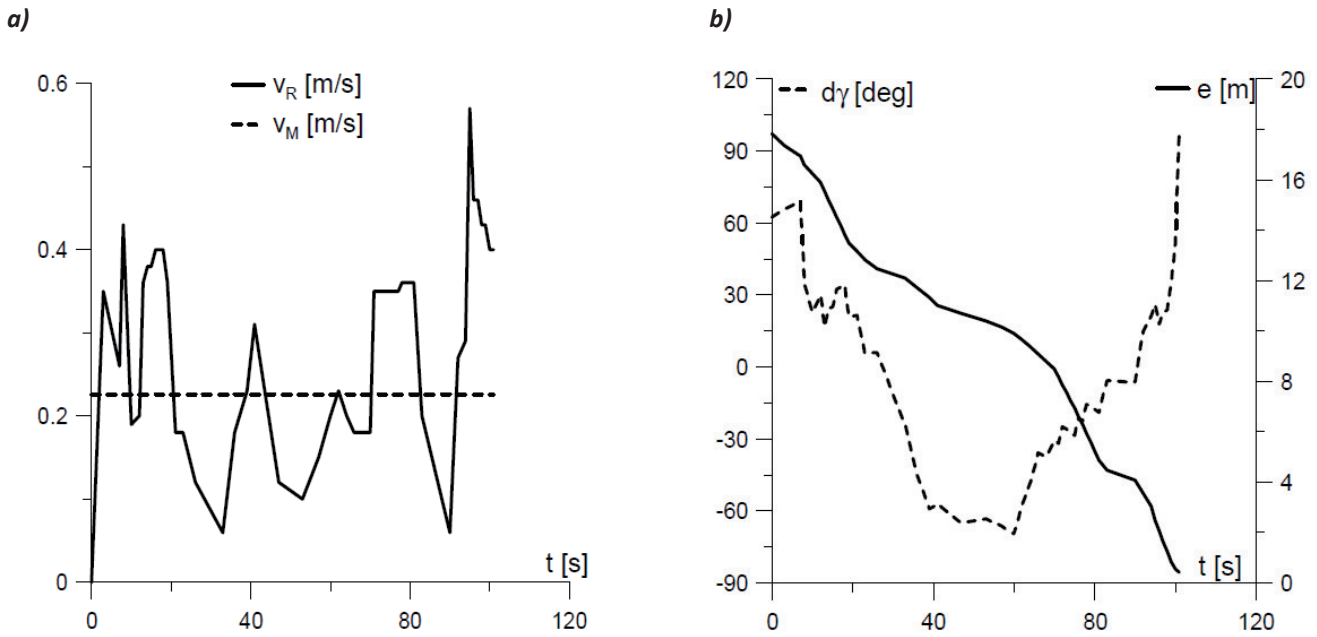


Fig. 12. Results of Experiment 2

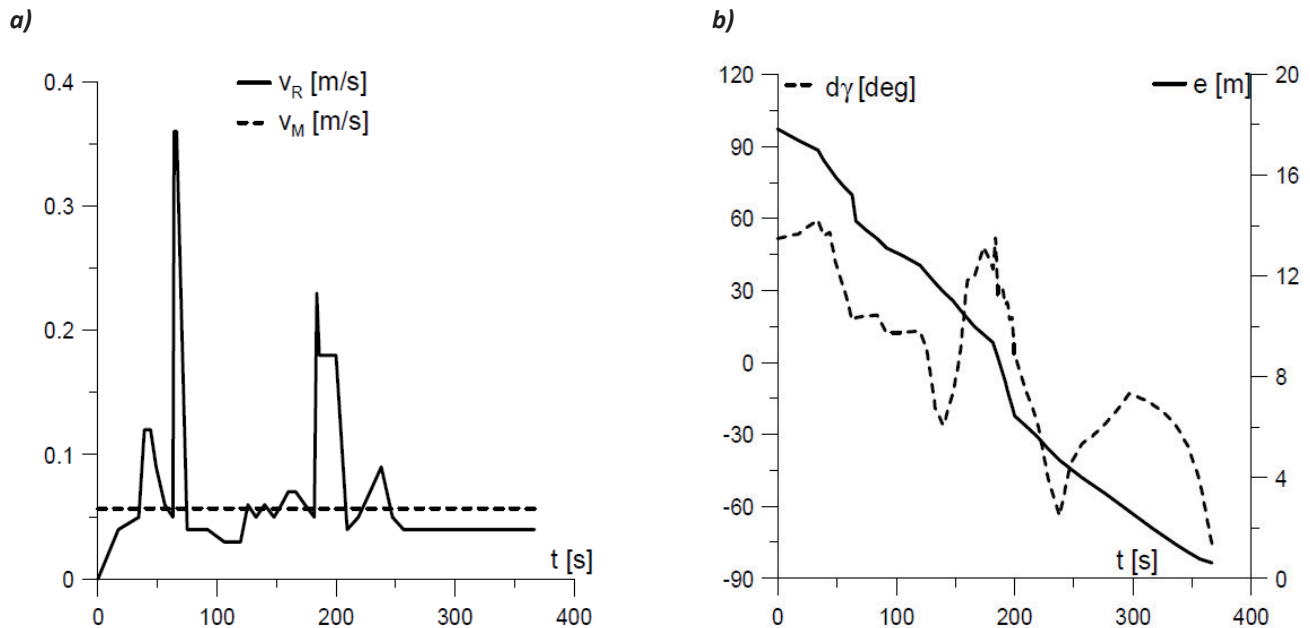


Fig. 13. Results of Experiment 3

5. Experimental Tests

Taking into account simulation conclusions and safety reasons the robot's speed in experimental tests has been reduced (especially in case of laser scanner failure). The experimental tests were conducted in similar environment as in simulation research.

Experiment 1 – all environmental sensors are working properly

The first experimental test was performed for the case of all environmental sensors working properly. The results of this experiment are illustrated in Figs. 10–11 and Table 2. The robot reached a destination point in about 100 s. During movement, robot decreases distance to the target (Fig. 11b) and moves

with variable speed, i.e. reduces it being closer to obstacle (Fig. 11a).

Experiment 2 – failure of the laser rangefinder marked as FL

In the second experimental test laser rangefinder FL failed. As a result (see Fig. 10, Fig. 11 and Table 2) robot reached the destination point in 101 [s]. All quality rates except of S were slightly worst in comparison to the first experiment.

Experiment 3 – failure of the laser scanner

In the last, third experiment the failure of laser scanner occurred. As one can notice (see Fig. 10, Fig. 13 and Table 2), the robot moves to the target by dif-

ferent route and with much less speed in comparison to previous two tests. Therefore quality rates E , T and v_M are the worst and S and s are the best from all experiments. Robot moves slowly but with relatively constant speed and reached the target accidentally using shorter route.

6. Summary

The results of experimental research have been shown in this paper. The developed semiautonomy algorithm and its sensitivity to sensors' failure were described. The videos recorded during experiments and the robot's animations can be found under address [10].

Further research will focus on simulating and testing the failure of large number of sensors and its influence on robot's movement.

ACKNOWLEDGEMENT

The paper is a result of PROTEUS Project (POIG.01.01.02-00-014/08) and is co-financed by The European Union from The European Regional Development Fund under the Operation Programme Innovative Economy, 2007–2013.

AUTHORS

Piotr Bigaj*, **Jakub Bartoszek**, **Maciej Trojnacki**
– Industrial Research Institute for Automation and Measurements (PIAP), Warsaw, Poland.

* Corresponding author (pbigaj@piap.pl)

REFERENCES

- [1] Wołoszczuk A., Andrzejczak, M., Szyrkarczyk, P., "Architecture of mobile robotics platform planned for intelligent robotic porter system – IRPS project", *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol.1, no. 3, 2007, 59–63.
- [2] Bouibed K., Aitouche A., Bayart M., "Sensor fault detection by sliding mode observer applied to an autonomous vehicle". In: *2009 International Conference on Advances in Computational Tools for Engineering Applications*, 621–626. DOI: <http://dx.doi.org/10.1109/ACTEA.2009.5227853>.
- [3] Rae G. J. S., Dunn S. E., "On-Line detection for AUV", *IEEE Symp. Autonomous Underwater Vehicle Technology*, 1994, 383–392.
- [4] Visinsky M. L., Cavallaro J. R., Walker, J. D., "Expert System Framework for Fault Detection and Fault Tolerance in Robotics", *Computers & Electrical Engineering*, vol. 20, no. 5, 1994, 421–435.
- [5] Soika M., "A sensor failure detection framework for autonomous mobile robots". In: *Proceedings of the 1997 IEEE/RSJ International Conference*

on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. (IROS '97), vol.3, 1997, 1735–1740. DOI: <http://dx.doi.org/10.1109/IROS.1997.656594>.

- [6] Healey A. J., "Toward an automatic health monitor for autonomous underwater vehicles using parameter identification", *Amer. Control Conf.*, 1993, 585–589.
- [7] http://www.antiterrorism.eu/combat_robot.php
- [8] Braitenberg V., *Vehicles: Experiments in synthetic psychology*, Cambridge, MA: MIT Press, 1984.
- [9] Ulrich I., Borenstein J., "VFH+: reliable obstacle avoidance for fast mobile robots", *1998 IEEE International Conference on Robotics and Automation*, 1998, 1572–1577.
- [10] <http://www.youtube.com/user/osmpiap>