SENSITIVITY ANALYSIS OF SEMIAUTONOMY ALGORITHM OF MOBILE ROBOT TO ENVIRONMENTAL SENSORS' FAILURE - SIMULATION RESEARCH AND EXPERIMENTAL TESTS

Piotr Bigaj*, Jakub Bartoszek, and Maciej Trojnacki

Industrial Research Institute for Automation and Measurements (PIAP), Warsaw, Poland
** Corresponding author (pbigaj@piap.pl)*

ABSTRACT: This work is concerned on sensitivity analysis of semiautonomy algorithm of mobile robot to environmental sensors' failures. The 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 takes also 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 were 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, Simulation Research, Experimental Tests, 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]. 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 (Fig. 1a) 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 [7].

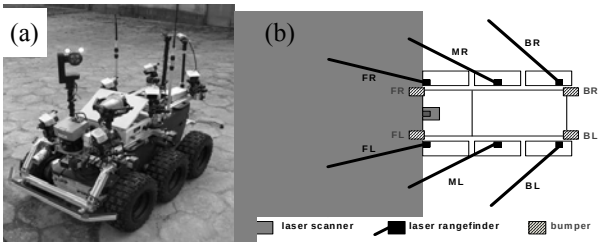

Fig. 1. IBIS mobile robot version for autonomy method development (a) [7], distribution of the sensors (b)

For research activity a new version of IBIS robot without manipulator has been designed (Fig.1b). The IBIS robot can perform in one of two modes: teleportation 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.2 – 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. 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 setting (subbehavior 4). Therefore there are four partial control signals that are combined as follows:

$$\mathbf{v}(t)=\begin{bmatrix} v_R(t) \\ v_L(t) \end{bmatrix}=f_a\left(\sum_{l=0}^{3}\mu_l\alpha_l\begin{bmatrix} v_{lR}(t) \\ v_{lL}(t) \end{bmatrix},\begin{bmatrix} v_R(t-1) \\ v_L(t-1) \end{bmatrix}\right), \qquad (1)$$

where: l - subbehavior number, t - iteration number, μ_l - subbehavior credibility factor, α_l - subbehavior basic weighting factor, $\mathbf{v}_l = [v_{lL}, v_{lR}]^T$ - partial control for particular subbehaviors, f_a - acceleration limiting function, $[v_R(t), v_L(t)]^T$, $[v_R(t-1), v_L(t-1)]^T$ - current and previous control vectors.

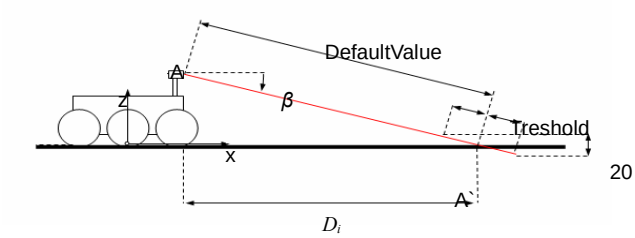

Fig. 3. Illustration of DefaultValue and Treshold

Current control vector is sent to the motor driver which executes the robot movement. Vectors $\mathbf{v}_0$, $\mathbf{v}_1$, $\mathbf{v}_2$ determine angular velocity partial control of the robot whilst $\mathbf{v}_3$ corresponds to its linear velocity. This is because all three: Braitenberg subbehavior, VFH subbehavior and azimuth setting subbehavior are calculated in a way that $v_{lL}=-v_{lR}$ by their nature. 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):

$$\mathbf{v}_0(t)=\begin{bmatrix} v_{0R}(t) \\ v_{0L}(t) \end{bmatrix}=\begin{bmatrix} W_{1,1}\ldots W_{1,6} \\ W_{2,1}\ldots W_{2,6} \end{bmatrix}\left[s_{RF,1}\ldots s_{RF,6}\right]^T \qquad (2)$$

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

where: i - laser rangefinder's number, $s_{RF,i}$ - laser rangefinder's reading, D_i, D_{idef} - a length of segment of vertical projection of segment |AA'| (Fig. 3) if a robot is placed on flat surface, $W_{w,i}$ - Braitenberg weighting factor that have following properties:

$$\forall i=1..6, W_{1,i}=-W_{2,i} \text{ and } \forall i=1..3, \forall w=1,2, W_{w,i}=-W_{w,7-i} \qquad (4)$$

where: w - wheel's index, equals 1 for right and 2 for left wheel. The s_i values take into consideration both convex and concave obstacles. In indoor conditions s_i can be simplified to:

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

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 $m/2$ weighting factors W.

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:

$$\mathbf{H}'=[s'_{s,0}\ldots s'_{s,100}],\ s'_{s,i}=1-\left(\left|D_{idef}-(D_i-W_{safe})\right|\%D_{idef}\right)/D_{idef} \qquad (6)$$

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

$$\mathbf{a}=[a_0..a_{100}],\ a_i=\operatorname{atan}\left\{W_{safe}\Big/\left(1-\frac{|D_{idef}-(D_i-W_{safe})|\%D_{idef}}{D_{idef}}\right)\right\} \quad (7)$$

where W_{safe} is a radius of safety region around the robot. Elements of **a** vector correspond directly with elements of $\mathbf{H}'$ vector. $s_{s,i}$ values are calculated in following steps:

a) A histogram vector $\mathbf{H}'$ is then widened according to robot's dimensions as following to create true histogram **H**:

$$\mathbf{H}=[s_{s,0}\ldots s_{s,100}],\ s_{s,i}=\max_{i=0..100}(\mathbf{S}_i),\ \mathbf{S}_i: s'_{s,j}\in\mathbf{S}_i$$

$$\Leftrightarrow (j-a_j\le i)\vee(j+a_j\ge i) \quad (8)$$

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

b) Vector **H** is then binarised with D_{th} threshold (i.e. where obstacles are further then D_{idef} - D_{th}) to vector $\mathbf{B}_H=[b_0,\ldots,b_{100}]$, and openings (valleys) are found (see Fig. 4). Then we establish a set of valleys:

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

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

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

$$A=\operatorname{atan2}(s_{L2-L1}c_{B2},c_{B1}s_{B2}-s_{B1}c_{B2}c_{L2-L1}) \quad (10)$$

where $(B1,L1)$ and $(B2,L2)$ are current robot's position and the destination point in WGS-84 coordinate system, s_x=sin(x), c_x=cos(x).

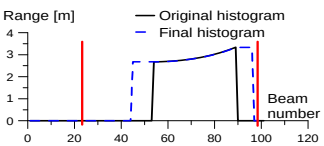

Fig. 4. Illustration of VFH method

For this value next control signals is equal to:

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

$\beta_1=0.01$ because control signals has to be of range <0,1>.

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

$$\mathbf{v}_2(t)=\begin{bmatrix}v_{2R}(t)\\v_{2L}(t)\end{bmatrix}=\begin{bmatrix}-\beta_2(A_C-A)\\\beta_2(A_C-A)\end{bmatrix} \quad (12)$$

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

The linear velocity control is calculated as follows:

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

Sensors' failures is a vector $\mathbf{F}=[\varphi_1,\ldots,\varphi_6,\varphi_s]$, $\varphi_{1..6}$ refer to rangefinders' failures (value 1 stands for failure, 0 - sensor is working properly), φ_s refers to scanner's failure. If all sensors work properly all subbehavior credibility factors are equal to 1. The value is changed as follows:

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

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

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

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:

$$f_a(\mathbf{v}(t),\mathbf{v}(t-1))=\begin{cases}\mathbf{v}(t) \text{ for } \dfrac{\mathbf{v}(t)-\mathbf{v}(t-1)}{\Delta t}\le[1,1]^T a_{th}\\[1,1]^T a_{th}\Delta t+\mathbf{v}(t-1) \text{ for others}\end{cases} \quad (16)$$

where: Δt - step time, a_{th} - threshold acceleration, equals 3 [m/s^2] when accelerating and 6 [m/s^2] when decelerating.

Weighting factors in our algorithm are not constant values. They are made so, because different subbehaviors need to

have different importance in particular situations e.g. when no obstacle occurs linear velocity control and goal orientation subbehavior need to have significant dominance over omitting obstacles subbehavior as there is nothing to omit. For the simplicity we have decided to use linear segment function to calculate $\boldsymbol{\alpha_i}$ which value depends on the distance to the nearest obstacle. They are shown in figures below.

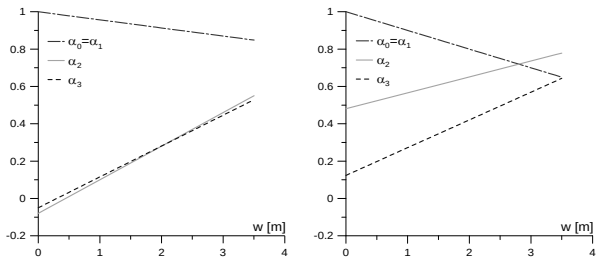

Fig. 5. Illustration of $\boldsymbol{\alpha_i}$ functions

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 robot preformed the complex „go to the target omitting obstacles" behavior. The mobile robot's semiautonomy algorithm has been evaluated in terms of sensitivity to the failure of selected environmental sensors.

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

$$E = \sum_{t=1}^{N} e_t^2 \Delta t \,, \tag{17}$$

where e_t – 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

$$S = \sqrt{\sum_{t=1}^{N} \left(v_t - v_M\right)^2 \Big/ \left(N-1\right)}, \tag{18}$$

where: v_t – robot's speed, v_M – robot's average speed,

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

$$s = \sum_{t=1}^{N} v_t \Delta t \,, \tag{19}$$

d) the time T it takes for the robot to reach the target, assuming the target is achieved for $e_t \leq 0.5$ [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 preformed with all of the robot's sensors working properly. The results of this simulation are presented in Fig. 6-7 and Table 1. They are used as reference for the next simulations, in which influence of selected sensors' failure is presented.

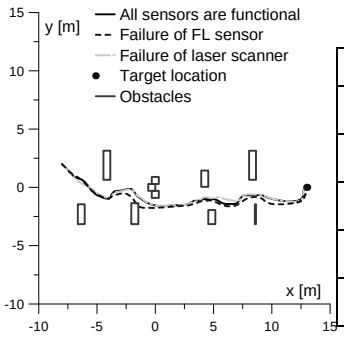

Fig. 6. Motion trajectories of robot for Simulation 1 - 3

	S1	S2	S3
E	7317	7681	7287
S	0.131	0.126	0.125
s	23.8	23.6	23.6
T	47.0	50.3	46.1
v_M	0.51	0.47	0.51

Table 1. Values of quality rates for Simulation 1 - 3

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. A performed simulation shown 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 (Fig7b).

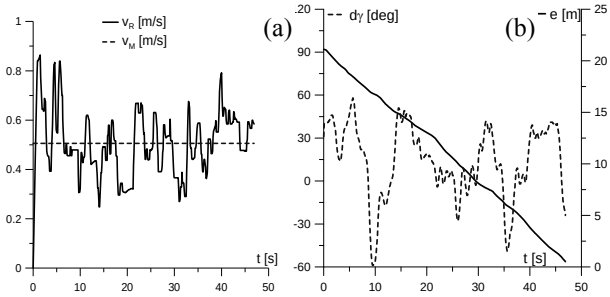

Fig. 7. Results of Simulation 1

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 semiautonomy algorithm the weighting was modified for the subbehavior related to the modified Braitenberg algorithm. This weighting have been reduced after failure detection.

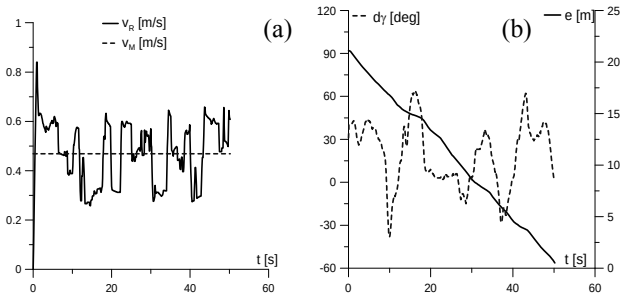

Fig. 8. Results of Simulation 2

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 a 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 also been 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.

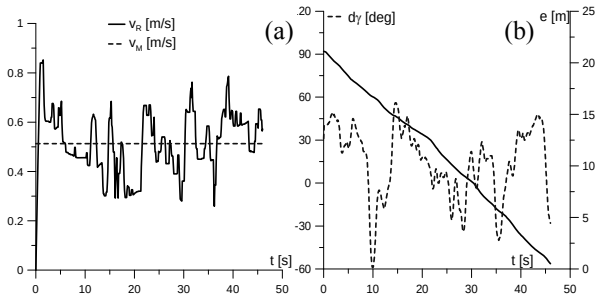

Fig. 9. Results of Simulation 3

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 were working properly. The results of this experiment are illustrated in Fig. 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. reducing it being closer to obstacle (Fig. 11a).

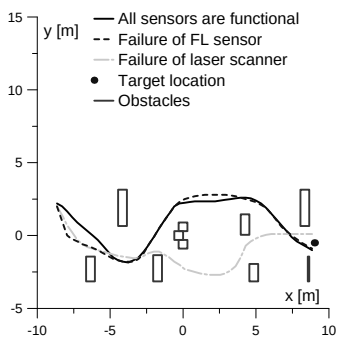

Fig. 10. Motion trajectory of

	E1	E2	E3
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

Table 2. Values of quality

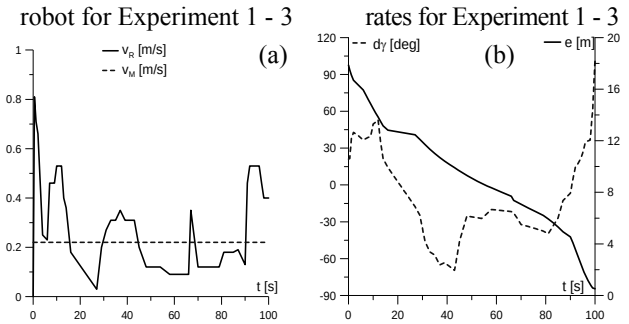

Fig. 11. Results of Experiment 1

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. 12 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.

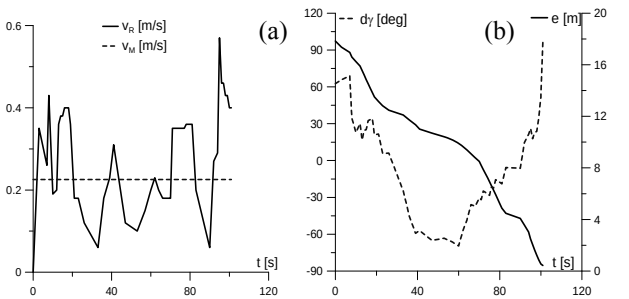

Fig. 12. Results of Experiment 2

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 different 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.

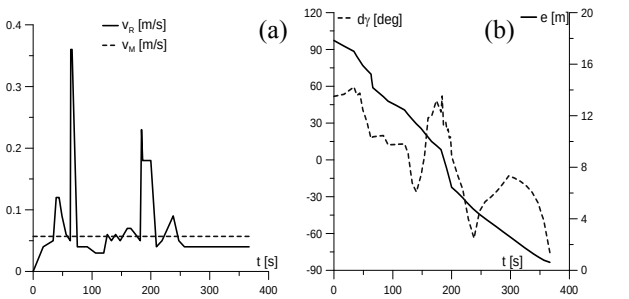

Fig. 13. Results of Experiment 3

6. SUMMARY

The results of experimental research have been shown in this paper. The developed semiautonomy algorithm and its sensitivity to sensors' failure was 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 to robot's movement.

REFERENCES

[1] Wołoszczuk, A., Andrzejczak, M., Szynkarczyk, P., "Architecture of mobile robotics platform planned for intelligent robotic porter system – IRPS project", *Journal of Automation, Mobile Robotics & Intelligent Systems*, Vol.1, pp.59-63, 2007.

[2] Bouibed, K., Aitouche, A., Bayart, M., "Sensor fault detection by sliding mode observer applied to an autonomous vehicle", *ACTEA '09*, pp. 621-626, 2009.

[3] Rae, G. J. S., Dunn, S. E., "On-Line detection for AUV", *IEEE Symp. Autonomous Underwater Vehicle Technology*, pp. 383-392, 1994.

[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(5), pp. 421-435, 1994.

[5] Soika, M., "A sensor failure detection framework for autonomous mobile robots", *IROS '97*, Vol.3, pp. 1735-1740, 1997.

[6] Healey, A. J., "Toward an automatic health monitor for autonomous underwater vehicles using parameter identification", *Amer. Control Conf.*, pp. 585-589, 1993.

[7] http://www.antiterrorism.eu/combat_robot.php

[8] Braitenberg, V. (1984). Vehicles: Experiments in synthetic psychology. Cambridge, MA: MIT Press.

[9] Ulrich, I., Borenstein, J., "VFH+: reliable obstacle avoidance for fast mobile robots", *Robotics and Automation, 1998 IEEE International Conference*, pp. 1572-1577, 1998.

[10] http://www.youtube.com/user/osmpiap