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

Piotr Bigaj*, Jakub Bartoszek, and Maciej Trojнакi

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].
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:

\[
v(t) = \begin{bmatrix} v_{L,R}^d(t) \\ v_{L,R}^i(t) \end{bmatrix} = \alpha \sum_{\mu \in \mathcal{M}, l \in \mathcal{L}} \begin{bmatrix} v_{L,R}^d(t) \\ v_{L,R}^i(t) \end{bmatrix}
\]

where: \( l \) - subbehavior number, \( t \) - iteration number, \( \mu \) - subbehavior credibility factor, \( \alpha \) - subbehavior basic weighting factor, \( v_i \) = \([v_{L,R},v_{L,R}]^T\) - partial control for particular subbehaviors, \( f_\alpha \) - acceleration limiting function, \( [v_{L,R}^d(t),v_{L,R}^i(t)]^T \) - current and previous control vectors.

Current control vector is sent to the motor driver which executes the robot movement. Vectors \( v_0, v_1, v_2 \) determine angular velocity partial control of the robot whilst \( v_3 \) corresponds to its linear velocity. This is because all three: Braitenberg subbehavior, VFH subbehavior and azimuth setting subbehavior are calculated in a way that \( v_{L,R}^d = -v_{L,R}^i \) 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):

\[
v_i(t) = \begin{bmatrix} v_{L_1,R_1}^d(t) \\ v_{L_1,R_1}^i(t) \end{bmatrix} = \begin{bmatrix} W_{1,1} & \ldots & W_{1,6} \\ W_{2,1} & \ldots & W_{2,6} \end{bmatrix} \begin{bmatrix} s_{RF1} \ldots s_{RF6} \end{bmatrix}
\]

\[
s_{RF,i} = 1 - \frac{D_{def} - D_i v_{def} / D_{def}}{D_{def}}
\]

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

\[
\forall i = 1..6, W_{ij} = -W_{ij} \quad \text{and} \quad \forall i = 1..3, \forall W = 1..2, W_{uj} = -W_{uj}\]

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:

\[
s_i = 1 - \frac{D_i}{D_{def}}
\]

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:

\[
H = [s_0^' \ldots s_{\beta}^']^T, \quad s_0^' = 1 - \frac{D_{def} - (D_i - W_{def}) v_{def}}{D_{def}}
\]
where $s'_{si}$ is a laser scanner’s reading and $a$ a angle width vector is then created:

$$a = [a_y, a_{z0}], a_i = \alpha_i \left\{ \frac{D_{safe} \cdot (D_i - W_{safe} \cdot p_{iD_{safe}})}{D_{safe}} \right\} \quad (7)$$

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

a) A histogram vector $H'$ is then widened according to robot's dimensions as follows to create true histogram $H$:

$$H = [s_{x,0}, ..., s_{x,100}], s_{x,i} = \max (S_i), S_i : s_{x,i} \in S_i$$

$$\Rightarrow (j - a_i \leq i) \land (j + a_i \geq i) \quad (8)$$

Vector $H$ is produced based on vectors $H'$ (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 destination point in WGS-84 coordinate system, $W_{safe}$ is a radius of safety region around the robot.

Then we establish a set of valleys:

$$A \rfloor \left\{ \sum_{i=1}^{6} \phi_i + c_1 \right\} + c_2 \quad (14)$$

$c_1, c_2$ 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. $c_1 = -3 + \sqrt{15}, c_2 = 1/3 + \sqrt{15}$

$$\mu_i = \begin{cases} 0 & \text{if } \phi_i = 1 \\ 1 & \text{if } \phi_i = 0 \end{cases}, \quad \mu_i = \mu_1 = \begin{cases} 0 & \text{if } \phi_1 \prod_{i \neq 1} \phi_i = 1 \\ 1 & \text{if } \phi_1 \prod_{i \neq 1} \phi_i = 0 \end{cases}$$

$f_a$ 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:

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

where: $\Delta t$ - 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 $a_i$, which value depends on the distance to the nearest obstacle. They are shown in figures below.

![Fig. 5. Illustration of $a_i$ functions](image)

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

$$E = \sum_{t=1}^{N} e_t^2 \Delta t,$$  \hspace{1cm} (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

$$S = \sqrt{\frac{\sum_{t=1}^{N} (v_t - v_M)^2}{(N - 1)}},$$  \hspace{1cm} (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

$$s = \sum_{t=1}^{N} v_t \Delta t;$$  \hspace{1cm} (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](image)

![Table 1. Values of quality rates for Simulation 1 - 3](image)

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 $\gamma$ 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 (Fig 7b).

![Graph](image1)

**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.

![Graph](image2)

**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.

![Graph](image3)

**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).

![Graph](image4)

**Fig. 10. Motion trajectory of Table 2. Values of quality**
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


