*Last edited: 2014.11.17. 11:17*

Periodica Polytechnica

*Civil Engineering*
*x(y) pp. x-x, (year)*
*doi: 10.3311/pp.tr.201x-x.xx*
*web: http://www.pp.bme.hu/ci*
*© Periodica Polytechnica year*

RESEARCH ARTICLE

# Mobile robot navigation tests for civil engineering considerations

Ferenc TAJTI[1,2], Géza SZAYER[1], Bence KOVÁCS[2], Mauricio A. P. BURDELIS[3]

Received 2014-10-31

***Abstract***

*In the last few years the market of robotics has significantly changed and the growing sector of service robotics requires new considerations in civil engineering. This paper investigates different room and furniture arrangements in the view of mobile robot navigation requirements. The described method provides robot motion and path planning cost functions for different furniture arrangements and floor maps. During mobile robot path planning ISO 7176-10:2008 and ISO 7176-5:2008 (electrically powered wheelchair) compatible buildings were considered to find the optimal solutions over the minimal standardized requirements. The proposed method can provide efficient results regarding navigation error, time and power consumption.*

***Keywords***

*mobile robot, robot compatible design, maneuvering space, obstacle avoidance, ground plan, cost functions*

***Acknowledgement***

*The authors wish to thank the support to the Hungarian Research Fund (OTKA K100951), and MTA-ELTE Comparative Ethology Research Group (MTA: 01 031).*

## 1 Introduction

The needs of automation and robotics solutions are everyday questions in the industrial sector but have been growing also in daily life situations. Engineering solutions in robotics have reached a technical level in which mobile robots can be used in the service sector to solve many problems and tasks such as cleaning, supply chain, etc. This field of mobile robotics requires an investigation from the civil engineering point of view. Since the wheelchair related standards (ISO 7176-10:2008 and ISO 7176-5:2008) were defined civil engineering has changed a lot to satisfy the needs of handicapped people. The basic concept in the design process of a service robot is that the robot should be able to move around the same obstacles and maneuvering space as wheelchairs, but within this definition there is no cooperation or feedback regarding civil engineering. The working environment of a robot has a significant effect on the power consumption and on the navigation performance. The aim of this paper is provide feedback to the field of civil engineering, regarding to these two performance criteria. Our recommendations are based on theoretical investigation and experimental results obtained by using differential and kiwi drive mobile robot platforms.

From the user point of view, the energy consumption of wheelchairs or robots is an important issue, and a deeply investigated design for the interior of a building can significantly extend battery life.

Mobile robots use odometry for low level positioning, and vision or laser sensors for higher level navigation. Odometry calculates robot's position based on wheel rotation using kinematic equations. Arising from its operation, it has an accumulative error, which is handled by the upper level sensors and algorithms. Odometry error has significant effect on motion planning and mapping, which can be improved by optimized interior design.

Different paths have different costs related to the odometry errors, energy consumption and travel time. The main contribution of this paper is to provide a numerical method to test and improve building maps and interior

[1]Department of Mechatronics Optics and Mechanical Engineering Informatics
Faculty of Mechanical Engineering,
Budapest University of Techology and Economics
H-1111 Budapest, Műegyetem rkp. 3, Hungary (e-mail: tajti@mogi.bme.hu, geza.szayer@mogi.bme.hu, bence.kovacs@mogi.bme.hu, korondi@mogi.bme.hu)

[2]MTA-ELTE Comparative Ethology Research Group MTA 01 031,
Office for Subsidized Research Units,
Hungarian Academy of Sciences
H-1117 Budapest, Pázmány Péter sétány 1C, Hungary (e-mail: tajti@mogi.bme.hu)

[3]Software Developer Fellow for the Brazilian National Institute of Historic and Artistic Heritage (IPHAN)
Avenida Professor Luciano Gualberto, 380 - Butanta, Sao Paulo – SP, Brazil
(e-mail: mburdelis@gmail.com)

designs for service robotics and wheelchair transportation [1].

The paper is organized as follows. Section 1 introduces the problem and describes the background. Section 2 introduces two basic types of mobile robot kinematics and a robot used for the theoretical investigation and measurements. Section 3 describes a path planning method based on the floor maps of different buildings. Section 4 describes the theoretical background of the odometry and energy related motion cost functions. Section 5 describes the experimental results and Section 6 concludes the paper.

## 2 Mobile robot motion theory

Mobile robots are usually controlled by position and angular position, or velocity and angular velocity references. Path planning algorithms provide references for the robot which is often controlled without position and orientation feedback. In this case, only the servo amplifiers have their own angular position feedback during the motion, and the references of the wheels can be calculated with the equations of the inverse kinematics. The actual position and orientation of the robot is calculated from the direct kinematics, however such technique cannot compensate for slips of the wheels and mechanical errors of the structure, like gearbox backlash.

In the view of motion on the ground plane a mobile robot can have 2 or 3 degrees of freedom (DoF) depending on the structure of the drive. In case of 3 DoF (e.g. holonomic drives), the robot can change the position and the orientation at the same time, but a 2 DoF robot (e.g. differential drives) cannot change position and orientation independently. We will investigate the motion-cost function in Section 4 trough a differential and a holonomic robot motion type. This section describes the theoretical background of the inverse kinematical and direct geometrical functions of the example robots. The transformation between the world coordinate system and the robot coordinate system can be described as (2.1) (See Figs. 1 and 2.),

$$T_{W\to R}\colon \Re^3 \xrightarrow{Rot(z,\varphi)^T\cdot Trans(x,\Delta x)^{-1}\cdot Trans(y,\Delta y)^{-1}} \Re^3 \quad (2.1)$$

where $Rot(z,\varphi)^T$ is the rotational transformation around axis z, $Trans(x,\Delta x)^{-1}$ and $Trans(y,\Delta y)^{-1}$ are the linear transformations along axes x and y. The transformation of a point from world coordinate system to robot coordinate system can be expressed as (2.2).

$$\boldsymbol{p}_R = (\boldsymbol{p}_W - \boldsymbol{a}_{offset}) \cdot \boldsymbol{Rot}(z,\varphi)^{-1} \quad (2.2)$$

Equation (2.2) can be expressed as (2.3),

$$\begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} = \left( \begin{bmatrix} x_W \\ y_W \\ z_W \end{bmatrix} - \begin{bmatrix} \Delta x \\ \Delta y \\ 0 \end{bmatrix} \right) \cdot \begin{bmatrix} cos\varphi & sin\varphi & 0 \\ -sin\varphi & cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

where index $R$ is related to robot coordinate system and index $W$ is related to world coordinate system. The inverse kinematical functions of a 2-wheeled robot with differential drive can be described as (2.4) and (2.5),

$$\omega_{W1} = \frac{1}{R_W}\left(v - \frac{1}{2}\cdot L\cdot \omega\right) \quad (2.4)$$

$$\omega_{W2} = \frac{1}{R_W}(R_W \cdot \omega_{W1} + L\cdot\omega) \quad (2.5)$$

where L is the distance between the wheels and $R_W$ is the radius of the wheels. In this case the robot has two wheels with drives $(\omega_{W1}, \omega_{W2})$ and one wheel with free rotation and free steering, $v$ and $\omega$ are the velocity and the angular velocity of the robot.

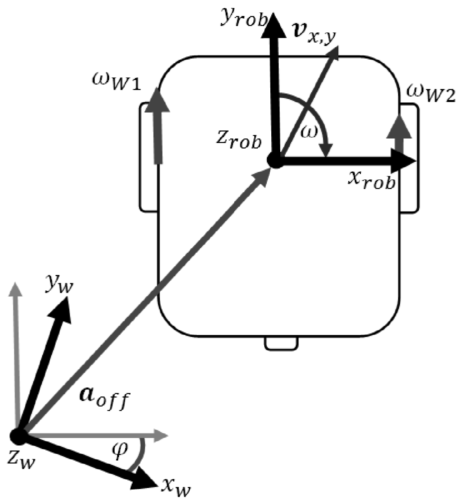


***Fig. 1*** The kinematic parameters of a differential drive, where $\omega_{W1}$ $\omega_{W2}$ are the angular velocity of the wheels, $x_{rob}$, $y_{rob}$, $z_{rob}$ are the robot coordinates, $x_W$, $y_W$, $z_W$ are the world coordinates, $\varphi$ is the orientation, $\boldsymbol{a}_{off}$ is the offset (position), $\boldsymbol{v}_{x,y}$ is the velocity and $\omega$ is the angular velocity

The position of the robot and the direct geometry can be expressed as (2.6-2.8),

$$\varphi = \int \frac{R_W\cdot\omega_{W2} - R_W\cdot\omega_{W1}}{2} dt + \varphi_0 \quad (2.6)$$

$$x = \int \frac{R_W\cdot\omega_{W2} + R_W\cdot\omega_{W1}}{2} \cdot \cos\varphi\, dt + x_0 \quad (2.7)$$

$$y = \int \frac{R_W\cdot\omega_{W2} + R_W\cdot\omega_{W1}}{2} \cdot \sin\varphi\, dt + y_0 \quad (2.8)$$

where $\varphi_0, x_0, y_0$ are the values of the start position and orientation of the robot and $\varphi, x, y$ are the actual robot position and orientation.

The inverse kinematical functions of a 3 wheeled holonomic drive (kiwi drive) can be described as (2.9-2.11), where $K$ and $C$ are constants used to obtain real units.

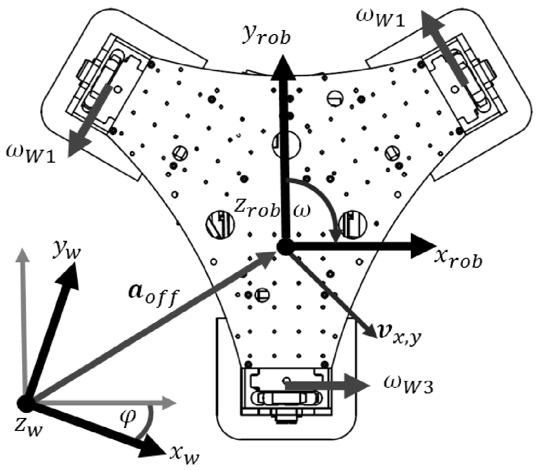


**Fig. 2** The kinematic parameters of a kiwi drive, where $\omega_{W1}$ $\omega_{W2}$ $\omega_{W3}$ are the angular velocity of the wheels, $x_{rob}$, $y_{rob}$, $z_{rob}$ are the robot coordinates, $x_W$, $y_W$, $z_W$ are the world coordinates, $\varphi$ is the orientation, $\boldsymbol{a}_{off}$ is the offset (position), $\boldsymbol{v}_{x,y}$ is the velocity and $\omega$ is the angular velocity

In this case the robot has three omnidirectional wheels with drives ($\omega_{W1}, \omega_{W2}, \omega_{W3}$).

$$\omega_{W1} = K\left(C\omega - v_x \sin 30° - v_y \cos 30°\right) \qquad (2.9)$$

$$\omega_{W2} = K\left(C\omega - v_x \sin 30° + v_y \cos 30°\right) \qquad (2.10)$$

$$\omega_{W3} = K(C \cdot \omega + v_x \cdot 2) \qquad (2.11)$$

The position of the robot and the direct geometry can be expressed as (2.12-2.14).

$$\varphi = \int \frac{\omega_{W1}+\omega_{W2}+\omega_{W3}\cdot\sin 30°}{K\cdot C\cdot(\sin 30°+2)} dt + \varphi_0 \qquad (2.12)$$

$$x = \int - \frac{\omega_{W1}+\omega_{W2}-2\cdot\omega_{W3}}{2\cdot K\cdot(\sin 30°+2)} dt + x_0 \qquad (2.13)$$

$$y = \int - \frac{\omega_{W1}-\omega_{W2}}{2\cdot K\cdot\cos 30°} dt + y_0 \qquad (2.14)$$

In the experimental results (Section 5) we have worked with Ethon which is a 3-wheeled holonomic type mobile robot. (See Fig. 3).

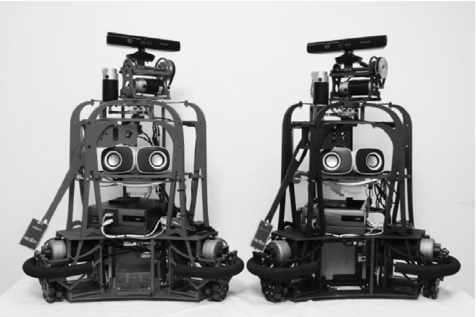


***Fig. 3*** Ethon robots

## 3 Path planning

Several path planning algorithms are generally used in engineering for motion in 3D space [2], but for 2D motion of mobile robots, commonly used methods are genetic algorithms, sampling-based motion planning, fuzzy systems, neural networks, and artificial potential field (APF) methods. Global path planning algorithms collect and use global information. Such algorithms find paths if they exist, however computation complexity limits their online usage and planning in dynamic environments. Global optimization, such as ant colony algorithms can be used, but convergence is also slow. Local algorithms often based on APF based methods have simple implementation and low processing needs.

APF methods are based on the idea that the target attracts the robot and obstacles generate repulsive force. An attractive potential function as (3.1) was proposed in [3]. It handles moving targets and provides soft landing. Providing soft landing means that the robot reaches the target with the same velocity as the target.

$$U_{att}(\boldsymbol{x}, \boldsymbol{v_R}, \boldsymbol{v_T}) =$$

$$\alpha_x |\boldsymbol{x}(t)|^m + \alpha_v \|\boldsymbol{v_R}(t) - \boldsymbol{v_T}(t)\|^n \qquad (3.1)$$

where $|\boldsymbol{x}(t)|$ is the Euclidean distance between robot and target, $\boldsymbol{v_R}(t), \boldsymbol{v_T}(t)$ denote the velocity of the robot and the target at time t, respectively; $\|\boldsymbol{v_R}(t) - \boldsymbol{v_T}(t)\|$ is the magnitude of the relative velocity between robot and target; $\alpha_x$, $\alpha_v$ are scalar positive parameters; and $m, n$ are none-negative constants. The force vector function pointing from the robot to the target is calculated by taking the derivative of the potential function according to equation (3.2).

$$F_{att}(\boldsymbol{x}, \boldsymbol{v_R}, \boldsymbol{v_T}) = -\nabla U_{att}(\boldsymbol{x}, \boldsymbol{v_R}, \boldsymbol{v_T}) =$$

$$\frac{\partial U_{att}(x,v_R,v_T)}{\partial x} \boldsymbol{n_{RT}} + \frac{\partial U_{att}(x,v_R,v_T)}{\partial (v_R(t)-v_T(t))} \boldsymbol{n_{VRT}} \qquad (3.2)$$

where $\boldsymbol{n_{RT}}$ is the unit vector pointing from the robot to the target and $\boldsymbol{n_{VRT}}$ denotes the unit vector pointing from the robot in the direction of the relative velocity of the robot with respect to the target (i.e. the velocity of the robot in the frame of reference of the target).

We propose a repulsive potential function as (3.3) in order to avoid collisions and handle moving obstacles:

$$U_{rep}(\boldsymbol{x}, \boldsymbol{v_R}, \boldsymbol{v_{OBS}}) =$$

$$\begin{cases} -\log\left(\delta(|\boldsymbol{x}(t)| + \boldsymbol{r}_{sec} + \boldsymbol{r}_{rob})\right) \\ \qquad -\zeta \frac{\|v_R - v_{OBS}\|^2}{2a_{MAX}} & if\ \delta(\boldsymbol{x} + \boldsymbol{r}_{sec} + \boldsymbol{r}_{rob}) < 1 \\ \qquad -\zeta \frac{\|v_R - v_{OBS}\|^2}{2a_{MAX}} & else \end{cases} \qquad (3.3)$$

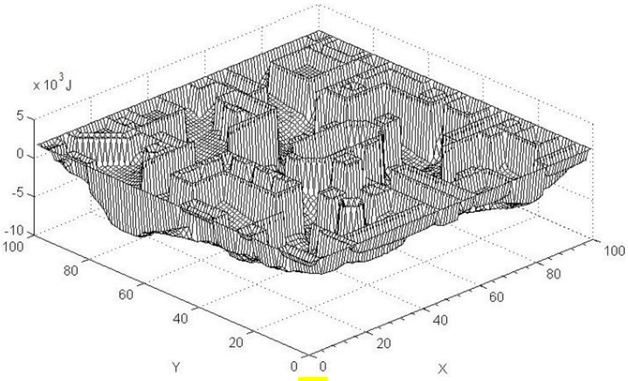
where $\boldsymbol{r}_{sec}$ is a constant expressing a safe distance between the robot and the obstacle, in order to avoid collisions. $\boldsymbol{r}_{rob}$ is the radius of the robot assuming a cylinder-shaped robot. $\delta, \zeta$ are non-negative constants, $\boldsymbol{v_R}, \boldsymbol{v_{OBS}}$ denotes the robot and obstacle velocity vectors respectively. The maximum acceleration of the robot is $a_{MAX}$. The repulsive vector force function pointing from the obstacle to the center of the robot is calculated as (3.4).

$$\boldsymbol{F}_{rep}(\boldsymbol{x}, \boldsymbol{v}_{\boldsymbol{R}}, \boldsymbol{v}_{\boldsymbol{OBS}}) = -\nabla U_{rep}(\boldsymbol{x}, \boldsymbol{v}_{\boldsymbol{R}}, \boldsymbol{v}_{\boldsymbol{OBS}}) =$$

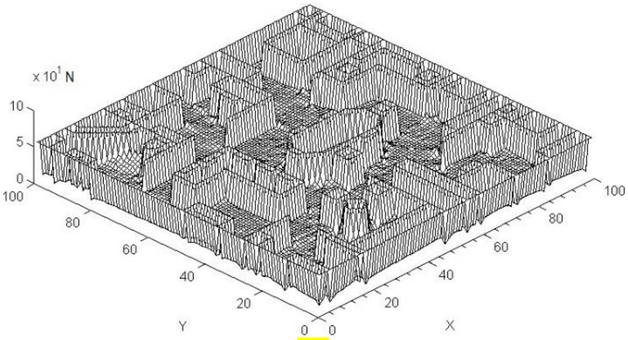
$$\frac{\partial U_{rep}(\boldsymbol{x},\boldsymbol{v}_{\boldsymbol{R}},\boldsymbol{v}_{\boldsymbol{OBS}})}{\partial \boldsymbol{x}}\boldsymbol{n}_{\boldsymbol{OR}} + \frac{\partial U_{rep}(\boldsymbol{x},\boldsymbol{v}_{\boldsymbol{R}},\boldsymbol{v}_{\boldsymbol{OBS}})}{\partial(\boldsymbol{v}_{\boldsymbol{R}}(t)-\boldsymbol{v}_{\boldsymbol{OBS}}(t))}\boldsymbol{n}_{\boldsymbol{VOR}} \quad (3.4)$$

where $\boldsymbol{n}_{\boldsymbol{OR}}$ is the unit vector pointing from the obstacle to the center of the robot and $\boldsymbol{n}_{\boldsymbol{VOR}}$ denotes the unit vector pointing to the relative velocity direction of the robot with respect to the obstacle (i.e. the velocity of the robot in the frame of reference of the obstacle).

MATLAB simulations were implemented in order to evaluate potential trajectories and their corresponding performance in an experimental environment. An experimental square layout flat was created and the corresponding sensor data was calculated. The 8x8 (eight times eight) meters evaluation flat was divided into 100x100 (hundred times hundred) simulation points. A static illustration of the resulting potential and force fields can be seen in Fig. 4. (a) and (b), respectively.



(a)



(b)

**Fig. 4** Generated fields inside an evaluation flat according to the proposed attractive (+) and repulsive (-) potential functions. The X-Y plane represents the flat layout scaled to 100x100 simulation points, while the Z axis represents (a) the potential field, and (b) the force field.

The artificial potential method is based on the idea that obstacles generate repulsive force, and the target generates attractive force to the robot. The aim of calculating potential and force fields is to calculate the path to a predefined target position. The sum of attractive and all repulsive forces is moving the robot according to Newton's laws as (3.5).

$$\ddot{\boldsymbol{x}} = \frac{\sum F_{att}(\boldsymbol{x},\boldsymbol{v}_{\boldsymbol{R}},\boldsymbol{v}_{\boldsymbol{T}})+F_{rep}(\boldsymbol{x},\boldsymbol{v}_{\boldsymbol{R}},\boldsymbol{v}_{\boldsymbol{OBS}})}{m_{rob}} \quad (3.5)$$

where $m_{rob}$ is the mass of the robot.

The path planner algorithm calculates the trajectory path, starting from an initial position with zero velocity and sequentially calculating the next position by taking the double integral of the acceleration in (3.5).

## 4 Cost functions

The performance criteria have to be defined to evaluate the results of the path planner output. Two main aspects are discussed in this chapter: the estimated energy consumption and the position losses of the odometry based navigation. These performance estimations are properly described with their corresponding cost functions.

The odometry-based navigation error can be intensely accumulated depending on the motion of the robot. As a simple example, wheel slipping cases are more frequent in case of acceleration states compared to a linear, straight movement with constant velocity.

The experimental amounts of typical odometry error, for different robots are in the same magnitude in case of a fine-tuned motion control. As an illustrative example, during 30~50 meters of motion in different directions, the Ethon robot odometry can accumulate 1~2 meter errors in any directions and 10~20° errors in orientation. [5] However, in practice, this unacceptable amount of error is corrected by the SLAM computer vision algorithm of the high-level robot control software. But the minimization of odometry error is still an important goal in real-time control systems, in order to obtain a better feedback at a much lower sampling rate compared to the vision based-solutions. For understanding position losses, the following root cause phenomena have to be considered:

- The slip of the wheels [4,6]
- The alternating contact points between the floor and the wheels in case of omni-wheels (see Fig. 5), or in case of a differential drive assembled with wheels having wide contact surfaces
- The production accuracy of the robot mechanics and wheels contains some ‰ of errors (For example in the case of Ethon, 0,1 mm difference between the diameters of the wheels causes at least 6‰ orientation error and 1‰ position error related to the travelled path)

In case of a kiwi platform, the constant parameters in the inverse kinematical functions express the distance between the center of the robot geometry and the contact points of the wheels, which alternates between $W_IA$ and $W_IB$ and it can cause 5,4% error in the robot position and orientation.

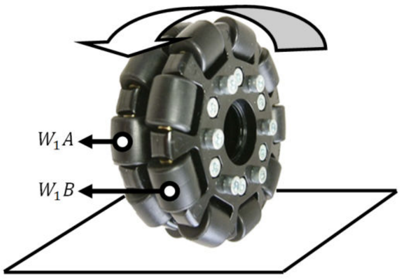
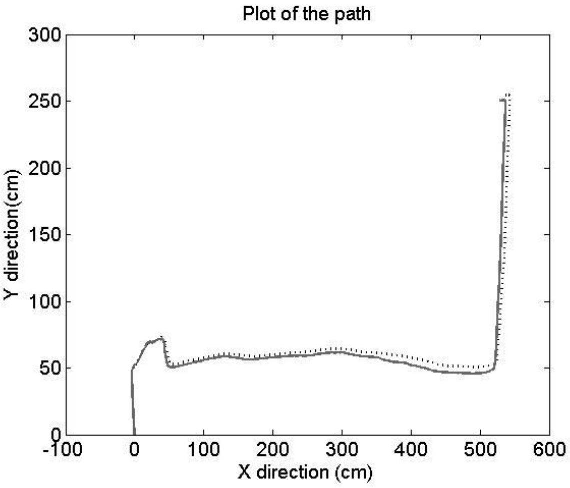
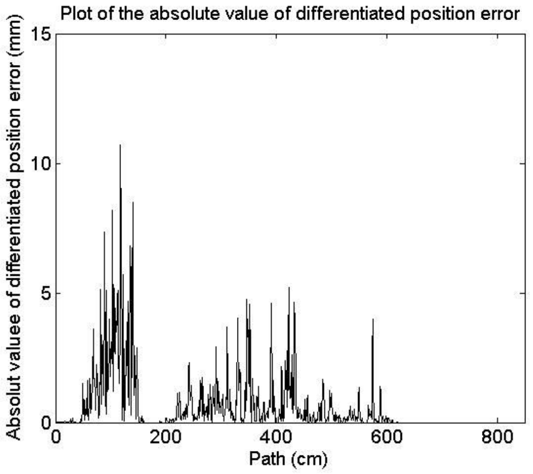


***Fig. 5*** Alternating wheel contact points

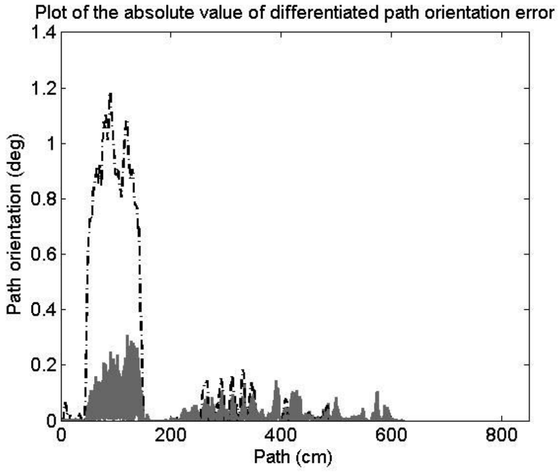
During robot motion 1-4° angular error causes a 3‰ ~1,5% position error that corresponds to 15 cm error after 10 meters of driving. 15 cm position error can cause serious problems (e.g. in the case of a room entrance). A similar problem occurs in the case of differential platforms, which have wheels with wider contact surfaces. The odometry errors were measured with different robots, in measurement experiments, which estimated the real position with better accuracy by using sensor fusion with optical flow position measurements based on ADNS9500 sensors, similar to the ones in [6, 7] and with the distributed camera vision system of the Mechatronics Department of the Budapest University of Technology and Economics [8]. The odometry error increase was obtained by getting the absolute value of the first derivetive of the odometry error.



(a)



(b)



(c)

***Fig. 6*** Odometry position (b) and orientation (c) errors along the bechmark trajectory (a) In (a) the normal line represents the reference path and the dotted line is the real path. In (c) the grey line is the absolute value of the derivative of the orientation error and the dotted line is the absolute value of the orientation.

Figure 6 (a) shows one of the benchmark trajectories, which was measured along an angular path, where the dotted line is the robot path and the normal line is the reference path, (b) shows the position odometry error increase along the trajectory, and finally, (c) shows the orientation error increase, where the dotted plot shows the angular position and the normal grey plot shows the derivative of the angular error. We use absolute values instead of signed values, because during the tests we are interested in the absolute amount of error. If we define positive and negative directions for motion, the robot will make the same errors in positive and also in negative directions, so signed values are not meaningful in this experiment. Furthermore during changes of orientation, the robot moves along the x and y axes and the positive and negative errors cannot compensate for each other and both have effects on the robot position. It can be clearly seen in (c) that the orientation error has bigger growth when the robot is turning along a sharp curve (in the beginning of the trajectory in (a)). For evaluation, benchmark tests were carried out over the past few years using many different robots along the same trajectories, but under different conditions (e.g. the wheels were changed, or motion control parameters were differently adjusted.) The test paths can be split into five different types of sections where the robot's motion state is clearly different. The data from diagrams (b) and (c) of Fig. 6 are also split into sections according to the motion states. Fig. 6 was made from measurements with Ethon holonomic drive based robot. Evaluations were made based on motion states under more benchmark measurements, with two different types of robots. Table 1. shows all these states with the results of average increase in odometry error at each state both for a holonomic and for a differential robot.

**Table 1 Odometry error increase related to different motion states**

| Robot Motion State | Differential | | Holonomic | |
|---|---|---|---|---|
| | Pos. ε [%] | Orient. ε [%] | Pos. ε [%] | Orient. ε [%] |
| Linear acceleration and deceleration<br>$\ddot{s} \neq 0$<br>$\dot{\varphi}, \ddot{\varphi} = 0$ | 0.85 | 1.1 | 1.15 | 1.55 |
| Linear straight movement with constant velocity<br>$\dot{s} \neq 0, \ddot{s} = 0$<br>$\dot{\varphi}, \ddot{\varphi} = 0$ | 0.14 | 0.3 | 0.19 | 0.35 |
| Turning in standstill position<br>$\dot{s}, \ddot{s} = 0$<br>$\dot{\varphi}, \ddot{\varphi} \neq 0$ | 0.35 | 0.7 | 0.45 | 0.95 |
| Driving along a smooth curve with head front<br>$\dot{s}, \ddot{s} \neq 0$<br>$\dot{\varphi}, \ddot{\varphi} \neq 0$ | 0.2 | 0.4 | 0.25 | 0.45 |
| Curve path with orientation change (holonomic only)<br>$\dot{s}, \ddot{s} \neq 0$<br>$\dot{\varphi}, \ddot{\varphi} \neq 0$ | n/a | n/a | 0.3 | 0.5 |

As previously mentioned a mobile robot with differential drive has only 2 DoF. It cannot change the position and the orientation at the same time so the curved path with orientation change data is not available in this case. The ratio of the orientation losses along different path sections of the holonomic platform can be seen on Fig. 7. It is possible to see by comparison that most of the errors occur during acceleration or deceleration states and during standstill turning. These numbers can be compared with very similar ratios to the position errors, also in case of differential drive robots.

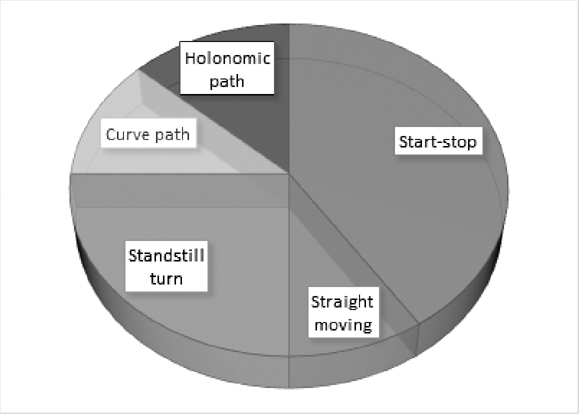


**Fig. 7** The proportion of orientation error increase along different path sections of the holonomic platform

Energy consumption is an important part of mobile robotics and wheelchair transportation [9, 10]. The energy consumption of the robot can be estimated from the motion path. During acceleration and angular acceleration the robot increases its motional and rotational energy. During deceleration and angular deceleration the robot decreases the energy of motion through motor braking. In the case of braking, the servo amplifiers change the direction of the electrical current in the servo, and this means that a strong or abrupt braking action can consume the same amount of energy as acceleration and angular acceleration. The time function of the energy consumptions is expressed in (4.1),

$$E(t) = \begin{cases} \dot{s}(t)\tau_{\dot{s}} + \dot{\varphi}(t)\tau_{\dot{\varphi}} + mg\Delta h(t) \\ \quad + \frac{1}{2}m\dot{s}(t)^2 + \frac{1}{2}\Theta\dot{\varphi}(t)^2 & \text{if } \dot{s}, \dot{\varphi} \neq 0,\ \ddot{s}, \ddot{\varphi} \neq 0 \\ \dot{\mathrm{s}}(\mathrm{t})\tau_{\dot{\mathrm{s}}} + \dot{\varphi}(\mathrm{t})\tau_{\dot{\varphi}} + mg\Delta h(t) & \text{if } \dot{s}, \dot{\varphi} \neq 0,\ \ddot{s}, \ddot{\varphi} = 0 \end{cases} \tag{4.1}$$

where $\Delta h(t)$ is the time function of the height of the path, $\tau_{\dot{s}}$ and $\tau_{\dot{\varphi}}$ are the linear and the angular power loss constants which include friction, rolling resistance, gear efficiency, etc. The power loss constants could be defined with measurements, but in this case are used as parameters. The total amount of energy consumption by following the path can be calculated from the sum of the energy consumption of all time intervals. During the path planning experiments we did not use ground plans with different heights, like wheelchair ramps ($\Delta h(t) = 0$) and we have investigated the energy as a function of the path instead of a function of time. In this case the total amount of energy can be calculated as (4.2 ),

$$\sum E = \int_s E(s) \tag{4.2}$$

where $E(s)$ is the path-related energy consumption function, which can be described as (4.3).

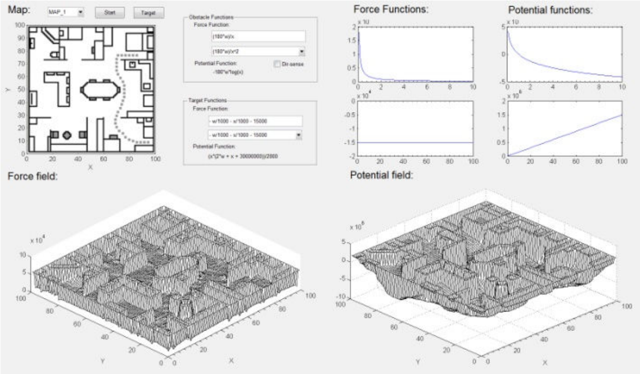
$$E(s) = \begin{cases} \dot{s}(s)\tau_{\dot{s}} + \dot{\varphi}(s)\tau_{\dot{\varphi}} + \frac{1}{2}m\dot{s}(s)^2 + \frac{1}{2}\Theta\dot{\varphi}(s)^2 & \text{if } \dot{s},\dot{\varphi} \neq 0,\ \ddot{s},\ddot{\varphi} \neq 0 \\ \dot{s}(s)\tau_{\dot{s}} + \dot{\varphi}(s)\tau_{\dot{\varphi}} & \text{if } \dot{s},\dot{\varphi} \neq 0,\ \ddot{s},\ddot{\varphi} = 0 \end{cases} \tag{4.3}$$

As previously mentioned, in the experiments described in the experimental results section we have used Ethon robots, and we could make long-term linear motion and rotational motion battery tests. During the tests the robot could move 13% more with angular rotation so the ratio of $\tau_{\dot{s}}$ and $\tau_{\dot{\varphi}}$ can be described as (4.4), what is important at the experimental results section to get the final numbers in Table 2.

$$\frac{\tau_{\dot{\varphi}}}{\tau_{\dot{s}}} = 1{,}13 \tag{4.4}$$

## 5 Experimental results

Different ground plans were tested with the potential field path planning algorithm, and also simulations were made with the same ground plan, with slightly different furniture arrangements. The goal of this paper is not restricted to a limited robotic task or functionality. The start and end points of the path can be chosen arbitrarily. The main contribution is to propose a method to measure the robotic compatibility of different options of ground plans. To run the tests we have designed a Graphical User Interface (GUI), where the ground plans can be imported from bitmap (.bmp) image files and the parameters of the path planning methods can be changed. (See Fig. 8.) From the calculated path, the results of the energy and odometry loss cost functions can be calculated for differential and also for holonomic drives.



Obstacle Functions

Force Function:

(100*w)/x

(100*w)/x^2

Potential Function: ☐ Dir-sense

-100*w*log(x)

Target Functions

Force Function:

- w/1000 - x/1000 - 15000

- w/1000 - x/1000 - 15000

Potential Function:

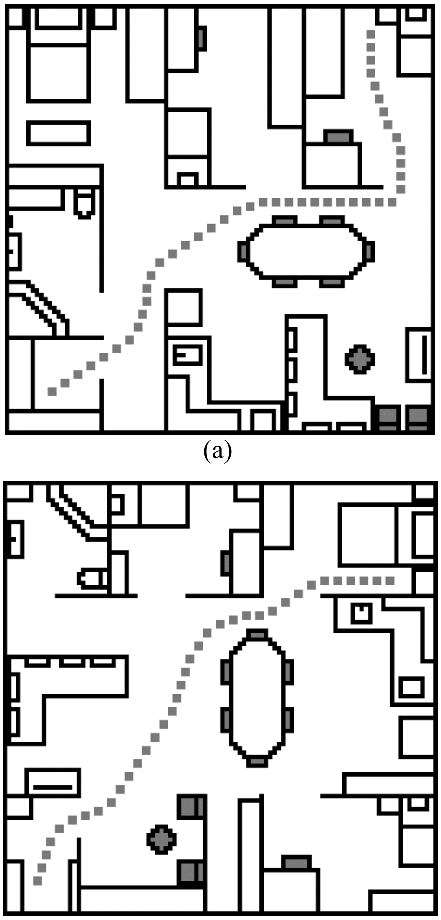
(x^2*w + x + 30000000)/2000

*Fig. 8* The GUI of the proposed path planning method

The absolute odometry losses and the energy consumption related to the whole path have to be divided by the Euclidean distance between the start and end points of the path, in order to get specific performance criteria indicators that are independent from the exact functionality. The results can be seen in Table 2. Initially, we have tested ground plans which contained more separate rooms, and then we have compared them to rooms that are less separated from each other (recent trend). One ground plan of each type can be seen on Fig. 9. We have made simulations with the same ground plans, considering that smaller lightweight furniture are moved very often (even though people can simply move them out of the way when they are crossing the ideal path, the robots are often unable to do that autonomously). A simulation with the same ground plans, but slightly different furniture arrangements can be seen on Fig. 10.

**Table 2 Odometry error and energy cost results**

| Energy and odometry cost results | Differential | | Holonomic | |
|---|---|---|---|---|
| | Odom. [%] | Energy [%] | Odom. [%] | Energy [%] |
| Fig. 9. (a) | 116 | $E_1$ | 124 | $1{,}08 \cdot E_1$ |
| Fig. 9. (b) | 109 | $0{,}72 \cdot E_1$ | 117 | $0{,}81 \cdot E_1$ |
| Fig. 10. (a) | 121 | $E_2$ | 129 | $1{,}09 \cdot E_2$ |
| Fig. 10. (b) | 138 | $1{,}13 \cdot E_2$ | 147 | $1{,}22 \cdot E_2$ |


(a)

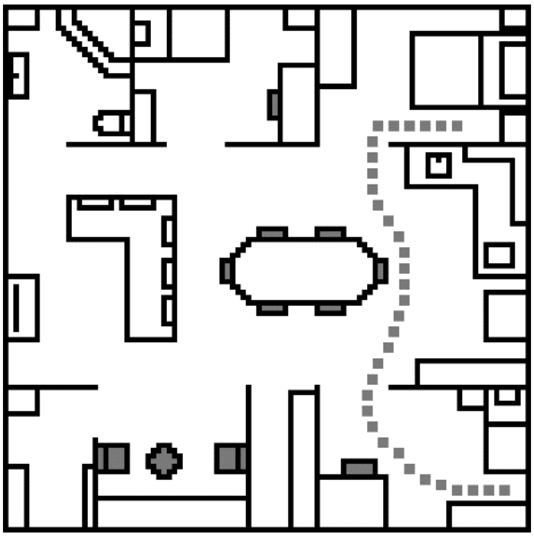
(b)

**Fig. 9** Generated paths for two different ground plans. The first map (a) has less inner walls and narrow places, therefore the generated path has smooth curves.

## 6 Conclusion

The generated test trajectories show about 30~65% differences between the estimated energy consumptions and the odometry position and orientation errors. The better paths have smoother curves, with less sharp-angle corners and start-stop positions. This can be explained by simple kinetic energy calculations, as well as the higher relative position and orientation errors related to the acceleration, deceleration and turning in standstill position motion states. As a first conclusion, we can state the obvious fact that maps consisting of sharp edges, narrow passages etc. are more likely to fail for hosting mobile robotic applications. Furthermore, we have found out that a robot in a household often crosses the path of humans, so narrow passages are more costly for robots and also less comfortable for living.

In many other cases, the main reason for the occurrence of sharp edges and narrow passages in the map is that people frequently change the position of smaller furniture, like seats or coffee tables. As an example, the results of cost functions related to energy and odometry loss can increase significantly if a dining table with many chairs is positioned near the optimal path of the robot. Two different test maps that illustrate this fact can be seen on Fig. 9 (a) and (b). So, as an important recommendation, we can advise to leave more free space around frequently moved furniture in order to give robots the opportunity of easy navigation while humans can also cross the ideal path.


(a)

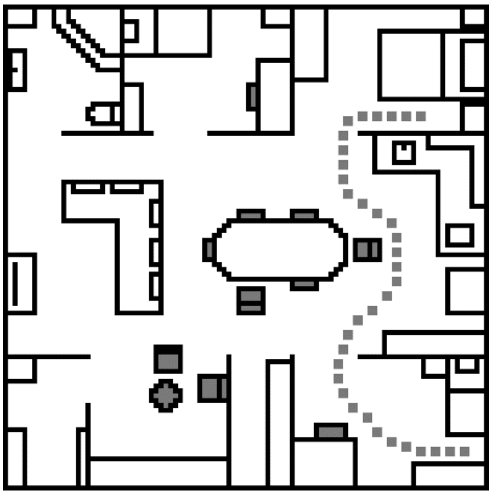

(b)

**Fig. 10** The same ground plan with slightly different furniture arrangements. The gray filled shapes correspond to easily and frequently moved furniture, such as chairs or coffee tables.

From the reasons mentioned above, we suggest that buildings for wheelchair transportation and mobile robotics should be investigated with similar methods to provide better environments for automated indoor transportation. The experimental results have proved that a deeply investigated ground plans and interior designs can save 30%...65% energy for electrical driven wheelchairs, which means much longer battery life over the minimal compulsory ISO standards. The other point of view relates to indoor mobile robotics, where the odometry loss cost is an additional benefit. With stronger communication and design feedback between different disciplines we can improve engineering designs to a higher level, where we can find new technical solutions for everyday problems.

## References

[1] Pin-Chun H, Won-jong K., *Autonomous robotic wheelchair with collision-avoidance navigation and real-time path planning*, in 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), 6-9 July 2010, Montreal, pp. 1396 - 1401, DOI: 10.1109/AIM.2010.5695790

[2] Paláncz B, *Numeric-Symbolic Solution for Satellite Trajectory Control by Pole Placement*, in Periodica Polytehcnica Civil Engineering, 2013, Vol. 57, No. 01, pp. 21-26, DOI: 10.3311/Pci.2138

[3] S.S. Ge, Y.J. Cui, *Dynamic Motion Planning for Mobile Robots Using Potential Field Method*, in Autonomous Robots 13, 2002, pp. 207–222. DOI: 10.1023/A:1020564024509

[4] Hyoung-Ki L, Kiwan C, Jiyoung P, Yeon-Ho K, Seokwon B, *Improvement of Dead Reckoning Accuracy of a Mobile Robot by Slip Detection and*

*Compensation using Multiple Model Approach*, in IEEE/RSJ International Conference on Intelligent Robots and Systems, (Nice, France), September 22-26, pp.1140-1147.DOI: 10.1109/IROS.2008.4650843

[5] Xiaojing S, Lakmal D.S, Kaspar A, Zibin S, *A Robust Slip Estimation Method for Skid-Steered Mobile Robots*, in Intl. Conf. on Control, Automation, Robotics and Vision, (Hanoi, Vietnam), December 17-20, 2008, pp. 279-284. DOI: 10.1109/ICARCV.2008.4795532

[6] Isaku N, Keigo W, Keiji N, Kazuya Y, *Noncontact Position Estimation Device with Optical Sensor and Laser Sources for Mobile Robots Traversing Slippery Terrains,* in IEEE/RSJ International Conference on Intelligent Robots and Systems, (Taipei, Taiwan), October 18-22, 2010, pp. 3422-3427. DOI: 10.1109/IROS.2010.5650346

[7] Ross R, Devlin J, Wang S, *Toward Refocused Optical Mouse Sensors for Outdoor Optical Flow Odometry*, in *IEEE SENSORS JOURNAL*, June, 2012, Vol. 12, No. 6, pp. 1925-1932. DOI: 10.1109/JSEN.2011.2180525

[8] Devecseri V, Dóka A, Molnár J, Tamás P, *An Ethological Motion Capture System,* in 12th IEEE International Symposium on Computational Intelligence and Informatics, CINTI November, 2011, Budapest, Hungary, IEEE Press 2011, pp. 487-491. DOI: 10.1109/CINTI.2011.6108555

[9] Tanohata N, Murakami H, Seki H, *Battery friendly driving control of electric power-assisted wheelchair based on fuzzy algorithm,* in Proceedings of SICE Annual Conference 2010, 18-21 Aug. 2010, Taipei, pp. 1595 - 1598, ISBN: 978-1-4244-7642-8

[10] Pei-Chung C, Yong-Fa K, *Residual traveling distance estimation of an electric wheelchair*, in 2012 5th International Conference on Biomedical Engineering and Informatics (BMEI), 16-18 Oct. 2012, Chongqing, pp. 790 - 794, DOI: 10.1109/BMEI.2012.6513075