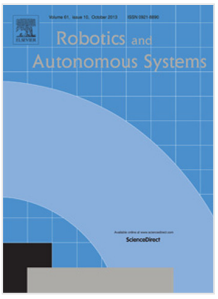
Accepted Manuscript



A novel potential field method for path planning of mobile robots by adapting animal motion attributes

Bence Kovács, Géza Szayer, Ferenc Tajti, Mauricio Burdelis, Péter Korondi

PII: S0921-8890(16)30215-9
DOI: http://dx.doi.org/10.1016/j.robot.2016.04.007
Reference: ROBOT 2627

To appear in: *Robotics and Autonomous Systems*

Received date: 22 June 2015
Revised date: 15 March 2016
Accepted date: 22 April 2016

Please cite this article as: B. Kovács, G. Szayer, F. Tajti, M. Burdelis, P. Korondi, A novel potential field method for path planning of mobile robots by adapting animal motion attributes, *Robotics and Autonomous Systems* (2016), http://dx.doi.org/10.1016/j.robot.2016.04.007

A Novel Potential Field Method for Path Planning of Mobile Robots by Adapting Animal Motion Attributes

Bence Kovács*, Géza Szayer, Ferenc Tajti, Mauricio Burdelis, Péter Korondi

Department of Mechatronics, Optics and Mechanical Engineering Informatics, Budapest University of Technology and Economics –Bertalan Lajos u. 4-6., 1111 Budapest, Hungary

HIGHLIGHTS

- Animal motion attributes were adapted to mobile robot path planning.
- The aim of the proposed method is to assist human-robot interaction.
- The proposed method is capable of on-line navigation in dynamic environments.
- Simulations were implemented and are accessible online for evaluation.
- The robotics implementation confirmed the simulation results.

ABSTRACT

This paper presents a concept for path planning of mobile robots in household environments. The proposed algorithm takes communicative means of locomotion into account in order to facilitate human-robot interaction. The well-known traditional artificial potential field method (APF) was extended by motion characteristics of household animals. The proposed algorithm includes velocity and orientation information and can be used in unknown, dynamic environments. The main contribution of this paper is the definition of an online, local path planning method by adapting animal motion attributes in order to assist human-robot interaction. The algorithm was implemented in an embedded system and evaluated on MOGI-ETHON, a holonomic drive mobile robot.

Keywords: Holonomic mobile robot; Potential field; On-line path planning; Laser-scanner; Animal motion attributes; Dynamic environment;

1. INTRODUCTION

Robots are mainly used in the industry sector, however, according to robotics trends, it is predicted that mobile robots will appear in average households in the following decades. One problem in this exponentially growing field is path planning in human-living areas. A robot's trajectory does not only have the function of changing location, but in some cases it also has communicative implications, therefore it has significant effects on human-robot interaction.

Robots in households should be operated easily and intuitively. This means that the user should not need to read complex user manuals or decode robot indications in order to be able to operate the robot. When smartphones emerged in the market, in just a few years they could be used without user manuals, due to their intuitive ways of use (e.g. scrolling pages of a document as if it were a book etc.). While smartphones have some similarities with books, there are some similarities between mobile robots and household animals. Our novel approach of path planning is based on the prediction that household animals live in a similar environment and are about the same size as household robots will be in the near future. As people lived together with household animals for thousands of years, people can interact with them more easily and can interpret the communicative implications of their locomotion. Therefore mobile robots can indicate certain intents or conditions to people by exhibiting the same motion attributes as animals. Ethological research studies addressed the same problems and proposed the implementation of the dog's visual communication elements [1,2] in order to enhance human-robot interactions, but robotics implementation of path planning in this context has not yet been studied.

Our objective is not to implement exact motions of certain animal species, but to design a path planning algorithm for mobile robots inspired by house-hold animals, with the aim of enhancing human-robot interaction.

Several research studies targeted mobile robot path planning [3], which led to such achievements as path planning based on Genetic algorithms [4], Sample-based motion planning [5], Neural networks [6], Fuzzy logic [7], Support Vector Machines (CVM) [8], and Artificial potential field (APF) [9–11]. Compact Internal Representation (CIR) [12] handles dynamic situations efficiently, but it requires constant speed along the path. Global path-planning techniques are guaranteed to find a path if one exists, however their computation time increases exponentially as the degrees of freedom (DOF) of the robot increase. This computational complexity limits their use for real-time path planning in dynamic environments. Ant colony algorithms [13,14] can also be used for global optimization, but convergence is also slow. [15] gives a wide overview of traditional and evolutionary approaches of both off-line and on-line path planning in the year 2012.

Local algorithms are usually based on APF due to its simple implementation and low processing needs. It can get input data directly from range sensors without the need for processor-consuming map calculations and estimations. It is suitable for underlying real-time control with the lack of global information, but it has a problem of local minima [9,10,16] which needs to be handled by global algorithms at a higher layer. All these studies were focused on finding the optimal path from the point of view of distance, time, or

*Corresponding author.
E-mail addresses: bence.kovacs@generalmechatronics.com (B. Kovács), geza.szayer@generalmechatronics.com (G. Szayer), tajti@mogi.bme.hu (F. Tajti), mauricio@mogi.bme.hu (M. Burdelis), korondi@mogi.bme.hu (P. Korondi).

energy. Many researches address path planning implications of human-robot interaction [17], but mainly focus on collision-free paths.

[18] suggests general methods for animal motion analysis and path planning, and shows an application for snake locomotion. Our research is new in terms of describing mobile robot trajectory planning by taking human-robot communication into account.

The APF method was an active research area until the decade of the 2000s, and this research field has been progressing until recent years, but it is a less active research topic nowadays as the problems of this area are already well described. However, we decided to use the APF method, because it fits the addressed problem well, and it is not yet explored from our approach.

One of the main characteristics of animal locomotion is that the body orientation and the velocity vector don't always point to the same direction. Therefore, the path planning algorithm was designed for a holonomic driven robot platform. Implementation was made on MOGI-ETHON (Fig. 1), a holonomic drive mobile robot.

Fig. 1. MOGI-ETHON, a holonomic mobile robot. Its omnidirectional drive allows the implementation and measurements of path-planning algorithms including orientation information. A laser scanner is built on the top of the robot providing real-time distance measurement data.

The rest of the paper is organized as follows. Section 2 focuses on the main motion attributes of the proposed animal-like path-planning. In Section 3 the proposed algorithm is presented along with its scope. Section 4 describes the experimental results, including simulations (4.1) and experimental implementation and results (4.2). Section 5 concludes the paper.

2. Adapted Animal Motion Attributes

This chapter points to three motion-related animal characteristics with communicative implications, which are proposed to be adapted to mobile robot path planning. People's interpretation of these animal motion attributes will be explained along with the proposed corresponding meaning in case of mobile robotics.

2.1 Object proximity-seeking

Objects often mean protection or hiding for animals. These are not just obstacles which have to be bypassed. Most visibly, cats often go close to the wall even if it results in a longer path as demonstrated on Fig. 2. This antipredator behavior specific to small mammals is well described by ethological studies. It is affected by anxiety-related medicines, therefore medical companies also analyze it in the widely used animal tests called open field test [19].

People's interpretation of object proximity-seeking behavior of animals is the avoidance of interaction. A mobile robot could behave in the same way if it does not intend to initiate an interaction, e.g. when the robot has an already defined task and does not expect new commands.

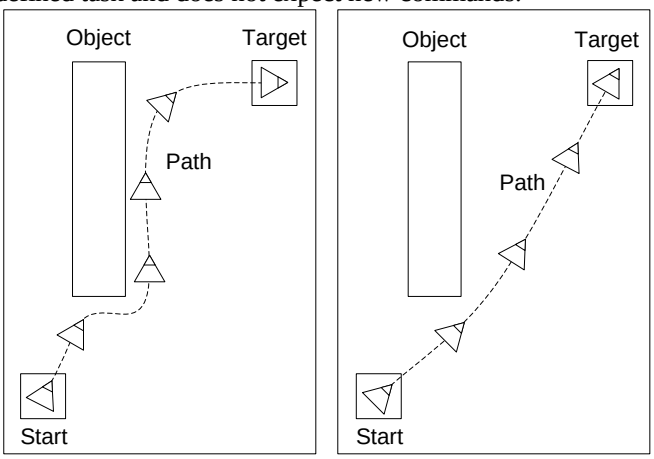


Fig. 2. The path (a) with and (b) without the object proximity-seeking motion characteristics. This nature is specific especially for small mammals, like cats or rodents, as they often stay close to objects for hiding and protection.

2.2 People approach

As ethological research points out [20], dogs distinguish between their owner, familiar people, and strangers. Dogs stay in closer proximity with their owner, and are more likely to interact with familiar people [2].

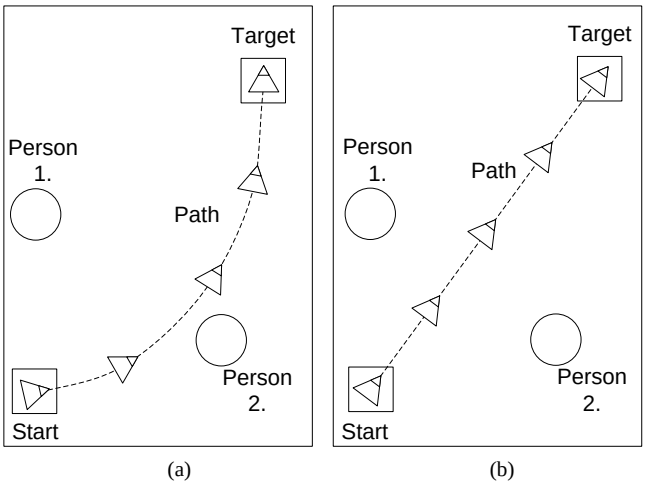


Fig. 3. Demonstration of different paths specific to dogs in case of (a) Person 2 has a more important role (e.g. Person 2 is the owner), (b) people approach doesn't differ.

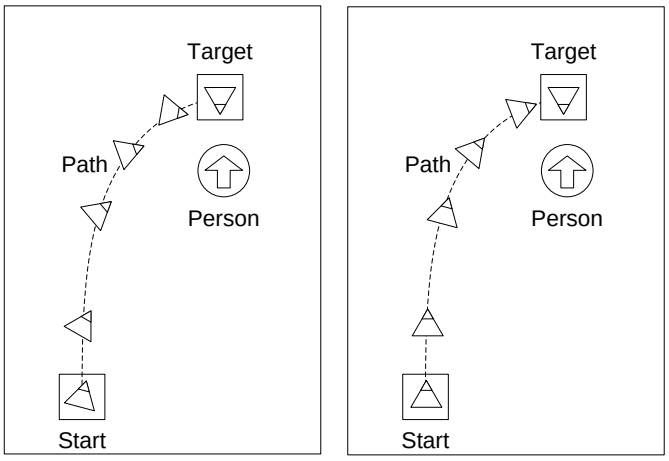
People intuitively perceive the dog's approach or avoidance, and expect interaction according to it. In case of robotics, this motion characteristic could be used to express that the robot expects getting a command from a specific person while doing its task and the robot is ready to process

the new task, or it could express the opposite, that the robot does not accept a command from a specific person. Therefore, we suggest assigning artificial values to people, which we have called people approach value. This value could be assigned to each person individually by a higher level algorithm. Based on that, our proposed path planning algorithm can behave towards different people using different degrees of approximation. Fig. 3 shows the effect of this parameter on the resulting path.

2.3 Target pointing orientation

Movement of animals is holonomic in the sense that animals can move to any direction and change orientation at the same time, independently. Angle difference between velocity and orientation often shows the intent or attention of the animal [2]. For example, Fig. 4(a) shows a typical path and orientation of a dog, which comes from behind a person to the person's front for initiating interaction, while Fig. 4(b) shows the same path without any difference between orientation and velocity direction.

Based on orientation information, people can get a concept about the target, attention, or future movement of the animal. Mobile robots could do the same to provide feedback to people about their objective and future movement. For example, if a robot is waiting for a person to initiate an interaction, it could indicate this intent by continuously facing that person even if the robot has to follow him or pass to the person's other side.



(a) (b)

Fig. 4. (a) A typical path and orientation of a dog, which comes from behind a person to the person's front for initiating interaction. Orientation and direction of movement are different, indicating the dog's objective. Fig. 4 (b) The same path in case of a not natural, non-holonomic movement, in which the movement of the dog would not indicate the robot's intent before reaching its target.

3. PROPOSED ALGORITHM

APF is based on the idea that the target attracts the robot, while the obstacles generate repulsive force fields in order to avoid collision. We propose a novel APF method in order to navigate mobile robots in unknown, dynamic environments, while adapting animal locomotion attributes (described in Section 2) including orientation information.

Contrary to conventional potential functions, our proposed repulsive potential function is not monotonic, which means that an obstacle can attract the robot depending on the distance and parameter range. However, for conventional reasons, the function of obstacles is named *repulsive function*; and the function of the target is called *attractive function*. We make the following assumptions:

Assumption 1. The robot geometry is approximated with a cylinder shape of radius r_{rob}.

Assumption 2. The position and velocity of both the robot and the target are well known.

Assumption 3. The position and velocity of obstacle surfaces which are not covered can be accurately measured on-line.

3.1 Attractive potential function

Conventional APF method defines attractive potential function as a function of the relative distance between the robot and the target. This conventional method cannot handle moving targets efficiently, and it cannot provide soft-landing solutions, consequently the robot reaches the target with non-zero velocity. An attractive potential function as stated in equation (1) was proposed [9] including velocity information for handling a moving target and providing soft landing.

$$U_{att}(x, \underline{v}_R, \underline{v}_T) = \alpha_x x(t)^m + \alpha_v \left\| \underline{v}_R(t) - \underline{v}_T(t) \right\|^n \qquad (1)$$

where $x(t)$ is the Euclidean distance between the robot and the target, $\underline{v}_R(t), \underline{v}_T(t)$ denote the velocity of the robot and the target at time t, respectively; $\left\| \underline{v}_R(t) - \underline{v}_T(t) \right\|$ is the magnitude of the relative velocity between robot and target; α_x, α_v are scalar positive parameters; and m, n are positive constants.

The main disadvantage of this attractive potential function from the point of view of animal motion planning is that the attraction force significantly depends on the distance to the target. It causes significantly different paths in case of middle- and long- distances to the target, which is not natural for animal motion. The extended attractive potential field is inspired by animal motion. The idea of taking velocity into account is adapted as follows:

$$U_{att}(x, \underline{v}_R, \underline{v}_T) = \frac{\alpha}{\beta^2} x + \frac{\alpha}{x + \beta} - \frac{\alpha}{\beta} + \gamma \left\| \underline{v}_R - \underline{v}_T \right\|^2 \qquad (2)$$

where α, γ are non-negative, β is a positive constant.

The term regarding velocity ($\gamma \left\| \underline{v}_R - \underline{v}_T \right\|^2$) is chosen to be quadratic, following the nature of robot kinematics energy, in order to avoid collisions and provide soft-landing efficiently.

It can be seen that $\frac{\alpha}{\beta^2} x + \frac{\alpha}{x+\beta} - \frac{\alpha}{\beta}$ and $\gamma \left\| \underline{v}_R - \underline{v}_T \right\|^2$ are both non-negative. This means that $U_{att}(x, \underline{v}_R, \underline{v}_T)$ approaches its minimum of zero if and only if both the relative velocity and the distance between the target and the robot approach zero.

Force vector function pointing from the robot to the target can be calculated by taking the derivative of the potential function:

$$F_{att}(x, \underline{v}_R, \underline{v}_T) = -\nabla U_{att}(x, \underline{v}_R, \underline{v}_T) =$$

$$\frac{\partial U_{att}(x,\underline{v}_R,\underline{v}_T)}{\partial x}\underline{n}_{RT}+\frac{\partial U_{att}(x,\underline{v}_R,\underline{v}_T)}{\partial(\underline{v}_R-\underline{v}_T)}\underline{n}_{VRT} \tag{3}$$

where $\underline{n}_{RT}$ is the unit vector pointing from the robot to the target and $\underline{n}_{VRT}$ denotes the unit vector pointing in the direction of the relative velocity of the robot with respect to the target.

$$\frac{\partial U_{att}(x,\underline{v}_R,\underline{v}_T)}{\partial x}=\frac{\alpha}{\beta^2}-\frac{\alpha}{(x+\beta)^2} \tag{4}$$

$$\frac{\partial U_{att}(x,\underline{v}_R,\underline{v}_T)}{\partial(\underline{v}_R-\underline{v}_T)}=2\gamma\left\|\underline{v}_R-\underline{v}_T\right\| \tag{5}$$

$$F_{att}(x,\underline{v}_R,\underline{v}_T)=\left(\frac{\alpha}{\beta^2}-\frac{\alpha}{(x+\beta)^2}\right)\underline{n}_{RT}+2\gamma\left\|\underline{v}_R-\underline{v}_T\right\|\underline{n}_{VRT} \tag{6}$$

The main advantage of the proposed attractive potential field is shown as solid line in Fig. 5: when the velocity is constant, the force field is almost constant in the cases that the robot is 'far' from the target. The force function in Fig. 5 is given by:

$$F_{att\,MAX}(\underline{v}_R,\underline{v}_T)=$$
$$=F_{att}(x,\underline{v}_R,\underline{v}_T)|_{x\to\infty}=\left|\frac{\alpha}{\beta^2}\underline{n}_{RT}+2\gamma\left\|\underline{v}_R-\underline{v}_T\right\|\underline{n}_{VRT}\right| \tag{7}$$

$$F_{att\,MIN}(\underline{v}_R,\underline{v}_T)=$$
$$=F_{att}(x,\underline{v}_R,\underline{v}_T)|_{x=0}=2\gamma\left\|\underline{v}_R-\underline{v}_T\right\| \tag{8}$$

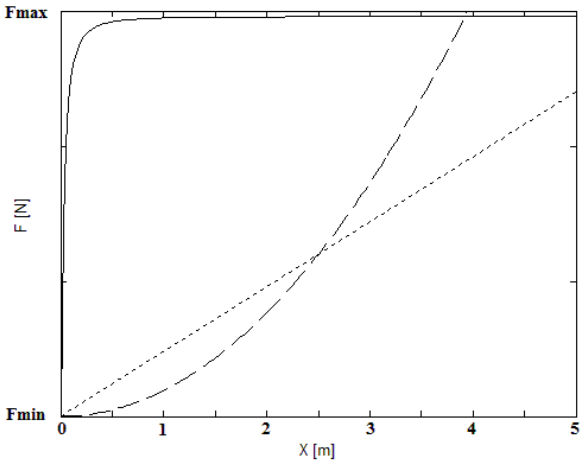


Fig. 5. The solid line shows the proposed attractive force function versus the corresponding values of robot-target distance, where the robot-target relative velocity is influencing the Fmax, Fmin parameters according to equations (7, 8). The dotted and the dashed lines respectively show the linear and exponential characteristics of attractive force functions widely used in the literature, e.g. equation (1). The main advantage of the proposed function is that the force field is almost constant, resulting in a path that is independent from the target distance when the robot is not close to the target, while the function still provides soft-landing.

3.2 Repulsive potential function

Implementing the object proximity-seeking property (Section 2.1, Fig. 2) is not possible with a monotonic repulsive potential field. In order to solve this problem while taking all mentioned animal motion attributes into account, we propose the repulsive potential function as follows:

$$U_{rep}(x,\underline{v}_R,\underline{v}_{OBS})=$$
$$\begin{cases}\delta\frac{1}{4}x^4-\delta\left(\frac{d_0+2d_1}{3}\right)x^3+\delta\left(\frac{d_1^2}{2}+d_0d_1\right)x^2- \\ -\delta d_0d_1^2x+\delta C+\zeta\frac{\left\|\underline{v}_R-\underline{v}_{OBS}\right\|^2}{2a_{MAX}}, if\ x\le d_1 \\ \zeta\frac{\left\|\underline{v}_R-\underline{v}_{OBS}\right\|^2}{2a_{MAX}}\quad, if\ x>d_1\end{cases} \tag{9}$$

where:

$$d_0=r_{sec}+r_{rob} \tag{10}$$

$$d_1=d_0+P\ O \tag{11}$$

$$C=d_0d_1^3\frac{1}{3}-d_1^4\frac{1}{12} \tag{12}$$

r_{sec} is a constant representing a security distance between the robot and an obstacle in order to avoid collision. r_{rob} is the radius of the robot assuming a cylinder-shaped robot. δ,ζ are non-negative constants, $\underline{v}_R,\underline{v}_{OBS}$ denote the robot and obstacle velocity vector respectively. The maximum acceleration of the robot is a_{max}. P and O are scalar parameters: P is related to people individually, meaning "People approach"; O is a parameter of robot behavior meaning "object proximity-seeking"; both as described in Section 2.

The effects of the parameters above mentioned and the advantage of the proposed potential function will be more visible by calculating the force function:

$$F_{rep}(x,\underline{v}_R,\underline{v}_{OBS})=-\nabla U_{rep}(x,\underline{v}_R,\underline{v}_{OBS})=$$
$$\frac{\partial U_{rep}(x,\underline{v}_R,\underline{v}_{OBS})}{\partial x}\underline{n}_{OR}+\frac{\partial U_{rep}(x,\underline{v}_R,\underline{v}_{OBS})}{\partial(\underline{v}_R-\underline{v}_{OBS})}\underline{n}_{VOR} \tag{13}$$

where $\underline{n}_{OR}$ is the unit vector pointing from an obstacle to the robot center and $\underline{n}_{VOR}$ denotes the unit vector pointing in the direction of the relative velocity of the robot with respect to the obstacle. The partial derivatives are:

$$\frac{\partial U_{rep}(x,\underline{v}_R,\underline{v}_{OBS})}{\partial x}=\delta x^3-\delta(d_0+2d_1)x^2+$$
$$+\delta(d_1^2+2d_0d_1)x+\delta d_0d_1^2=\delta(x-d_0)(x-d_1)^2 \tag{14}$$

$$\frac{\partial U_{rep}(x,\underline{v}_R,\underline{v}_{OBS})}{\partial(\underline{v}_R-\underline{v}_{OBS})}=\zeta\frac{\left\|\underline{v}_R-\underline{v}_{OBS}\right\|}{a_{MAX}} \tag{15}$$

Substituting (14) and (15) into (13) we have:

$$F_{rep}(x,\underline{v}_R,\underline{v}_{OBS})=$$
$$=\begin{cases}-\delta(x-d_0)(x-d_1)^2\underline{n}_{RO}+\zeta\frac{\left\|\underline{v}_R-\underline{v}_{OBS}\right\|}{a_{MAX}}\underline{\mathrm{n}}_{VRO} \\ \qquad\qquad\qquad\qquad, if\ x\le d_1 \\ \zeta\frac{\left\|\underline{v}_R-\underline{v}_{OBS}\right\|}{a_{MAX}}\underline{\mathrm{n}}_{VRO}, \qquad\qquad, if\ x>d_1\end{cases} \tag{16}$$

Solid line of Fig. 6 shows the shape of the proposed force function and its main parameters d_0,d_1. In case $x>d_1$, the artificial force function is not a function of x, it is affected only by relative obstacle-robot velocity marked on Fig. 6 and calculated as:

$$F_{lim}(\underline{v}_R, \underline{v}_{OBS}) = F(\underline{v}_R, \underline{v}_{OBS})|_{x>d_1} = \zeta \frac{\|\underline{v}_R - \underline{v}_{OBS}\|}{a_{MAX}} \underline{n}_{VRO} \quad (17)$$

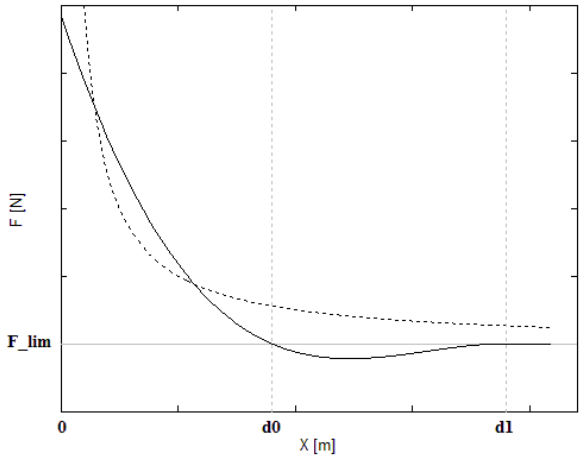


Fig. 6. The solid line shows the proposed repulsive force function as a function of the robot-obstacle distance. The dotted line shows the exponential characteristics of functions mainly used in the technical literature. The main advantage of the proposed function is that all attributes described in Section 2 can be adapted, including the attraction interval of repulsive force function (between d0 and d1), while the function is suitable for real-time control in a dynamic environment.

3.3 Control law

Velocity and angular velocity of holonomic robots can be controlled independently. Linear and angular motion influence the maximum limits of each other, but that is not in the scope of this article. We assume the maximum speed and the acceleration range of operation in which linear and angular motion limits can be achieved independently.

The forces are calculated in real-time as the sum of attractive function and repulsive functions based on equations (6) and (16) respectively.

$$F_{SUM} = F_{att}(x, \underline{v}) + \sum F_{rep}(x, \underline{v}) \quad (18)$$

The control method was designed to use 2D laser sensor data directly, and the obstacle surfaces were subdivided into surface units. As a consequence, it is considered that, for each sensor-visible (i.e. non-obstructed) obstacle surface, each surface unit of the obstacle will apply a repulsive force to the robot, according to equation (16).

Assuming that the mass of robot m_{rob} is known, the robot's acceleration can be determined by Newton's second law:

$$\ddot{x} = \frac{F_{SUM}}{m_{rob}} \quad (19)$$

The orientation reference is calculated by taking the target's position and the robot's actual velocity direction into account:

$$\varphi_{ref} = a\, \varphi_v + b\, \varphi_T \quad (20)$$

where φ_{ref} is the reference orientation of the robot; φ_T is the angle from the robot to the point called focus. The focus point is the point which the robot will be facing after it arrives at the target position and orientation (e.g. the point where the robot's "owner" is). φ_v is the orientation of the robot's actual velocity vector, and a, b denote the non-negative constants of weighted average. It is proposed that a, b be such that $a + b = 1$.

A simple torque function based on a P-type controller is proposed for adapting the orientation information as described in Section 2.3:

$$\underline{\tau} = \sigma(\varphi_{rob} - \varphi_{ref})\underline{n}_z \quad (21)$$

where $\underline{n}_z$ is the unit vector corresponding to the axis of rotation; φ_{rob} is the robot's orientation $(-\pi < \varphi_{rob} < \pi)$,$\sigma$ is a non-negative constant, and $\underline{\tau}$ denotes the artificial torque applied to the robot.

Assuming that the inertia of the robot I_{rob} is known, the robot's angular acceleration can be determined by Newton's second law:

$$\dot{\omega} = \frac{\underline{\tau}}{I_{rob}} \quad (22)$$

The path planner algorithm calculates the trajectory path, starting from the initial position with zero velocity and angular velocity, and then sequentially calculates the next position by taking the double integral of the acceleration (19) and angular acceleration (22).

3.4 Local minima problem

The APF method, like any local path planning algorithm, has the local minima problem. It means that, if there is a local minimum of the potential field and the robot stops there, the robot can never reach the target. APF based path planning has three basic cases of local minima as shown in Fig. 7.

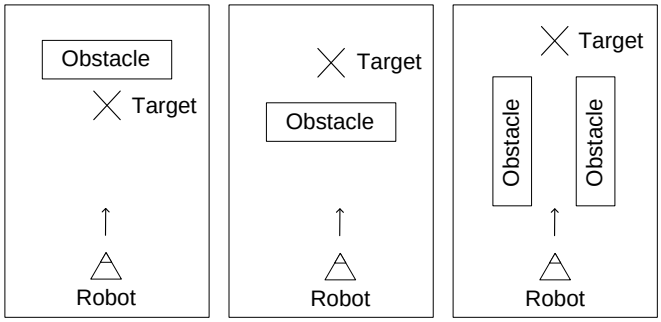


Fig. 7. Three basic cases of APF local minima, (a-c) from left to right, respectively. In case (a), the target is close to the repulsive filed of an obstacle. In case (b), there is an obstacle between the target and the robot. This case is also called a U-trap, because the robot often gets trapped in if the obstacle has a U shape. In case (c) there is a narrow passageway between two obstacles and the repulsive forces of the obstacles do not allow the robot to pass through them.

This problem can be solved by a global algorithm at a higher software layer. There are basically two kinds of solutions.

One group of possible solutions tries to generate potential field that is free of local minima [21], or modify the potential field in order to get out of the trap. This group of methods also includes those that add virtual obstacles or virtual targets in order to modify the potential field. These algorithms often need global information, and they may be computationally expensive, and they are usually not guaranteed to find a path even if it exists.

Another group of algorithms changes behavior when a local minimum is detected, and changes back to the APF method when escape conditions are met. Most of them are based on wall-following methods with sophisticated decision making and escape conditions [16].

The local minima problem is investigated and well-described in the technical literature [9–11,16,21], and also many corresponding articles were reviewed in [15]. Fig. 8 shows a block diagram of our robotic implementation, which is overlaid by a traditional wall-following method based on the BUG algorithm.

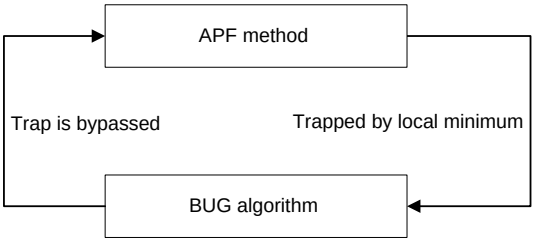


Fig. 8. Block diagram of APF method with a BUG algorithm at a higher layer. APF is changed to BUG algorithm in case APF stops at a local minimum. Path planning changes back to APF after the trap is bypassed.

When the robot stops at a local minimum, the potential of the local minimum is saved and the APF method is replaced by the BUG algorithm. In that mode, the robot goes straight to the target. If obstacles are in the way, they are bypassed by wall-following method, then the robot continues on a straight line to the target. The BUG algorithm changes back to APF when the potential of the actual position is lower than the potential saved in the local minimum (i.e. robot bypassed local minimum).

4. Experimental results

4.1 Simulation

Before real robotics experimentation, a simulation was implemented in MATLAB in order to tune parameters and evaluate results. The simulation calculates the potential and force functions proposed above, and calculates the path. The MATLAB simulation with graphical user interface was compiled to a Windows executable (exe). It can be downloaded from the link in the appendix. Robotics implementations of local path planning algorithms are usually based on distance sensors. The noise of these measurements are often not negligible. The distance measurement error of a laser scanner is usually under +/-5 mm, but low cost laser sensors have as high as about +/-25 mm deviation in the typical indoor distance range which is smaller than or equal to approximately 6 meters. In order to model the measurement error, white noise with zero mean and 25 mm deviation was added to the simulated distances.

An obstacle map can be loaded in the simulation program in square grayscale bitmap image form, which corresponds to a 10 meters square area. White pixels mean free area, while darker pixels represents obstacles. Grey pixels represent people, with different people approach values, as proposed in Section 2.2, and referred to as P in equation (11). In our simulation, one pixel in the map refers to a 10 cm square shape obstacle, which we call an "obstacle primitive". Each of the four sides of an obstacle primitive is called a "surface primitive". Virtual forces are generated by obstacle surface primitives, which are not covered by other obstacle surfaces. The virtual forces of obstacle primitives are calculated based on equation (7), and proportionally scaled by the surface primitive size depending on the view angle of the surface from the point of view of the robot.

Based on map and parameter settings, force and potential fields are calculated and shown on the simulator program window as Figs. 10-11, and Figs. 13-14. These representations correspond to zero robot velocity. At the next simulation step, the path is iteratively calculated from the start position according to the applied force of the APF algorithm. This calculation takes velocity information into account according to equations (8, 16). The calculated path is shown on the map, at the force field graph and the potential field graph as well.

4.1.1 Simulation 1. - Object proximity seeking

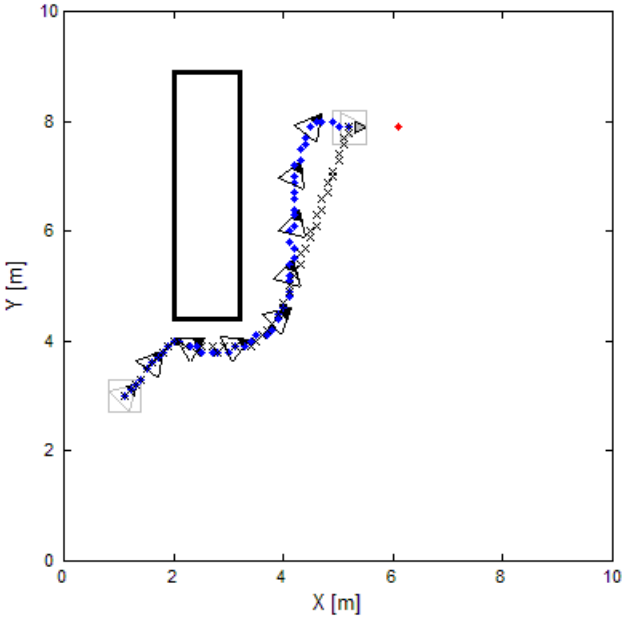


Fig. 9. Map and calculated path of the first scenario. The "diamond-shaped" dots show the path that demonstrates object proximity-seeking characteristics of the proposed algorithm, while the "crosses" show the resulting path of the traditional APF method. By comparing these results it is possible to see the obstacle-adjacent path of the proposed method, which was achieved by applying a non-monotone repulsive force (i.e. the obstacle attracts the robot in certain distance and velocity ranges).

Fig. 9 shows the map of the first scenario in order to present an object proximity-seeking adaptation for reflecting interaction avoidance. The resulting path of the traditional APF method is also presented for comparison.

The parameters were chosen as follows. Attractive APF function constants: $\alpha = 100, \beta = 1.2, \gamma = 4$; repulsive function constants: $\delta = 15, \zeta = 0.5, O = 1\ m$; robot parameters: $r_{rob} = 0.3\ m,\ r_{sec} = 0.6\ m, v_{max} = 1.5\ m/s,$ $a_{max} = 3\ m/s^2,\ m = 20\ kg$; and orientation constants: $a = 0.66,\ b = 0.34$. Figs. 10-11 show force field and potential field graphs, respectively, in the first test scenario.

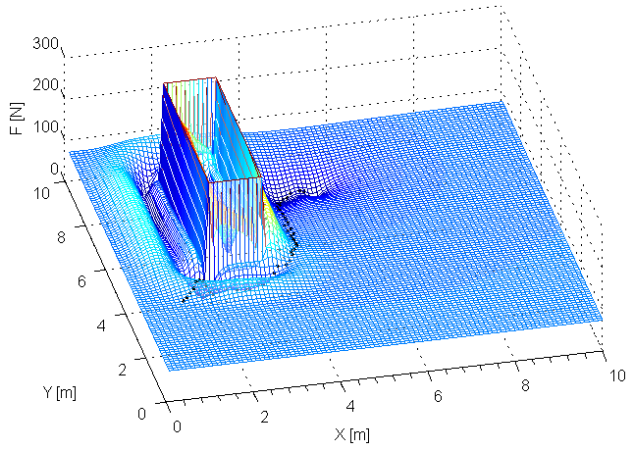


Fig. 10. Force field representation of object proximity-seeking scenario at zero velocity.

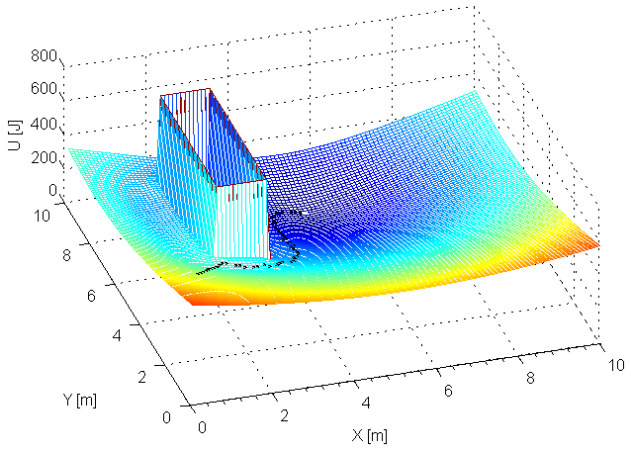


Fig. 11. Potential field representation of object proximity-seeking scenario at zero velocity.

The calculated path of the first scenario shows that object proximity-seeking characteristics described in section 2.1 were successfully implemented. The magnitude of these characteristics can be changed by adjusting the repulsive force function parameter O as shown in equations (11, 16). The distance between the robot and the obstacle can be also changed by the parameter r_{sec}.

4.1.2 Simulation 2. - People approach

Fig. 12 shows the map of the second scenario in order to demonstrate people approach characteristics. Both the path that results from our proposed method and a path computed by the traditional APF method are shown for comparison.

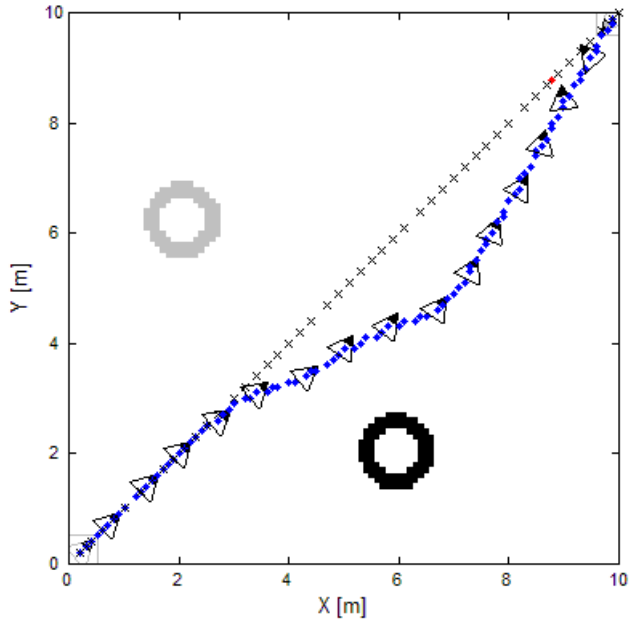


Fig. 12. Map and calculated path of the second scenario. The "diamond-shaped" dots show a path demonstrating people proximity characteristics of the proposed algorithm, while the "crosses" shows the resulting path of the traditional APF method. By comparing these results it is possible to see that the proposed method assigns different function parameters to different obstacles, resulting in a selective approach to different people.

The parameters of proposed method were set as follows. Attractive APF function constants: $\alpha = 100,\ \beta = 1.2,\ \gamma = 4$; repulsive function constants: $\delta = 3,\ \zeta = 0.5, O = 2\ m$; robot parameters: $r_{rob} = 0.3\ m,\ r_{sec} = 1\ m,\ v_{max} = 1.5\ m/s,\ a_{max} = 3\ m/s^2,\ m = 20\ kg$; and orientation constants $a = 0.22,\ b = 0.78$.

Individual people approach values are set as greyscale brightness on the bitmap of the simulation program (Fig. 12). Black means parameter P=1, while light grey (nearly white) would represent P=0. The evaluated map (Fig. 12) represents people with approach value 0.24 to the left, and 1 to the right. Figs. 13-14 shows force field and potential field graphs respectively.

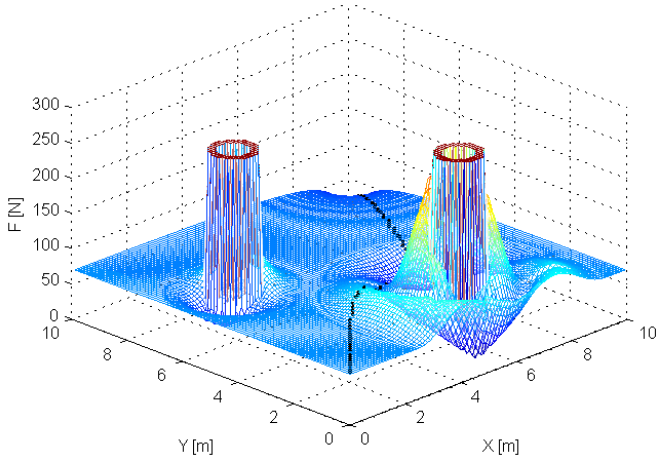


Fig. 13. Force field representation of people proximity characteristics evaluation scenario at zero velocity.

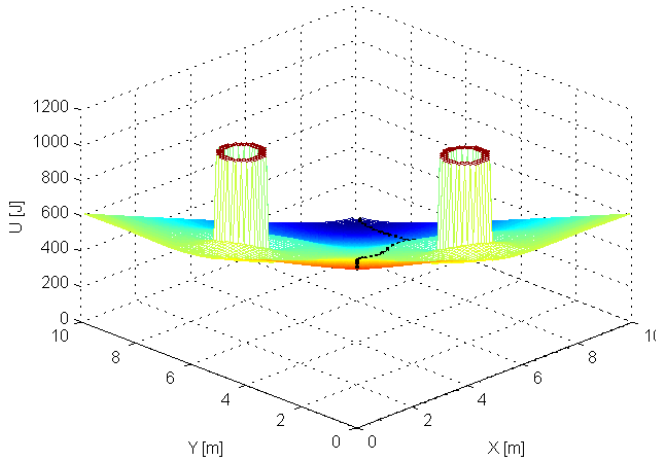
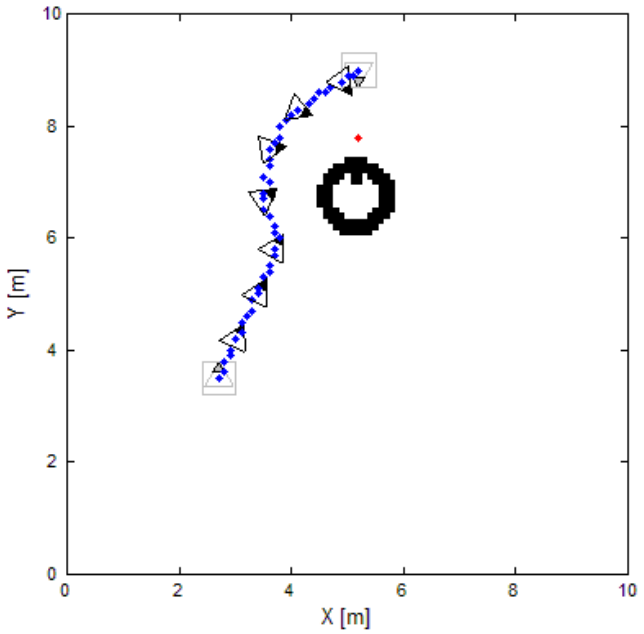


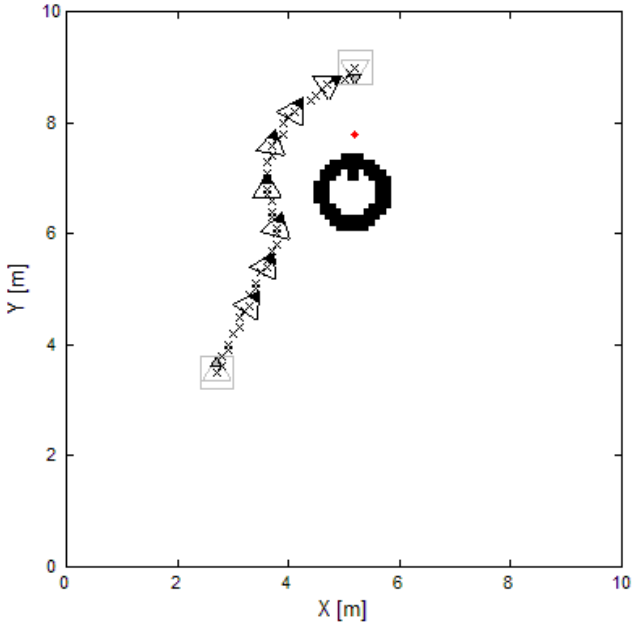
Fig. 14. Potential field representation of people proximity characteristics evaluation scenario at zero velocity.

4.1.3 Simulation 3. - Target pointing orientation

Fig. 15 (a) shows the map and the calculated path of a target pointing scenario. Fig. 15 (b) shows a traditional, non-holonomic path for comparison. Target pointing means that the robot is continuously considering a focus point, the point which the robot will be facing after it arrives at the target position and orientation (as explained in section 2.3). For simplification, the focus point was set in a way that, when the robot reaches the final position, the distance between robot and the focus point is equal to the robot's diameter. The focus point is indicated as a "diamond-shaped" dot on the map.



(a)



(b)

Fig. 15. Map and calculated path of the third scenario. Fig. (a) shows a path that includes orientation information for demonstrating target pointing characteristics of the proposed algorithm, while Fig. (b) shows the resulting path and orientation of the traditional APF method (where the orientation is defined as the direction of the velocity). By comparing these paths it is possible to see that the proposed method takes both velocity and target positon into consideration, providing feedback to people about target and future movement of the robot.

The parameters were chosen as follows. Attractive APF function constants: $\alpha = 100,\ \beta = 1.2,\ \gamma = 4$; repulsive function constants: $\delta = 15,\ \zeta = 0.5, O = 0.6\ m$; robot parameters: $r_{rob} = 0.3\ m,\ r_{sec} = 1.1\ m, v_{max} = 1.5\ m/s, a_{max} = 3\ m/s^2,\ m = 20\ kg$; and orientation constants $a = 0.02, b = 0.98$.

The simulations have proved that, with appropriate parameters, the proposed algorithm is capable to adapt the animal motion attributes described in section 2 to the robot's behavior.

4.1.4 Simulation 4. – Dynamic environment

This section evaluates the proposed method in two different dynamic scenarios. Fig 16 shows the first dynamic scenario with a single moving target. The target has a constant velocity of $v_{tar} = [0, -1.3]\ m/s$. The simulation results confirm that the attractive potential function described in Section 3.1 is capable to approach a dynamic target, i.e. the robot reaches the moving target without a collision with the target (with zero relative velocity).

The parameters were chosen as follows. Attractive APF function constants: $\alpha = 100,\ \beta = 1,\ \gamma = 14$; repulsive function parameters had no effect in this scenario; robot parameters: $r_{rob} = 0.3\ m,\ r_{sec} = 0.4\ m, v_{max} = 1.5\ m/s, a_{max} = 3\ m/s^2,\ m = 20\ kg$; and orientation constants $a = 0.13, b = 0.87$.

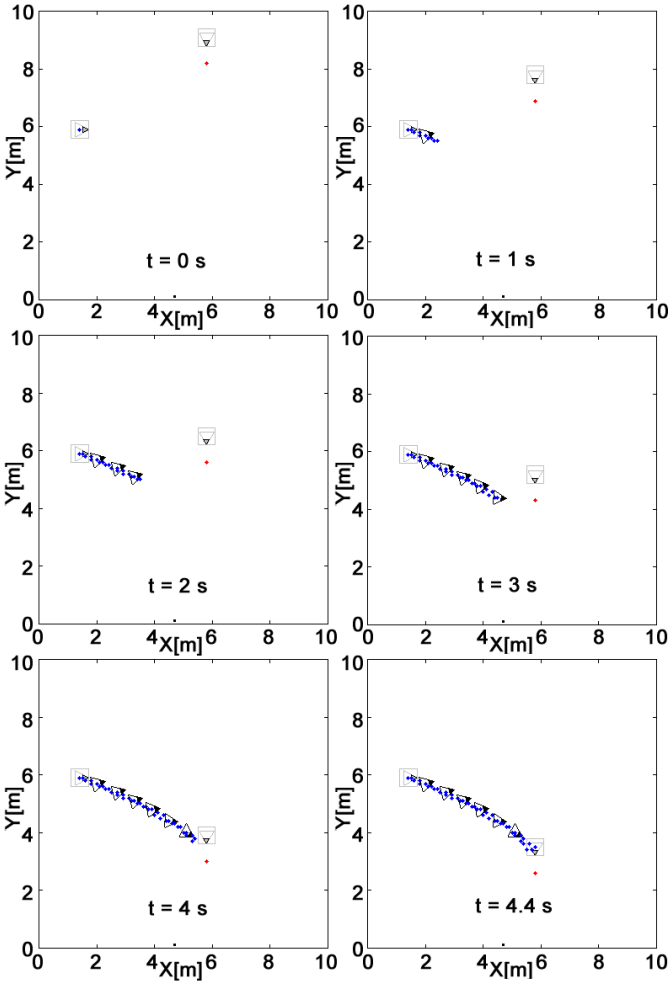


Fig. 16. Consecutive instants in the map and simulated path in the first dynamic scenario with a moving target.

Fig 17 represents the second dynamic scenario including a moving target and moving obstacles as well. The target has a constant velocity of $v_{tar} = [0, -0.4]\ m/s$, and all obstacles have constant velocity of $v_{obst} = [0,\ 0.5]\ m/s$. The simulation results show both the dynamic obstacle avoidance, and the moving target approach capability, using the proposed potential functions described in Section 3.1-3.2.

The parameters were chosen as follows. Attractive APF function constants: $\alpha = 100,\ \beta = 1,\ \gamma = 14$; repulsive function constants: $\delta = 60,\ \zeta = 50, O = 0.6\ m$; robot parameters: $r_{rob} = 0.3\ m,\ r_{sec} = 0.4\ m, v_{max} = 2\ m/s, a_{max} = 3\ m/s^2,\ m = 20\ kg$; and orientation constants $a = 0.1, b = 0.9$.

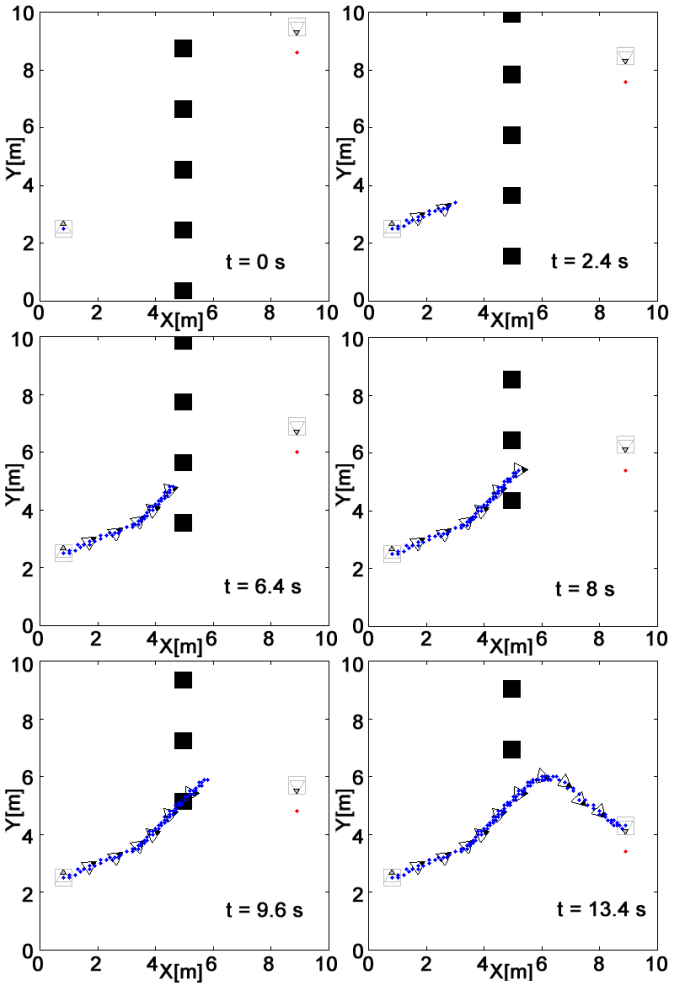


Fig. 17. Consecutive instants in the map and simulated path in the second dynamic scenario with a moving target and moving obstacles.

4.2 Robotic implementation and measurement

Based on simulation results, robotic implementation was developed and evaluated on MOGI-ETHON (Fig. 1), a holonomic drive mobile robot [22]. The proposed algorithm needs real-time information on the robot, the obstacles, and the target motion states. Therefore MOGI-ETHON was equipped with a RPLIDAR 360° type laser scanner.

The proposed algorithm assumes that the position and the velocity of obstacle surface primitives which are not covered by other obstacles can be accurately measured on-line. The positions of surface primitives were approximated from two consecutive range measurements with a small 1° angle difference. From that the average distance and size of the surface is calculated for further calculation of virtual force. Using laser range finder in static environment (e.g. to measure the position of an obstacle) is simple, but using range finders to determine the velocity of obstacle surface primitives is a challenging task, which needs to find matching points of obstacles in consecutive range measurements. Our article assumes that it is given, and does not deal with this problem. Obstacles in our dynamic experiment measurements were moving with constant velocity, which was previously given to the algorithm.

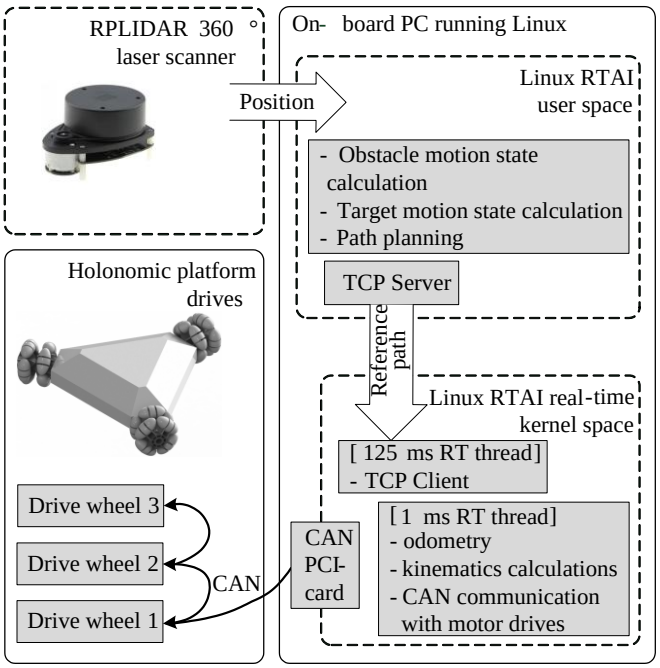


Fig. 18. Block diagram of the implementation. Distances were measured by a RPLIDAR 360° type laser scanner, calculations were performed in Linux RTAI real-time operating system, and DC motors with custom drives are moving the holonomic robot platform.

The architecture of the robot can be seen on Fig. 18. An on-board Linux computer reads the sensor data every 180 ms triggered by Linux RTAI timers. Path planning calculations are processed in user space at the same rate. The calculated paths are sent to a 1 ms robot kinematics thread, which is responsible for kinematics calculations, odometry, and communication with the motor drives. PCI motion control card [23] provides interface between Linux RTAI and DC motor drives via CAN BUS.

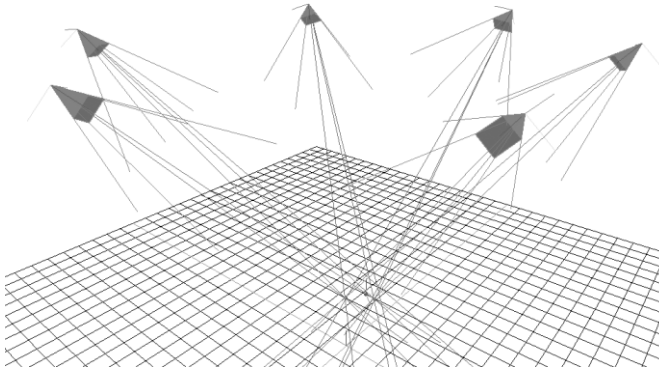


Fig. 19. Schematics representation of cameras in the MOCAP motion tracking vision system.

Robot path was measured with the latest version of MOCAP [24] six camera motion capture system. Fig. 19 represents the camera setup in the MOCAP system. The position of the cameras are well known. From each camera a three-dimensional ray can be calculated with the camera focus as initial-point. Theoretically, the position of a marker in three dimensions is calculated with triangulation as the intersection of these rays.

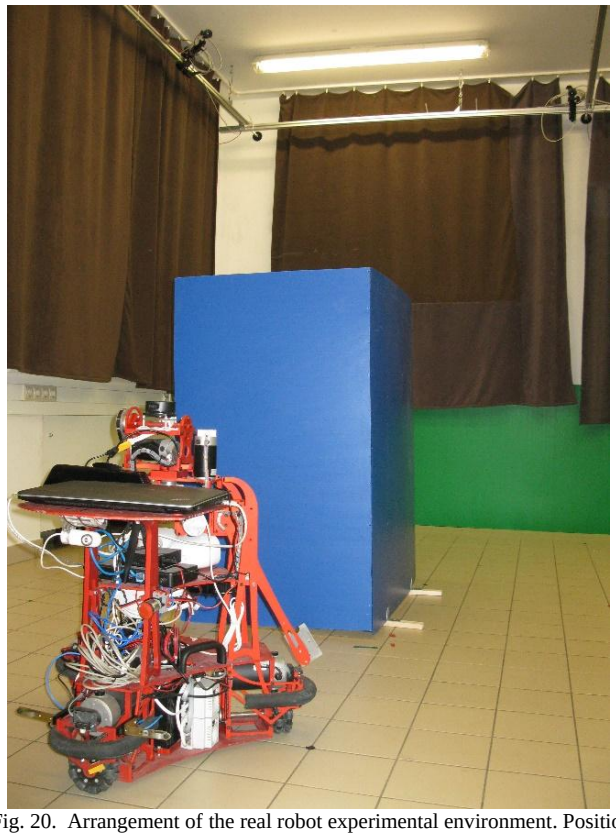


Fig. 20. Arrangement of the real robot experimental environment. Position and orientation of the robot were tracked by a six camera vision system. Two of the cameras can be seen at the upper corners of the picture.

Fig. 20 shows the experimental environment. Start and target position were given in a similar manner to Figs. 2, 9. The parameters were chosen as follows. Attractive APF function constants: $\alpha = 120, \beta = 1.2, \gamma = 4$; repulsive function constants: $\delta = 15, \zeta = 1.2, O = 1.2\ m$; robot parameters: $r_{rob} = 0.3\ m, r_{sec} = 0.5\ m, v_{max} = 1.5\ m/s, a_{max} = 3\ m/s^2,\ m = 20\ kg$; and orientation constants: $a = 0.02, b = 0.98$.

Two lighting optical markers were placed on the robot in order to track both the position and the orientation. The measured path from the vision system was imported to MATLAB in order to compare simulation and real environment results as shown in Fig. 21.

The most significant error is an accumulative offset, which is caused by robot positioning error. Robot position and orientation are calculated by an odometry method. Odometry uses data from the wheel encoders to estimate changes in position over time. It has an accumulative error, which is mainly caused by wheel slip. This error can be suppressed by periodically calibrating the robot position with laser sensors or vision system.

Other significant errors have nondeterministic nature, mainly caused by laser sensor noise. Fig. 22 shows position and orientation errors of measurement.

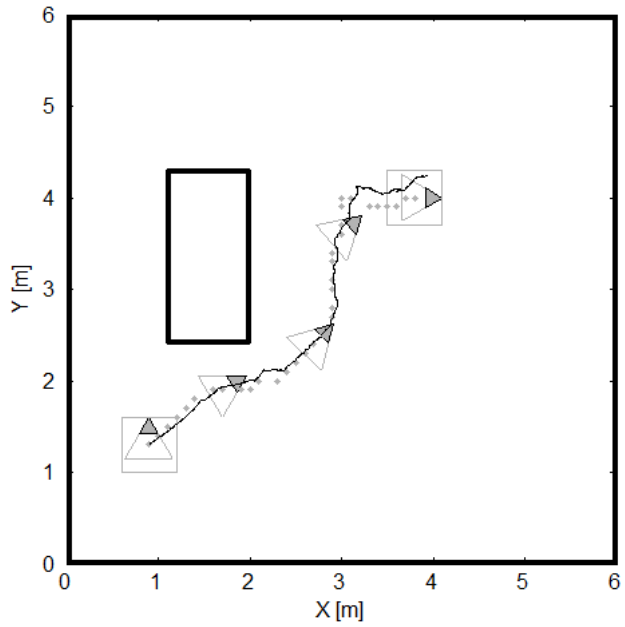
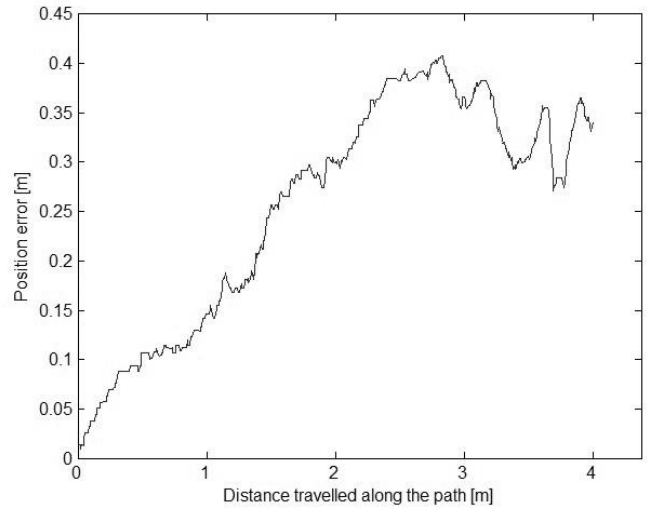
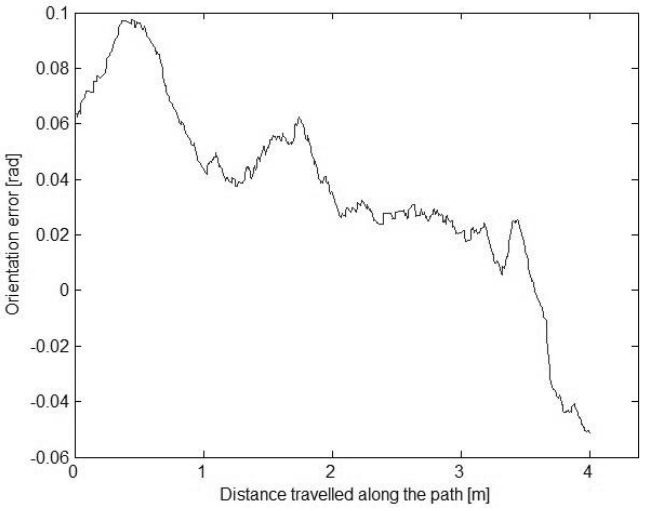


Fig. 21 Measured path and simulated path. The dotted grey line is the simulated path and the continuous black line shows the measured path.



(a)



(b)

Fig. 22 Absolute (a) position, and (b) orientation errors along the path.

The camera vision system also has inaccuracy, but it has smaller magnitude. The theoretically intersecting rays calculated from the cameras are not intersecting, they are skew due to camera position errors, limited picture resolution, and camera distortion. This problem is solved by a marker positioning algorithm, which approximates the marker position as the closest point approached by these rays.

Three more measurements were taken on the same path with similar results. The two-dimensional correlation coefficient of x-y coordinates between simulated and real paths is above 99%, thus the robotics implementation validates the simulated results.

5. CONCLUSIONS

In this paper a novel motion planning method based on artificial potential fields (APF) was proposed. The main contribution of this paper is the definition of a path planning method by implementing animal motion attributes as new elements for the locomotive behavior of artificial mobile robots. The main advantage of the proposed method is that, by adapting natural motion attributes of animals to robot motion, the human-robot interaction becomes much more natural and intuitive. It also becomes easy to understand the robot's current state and future intentions.

The experimental implementation confirmed the simulation results and demonstrated that the simulated ideas are feasible in real robots. Future work could extend this concept to include more animal motion attributes to robot motion. Another possible future development direction is the application of our proposed method on robots with animal inspired kinematics, together with the animal-specific trajectory control proposed in [18]. This is a promising line of research in the field of consumer mobile robotics and an important step into the evolution of human-robot interaction.

ACKNOWLEDGEMENT

The project acknowledges the financial support of Hungarian Research Fund (OTKA K100951).

This work is connected to the scientific program of the " Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Hungary Development Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

APPENDIX

The MATLAB simulation source code, and also the compiled Windows executable (.exe) is available as supplementary material on www.sciencedirect.com, or it can be downloaded directly from authors website: www.generalmechatronics.com/doc/publ/PathPlanning_Elsevier2015.zip

REFERENCES

[1] K.L. Koay, G. Lakatos, D.S. Syrdal, M. Gacsi, B. Bereczky, K. Dautenhahn, et al., Hey! There is someone at your door. A hearing robot using visual communication signals of hearing dogs to

communicate intent, Artif. Life (ALIFE), 2013 IEEE Symp. (2013) 90–97. doi:10.1109/ALIFE.2013.6602436.

[2] M.G. Tamás Faragó, Ádám Miklósi, Beáta Korcsok, Judit Száraz, Social behaviours in dog-owner interactions can serve as a model for designing social robots, Interact. Stud. (2014) 143–172.

[3] F. Tajti, G. Szayer, B. Kovács, M.A.P. Burdelis, Mobile Robot Performance Analysis for Indoor Robotics, Period. Polytech. Civ. Eng. (2015) 123–131.

[4] P.S. Xuan Zou, Bin Ge, Improved Genetic Algorithm for Dynamic Path Planning, Int. J. Inf. Comput. Sci. IJICS. 1 (2012) 16–20.

[5] K.I. Tsianos, I. a. Sucan, L.E. Kavraki, Sampling-based robot motion planning: Towards realistic applications, Comput. Sci. Rev. 1 (2007) 2–11. doi:10.1016/j.cosrev.2007.08.002.

[6] J. Ni, X. Li, X. Fan, A Dynamic Risk Level Based Bioinspired Neural Network Approach for Robot Path Planning, in: World Autom. Congr., 2014: pp. 829 – 833. doi:10.1109/WAC.2014.6936167.

[7] N.A. Mai, K. Janschek, Generic system architecture for behavior-based mobile robot control using fuzzy logic, 2012 Int. Conf. Control. Autom. Inf. Sci. (2012) 253–258. doi:10.1109/ICCAIS.2012.6466598.

[8] K. Charalampous, I. Kostavelis, A. Gasteratos, Thorough robot navigation based on SVM local planning, Rob. Auton. Syst. 70 (2015) 166–180. doi:10.1016/j.robot.2015.02.010.

[9] S. Ge, Y. Cui, Dynamic motion planning for mobile robots using potential field method, IEEE Mediterr. Conf. Control Autom. (2002) 207–222. http://www.springerlink.com/index/J6148PQ6724Q7871.pdf.

[10] T. Zhang, Y. Zhu, J. Song, Real-time motion planning for mobile robots by means of artificial potential field method in unknown environment, Ind. Robot An Int. J. 37 (2010) 384–400. doi:10.1108/01439911011044840.

[11] M. a P. Castañeda, J. Savage, A. Hernández, Local Autonomous Robot Navigation using Potential Fields, Motion Plan. (2008) 1–23. doi:10.5772/6022.

[12] J.A. Villacorta-Atienza, V.A. Makarov, Neural network architecture for cognitive navigation in dynamic environments, IEEE Trans. Neural Networks Learn. Syst. 24 (2013) 2075–2087. doi:10.1109/TNNLS.2013.2271645.

[13] J.-W. Lee, J.-J. Kim, J.-J. Lee, Improved Ant Colony Optimization Algorithm by Potential Field Concept for Optimal Path Planning, Humanoid Robot. (2008) 662–667. doi:10.1109/ICHR.2008.4756022.

[14] Z.S.Z. Shaogang, L.M.L. Ming, Path Planning of Inspection Robot Based on Ant Colony Optimization Algorithm, Electr. Control Eng. (ICECE), 2010 Int. Conf. (2010) 1474–1477. doi:10.1109/iCECE.2010.1438.

[15] P. Raja, S. Pugazhenthi, Optimal path planning of mobile robots: A review, Int. J. Phys. Sci. 7 (2012) 1314–1320. doi:10.5897/IJPS11.1745.

[16] Y. Zhu, T. Zhang, J. Song, An Improved Wall-Following Method for Escaping from Local Minimum in Artificial Potential Field Based Path Planning, Control. (2009) 6017–6022.

[17] A. Cherubini, R. Passama, P. Fraisse, A. Crosnier, A unified multimodal control framework for human-robot interaction, Rob. Auton. Syst. 70 (2015) 106–115. doi:10.1016/j.robot.2015.03.002.

[18] C. Panagiotakis, G. Tziritas, Snake terrestrial locomotion synthesis in 3D virtual environments, Vis. Comput. 22 (2006) 562–576. doi:10.1007/s00371-006-0035-1.

[19] D. Lipkind, A. Sakov, N. Kafkafi, G.I. Elmer, Y. Benjamini, I. Golani, New replicable anxiety-related measures of wall vs center behavior of mice in the open field., J. Appl. Physiol. 97 (2004) 347–359. doi:10.1152/japplphysiol.00148.2004.

[20] A. Kerepesi, A. Dóka, Á. Miklósi, Dogs and their human companions: The effect of familiarity on dog-human interactions, Behav. Processes. 110 (2014) 27–36. doi:10.1016/j.beproc.2014.02.005.

[21] J.-O. Kim, P.K. Khosla, Real-time obstacle avoidance using harmonic potential functions, Proc. 1991 IEEE Int. Conf. Robot. Autom. 8 (1992) 338–349. doi:10.1109/70.143352.

[22] F. Tajti, G. Szayer, B. Kovács, P. Korondi, Robot base with holonomic drive, in: Int. Fed. Autom. Control 19th World Congr., 2014: pp. 5715–5720. doi:10.3182/20140824-6-ZA-1003.00785.

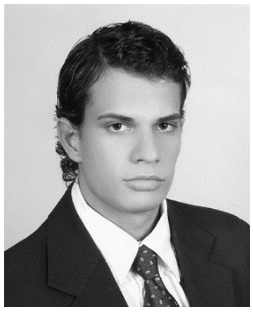
[23] B. Kovács, G. Szayer, F. Tajti, Design of a universal robot controller, Period. Polytech. Mech. Eng. 55 (2011) 95–100. doi:10.3311/pp.me.2011-2.06.

[24] V. Devecseri, A. Dóka, J. Molnár, P. Tamás, An ethological motion capture system, 12th IEEE Int. Symp. Comput. Intell. Informatics, CINTI 2011 - Proc. (2011) 487–491. doi:10.1109/CINTI.2011.6108555.

Bence Kovács received MSc. degree in 2011 in Mechatronics at Budapest University of Technology and Economics. Before applying to the PhD course of the Department of Mechatronics Optics and Engineering Informatics at the Faculty of Mechanical Engineering of BME, he had worked for two years in the Budapest R&D Center of Ericsson. During this time, he worked as software developer, then as hardware developer of high speed network

systems. He established his own company—General Mechatronics Ltd.—in 2012 together with Géza Szayer.

Géza Szayer has received MSc. in Mechatronics at Budapest University of Technology and Economics (BME). He had worked for 1.5 years in the Budapest R&D Center of Knorr-Bremse as an embedded software engineer before applying to the PhD course of the Department of Mechatronics Optics and Engineering Informatics at the Faculty of Mechanical Engineering of BME. He established his own company—General Mechatronics Ltd.—in 2012 together with Bence Kovács.



Ferenc Tajti has received MSc. in the field of Mechatronics at Budapest University of Technology and Economics. Since then he has been a Ph.D. student at the Department of Mechatronics Optics and Mechanical Engineering Informatics. Since 2012, he has been working at the MTA-ELTE Comparative Ethological Research Group (MTA 01 031) of the Hungarian Academy of Sciences.

Mauricio Burdelis has received a Ph.D. in engineering from the Nara Institute of Science and Technology, Japan, in 2013. He has a MSc. in computer engineering from the Polytechnic School of the University of Sao Paulo, Brazil, in 2009. Since 2015 he has been working as a Senior Lecturer at the Budapest University of Technology and Economics (formerly the Technical University of Budapest).

Péter Korondi (M'98-SM'11) received the Dipl. Eng. and Ph.D. degrees in electrical engineering from the Technical University of Budapest, Budapest, Hungary, in 1984 and 1995, respectively.

Since 1986, he has been with Budapest University of Technology and Economics (formerly the Technical University of Budapest). From April 1993 to April 1995, he worked in the laboratory of Prof. Harashima and Prof. Hashimoto at the Institute of Industrial Science, The University of Tokyo, Japan, where he continues to spend a month each year, working on joint research. As a result of this cooperation, the Intelligent Integrated System Japanese–Hungarian Joint Laboratory was founded in 2001. His research interests include telemanipulation and motion control. Dr. Korondi is a founding member of the International PEMC Council and Senior Member of IEEE.

HIGHLIGHTS

· Animal motion attributes were adapted to mobile robot path planning.
· The aim of the proposed method is to assist human-robot interaction.
· The proposed method is capable of on-line navigation in dynamic environments.
· Simulations were implemented and are accessible online for evaluation.
· The robotics implementation confirmed the simulation results.