Reactive Magnetic-field-inspired Navigation for Non-holonomic Mobile Robots in Unknown Environments*

Ahmad Ataka[1], Hak-Keung Lam[1], and Kaspar Althoefer[2]

***Abstract*— In this paper, we present a reactive robot navigation method for a non-holonomic mobile robot taking inspiration from the phenomena observed in magnetic fields. The algorithm is shown to be able to guide mobile robots in arbitrary-shaped convex environment without being trapped in local minima by exploiting the local sensory information without priori knowledge about the environment. A preliminary validation study involving simulation of and experiments with a TurtleBot mobile robot platform show the advantage of the proposed method over existing ones.**

I. Introduction

In recent years, robotics applications have moved from static, well-defined environments of factory floors to environments that are dynamic, often unknown environment and where the close proximity or even physical interaction between robots and humans are unavoidable. This new way of making use of robots has led to a shift of paradigm in many areas of robotics, including the field of robot navigation. The classic planning technique [1], [2], which aims to find a geometrical collision-free path from initial configuration to the final configuration under the assumption of perfect environmental knowledge, is no longer sufficient. The use of reactive navigation, which attempts on simultaneously solving planning and control problem on-line using feedback signals and sensory information, is preferable since it does not rely on a priori information of obstacles, avoids computationally expensive and time-consuming calculations, and hence, makes it perfectly suited for unknown and unpredictable environment with strong real-time constraints.

One of the most popular examples of reactive navigation is the electric-field-inspired Artificial Potential Field (APF) method [3]. The APF method uses the gradient of a potential function as a feedback signal to guide the robot's movement, whereby obstacles show a repelling behavior while the goal has an attractive influence on the robot. Despite its simplicity, the APF method and a lot of its variances (such as those reported in [4]-[5]) suffer from local minima where the robot gets stuck in the saddle points of the potential [6].

*This work was supported in part by King's College London, the EPSRC in the framework of the NCNR (National Centre for Nuclear Robotics) project (EP/R02572X/1), and the Indonesia Endowment Fund for Education, Ministry of Finance Republic of Indonesia.

[1]A. Ataka and H.K. Lam are with the Centre for Robotics Research (CoRe), Department of Informatics, King's College London, WC2R 2LS, United Kingdom. (Corresponding author e-mail: ahmad_ataka_awwalur.rizqi@kcl.ac.uk)

[2]K. Althoefer is with the Centre for Advanced Robotics @ Queen MAry (ARQ), Faculty of Science and Engineering, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom.

On attempting to solve the local minimum problem while maintaining the reactive property, the use of a vortex field, in which the vector field is designed to circulate rather than to emanate outwardly from the obstacle, was proposed for mobile robot [7] and multiple flying robots [8]. These methods, however, do not present a collision avoidance or goal convergence guarantee in the presence of arbitrary-shaped obstacles. A recent reactive navigation method used the power diagram to design a globally stable feedback planner with unique attractors in each of the saddle points besides the final goal position [9]. Nonetheless, this approach is only suitable for a topologically simple spherical environment.

The idea of creating a vector field which deviates rather than repels the robot has motivated the use of artificially-induced magnetic fields for mobile robots [10]-[11] and navigation methods for rigid-link manipulators [12]. Attempts to make this type of vector field reactive were made, for example by using a 'gyroscopic force' employed to guide a planar point-like robot for boundary following [13], [14], obstacle avoidance [15] and formation control of multiple vehicles [16]. Efforts to extend these works into 3D environments have also been made recently for obstacle avoidance purpose [17] and formation control [18]-[20]. Albeit successfully creating a vector field without a local minimum, some of the works still need a priori information, such as the obstacle's geometry [10]-[11] and location of its center point [12], or these approaches are only applicable to certain specific shapes of obstacles [17]. Meanwhile, other navigation methods could only be used for specific purposes such as boundary following [13], [14], or collision avoidance among point-like agents [18]-[20], incapable of coping with environments with large and arbitrary-shaped obstacles.

In this paper, a magnetic-field-inspired reactive navigation is presented. Relying only on a local sensory information, a robot will induce an artificial electric current on the surface of the closest obstacle, which in turn will produce an artificial magnetic field with the ability to alter the robot's direction of motion. The algorithm is shown to work without affecting the robot's energy, and hence, leave the global convergence property of the attractive field to the goal unchanged. Moreover, obstacle avoidance is guaranteed for any types of convex-shaped obstacles. This paper focuses on applying the basic navigation algorithm to the case of a non-holonomic mobile robot (TurtleBot) with limited sensor field of view.

II. Motivating Phenomena

The physical phenomena which motivated the creation of this algorithm can be described by the theory of classical

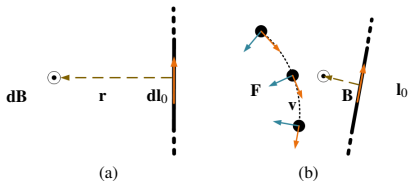


Fig. 1. (a) A current-carrying wire creates a magnetic field in the surrounding space. (b) The magnetic field induced by electrical current affects the motion of a moving charged particle.

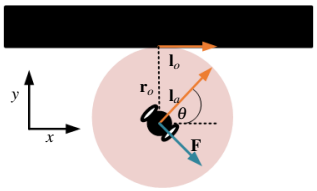


Fig. 2. The non-holonomic mobile robot with 360° range sensor (magenta area) in the vicinity of flat-surfaced obstacle (drawn in black).

electromagnetism. According to Biot-Savart law, an electrical current i_o flowing along a straight wire with infinitesimal length $\mathbf{dl}_o$ produces magnetic field $\mathbf{dB}$ in a point whose position denoted by $\mathbf{r}$ with respect to the wire, as illustrated in Fig. 1a. The corresponding magnetic field is given by

$$\mathbf{dB} = \frac{\mu_0}{4\pi}\frac{i_o \mathbf{dl}_o \times \mathbf{r}}{|\mathbf{r}|^3}, \tag{1}$$

where μ_0 stands for a permeability constant while $\times$ stands for the vector cross product operation.

As shown in Fig 1b, this magnetic field will influence the movement of a charged particle q, producing a force $\mathbf{F}$ which is perpendicular to both the magnetic field $\mathbf{B}$ and the particle's velocity $\mathbf{v}$ as follows

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B}. \tag{2}$$

Abandoning the infinitesimal notation in (1), the force, popularly known as Lorentz force, can be expressed as follows

$$\mathbf{F} = \frac{\mu_0 q i_o}{4\pi}\frac{\mathbf{v} \times (\mathbf{l}_o \times \mathbf{r})}{|\mathbf{r}|^3}. \tag{3}$$

As illustrated in Fig. 1b, this type of force works by altering the particle's direction of motion, in contrast to the repelling nature of an electric field. Hence, applying this characteristic to the problem of robot navigation, a robot can be seen as a charged particle whereas the obstacle surface can be viewed as a current-carrying wire. The produced force can be regarded as a vector field which works as a control signal to control the robot's motion in avoiding the obstacle.

Applying this perspective, a robot located at position $\mathbf{p}$ will induce an artificial current $\mathbf{l}_o$ on the closest obstacle surface located at vector position $\mathbf{r}_o$ with respect to the robot, which in turn produces the artificial magnetic field $\mathbf{B}$ with the ability to affect the robot's movement by a force $\mathbf{F}$. Considering the fact that $\mathbf{r}_o = -\mathbf{r}$, the force in (3) can be rewritten to be

$$\mathbf{F} = c\ \ \mathbf{l}_a \times (\mathbf{r}_o \times \mathbf{l}_o)\ \ f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|), \tag{4}$$

where $c > 0$ is a positive constant, $\mathbf{l}_a$ denotes the robot's direction, while $f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) \geq 0$ is a function of robot-to-obstacle distance $|\mathbf{r}_o|$ and the robot's speed $|\dot{\mathbf{p}}|$.

III. Magnetic-field-inspired Navigation

Suppose we have a point-like non-holonomic mobile robot in $\mathbb{R}^2$ with a 360° range sensor moving in proximity to the flat-surfaced obstacle as illustrated in Fig. 2. For the robot to avoid collision, the desired direction of motion is chosen to be parallel to the obstacle's surface. Among the two possible choices, the artificial current direction is chosen to be the one with the smallest angle to the robot's velocity direction $\mathbf{l}_a$ to minimize oscillation due to the change in the direction of motion. To fulfill this requirement, the artificial induced current $\mathbf{l}_o$ is chosen to be the projection of the robot's direction of motion $\mathbf{l}_a$ on to the closest obstacle surface, as depicted in Fig. 2. This can be accomplished using the following expression

$$\mathbf{l}_o = \mathbf{l}_a - \frac{(\mathbf{l}_a^T \mathbf{r}_o)\mathbf{r}_o}{|\mathbf{r}_o|^2}. \tag{5}$$

In order to force the robot to follow the desired direction in (5), the original force equation in (4) is changed to

$$\mathbf{F} = c\ \ \mathbf{l}_a \times (\mathbf{l}_o \times \mathbf{l}_a)\ \ f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|), \tag{6}$$

where the robot's movement's direction is defined as $\mathbf{l}_a = \frac{\dot{\mathbf{p}}}{|\dot{\mathbf{p}}|}$.

Lemma 1: The vector field $\mathbf{F}$ in (6) does not change the mobile robot's linear speed $v = |\dot{\mathbf{p}}|$, but will affect the angular speed whose magnitude is described as follows

$$\omega_a = \frac{|\mathbf{F}|}{mv}, \tag{7}$$

in which m denotes the mass of the robot.

Proof: Without loss of generality, we can safely assume that $cf(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) = 1$. Suppose we define $\mathbf{l}_o \times \mathbf{l}_a = \mathbf{a}$, by definition of vector product operation, we get

$$\mathbf{l}_a^T \mathbf{F} = \mathbf{l}_a^T (\mathbf{l}_a \times \mathbf{a}) = 0. \tag{8}$$

The dynamics of the robot is expressed as

$$\mathbf{F} = m\frac{d\dot{\mathbf{p}}}{dt} = m(\frac{dv}{dt}\mathbf{l}_a + v\frac{d\mathbf{l}_a}{dt}). \tag{9}$$

Substituting the vector field $\mathbf{F}$ to (8) and the fact that $\mathbf{l}_a^T \frac{d\mathbf{l}_a}{dt} = 0$ will lead to $\mathbf{l}_a^T \mathbf{F} = m\frac{dv}{dt} = 0$, which will converge to the rate of speed equals zero ($\frac{dv}{dt} = 0$), meaning that the linear speed is constant. The only component of vector field $\mathbf{F}$ relates to the change of direction given by $\mathbf{F} = mv\frac{d\mathbf{l}_a}{dt}$. The robot's angular speed is defined as the rate of change of the robot's direction $\mathbf{l}_a$, i.e. $\omega_a = \frac{d\mathbf{l}_a}{dt}$. Thus, we can easily conclude that $\omega_a = \frac{|\mathbf{F}|}{mv}$. ■

Remark 1: The fact that the vector field $\mathbf{F}$ does not change the speed v leads to an important feature: the vector field has no effect on the robot's energy. Thus, the proposed vector field $\mathbf{F}$ can be applied in addition to any attractive field with a globally stable property without losing its stability property.

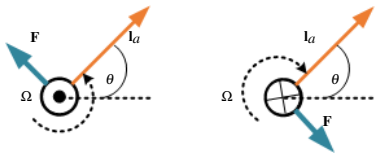


Fig. 3. The diagram showing how the vector field $\mathbf{F}$ (blue arrow) relates to the velocity $\mathbf{l}_a$ (orange arrow) and angular speed Ω.

Lemma 2: The vector field $\mathbf{F}$ will change the mobile robot's heading angle with the rate of change described as

$$\dot{\theta} = \frac{|\mathbf{F}|}{mv}\Omega_z, \tag{10}$$

where Ω_z is a component of $\Omega = \begin{bmatrix}0 & 0 & \Omega_z\end{bmatrix}^T = \frac{\mathbf{l}_a \times \mathbf{F}}{|\mathbf{l}_a \times \mathbf{F}|}$.

Proof: Lemma 1 gives us an expression for the angular speed ω_a in (7). From the kinematic model of a mobile robot, we can easily conclude that this will serve as the magnitude of the heading angle's rate of change $|\dot{\theta}|$. For the scenario illustrated in Fig. 2, we can observe that the produced vector field $\mathbf{F}$, which is perpendicular to the velocity direction $\mathbf{l}_a$, tends to decrease the heading angle θ, yielding a negative rate of change. Hence, we can conclude that the sign of $\dot{\theta}$ relates to the vector cross product operation between robot direction $\mathbf{l}_a$ and vector field $\mathbf{F}$, which is defined as a unit vector $\Omega = \frac{\mathbf{l}_a \times \mathbf{F}}{|\mathbf{l}_a \times \mathbf{F}|}$ as illustrated in Fig. 3. As a consequence of considering a $\mathbb{R}^2$ environment, the only non-zero component of Ω is Ω_z and with the help of illustration in Fig. 2 and Fig. 3, we can easily conclude that $\text{sgn}(\dot{\theta}) = \text{sgn}(\Omega_z)$. We can complete the description of $\dot{\theta}$ by weighting the magnitude of ω_a in (7) by Ω_z and hence, completing the proof. ■

Lemma 3: Assuming that the robot does not hit an obstacle, the vector field $\mathbf{F}$ with the direction described in (6) will change the robot's direction of movement such that its final direction is parallel to the artificial current $\mathbf{l}_o$.

Proof: From the scenario illustrated in Fig. 2, we could express the closest obstacle point's location with respect to the robot as $\mathbf{r}_o = \begin{bmatrix}0 & r_o\end{bmatrix}^T$ and the robot's velocity direction as $\mathbf{l}_a = \begin{bmatrix}\cos\theta & \sin\theta\end{bmatrix}^T$. The induced artificial current in (5) can then be written as $\mathbf{l}_o = \begin{bmatrix}\cos\theta & 0\end{bmatrix}^T$. After setting the scalar $f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) = 1$, the vector field $\mathbf{F}$ can be expressed as

$$\mathbf{F} = c\begin{bmatrix}\sin^2\theta\cos\theta & -\sin\theta\cos^2\theta\end{bmatrix}^T, \tag{11}$$

where the magnitude can be simply written as $|\mathbf{F}| = c\sin\theta\cos\theta$. From the angular speed expression in (10), we can write

$$\dot{\theta} = -\frac{c}{mv}\sin\theta\cos\theta. \tag{12}$$

To get the final angle θ_f, we integrate the above equation until infinite time as follows

$$\int_{\theta_0}^{\theta_f} \frac{1}{\cos\theta\sin\theta}d\theta = -\int_0^{\infty}\frac{c}{mv}dt, \tag{13}$$

which will yield to $\theta_f = 0$, meaning that the robot's direction of movement is parallel to the obstacle surface and artificial current $\mathbf{l}_o$. ■

To ensure obstacle avoidance, the scalar function $f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|)$ is designed to be as follows

$$f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) = \frac{|\dot{\mathbf{p}}|}{|\mathbf{r}_o|}. \tag{14}$$

Lemma 4: The vector field $\mathbf{F}$ will avoid the robot to touch any convex-shaped obstacle surface as long as the robot's initial direction $\mathbf{l}_a$ is not perpendicular to the obstacle surface.

Proof: Suppose we start with simple scenario as illustrated in Fig. 2. Combining (5), (6), and (14), the vector field magnitude can be written as

$$|\mathbf{F}| = \frac{cv}{r}\sin\theta\cos\theta, \tag{15}$$

where $r = |\mathbf{r}_o|$. From (10) and the fact that $\Omega_z = -1$, the heading angle's rate of change is expressed as

$$\dot{\theta} = -\frac{c}{mr}\sin\theta\cos\theta. \tag{16}$$

Focusing on movements along the y-axis, the rate of decrease in distance r is equal to the velocity component in y direction

$$v\sin\theta = -\dot{r} \Leftrightarrow \frac{dr}{d\theta}\dot{\theta} = -v\sin\theta. \tag{17}$$

Combined with (16) and the fact that the closest distance to the obstacle occurs when $\dot{r} = -v\sin\theta = 0$, meaning that $\theta = 0$ for $\theta \in \begin{bmatrix}0 & \frac{\pi}{2}\end{bmatrix}$, the integration can be simplified to

$$\int_{r_0}^{r_f}\frac{1}{r}dr = \frac{mv}{c}\int_{\theta_0}^{0}\frac{1}{\cos\theta}d\theta, \tag{18}$$

in which r_0 and r_f stand for the initial and final distance between the robot and the obstacle respectively while θ_0 stands for the initial angle between the robot's direction of movement and the obstacle surface. This will lead to

$$r_f = \frac{r_0}{|\sec\theta_0 + \tan\theta_0|^C}. \tag{19}$$

where $C = \frac{mv}{c}$. The final distance $r_f \geq 0$ can then only be zero when the initial angle $\theta_0 = \frac{\pi}{2}$, i.e. when the robot's direction of movement is exactly perpendicular to the obstacle surface. Apart from this case, the robot's final distance to the flat-surfaced obstacle will never be zero, and this will also be the case for any convex-shaped obstacle due to its convexity. ■

Lemma 5: The final robot's direction of movement presented in Lemma 3 ($\theta_f = 0$) is globally asymptotically stable.

Proof: Suppose the positive semi-definite Lyapunov function candidate $V \geq 0$ is defined as

$$V = -\ln(\cos\theta), \tag{20}$$

where $\theta \in \begin{bmatrix}0 & \frac{\pi}{2}\end{bmatrix}$. Substituting $\dot{\theta}$ from (16), we get

$$\dot{V} = -\frac{c}{mr}\sin^2\theta. \tag{21}$$

Recalling the fact that $c > 0$, $m > 0$, and $r > 0$ according to Lemma 4, thus the condition $\dot{V} \leq 0$ will always hold. Globally stable equilibrium occurs when $V = \dot{V} = 0$, which only happens when $\theta = \theta_f = 0$, and hence, concluding the proof. ■

Remark 2: All the features presented in Lemma 3-5 are true under the assumption that the robot has a necessary energy to produce the desired control signal.

IV. Implementation on TurtleBot

A non-holonomic mobile robot is modeled as

$$\begin{aligned}\dot{\mathbf{p}} &= v\begin{bmatrix}\cos\theta & \sin\theta\end{bmatrix}^T \\ \dot{\theta} &= \omega\end{aligned}. \tag{22}$$

The input for the robot consists of linear and angular speed defined as $\mathbf{u} = \begin{bmatrix} v & \omega \end{bmatrix}^T$. The linear speed is influenced by the goal located at $\mathbf{p_g}$ as follows

$$v = \begin{cases} K_P\delta & \text{if } |\mathbf{p}-\mathbf{p_g}| \geq \delta \\ K_P|\mathbf{p}-\mathbf{p_g}| & \text{if } |\mathbf{p}-\mathbf{p_g}| < \delta \end{cases}, \tag{23}$$

where $K_P > 0$ denotes a proportional constant.

The angular speed will be influenced by both the goal and the closest obstacle. The angular speed terms induced by an obstacle is given in (10) as $\omega_o = \frac{|\dot{\mathbf{F}}|}{mv}\Omega_z$. The vector field $\mathbf{F}$ is chosen to influence the robot only when the robot is closer than distance r_l from the obstacle as follows

$$\mathbf{F} = \begin{cases} c\ \mathbf{l}_a \times (\mathbf{l}_o \times \mathbf{l}_a)\ \frac{|\dot{\mathbf{p}}|}{|\mathbf{r}_o|} & \text{if } |\mathbf{r}_o| < r_l \\ \mathbf{0} & \text{if } |\mathbf{r}_o| \geq r_l \end{cases}. \tag{24}$$

To guide the robot towards the direction of the goal without suffering from parametric singularity, the geometric control term, modified from work shown in [21], is employed as follows

$$\omega_g = -K_\omega \log(R_e), \tag{25}$$

where $K_\omega \geq 0$ stands for a constant and $R_e \in SO(2)$ is defined as an error matrix $R_e = R_g^T R$. $R \in SO(2)$ and $R_g \in SO(2)$ represent the homogeneous transformation matrices of the current heading angle θ and the desired goal angle $\theta_g = \text{atan2}(y-y_g, x-x_g)$. The operator $\log(g)$ for any $g \in SO(2)$ is defined as $\log(g) = \frac{\alpha}{2\sin\alpha}(g-g^T)$, where $\alpha = \arccos(\frac{\text{tr}(g)}{2})$. The total angular speed can be expressed as $\omega = \omega_g + \omega_o$.

The artificial current in (5) works under the assumption that the robot has a 360° laser sensor. Suppose we have a robot whose sensor field of view is characterized by angle ϕ as illustrated in Fig. 4a. To estimate the obstacle's surface, we chose n number of closest points on the obstacle sensed by the robot. For practical purpose, these points are chosen such that they are separated from each other by a minimum distance d to make sure that all n points are distinct. These points are then used to estimate the unit vector $\mathbf{s}_o$ describing the obstacle's surface direction as follows

$$\begin{aligned} b &= \frac{\sum_{i=0}^{n} x_i y_i - n\bar{x}_i\bar{y}_i}{\sum_{i=0}^{n} x_i^2 - n\sum_{i=0}^{n}\bar{x}_i^2}, \\ \mathbf{s}_o &= \begin{bmatrix} \frac{1}{\sqrt{1+b^2}} & \frac{b}{\sqrt{1+b^2}} \end{bmatrix}^T, \end{aligned} \tag{26}$$

where x_i and y_i denote the component of vector $\mathbf{r}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T \in \mathbf{R_s}$, $\mathbf{R_s} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \dots & \mathbf{r}_n \end{bmatrix}$ represents the set of positions of n closest sensed points, while $\bar{x}_i$ and $\bar{y}_i$ denote their average position, all expressed in global frame. The artificial current and closest distance to the obstacle can be expressed as

$$\begin{aligned} \mathbf{l}_o &= (\mathbf{l}_a^T \mathbf{s}_o)\mathbf{s}_o, \\ r &= \min_{\forall \mathbf{r}_i \in \mathbf{R_s}} |\mathbf{r}_i - \mathbf{p}|. \end{aligned} \tag{27}$$

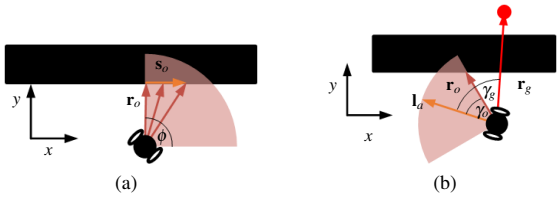


Fig. 4. (a) The diagram showing the way to estimate the obstacle surface $\mathbf{s}_o$, used as artificial current, for the case of limited sensor's field of view. (b) The angle γ_g and γ_o between the robot's direction $\mathbf{l}_a$ to a robot-to-goal vector $\mathbf{r}_g$ and robot-to-obstacle vector $\mathbf{r}_o$ respectively.

Another modification is made to take into account the special case when the robot's direction of motion is almost perpendicular to the obstacle surface. The artificial current in (27) denoted by $\mathbf{l}_{o,i}$ is modified as follows

$$\mathbf{l}_o = \begin{cases} \mathbf{l}_{o,i} & \text{if } |\mathbf{l}_{o,i}| > \varepsilon \\ \frac{\mathbf{l}_{o,i}}{|\mathbf{l}_{o,i}|} & \text{if } |\mathbf{l}_{o,i}| \leq \varepsilon \end{cases}, \tag{28}$$

in which ε denotes a positive constant.

The last modification relates to the case in which the robot's boundary-following movement is disturbed by the goal attraction caused by high difference in the robot's heading angle θ and the desired angle θ_g. This could cause an unwanted oscillation, force the robot body to be too close to an obstacle, and potentially lead to a collision. To solve this problem, we employ an angular speed relaxation to decrease the constant K_ω in (25) such that the attractive field contribution to the angular speed will decrease as the robot gets closer to the obstacle while the goal is still occluded by the obstacle. This is realized using the following equation

$$K_\omega = K_0(1-e^{-\frac{r_o}{r_c}})(\frac{1}{1+e^{\upsilon w}}), \tag{29}$$

$$w = \sin\gamma_g \sin\gamma_o = (\frac{\mathbf{l}_a \times \mathbf{r}_g}{|\mathbf{r}_g|})^T(\frac{\mathbf{l}_a \times \mathbf{r}_o}{|\mathbf{r}_o|}), \tag{30}$$

where $K_0 > 0$, $r_c > 0$, $\upsilon > 0$ stand for positive constants, $\mathbf{r}_g = \mathbf{p_g} - \mathbf{p}$ is a goal position with respect to the robot, and w denotes the weight relating to whether the goal is occluded by obstacle. γ_g and γ_o denote the angle between the robot's direction of motion $\mathbf{l}_a$ to vector $\mathbf{r}_g$ and $\mathbf{r}_o$ respectively as depicted in Fig. 4b. The sign of the angles are determined from the cross product operation in (30). It is noted that eq. (29)-(30) are only used when the distance to the obstacle is smaller than a limit r_l.

V. Results and Analysis

The proposed algorithm is implemented on a model of a planar mobile robot, specifically a TurtleBot robot platform equipped with a gyroscope sensor, odometry and a 3D sensor camera; the robot with all its sensors and actuators has been modelled in the Gazebo simulator. The experimental setup places the robot in the corridor environment with convex obstacle as shown in Fig. 5. The performance of the proposed algorithm are compared to the APF [3], circular field (CF)

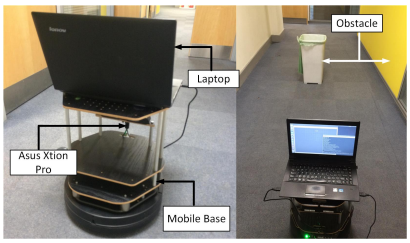
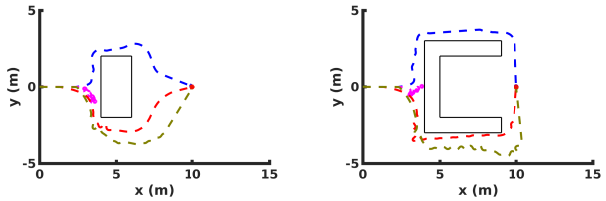


Fig. 5. The robot configuration is shown on the left and the experimental setup is shown on the right.



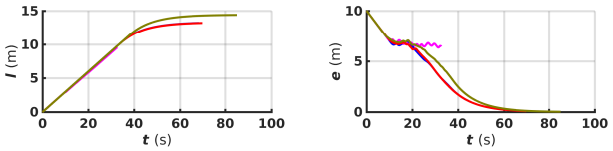
(a) (b)

Fig. 6. The robot's trajectory towards the goal (red dot) for (a) rectangular and (b) an N-shaped obstacle (black). The blue, magenta, red, yellow denote the proposed algorithm, APF, CF, and GF respectively.

[12], and gyroscopic force (GF) method [15]. All simulation and experimental scenarios are implemented in the Robot Operating System (ROS) framework [22]. The parameters are chosen to be $K_P = 0.1$, $c = 2$, $\varepsilon = 0.01$, $r_c = 1.5$ m, $\upsilon = 10$, $m = 1$ kg,$\delta = 3$ m, and $K_0 = 1$. For simulation: $r_l = 2$ m and $d = 0.0001$ m; for experiment: $r_l = 0.8$ m and $d = 0.01$ m. A video attachment is also available in http://ieeexplore.ieee.org.

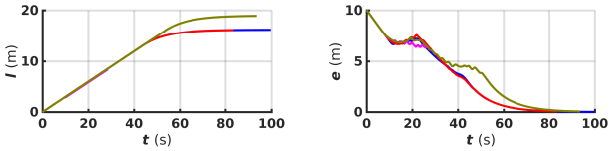
Fig. 6 shows the robot's path for a rectangular (Fig. 6a) and an N-shaped obstacle (Fig. 6b). In Fig. 6a, we can see that the proposed algorithm (blue line), the CF method (red line), and the GF method (yellow line) successfully navigate the robot towards the goal, while the APF method (magenta line) is unable to overcome the local minimum in the saddle point of the potential function. The shape of the obstacle and the position of the goal for scenario shown in Fig. 6b enable us to better observe whether the robot could move out from the boundary following movement once the goal is not occluded by a nearby obstacle. We can see that the proposed algorithm (blue line), the CF method (red line), and the GF method (yellow line) successfully navigate the robot towards the goal (red dot), while the APF (magenta line) is unable to overcome the local minimum. This also indicates that the proposed angular speed relaxation described in eq. (29)-(30) enables the robot to quickly move out from the boundary-following motion due to the bigger contribution of the go-to-goal angular speed once the goal is not occluded by obstacle.

In Fig. 7, the plot of the covered path (Fig. 7a) and the distance to the goal (Fig. 7b) are shown for the rectangular obstacle. We can observe that the algorithms with the shortest trajectory and the shortest time to reach the goal are the proposed algorithm (blue line) and the CF method (red line).



(a) (b)

Fig. 7. The plot shows (a) the covered path $l(t)$ and (b) the distance to the goal $e(t)$ for rectangular obstacle. The blue, magenta, red, yellow denote the proposed algorithm, APF, CF, and GF respectively.



(a) (b)

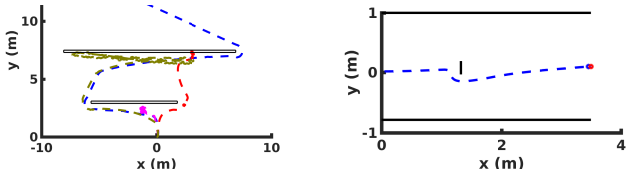
Fig. 8. The plot shows (a) the covered path $l(t)$ and (b) the distance to the goal $e(t)$ for an N-shaped obstacle. The blue, magenta, red, yellow denote the proposed algorithm, APF, CF, and GF respectively.

For an N-shaped obstacle, in Fig. 8, we also see that in terms of the covered path (Fig. 8a) and the time to reach the goal (Fig. 8b), the proposed algorithm and CF have a better performance when compared to the other methods. While the CF method still relies on a priori information of the position of the obstacle's center point, the proposed algorithm works without any prior knowledge about the environment and performs as well as the CF method.

In Fig. 9a, we introduce a corridor-like environment consisting of two long walls. The APF method (magenta line) fails to even circulate the first wall due to the local minimum problem. We can see that the CF (red line) and GF (yellow line) methods also fail to reach the goal. The CF method fails to avoid part of the robot's body from collision as the robot travels too far from the goal in following the obstacle's boundary. For the GF method, the long trajectory could bring the robot too close to the obstacle, causing the robot to sharply change its course away from the obstacle surface and reverse its direction altogether. This could lead to the condition where the robot travels back and forth along the wall without being able to reach the goal. Hence, in this type of environment, the proposed algorithm becomes the only one which can navigate the robot towards the goal.

In Fig. 9b, we show the trajectory of the real TurtleBot in the experimental setup shown in Fig. 5 where the environment consists of walls in the left and right side of the robot and a bin as an arbitrary-shaped obstacle. We can see that the proposed algorithm is able to navigate the robot without colliding with surrounding environment towards the goal.

We present the summary of results in Table I. The comparison is made in terms of the ability of the algorithm to successfully navigate the robot towards the goal, the length of the traveled path and the time to reach a goal. The robot is assumed to reach the goal once its distance to the goal equals 5% of the initial distance to the goal. From the table, we can conclude that the proposed magnetic-field-inspired



(a) (b)

Fig. 9. (a) The robot's path for corridor-like simulated environment. The blue, magenta, red, yellow denote the proposed algorithm, APF, CF, and GF respectively. (b) The trajectory of real TurtleBot (dashed blue line) moving towards the goal (red dot) with simplified representation of an arbitrary-shaped obstacle bin and walls.

TABLE I
RESULTS COMPARISON

Obstacle	Algorithm	Success	Covered Path (m)	Time (s)
Rectangle	APF	×	-	-
	CF	✓	13.21	52.15
	GF	✓	14.40	55.83
	MFI	✓	13.22	52.14
N-shaped	APF	×	-	-
	CF	✓	16.17	61.93
	GF	✓	18.91	71.02
	MFI	✓	16.23	61.81
Corridor	APF	×	-	-
	CF	×	-	-
	GF	×	-	-
	MFI	✓	34.82	121.93

(MFI) algorithm outperforms the APF since the latter always fail to navigate the robot towards the goal in the selected environments. The proposed algorithm also betters the GF in terms of the faster speed to reach the goal and shorter trajectory. Compared to the CF algorithm, the proposed algorithm produces similar performance in terms of time to the goal and length of trajectory. However, our algorithm does so without any priori knowledge of the environment. Our algorithm also becomes the only successful method for corridor-like environment. For the scenario of unknown arbitrary-shaped convex obstacle, our algorithm is also able to guide the real TurtleBot platform towards the goal.

VI. CONCLUSION

The paper presents a magnetic-field-inspired reactive navigation method for non-holonomic mobile robot navigating unknown arbitrary-shaped convex environment. The superiority of our method compared to the standard APF method is represented by the characteristics of the vector field which does create only one global minimum, the goal, and otherwise does not create local minima. The proposed algorithm has outperformed the previous magnetic-field-inspired algorithm in terms of its reactive property and ability to work in the absence of a priori information. The simulation and experimental results confirm the performance of the algorithm for the case of TurtleBot platform with its limited sensor's field of view. Future works will focus on the extension for the case of non-convex obstacle and to take into account the saturation in the the robot's actuators.

REFERENCES

[1] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[2] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, Jun. 2005.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Mar. 1985, pp. 500–505.

[4] C.-C. Wang, V. Kumar, and G.-M. Chiu, "A motion control and obstacle avoidance algorithm for hyper-redundant manipulators," in *Proc. Int. Symp. Underwater Technology*, Apr. 1998, pp. 466–471.

[5] D. H. Park, H. Hoffmann, P. Pastor, and S. Schaal, "Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields," in *Proc. IEEE-RAS Int. Conf. Humanoid Robots*, Dec. 2008, pp. 91–98.

[6] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom.* IEEE, 1991, pp. 1398–1404.

[7] A. D. Luca and G. Oriolo, "Local incremental planning for nonholonomic mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1994, pp. 104–110 vol.1.

[8] A. A. A. Rizqi, A. I. Cahyadi, and T. B. Adji, "Path planning and formation control via potential function for UAV Quadrotor," in *Proc. Int. Conf. Adv. Robot. Intell. Syst.*, Jun. 2014, pp. 165–170.

[9] O. Arslan and D. E. Koditschek, "Exact robot navigation using power diagrams," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 1–8.

[10] L. Singh, H. Stephanou, and J. Wen, "Real-time robot motion control with circulatory fields," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, Apr. 1996, pp. 2737–2742 vol.3.

[11] L. Singh, J. Wen, and H. Stephanou, "Motion planning and dynamic control of a linked manipulator using modified magnetic fields," in *Proc. IEEE Int. Conf. Control Applicat.*, Oct. 1997, pp. 9–15.

[12] S. Haddadin, R. Belder, and A. Albu-Schffer, "Dynamic motion planning for robots in partially unknown environments," in *IFAC World Congr.*, vol. 18, 2011.

[13] F. Zhang, E. Justh, and P. Krishnaprasad, "Boundary following using gyroscopic control," in *Proc. IEEE Conf. Decision Control*, vol. 5. IEEE, 2004, pp. 5204–5209.

[14] F. Zhang, A. O'Connor, D. Luebke, and P. Krishnaprasad, "Experimental study of curvature-based control laws for obstacle avoidance," in *IEEE Int. Conf. Robot. Autom.*, vol. 4. IEEE, 2004, pp. 3849–3854.

[15] D. E. Chang and J. E. Marsden, "Gyroscopic forces and collision avoidance with convex obstacles," in *New trends in nonlinear dynamics and control and their applications*. Springer, 2003, pp. 145–159.

[16] D. E. Chang, S. C. Shadden, J. E. Marsden, and R. Olfati-Saber, "Collision avoidance for multiple agent systems," in *Proc. IEEE Conf. Decision Control*, vol. 1, Dec. 2003, pp. 539–543 Vol.1.

[17] G. Garimella, M. Sheckells, and M. Kobilarov, "A stabilizing gyroscopic obstacle avoidance controller for underactuated systems," in *Proc. IEEE Conf. Decision Control*, Dec. 2016, pp. 5010–5016.

[18] E. W. Justh and P. Krishnaprasad, "Equilibria and steering laws for planar formations," *Syst. & control Lett.*, vol. 52, no. 1, pp. 25–38, 2004.

[19] L. Sabattini, C. Secchi, and C. Fantuzzi, "Collision avoidance using gyroscopic forces for cooperative Lagrangian dynamical systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 953–958.

[20] ——, "Collision avoidance for multiple Lagrangian dynamical systems with gyroscopic forces," *Int. J. Adv. Rob. Syst.*, vol. 14, no. 1, p. 1729881416687109, 2017. [Online]. Available: http://dx.doi.org/10.1177/1729881416687109

[21] F. Bullo and R. M. Murray, "Proportional Derivative (PD) Control On The Euclidean Group," in *In European Control Conference*, 1995, pp. 1091–1097.

[22] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System," in *ICRA workshop on open source software*, vol. 3, 2009, p. 5.