frontiers in Robotics and AI

ORIGINAL RESEARCH
published: 18 February 2022
doi: 10.3389/frobt.2022.834177

Magnetic-Field-Inspired Navigation for Robots in Complex and Unknown Environments

Ahmad Ataka[1]*, *Hak-Keung Lam*[2] *and Kaspar Althoefer*[3]

[1]*Signal Processing Laboratory, Department of Electrical and Information Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia,* [2]*The Centre for Robotics Research (CoRe), Department of Engineering, King's College London, London, United Kingdom,* [3]*The Centre for Advanced Robotics @ Queen Mary (ARQ), Faculty of Science and Engineering, Queen Mary University of London, London, United Kingdom*

Over the course of the past decade, we have witnessed a huge expansion in robotic applications, most notably from well-defined industrial environments into considerably more complex environments. The obstacles that these environments often contain present robotics with a new challenge - to equip robots with a real-time capability of avoiding them. In this paper, we propose a magnetic-field-inspired navigation method that significantly has several advantages over alternative systems. Most importantly, *1*) it guarantees obstacle avoidance for both convex and non-convex obstacles, *2*) goal convergence is still guaranteed for point-like robots in environments with convex obstacles and non-maze concave obstacles, *3*) no prior knowledge of the environment, such as the position and geometry of the obstacles, is needed, *4*) it only requires temporally and spatially local environmental sensor information, and *5*) it can be implemented on a wide range of robotic platforms in both 2D and 3D environments. The proposed navigation algorithm is validated in simulation scenarios as well as through experimentation. The results demonstrate that robotic platforms, ranging from planar point-like robots to robot arm structures such as the Baxter robot, can successfully navigate toward desired targets within an obstacle-laden environment.

Keywords: magnetic-field-inspired navigation, reactive navigation, motion control, path planning for manipulators, obstacle advoidance

OPEN ACCESS

Edited by:
Yongping Pan,
National University of Singapore,
Singapore

Reviewed by:
Jishnu Keshavan,
Indian Institute of Science (IISc), India
Kai Guo,
Shandong University, China

****Correspondence:***
Ahmad Ataka
ahmad.ataka.ar@ugm.ac.id

Specialty section:
This article was submitted to
Robotic Control Systems,
a section of the journal
Frontiers in Robotics and AI

Received: *13 December 2021*
Accepted: *21 January 2022*
Published: *18 February 2022*

Citation:
Ataka A, Lam H-K and Althoefer K
(2022) Magnetic-Field-Inspired
Navigation for Robots in Complex and
Unknown Environments.
Front. Robot. AI 9:834177.
doi: 10.3389/frobt.2022.834177

1 INTRODUCTION

Over the past decade, robotic technologies have found their way into an ever-growing number of fields. No longer limited to the well defined and structured environments commonly found in industrial sites, robotic devices have moved into all manner of environments as they become key "players" in a variety of domestic and care tasks, as well as in post-disaster evacuation and nuclear decommissioning. In line with this shift in robotic applications, the need to equip a robot with the ability to navigate arbitrarily-shaped obstacles within unknown environments, while relying on limited sensor information, becomes increasingly important.

The problems of robot navigation, path planning, and obstacle avoidance have been widely explored over the last 30 years (Latombe, 1991), (Choset, 2005). In the early years, solutions focused on producing geometrical paths for the robot to take to reach its target. While theoretically elegant, this type of planning was found to be rather impractical due to the costly numerical computation that was needed to create the configuration space (C-Space) (LaValle, 2011). This prompted a new approach, known as sampling-based planning which randomly samples and checks whether a configuration lies in free space or not (Choset, 2005). Other algorithms produce initial paths for static environments that are based on a known map and modify those paths online (Brock and Khatib, 2002; Vannoy and Xiao, 2008). Another approach that employs potential function with a single global minimum has been proposed. Examples of this kind of navigation method include a harmonic function (Garrido et al., 2010) and a fast-marching method (Valero-Gomez et al., 2013). The properties of magnetic fields have also inspired several researchers to create navigation systems that mimic the behaviour of magnetic fields. Examples include those reported in (Singh et al., 1996, 1997) and (Haddadin et al., 2011). However, and significantly, most of these algorithms rely on perfect knowledge of the environment prior to the onset of any motion on the part of the robotic device.

For applications in which no prior knowledge of the environment is available, a reactive sensor-based navigation system is employed, in which the robot senses the environment and the acquired sensor information is used to produce motion signals in real-time (Valle, 2011). An example of widely-used reactive navigation is the Artificial Potential Field (APF) method in which obstacles produce a repulsive behaviour repelling the robot while the target attracts it (Khatib, 1985). The main advantage of this method is its simplicity which indeed has led to its widespread application (Choset, 2005) and to the emergence of a variety of similar methods (Park et al., 2008). These methods, however, all suffer from the well-documented problem of local-minima often causing the robot to get stuck in undesired configurations and fail to reach its target (Choset, 2005). To avoid this problem without diminishing its reactivity, a vortex field, designed to circulate around the obstacle surface, is incorporated (Rizqi et al., 2014), although in this case, the authors present no guarantees in terms of stability or collision avoidance capability. Inspiration from computational geometry has also led to the use of power diagrams as a route to solving the problem of reactive navigation in an unknown environment (Arslan and Koditschek, 2016). Despite being able to guarantee obstacle avoidance and target convergence, this approach is however only suitable for a topologically simple environment consisting of spherical obstacles in a planar world.

“Circular fields” or “gyroscopic forces” (Singh et al., 1996, 1997; Haddadin et al., 2011), which work by redirecting rather than repelling the robot, have been researched extensively during the last decade and an algorithm based on this has been used in a planar point-like robot to achieve tasks such as boundary following (Zhang et al., 2004a,b), obstacle avoidance (Chang and Marsden, 2003), and formation control for multi-agent systems (Chang and Marsden, 2003). However, obstacle avoidance and goal convergence are only guaranteed in environments with convex obstacles, limiting its applications in realistic environments. Applying these techniques for non-convex obstacles is not straightforward because simply redirecting the direction of the robot’s movement to follow non-convex obstacles’ surface is not sufficient to guarantee collision avoidance. Adding a repulsive term will help to guarantee collision avoidance. However, it affects the goal convergence and potentially leads the robot towards a local minimum. More recent work has explored the possibility of using a gyroscopic control method for maze-like environments (Matveev et al., 2013) as well as environments with dynamic obstacles (Savkin and Wang, 2013, 2014) and those with deforming obstacles (Matveev et al., 2012, 2015). These methods, summarized in (Savkin and Wang, 2015), were however all specifically designed for planar unicycle-like mobile robots. Subsequent to that, efforts have been focused on the application of gyroscopic forces in fully-actuated or under-actuated robotics systems in 3D environments (Garimella et al., 2016). This method, though, is specifically designed for environments consisting of spherical and cylindrical obstacles. While efforts to apply gyroscopic force algorithms to formation control of multiple robots in 3D have also been reported (Justh and Krishnaprasad, 2004; Sabattini et al., 2013, 2017), these algorithms only provide solutions for collision avoidance in point-like robots.

In this paper, we present a reactive magnetic-field-inspired navigation system for robots in 3D environments. The moving robot generates an artificially-induced electric current on the surface of an obstacle which, in turn, induces a magnetic field on the robot. Although reliant on local sensor information only, indeed without any required knowledge of the geometry or position of any obstacle in the given environment, the proposed algorithm has the ability to navigate point-like robots to their desired targets without any local minima issues. Our algorithm outperforms other magnetic-field-inspired navigation algorithms (Singh et al., 1996, 1997; Haddadin et al., 2011) as no requisite knowledge of the environment is required. It is also superior to the reactive gyroscopic force methods (Zhang et al., 2004a,b; Chang and Marsden, 2003; Garimella et al., 2016; Justh and Krishnaprasad, 2004; Sabattini et al., 2013, 2017), indeed including our own recent works (Ataka et al., 2018a,b), as obstacle avoidance and goal convergence are guaranteed in environments consisting of convex and non-maze concave obstacles. Moreover, our algorithm is not limited to 2D environments, as is the case for methods presented in (Savkin and Wang, 2015), but also enables the guidance of robots to their target within 3D environments. Alongside these advantages, our algorithm has a further benefit, which relates to its generic nature - and its consequential applicability to a variety of robotic platforms operating in a wide range of settings. To the best of our knowledge, we believe that this paper is the first to propose a reactive, magnetic-field based robot navigation system that is capable of guiding robots around arbitrary-shaped and unknown convex and non-maze concave obstacles to its target, within 3D environments.

2 INSPIRATION FROM NATURE

Our ideas are inspired by phenomena observed in classical electromagnetism. Prominent among its underlying theorems is the Biot-Savart law which describes the relationship between a current-carrying conductor and the surrounding magnetic field

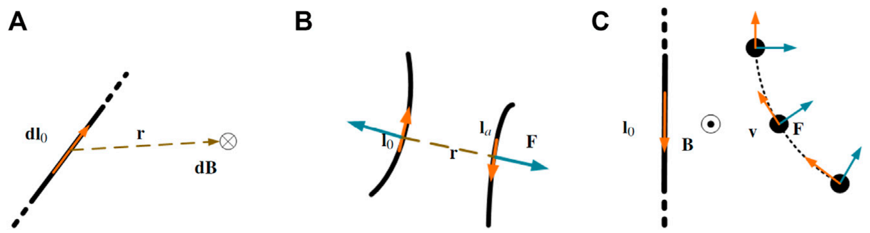

FIGURE 1 | (A) A current-carrying wire with electric current in the direction of $\mathbf{dl}_o$ will produce magnetic field **dB** in the space around it. In this configuration, **dB** points inside the paper. **(B)** Two wires with electric currents $\mathbf{l}_a$ and $\mathbf{l}_o$, each flowing towards opposite direction of each other, will generate a repulsive force **F** on both wires directed away from each other. **(C)** A positively-charged particle is affected by Lorentz force **F** due to the presence of magnetic field **B** induced by current $\mathbf{l}_o$.

that it produces. A current-carrying wire segment with an infinitesimal length $\mathbf{dl}_o$ and electrical current i_o will induce a magnetic field **dB**. This magnetic field, whose direction is illustrated in **Figure 1A**, is expressed by the following equation (Halliday et al., 2013)

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

Where **r** specifies the position of an arbitrary point in the surrounding space of the wire segment relative to the current-carrying wire, μ_0 specifies a permeability constant, while × specifies the operation of vector cross product. The total magnetic field produced by the wire can be derived by integrating the above operation over a wire length l which will depend on the wire configuration.

This magnetic field will produce a force **dF** on any other current-carrying wire with an infinitesimal length $\mathbf{dl}_a$ and current i_a flowing in the opposite direction, as illustrated in **Figure 1B**, and expressed by the following equation

$$\mathbf{dF} = i_a\mathbf{dl}_a \times \mathbf{B}. \tag{2}$$

The direction of the force is perpendicular to the direction of both the electric current i_a (in the direction of the vector $\mathbf{dl}_a$) and magnetic field vector **B**. This force also generates a repulsive behaviour in the pair of wires, such that it drives the wires apart.

In a similar manner, as illustrated in **Figure 1C**, the movement of a charged particle q will be influenced by the magnetic field as it moves into the vicinity of the current-carrying wire. This magnetic field will apply a force **F** to the moving particle in a direction perpendicular to the particle's velocity vector **v** and the magnetic field vector **B**. This force, usually known as Lorentz force, is expressed by the following equation (Halliday et al., 2013)

$$\mathbf{F} = q\mathbf{v} \times \mathbf{B}. \tag{3}$$

Substituting the expression for **B** from **(1)** and abandoning the infinitesimal notation, the force both in **Eqs 2, 3** can be expressed as

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

Where $\mathbf{l}_a$ denotes either the direction of the current in the case of the wire in **Figure 1B** or the particle's velocity in the case of the charged particle in **Figure 1C**.

As shown in **Figure 1C**, the interaction between the magnetic field and the moving charged particle produces a force whose direction is perpendicular to that of the particle's motion. The net result is that the particle's direction of movement is altered, rather than its velocity decreased. Taking inspiration from this behaviour pattern, a robotic arm can be envisaged as a moving charged particle whose velocity is expressed by vector **v** while the surface of a potential obstacle can be thought of as a current-carrying wire. Collision avoidance can then be achieved if the robot induces an artificial current $\mathbf{l}_o$ that flows over the surface of an obstacle expressed by position vector $\mathbf{r}_o$ (with respect to the robot's own position). This artificial current will then apply a force **F** onto the robot, pushing the robot away from and around the surface of any obstacles in its pathway.

Using our definition of the positional vector, we get $\mathbf{r}_o = -\mathbf{r}$. Thus, we can rewrite the **Eq. 4** in the following form

$$\mathbf{F} = c\ \ \mathbf{l}_a \times (\mathbf{r}_o \times \mathbf{l}_o)\ \ f\left(|\mathbf{r}_o|, |\dot{\mathbf{p}}|\right), \tag{5}$$

In which $c > 0$ is a positive constant, $\mathbf{l}_a$ is defined as the direction of the robot's velocity vector, while $f\left(|\mathbf{r}_o|, |\dot{\mathbf{p}}|\right) \geq 0$ is defined as a function which is dependent upon the distance between the obstacle and the robot $|\mathbf{r}_o|$ and/or speed of the robot $|\dot{\mathbf{p}}|$. To help highlight the key characteristics of the algorithm, a skew-symmetric matrix $\hat{\mathbf{l}}$ is used to replace the operation of vector cross product $\mathbf{l}\times$ of an arbitrary vector $\mathbf{l} = [\,l_x\ \ l_y\ \ l_z\,]^T$. This matrix is defined as

$$\hat{\mathbf{l}} = \begin{bmatrix} 0 & -l_z & l_y \\ l_z & 0 & -l_x \\ -l_y & l_x & 0 \end{bmatrix}. \tag{6}$$

We can design this artificial vector field to produce a desired robot behaviour by defining the current direction $\mathbf{l}_o$ on the obstacle surface and the scalar function $f\left(|\mathbf{r}_o|, |\dot{\mathbf{p}}|\right)$ as per the explanation in **Section 4**.

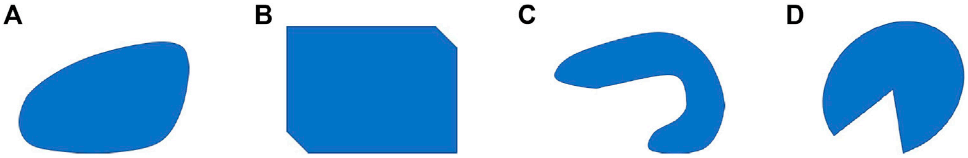

FIGURE 2 | The environment considered in the paper can consists of any of the following obstacles: **(A)** a smooth convex obstacle, **(B)** a non-smooth convex obstacle, **(C)** a smooth concave obstacle, and **(D)** a non-smooth concave obstacle.

3 PROBLEM FORMULATION

Let us consider a point-mass robot in a bounded work-space $\mathcal{W} \subset \mathbb{R}^2$ or $\mathcal{W} \subset \mathbb{R}^3$ whose position is described by a position vector $\mathbf{p} \in \mathcal{W}$. The robot is assumed to be equipped with a local sensor that is sensitive to the surrounding environment within a sphere of radius r_s centred at the robot's position $\mathbf{p}$. The robot is also able to measure, in real-time, its velocity $\dot{\mathbf{p}}$, whose direction is defined as

$$\mathbf{l}_a = \frac{\dot{\mathbf{p}}}{|\dot{\mathbf{p}}|}. \tag{7}$$

Lastly, the robot's actuators are assumed to be non-saturatable, i.e. they will always have the necessary energy to produce the required movement. Significantly, the environment is deemed completely unknown to the robot prior to its actuation.

We assume that the environment of the robot consists of $m \in \mathbb{N}$ number of fixed obstacles $\mathcal{O}_i$. The obstacle $\mathcal{O}_i$ can be any of the following types:

1. a member of a convex set with smooth boundary, such as illustrated in **Figure 2A**,
2. a member of a convex set with non-smooth boundary, such as illustrated in **Figure 2B**,
3. a member of a *simple* concave set with smooth boundary, such as illustrated in **Figure 2C**,
4. a member of a *simple* concave set with non-smooth boundary, such as illustrated in **Figure 2D**.

The term *simple* refers to non-maze geometry in which the obstacle surface will not require the robot to change its direction of motion more than 180° from its initial trajectory towards the target while following the obstacle boundary. The free space is then formally defined as $\mathcal{F} = \mathcal{W} \backslash \bigcup_{i=1}^{m} \mathcal{O}_i$. We assume that the robot starts moving from an initial position $\mathbf{p_s} \in \mathcal{F}$ to the desired target position $\mathbf{p_g} \in \mathcal{F}$ located at some distance from any of the surfaces of an obstacle. Assuming a double integrator system with dynamics described as

$$\ddot{\mathbf{p}} = \mathbf{u}, \tag{8}$$

We want to determine the control signal $\mathbf{u}$ which will guide the robot's position $\mathbf{p}(t)$ towards the desired position $\mathbf{p_g}$ as $t \to \infty$ while keeping the robot free from collision throughout its entire route, which is formally defined as

$$\mathbf{p}(t) \in \mathcal{F}, \forall t. \tag{9}$$

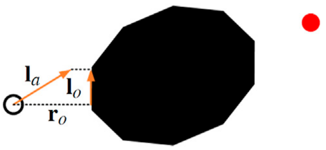

FIGURE 3 | The workspace of a robot on its way towards the target in the close proximity of the polygonal obstacle. The artificial current on the surface of obstacle $\mathbf{l}_o$ is designed to have the same direction as the projection of the robot's velocity $\mathbf{l}_a$ on to the obstacle surface.

4 PROPOSED ALGORITHM

4.1 General Algorithm

To ensure collision avoidance while moving toward the target, we introduce a magnetic-field-inspired vector field $\mathbf{F_o}$ consisting of two terms: a boundary-following vector field $\mathbf{F_b}$ and a collision-avoidance vector field $\mathbf{F_a}$ as follows

$$\mathbf{F_o} = \mathbf{F_b} + \mathbf{F_a}. \tag{10}$$

The magnetic-field-inspired vector fields $\mathbf{F_b}$ and $\mathbf{F_a}$ are created by artificial currents on the obstacle surface induced by the moving robot, $\mathbf{l}_o$ and $\mathbf{l}_{o\perp}$ respectively.

To induce boundary-following behaviour, so that the robot follows the direction of artificial current $\mathbf{l}_o$, the vector field $\mathbf{F_b}$ is defined as

$$\mathbf{F_b} = c\ \ \mathbf{l}_a \times (\mathbf{l}_o \times \mathbf{l}_a)\ \ f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|). \tag{11}$$

The behaviour of the moving robot under the influence of the vector field $\mathbf{F_b}$ mimics the behaviour of a charged particle moving in the proximity of a current-carrying wire as illustrated in **Figure 1C**.

In order to generate an artificial current $\mathbf{l}_o$ on the surface of an obstacle, let us start with a point-mass robot in $\mathbb{R}^2$ moving in an environment as illustrated in **Figure 3**.

In this planar environment, the current on the obstacle surface needs to be induced in such a way that the robot safely circumvents the obstacle by following the obstacle boundary in either an anti-clockwise or a clockwise direction. At the same time, the effect of the current should not dramatically alter the path that the robot is inclined to take to reach its target - it simply needs to ensure that collisions are avoided and unnecessary oscillations are mitigated. To satisfy this requirement, we define the proposed artificial current direction $\mathbf{l}_o$ as a projection of the robot's velocity direction $\mathbf{l}_a$ on to the surface of a nearby obstacle, as illustrated in **Figure 3**.

Despite having no knowledge of the obstacle's geometry, we can still make a practical assumption regarding the surface of a nearby obstacle using local sensory information only.

Suppose that the robot senses the closest point of the obstacle surface located at position $\mathbf{r}_o$ relative to the robot's current position. The surface of the obstacle at this specific point is then assumed to have a normal vector $\mathbf{n}_o$ in the opposite direction of $\mathbf{r}_o$, i.e. the surface is assumed to be perpendicular to the vector $\mathbf{r}_o$. The artificial current $\mathbf{l}_o$ can therefore be described as:

$$\mathbf{l}_o = \mathbf{l}_a - \frac{\left(\mathbf{l}_a^T\mathbf{r}_o\right)\mathbf{r}_o}{|\mathbf{r}_o|^2}. \tag{12}$$

Finally, to complete the definition of the boundary-following vector field $\mathbf{F_b}$, the scalar function $f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|)$ is chosen as:

$$f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|) = \frac{|\dot{\mathbf{p}}|}{|\mathbf{r}_o|}. \tag{13}$$

To keep the robot at a safe distance from the obstacle surface, the vector field $\mathbf{F_a}$ behaves much in the same way as a pair of current-carrying wires in which the current flowing in one wire is in the opposite direction to that of the other - *see* **Section 2** (**Figure 1B**). The robot moving in a direction of $\mathbf{l}_a$ will induce an artificial current $\mathbf{l}_{o\perp}$ in the opposite direction of current $\mathbf{l}_o$ described in **(12)** as follows:

$$\mathbf{l}_{o\perp} = -\left(\mathbf{l}_a - \frac{\left(\mathbf{l}_a^T\mathbf{r}_o\right)\mathbf{r}_o}{|\mathbf{r}_o|^2}\right). \tag{14}$$

The field equation $\mathbf{F_a}$ can then be derived following the general equation in **(5)** with the scalar function $f(|\mathbf{r}_o|, |\dot{\mathbf{p}}|)$ chosen to be inversely proportional to the robot-obstacle distance r as follows

$$\mathbf{F_a} = \mathbf{l}_a \times \left(\frac{\mathbf{r}_o}{r} \times \mathbf{l}_{o\perp}\right)\ \frac{c_\perp}{r}, \tag{15}$$

Where $c_\perp$ denotes a positive constant.

4.2 Properties of the Proposed Algorithm

In the following section, we outline the key properties of our navigation algorithm. To simplify the analysis, we describe the properties inherently possessed by each of the two vector field components: the boundary-following vector field $\mathbf{F_b}$ and the collision-avoidance vector field $\mathbf{F_a}$.

Several properties of the boundary-following vector field $\mathbf{F_b}$, outlined in previous work by the authors (Ataka et al., 2018a), can be summarised in Lemma 1 to Lemma 4. Please refer to our previous work in (Ataka et al., 2018a) for the proofs of these lemmas.

Lemma 1. *The force* $\mathbf{F_b}$ *described in* **(11)** *does not affect the speed of the robot* $v = |\dot{\mathbf{p}}|$.

Lemma 2. S*uppose that the robot is in the vicinity of a flat obstacle surface and it is located far enough from the obstacle such that no collision can occur during its movement. The force* $\mathbf{F_b}$ *will guide the robot to reach a direction of the artificial current* $\mathbf{l}_o$, *causing the robot's direction to be parallel to the obstacle surface.*

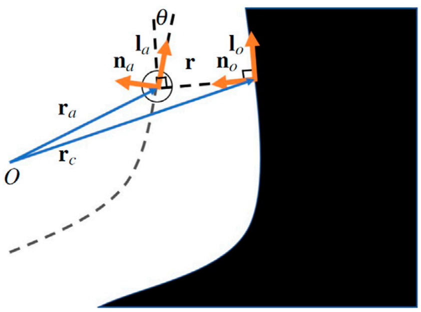

FIGURE 4 | The scenario of a robot moving in the vicinity of a concave obstacle which has a continuous surface.

Lemma 3. *Suppose that the closest surface of a convex-shaped obstacle is located at initial distance* $r_0 = |\mathbf{r}_o|$ *from the robot. The robot will never touch the surface of the obstacle as long as its initial direction* $\mathbf{l}_a$ *is not in line with the direction of vector* $\mathbf{r}_o$ *connecting the robot and the obstacle.*

Lemma 4. T*he robot's direction of movement will asymptotically reach a direction in line with the direction of the artificial current. This equilibrium direction is globally asymptotically stable.*

Lemma 3–4 show that the boundary-following vector field $\mathbf{F_b}$ ensures that the robot follows a direction parallel to the nearby obstacle surface and hence avoids collision with a convex obstacle. However, these features do not directly control the distance to the obstacle surface. This creates a problem for environments with concave obstacles as maintaining the robot's direction alone is not sufficient to avoid collisions. It is here that the collision-avoidance vector field $\mathbf{F_a}$ comes into play. The properties of this vector field are as follows:

Lemma 5. *The force* $\mathbf{F_a}$ *in* (15) *will not change the magnitude of robot's velocity* $v = |\dot{\mathbf{p}}|$.

Proof. We introduce vector $\mathbf{a_a} = \left(\frac{\mathbf{r}_o}{r} \times \mathbf{l}_{o\perp}\right)\ \frac{c_\perp}{r}$. From the skew-symmetric definition in **(6)**, we conclude that

$$\mathbf{l}_a^T\mathbf{F_a} = \mathbf{l}_a^T\left(\hat{\mathbf{l}}_a\mathbf{a_a}\right) = 0, \tag{16}$$

The dynamics of a point-mass robot characterized by mass m and speed $v = |\dot{\mathbf{p}}|$ is

$$\mathbf{F_a} = m\frac{d\dot{\mathbf{p}}}{dt} = m\left(\frac{dv}{dt}\mathbf{l}_a + v\frac{d\mathbf{l}_a}{dt}\right). \tag{17}$$

Substituting $\mathbf{F_a}$ into **(16)** and considering $\mathbf{l}_a^T\frac{d\mathbf{l}_a}{dt} = 0$, we conclude that

$$\mathbf{l}_a^T\mathbf{F_a} = m\frac{dv}{dt} = 0. \tag{18}$$

The only solution to the previous equation is $\frac{dv}{dt} = 0$, which proves the claim that the robot's speed is not affected by the force $\mathbf{F_a}$.

To prove obstacle avoidance, let us imagine an environment in which the obstacle has a concave continuous surface; this means that the surface is smooth with no discontinuity and each point on the obstacle surface can be described by a curvature parameter κ, which is a reciprocal of a radius of a surface. Suppose we have an obstacle whose closest point to the robot is located at vector $\mathbf{r}_\mathrm{c}$ and the robot's location is defined by vector $\mathbf{r}_\mathrm{a}$ with respect to a static reference point O as illustrated in **Figure 4**. According to the Frenet-Serret formula in $\mathbb{R}^2$ space (Millman and Parker, 1977), the surface of the obstacle can be described by the following equations

$$\frac{\partial \mathbf{r}_c}{\partial s} = \mathbf{l}_o, \frac{\partial \mathbf{l}_o}{\partial s} = \kappa \mathbf{n}_o, \frac{\partial \mathbf{n}_o}{\partial s} = -\kappa \mathbf{l}_o, \tag{19}$$

Where s represents curve segment, while $\mathbf{n}_\mathrm{o}$ and $\mathbf{l}_\mathrm{o}$ represent unit vectors whose direction is normal and perpendicular respectively to the closest obstacle surface as shown in **Figure 4**. *Via* the chain rule, we get the following set of equations

$$\dot{\mathbf{r}}_c = \frac{ds}{dt}\mathbf{l}_o, \dot{\mathbf{l}}_o = \frac{ds}{dt}\kappa \mathbf{n}_o, \dot{\mathbf{n}}_o = -\frac{ds}{dt}\kappa \mathbf{l}_o. \tag{20}$$

Since the robot's speed is shown to be unaffected by the vector field, the robot's motion can be described by

$$\dot{\mathbf{r}}_a = v\mathbf{l}_a, \dot{\mathbf{l}}_a = \omega \mathbf{n}_a, \dot{\mathbf{n}}_a = -\omega \mathbf{l}_a, \tag{21}$$

With $\omega = \frac{|\mathbf{F}_a|}{mv}$.

Lemma 6. For a concave obstacle which produces vector field $\mathbf{F_a}$ expressed in **(15)**, the rate of angle **θ** between the robot's direction and the obstacle surface (as illustrated in **Figure 4**) is given by

$$\dot{\theta} = \left(\frac{\kappa v}{1-\kappa r} - \frac{c_\perp}{mvr}\right)\cos\theta. \tag{22}$$

Proof from **Figure 4**, we can infer that

$$\mathbf{l}_o^T \mathbf{n}_a = \sin\theta. \tag{23}$$

Differentiating the equation with respect to time, we have

$$\dot{\mathbf{l}}_o^T \mathbf{n}_a + \mathbf{l}_o^T \dot{\mathbf{n}}_a = \dot{\theta}\cos\theta. \tag{24}$$

Combining with **20**, **21**, we get

$$\frac{ds}{dt}\kappa \mathbf{n}_o^T \mathbf{n}_a + \mathbf{l}_o^T(-\omega \mathbf{l}_a) = \dot{\theta}\cos\theta. \tag{25}$$

Recalling that $\mathbf{n}_o^T \mathbf{n}_a = \mathbf{l}_o^T \mathbf{l}_a = \cos\theta$, we can simplify the equation into

$$\dot{\theta} = \frac{ds}{dt}\kappa - \omega. \tag{26}$$

Using $|\mathbf{l}_a \times (\frac{\mathbf{r}_o}{r} \times \mathbf{l}_{o\perp})| = |\mathbf{l}_{o\perp}| = \cos\theta$, we can provide a simplified representation of ω as follows,

$$\omega = \frac{|\mathbf{F_a}|}{mv} = \frac{c_\perp |\mathbf{l}_a \times \left(\frac{\mathbf{r}_o}{r} \times \mathbf{l}_{o\perp}\right)|\frac{1}{r}}{mv} = \frac{c_\perp \cos\theta}{mvr}. \tag{27}$$

From **Figure 4**, we also get

$$\mathbf{r}^T \mathbf{l}_o = 0. \tag{28}$$

Differentiating with respect to time, we get

$$\dot{\mathbf{r}}^T \mathbf{l}_o + \mathbf{r}^T \dot{\mathbf{l}}_o = 0. \tag{29}$$

Combining with **20**, **21** and noting that $\dot{\mathbf{r}} = \dot{\mathbf{r}}_c - \dot{\mathbf{r}}_a$, we get

$$\left(\frac{ds}{dt}\mathbf{l}_o - v\mathbf{l}_a\right)^T \mathbf{l}_o + \kappa\frac{ds}{dt}\mathbf{r}^T \mathbf{n}_o = 0. \tag{30}$$

Combining the equation with the fact that $\mathbf{l}_a^T \mathbf{l}_\mathrm{o} = \cos\theta$ and $\mathbf{r}^\mathrm{T}\mathbf{n}_\mathrm{o} = -\mathrm{r}$, we get

$$\frac{ds}{dt} = \frac{v\cos\theta}{1-\kappa r}. \tag{31}$$

Substituting the value of $\frac{ds}{dt}$ in **(31)** and ω in **(27)** to **(26)**, we get

$$\dot{\theta} = \left(\frac{\kappa v}{1-\kappa r} - \frac{c_\perp}{mvr}\right)\cos\theta. \tag{32}$$

Lemma 7. Force $\mathbf{F_a}$ *in* **(15)** *will guarantee collision avoidance with any continuous concave surface if the following conditions are satisfied:*

1. *the initial direction of* $\boldsymbol{l}_a$ *is not in the direction of* $\mathbf{r}_o$*, and*
2. *the surface's curvature κ and the robot's distance to the surface r follows the following condition at all times*

$$\kappa < \frac{1}{r}. \tag{33}$$

Proof Using the chain rule of derivative, **Eq. 22** can be modified into

$$\frac{d\theta}{dr}\dot{r} = \left(\frac{\kappa v}{1-\kappa r} - \frac{c_\perp}{mvr}\right)\cos\theta. \tag{34}$$

The robot's velocity component in y-axis is equal to $-\dot{r}$ as follows

$$\dot{r} = -v\sin\theta. \tag{35}$$

Inserting the value of $\dot{r}$ from **(35)**, we get

$$\int \frac{\sin\theta}{\cos\theta}d\theta = \int\left(-\frac{\kappa}{1-\kappa r} + \frac{c}{mv^2 r}\right)dr. \tag{36}$$

Details on how to solve the integral are provided in the **Appendix**. The final equation has the following form

$$(1-\kappa r)r^{\frac{c}{mv^2}}\cos\theta = D, \tag{37}$$

Where D denotes the constant relating to the initial condition. We assume that the robot is in an initial condition described by:

1. r > 0, i.e. the robot does not touch the obstacle, and
2. (1 − κr) > 0, i.e. there is a single point on the obstacle surface that has the shortest distance to the robot.

Then, recalling the assumption that the initial angle $\theta \neq \pm\frac{\pi}{2}$ and the fact that $\theta \in (-\frac{\pi}{2}, \frac{\pi}{2})$, we can conclude that cos θ > 0, which results in D > 0. In other words, in order that the robot-to-obstacle distance maintains the condition r > 0 (i.e. no contact

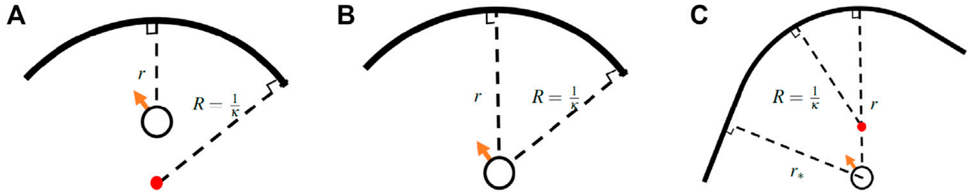

FIGURE 5 | Three possibilities of a robot in the vicinity of a smooth concave obstacle: **(A)** the case where $\kappa < \frac{1}{r}$, **(B)** the case where $\kappa = \frac{1}{r}$, and **(C)** the case where $\kappa > \frac{1}{r}$.

with the obstacle surface), it is necessary for (1 − κr) > 0. This concludes the proof.

There are three possibilities regarding the relation between obstacle-to-robot distance r and curvature of the obstacle surface κ:

1. The first case, where $\kappa < \frac{1}{r}$, describes the situation where there is only a single point on the obstacle surface closest to the robot, as illustrated in **Figure 5A**. Lemma 7 guarantees that in this condition, the algorithm makes sure that the robot will be able to avoid colliding with the obstacle.

2. The second case, $\kappa = \frac{1}{r}$, refers to a condition where the robot is located at a distance r which is equal to the radius of curvature $\frac{1}{\kappa}$ of the obstacle surface, as illustrated in **Figure 5B**. This means that there are several points along the obstacle surface, each equidistant from the robot. In this case, with multiple closest points, the algorithm, as things stand, will not be able to guarantee collision avoidance.

3. The final case, where $\kappa > \frac{1}{r}$, refers to a condition in which the point on the obstacle surface, where vector $\mathbf{r}$ is perpendicular to vector $\mathbf{l}_o$, is not the obstacle's closest point to the robot, i.e. there are other points (at least one) along the obstacle surface which are closer to the robot, as illustrated in **Figure 5C**. Those other points are characterized by distance vector $\mathbf{r}_*$ (with $|\mathbf{r}_*| < |\mathbf{r}|$), which is also perpendicular to vector $\mathbf{l}_o$. The actual point on the obstacle surface that is closest to the robot will then fall under one of the previous two cases.

Remark. From these three cases, we can conclude that Lemma 7 actually guarantees obstacle avoidance for any concave obstacle with a continuous surface and a single point that is closest to the robot. Cases with multiple closest points will be explored in **Section 4.4**.

4.3 Properties of Goal Convergence

For the chosen point-mass robot model (as described in **Section 3**), the control law for navigating the robot past obstacles towards the desired goal position is given by:

$$\mathbf{u} = \mathbf{F_g} + \mathbf{F_o}. \tag{38}$$

The control law in **(38)** consists of an obstacle avoidance term $\mathbf{F_o}$ in **(10)** and a goal attraction term $\mathbf{F_g}$. For simplicity, in this section, we use a proportional-derivative (PD) controller for $\mathbf{F_g}$ with goal position $\mathbf{p_g}$ as the equilibrium point, represented as follows:

$$\mathbf{F_g} = -K_P\left(\mathbf{p} - \mathbf{p_g}\right) - K_D\dot{\mathbf{p}}. \tag{39}$$

Here, K_P and K_D are both positive constants.

To prove goal convergence, we assume that the robot is in an environment with small-sized non-maze obstacles such that the robot's distance to the goal, when following the boundary of the obstacle surface, is never greater than its initial distance to the goal. By setting this constraint, we are assuming that the robot's speed will never be zero as it follows the boundary. In this case, the control law in **(38)** is globally asymptotically stable.

Lemma 8. The control law in **Eqs 38**, **39** when applied to a point-like robot using the dynamic model described in **(8)**, has a globally asymptotically stable equilibrium at the goal position $\mathbf{p_g}$. This assumes that the robot never stops while following the boundary, and that its direction of motion is not orthogonal to the closest obstacle surface.

Proof.The proof begins with the following Lyapunov function candidate

$$V = \frac{1}{2}\dot{\mathbf{p}}^T\dot{\mathbf{p}} + \frac{1}{2}K_P\mathbf{e}^T\mathbf{e}, \tag{40}$$

In which we define error vector as $\mathbf{e} = (\mathbf{p} - \mathbf{p_g})$. The rate of change of V is given by

$$\dot{V} = \dot{\mathbf{p}}^T\ddot{\mathbf{p}} + K_P\mathbf{e}^T\dot{\mathbf{e}}. \tag{41}$$

Substituting **(10)**, **(38)**, **(39)** into **(8)** and considering that $\dot{\mathbf{e}} = \dot{\mathbf{p}}$ while, according to Lemma 1 and Lemma 5, $\dot{\mathbf{p}}^T\mathbf{F_o} = \dot{\mathbf{p}}^T\mathbf{F_b} + \dot{\mathbf{p}}^T\mathbf{F_a} = 0$, the equation can be simplified into

$$\dot{V} = -K_D\dot{\mathbf{p}}^T\dot{\mathbf{p}}. \tag{42}$$

On this basis, the condition $\dot{V} \leq 0$ always holds including at $\mathbf{p} = \mathbf{p_g}$ and $\dot{\mathbf{p}} = \mathbf{0}$, which is the equilibrium point where $V = \dot{V} = 0$. Therefore, we can therefore conclude that this goal position is indeed a globally asymptotically stable equilibrium.

4.4 Extension for Special Cases Condition

Thus far, the proposed vector field $\mathbf{F_o}$ guarantees obstacle avoidance for: *1*) an arbitrary-shaped convex obstacle (irrespective of whether it has a continuous or discontinuous surface) and *2*) a concave obstacle provided that it has a continuous surface and a unique closest point at all times. The real world however can be uncooperative in this regard and is likely to present us with obstacles that do not adhere to these requisite constraints. There are objects with concave sharp corners or with surfaces whose curvature κ changes abruptly. We therefore need to consider situations in which we may not

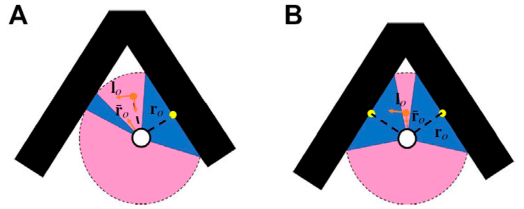

FIGURE 6 | The averaging technique employed to post-process the sensory information can be used to solve the problem of a concave sharp corner where **(A)** the curvature surface is discontinuous and **(B)** there are non-unique closest points.

have a unique point on the obstacle's surface that is closest to the robot. To mitigate this issue, we propose an extension to our original algorithm.

Previously, we had assumed that the robot identifies the closest point on the obstacle surface from its sensor readings by simply choosing the point with the smallest distance value out of a selection of readings. In the revised scenario, the robot obtains this value by computing the average over several sensed distances. The robot chooses a number of sensed distances from current sensor readings which are smaller than a specified threshold δr. It then calculates the average distance value from these readings, ($\bar{\mathbf{r}}_o$), which it uses for subsequent calculations. For a concave obstacle, this distance value ($\bar{\mathbf{r}}_o$) will always be smaller in magnitude than the closest sensed position, i.e. $|\bar{\mathbf{r}}_o| < |\mathbf{r}_o|$, as illustrated in **Figure 6**, and so we use this value as the new closest point. For a convex obstacle, however, the condition is reversed, i.e. $|\bar{\mathbf{r}}_o| \geq |\mathbf{r}_o|$, and we can therefore revert to the standard position data $\mathbf{r}_\mathrm{o}$ as the vector describing the closest point between the obstacle and the robot.

The robot is now able to navigate around both convex and non-convex obstacles, even when surfaces are not continuous (such as a sharp corners, as in **Figure 6**). This holds true as long as the robot's initial direction is not entirely in line with the vector connecting the robot and the closest obstacle point ($\theta \neq \frac{\pi}{2}$). This configuration could theoretically cause the artificial current $\mathbf{l}_\mathrm{o}$, a projection of the robot's speed $\mathbf{l}_\mathrm{a}$ towards the obstacle surface, to drop to zero. The obstacle would then produce no magnetic field to repel the advancing robot. From a practical perspective, this would be unlikely to happen as noise in the sensor's measurements would naturally yield a non-zero angle (even if it is a small one) between the two vectors $\mathbf{l}_\mathrm{a}$ and $\mathbf{r}_\mathrm{o}$. To reduce the effect of this problem, we use only the unit vector of the previously proposed artificial current when the magnitude of the standard artificial current is less than a positive constant ϵ. Another special case is when the environment consists of a large-size obstacle such that the robot's distance to the goal when circumnavigating the obstacle could be larger than its initial distance to the goal. Using only the PD control in **(39)**, in this type of environment, the robot could arrive at a zero-speed situation, i.e. when the robot loses most of its kinetic energy due to the dissipation term in the attractive field. To overcome this problem, we add a goal relaxation (GR) mechanism, having the property of decreasing the goal attraction $\mathbf{F}_\mathbf{g}$ when the robot is close to the surface of the obstacle, while the goal is still occluded by the obstacle, and, vice versa, to increase the goal attraction when the obstacle stops obstructing the desired position. Please refer to (Ataka et al., 2018a) for more details on strategies for the last two special cases.

TABLE 1 | List of parameter values used in the algorithm.

Param	Point-like robot	Baxter simulation	Baxter experiment
c	10	4	20
$c_\perp$	20	0.08	0.08
K_P	0.04	20	5
K_D	0.5	10	10
r_l	3 m	0.3 m	0.3 m
r_{la}	2 m	0.2 m	0.2 m
ϵ	3×10^{-6}	0.05	0.05

5 IMPLEMENTATION

We implemented our magnetic-field-inspired navigation algorithm to steer a dynamic model of a simulated point-like robot using the control law described in **(38)**. We first used PD control as described in **(39)** as a goal attraction term $\mathbf{F}_\mathbf{g}$ to guide the robot towards its desired target. Constants $\mathrm{K_P}$ and $\mathrm{K_D}$ are determined *via* trial and error, given the maximum actuating capability of the robot.

The second term $\mathbf{F}_\mathbf{o}$ of the control law **(38)** relates to obstacle avoidance, and consists of a boundary-following vector field $\mathbf{F}_\mathbf{b}$ described in **11–13** and a collision-avoidance vector field $\mathbf{F}_\mathbf{a}$ described in **14**, **15**. In our algorithm implementation, the boundary-following vector field is only used when the robot's closest distance to the obstacle falls below a specified limit $\mathrm{r_l}$, and the collision-avoidance vector field is only used when the robot get closer still, its threshold level being the distance $\mathrm{r_{la}}$.

$$\mathbf{F}_\mathbf{o} = \begin{cases} \mathbf{F}_\mathbf{b} + \mathbf{F}_\mathbf{a} & \text{if } |\mathbf{r}_o| < r_{la} \\ \mathbf{F}_\mathbf{b} & \text{if } |\mathbf{r}_o| \geq r_{la} \text{ and } |\mathbf{r}_o| < r_l \, . \\ \mathbf{0} & \text{if } |\mathbf{r}_o| \geq r_l \end{cases} \tag{43}$$

The PD control law referred to in **(39)**, however, does not guarantee constant robotic speed. A scenario could occur in which attraction to the goal causes the robot's speed v to increase when under the influence of a nearby obstacle. To prevent this from occurring, we introduced a geometric control term as an alternative to guide the robot towards the goal, loosely based on the work described in (Bullo and Murray, 1995) for SO(3). This control term has an interesting property: it is shown to globally asymptotically guide the robot's movement (toward the goal position $\mathbf{p}_\mathrm{g}$) without affecting the robot's speed. A full description of geometric control implementation is given in (Ataka et al., 2018b) for mobile robot and in (Ataka et al., 2019) for quadcopter robot.

In order to implement the algorithm on a 7-DOFs Baxter arm, we applied the vector field in the task space of the robot and used the dynamic model of the robot to produce the joint accelerations.

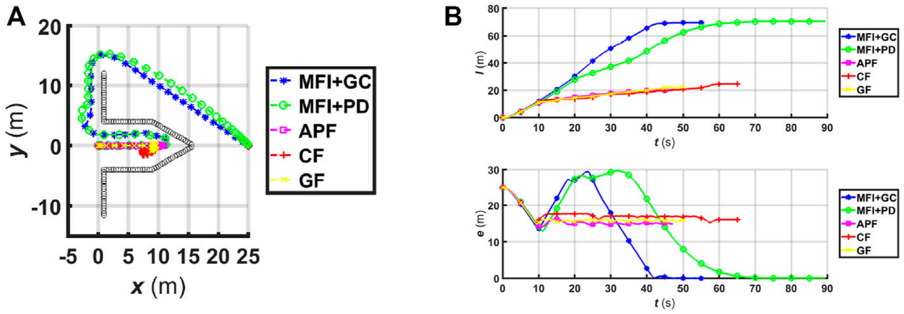

FIGURE 7 | (A) The robot's trajectory drawn in dashed lines. The environment consists of a sharp corner obstacle. **(B)** The plot of the covered trajectory l(t) and the positional error e(t) as function of time for 2D sharp corner environments.

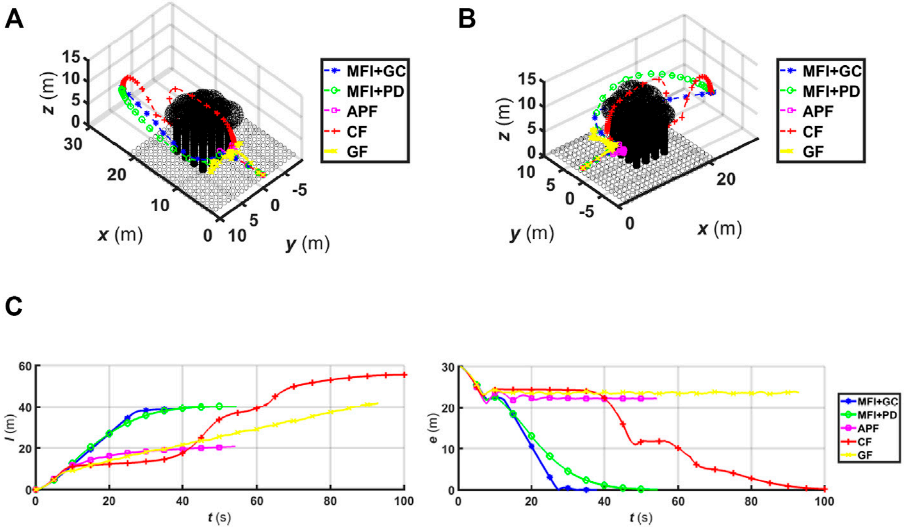

FIGURE 8 | The robot's trajectory drawn in dashed lines from two different perspectives (**(A,B)**). The environment consists of a forest-like environment. **(C)** The plot of the covered trajectory l(t) and the positional error e(t) as function of time for 3D forest environments.

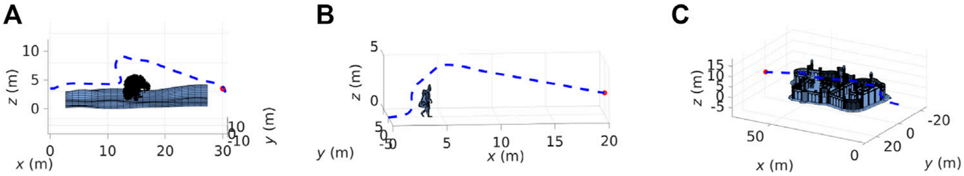

FIGURE 9 | The robot's trajectory using the magnetic-field-inspired navigation with geometric control in different environments populated with **(A)** a tree, **(B)** a human, and **(C)** a castle.

The vector field used to guide the point-like robot in (**38**) is redeployed to guide the tip of the Baxter arm towards its goal while avoiding obstacles along the way. A repulsive field is also applied to the body of the manipulator to mitigate potential impact, but to ensure that it doesn't affect the behaviour of the manipulator tip, we apply the torque in the null space of the Jacobian as described in (Brock and Khatib, 2002).

6 RESULTS AND ANALYSIS

To analyze and evaluate the performance of the proposed algorithm, we present our results from both simulations and an experimental study. Our magnetic-field-inspired navigation system was applied to a point-like robot model and tested in several scenarios in $\mathbb{R}^2$ and $\mathbb{R}^3$. The algorithm was also tested as

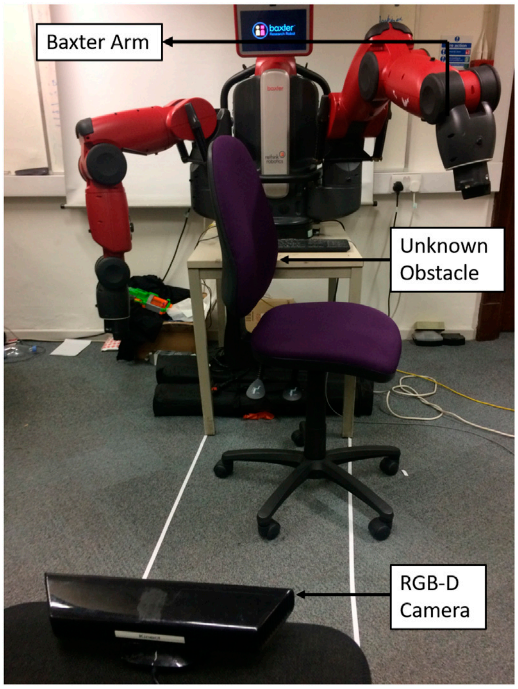

FIGURE 10 | The figure depicts experimental setup consisting of a seven DOFs Baxter manipulator. The environment consists of a chair detected by RGB-D camera.

a guiding method in a Baxter manipulator tip, a 7-DOFs industrial robot platform. The task was for the tip to avoid obstacles in $\mathbb{R}^3$, while the manipulator body itself was prevented from collisions by the application of a repulsive potential. We assume that the robot is only able to detect the surrounding environment within a range $r_s = r_l = 0.3$ m in all directions relative to its position. Some obstacles are generated as point clouds from a 3D model of the real environment; others are specifically designed to highlight the advantages of our algorithm when compared to other methods. This comparative study includes our magnetic-field-inspired (MFI) navigation system (with both the PD and geometric controllers) alongside several other navigation methods presented in the literature - specifically the standard APF (Khatib, 1985), the circular field (CF) (Haddadin et al., 2011), and the gyroscopic force (GF) (Sabattini et al., 2013). These methods were chosen as they share the same properties as the proposed algorithm, i.e. they are all reactive navigation methods, able to operate without the need for prior environmental mapping and are capable of operating in 3D environments populated with arbitrarily shaped obstacles. The system comparisons focus on trajectory covered and time needed to reach the goal while avoiding obstacles en route. All our simulations and experiments use the Robot Operating System (ROS) as a programming framework (Quigley et al., 2009). The parameter values of the algorithm are summarised in **Table 1**. These parameters are retrieved from trial and error. We choose the parameters which produce the most desirable trajectory in terms of the path length and the time required to cover the path.

6.1 Simulation Results for Point-like Robot

In the first simulation, depicted in **Figure 7**, our MFI algorithm is implemented in a point-like robot moving in a 2D plane containing a sharp corner. In this scenario, **Figure 7A**, we see that it is able to guide the robot towards its goal, using either geometric (MFI+GC) or PD control (MFI+PD). The other three algorithms, namely the APF, CF, and GF all fail to reach the goal. The APF fails due to the cancelling out of the attractive terms towards the goal by the repulsive field from the obstacle. The CF and GF, on the other hand, fail due to the zero speed condition caused by the attractive field towards the goal as seen from the saturated behaviour of the robot trajectory l(t) and positional error e(t) in **Figure 7B**. The reason this issue does not arise with our algorithm (whether using PD or geometric control) is that the collision-avoidance vector field $\mathbf{F_a}$ in **(15)** has the property of repelling the robot without affecting its speed. This is a major advantage over the CF and GF methods. In this 2D scenario, we also observe how the averaging technique described in **Section 4.4** helps the robot avoid the sharp corner despite it having a number of non-unique points of closest distance to the robot.

To test the methods in a more realistic 3D scenario, we designed a forest-like environment consisting of cylinders and spheres that mimic trunks and leaves as shown in **Figures 8A,B**. As before, we can see that the proposed algorithm (whether under geometric or PD control) is able to successfully navigate the robot amidst arbitrarily shaped obstacles towards the desired goal position. The APF and GF methods both fail to circumvent the trees in this environment due to the effect of local minima (in the case of APF) and of the zero-speed problem (in the case of

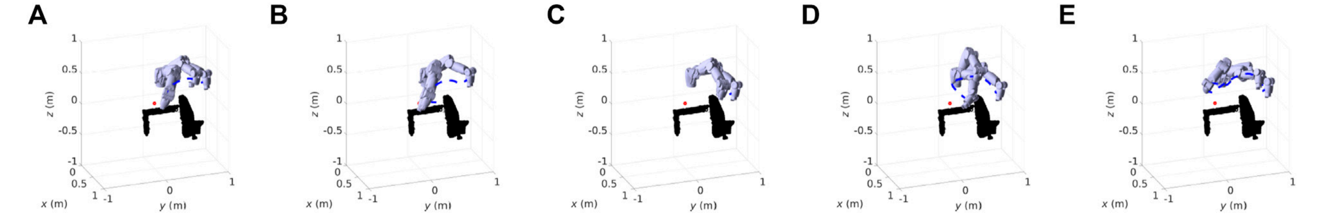

FIGURE 11 | The motion of the Baxter arm in a real scenario with environment consisting of black chair and table (drawn in black) using **(A)** the MFI algorithm with geometric control, **(B)** the MFI algorithm with PD control, **(C)** standard APF method, **(D)** CF method and **(E)** GF method, respectively. The dashed lines are the tip's trajectory.

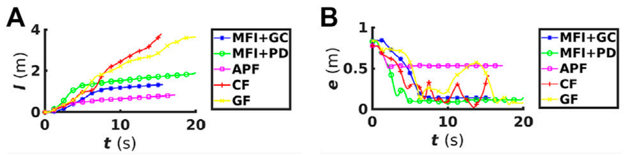

FIGURE 12 | (A) The plot of the trajectory covered by the Baxter's tip l(t) and **(B)** the position error e(t) as function of time for the case of real environment with chair as an obstacle.

GF). The CF method is able to guide the robot to its goal, but does so at the expense of a longer trajectory and convergence time as shown in **Figure 8C** despite its knowledge of the obstacle's centre point.

To further demonstrate the ability of the proposed algorithm in handling realistic scenarios, **Figure 9** shows the trajectory of the robot in more challenging environments that incorporate a tree (**Figure 9A**), a human (**Figure 9B**), and a castle (**Figure 9C**). In all these scenarios, we see that the proposed MFI algorithm is able to navigate the robot through complex environments that are cluttered with a range of obstacles differing in shape, convexity and surface continuity, while relying only on a range sensor and without any pre-knowledge of the environment. This last point is crucial to the algorithms potential application in which environment maps are unavailable such as in post-disaster sites or open spaces with unpredictable landscapes.

6.2 Experimental Results Using Baxter Arm

In our experimental implementation, the proposed algorithm's guidance capabilities were tested to guide the tip of a 7-DOFs Baxter Arm as depicted in **Figure 10**. The environment consisted of a chair whose location was unknown to the robot at the outset. The Kinect RGB-D camera was used to detect the position of the obstacle and to feed that information to the robot only once the tip of the robot's arm got closer to the obstacle than limit distance r_s. **Figure 11** shows the movement of the Baxter arm using the proposed MFI with geometric control (**Figure 11A**), MFI with PD control (**Figure 11B**), the APF method (**Figure 11C**), the CF method (**Figure 11D**), and the GF method (**Figure 11E**). We see that the proposed algorithm with geometric control (**Figure 11A**) is able to guide the robot towards the target while smoothly avoiding the obstacle. The APF method (**Figure 11E**) results in local minima issues, while the other algorithms produce non-smooth robotic arm movements.

To further highlight the superiority of the proposed algorithm, we present the performance comparison between the five methods in **Figure 12**. Bar the APF method, which fails to guide the robot to the goal, the others show similar performance to our MFI algorithm in terms of the time the robot needs to reach the goal - as evidenced by the plot of error e(t). It is noted, however, that the CF and GF algorithms fail to fluidly guide the tip to its goal in the real scenario - as evident from the unstable value of error e(t) in **Figure 12**. This is due to the non-existence of the collision-avoidance vector field in the case of CF and the use of the repulsive term causing the tip to be repelled too far from its path in the case of GF. In terms of the trajectory taken by the tip, we can see that the proposed algorithm with geometric control and the same with PD control take the shortest and second shortest paths, respectively.

6.3 Discussion

Table 2 summarizes the simulation and experimental results, comparing the artificial potential field (APF) method, the circular field (CF) method, the gyroscopic force (GF) method, and our magnetic-field-inspired (MFI) algorithm both with PD control and with geometric control (GC). The comparisons among these navigation methods are conducted in terms of their abilities in obstacle avoidance, goal reaching, and path quality. The latter parameter is judged by the length of trajectory l(t) and convergence time, i.e. a time needed for the distance error e(t) to fall below a specified value e_b. When an algorithm fails to navigate the robot to its desired target, these variables are not taken into account. We chose the distance threshold

TABLE 2 | Summary of Results.

Robot	Obstacle	Algorithm	Success	Covered path (m)	Time (s)
Point-Like Robot 2D	Sharp Corner	APF	×	—	—
		CF	×	—	—
		GF	×	—	—
		MFI + PD	✓	70.51	61.85
		MFI + GC	✓	69.64	41.03
Point-Like Robot 3D	Forest	APF	×	—	—
		CF	✓	55.80	86.13
		GF	×	—	—
		MFI + PD	✓	40.46	37.95
		MFI + GC	✓	38.91	26.08
Baxter Experiment	Chair	APF	×	—	—
		CF	✓	3.80	6.58
		GF	✓	3.69	16.3
		MFI + PD	✓	2.03	3.08
		MFI + GC	✓	1.35	6.21

parameter to be set at $e_b = 0.05r_{gi}$ in a simulation scenario in which r_{gi} denotes the initial distance between the robot and its target at the outset, but opted for e_b = 0.2 m for the experimental setup, taking into account the steady-state positional error of the Baxter's tip.

Table 2 indicates that in both the simulation and the experimental setup, our proposed reactive magnetic-inspired-field navigation algorithm outperforms other navigation methods in almost every tested scenario. This is of particular note in regard to two aspects: the ability of the algorithm to successfully navigate the robot to its desired target amidst an obstacle-laden environment and its fast convergence time of the positional error. We see that the proposed MFI algorithms, both with PD control or geometric control, are successful in every scenario. Our algorithms are significantly more successful in environments with concave obstacles, such as those with sharp corners. They also outperform others in environments consisting of arbitrarily shaped obstacles, such as a forest configuration. In all simulation scenarios, the proposed method with GC control invariably comes first in terms of its convergence speed. This is principally due to the fact that the whole vector field (both the goal attraction and the obstacle avoidance terms) does not influence the robot's speed, other than at the very beginning and the final part of the robot's motion. Indeed it is this constant speed that facilitates obstacle avoidance, enabling fast and fluid movement around the boundary of any obstacle. In addition to convergence speed, the proposed algorithms also achieve the shortest trajectory - both in the simulation as well as the experimental setup.

Most importantly, our MFI navigation algorithm does not rely on prior knowledge of the environment, such as obstacle geometry or location, as is the case with other magnetic-field-inspired navigation methods (Singh et al., 1996; Haddadin et al., 2011). Our algorithm only requires information from a local sensor able to measure the spatial distance to nearby obstacles along with information regarding the robot's speed. Compared to other gyroscopic-based navigation methods, the proposed algorithm is superior since it can be applied not only in 2D environments, but also in 3D environments, with convex obstacles or non-maze concave obstacles, including those obstacles with sharp corners. In terms of computational burden, our algorithm (summarised in **Eq. 11** and **Eq. 15**) only uses simple matrix multiplication process. The same process is also employed by gyroscopic-based methods. In conclusion, our algorithm achieves better performance in general despite having similar computational burden to other reactive navigation methods.

7 CONCLUSION

This paper presents and examines a reactive navigation method that can guide a robot toward a target position within a 3D environment that is laden with arbitrarily shaped convex and non-maze concave obstacles. Drawing inspiration from magnetic field laws, we produce an algorithm capable of steering a robot away from and around obstacles that lie en route to the target. It is demonstrably superior than the standard APF method, as it is free from local minima in environments consisting of both convex and non-maze concave obstacles. Compared to other magnetic-field-inspired navigation methods, ours has a further key advantage in that requires no prior knowledge of the environment, in terms of geometry or obstacle location. The algorithm is shown to successfully navigate a point-like robot model in both $\mathbb{R}^2$ and $\mathbb{R}^3$ and a 7-DOF Baxter arm towards a specified goal in the presence of previously unknown obstacles. The results show that our navigation algorithm has the potential to be used in a wide range of devices, such as flying robots, underwater robots, soft continuum manipulators, and even in swarming multi-robot systems. Future work will consider the limitations of robotic actuators and make in-roads into using such systems in unknown maze-like environment and unknown environments with dynamics obstacles.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

AA and KA contributed to the idea of the algorithm. AA designed the simulation, carried on the experiments, provided mathematical derivations, and prepared the draft of the paper. H-KL provided insightful suggestions on the mathematical proofs behind the algorithm and the structure of the paper. KA supervised the research, provided funding, and help in the draft preparation. All authors read and approved the submitted version.

FUNDING

This work was supported in-part by King's College London, the STIFF-FLOP project grant from the European Communities Seventh Framework Programme under grant agreement 287 728, the EPSRC in the framework of the NCNR (National Centre for Nuclear Robotics) project (EP/R02572X/1), q-bot led project WormBot (2308/104 059), and the Indonesia Endowment Fund for Education, Ministry of Finance Republic of Indonesia.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2022.834177/full#supplementary-material

REFERENCES

Arslan, O., and Koditschek, D. E. (2016). "Exact Robot Navigation Using Power Diagrams," in Proc. IEEE Int. Conf. Robot. Autom., Stockholm, Sweden, May 16–21, 2016. doi:10.1109/ICRA.2016.7487090

Ataka, A., Lam, H.-K., and Althoefer, K. (2018a). Reactive Magnetic-Field-Inspired Navigation Method for Robots in Unknown Convex 3-D Environments. *IEEE Robot. Autom. Lett.* 3, 3583–3590. doi:10.1109/LRA.2018.2853801

Ataka, A., Lam, H. K., and Althoefer, K. (2019). "Magnetic-Field-Inspired Navigation for Quadcopter Robot in Unknown Environments," in International Conference on Robotics and Automation (ICRA), Montreal, Canada, May 20–24, 2019, 6165–6171. doi:10.1109/ICRA.2019.8793682

Ataka, A., Lam, H. K., and Althoefer, K. (2018b). "Reactive Magnetic-Field-Inspired Navigation for Non-holonomic Mobile Robots in Unknown Environments," in Proc. IEEE Int. Conf. Robot. Autom., Brisbane, Australia, May 21–25, 2018, 6983–6988. doi:10.1109/icra.2018.8463203

Brock, O., and Khatib, O. (2002). Elastic Strips: A Framework for Motion Generation in Human Environments. *Int. J. Robotics Res.* 21, 1031–1052. doi:10.1177/0278364902021012002

Bullo, F., and Murray, R. M. (1995). "Proportional Derivative (PD) Control on the Euclidean Group," in In European Control Conference, Rome, Italy, September 5–8, 1995, 1091–1097.

Chang, D. E., and Marsden, J. E. (2003). "Gyroscopic Forces and Collision Avoidance with Convex Obstacles," in *New Trends in Nonlinear Dynamics and Control and Their Applications* (Springer), 145–159.

Choset, H. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA: MIT Press.

Park, D. H., Hoffmann, H., Pastor, P., and Schaal, S. (2008). "Movement Reproduction and Obstacle Avoidance with Dynamic Movement Primitives and Potential fields," in Proc. IEEE-RAS Int. Conf. Humanoid Robots., Daejon, South Korea, December 1–3, 2008, 91–98. doi:10.1109/ICHR.2008.4755937

Garimella, G., Sheckells, M., and Kobilarov, M. (2016). "A Stabilizing Gyroscopic Obstacle Avoidance Controller for Underactuated Systems," in Proc. IEEE Conf. Decision Control, Las Vegas, NV, United States, December 12–14, 2016, 5010–5016. doi:10.1109/CDC.2016.7799035

Garrido, S., Moreno, L., Blanco, D., and Martín Monar, F. (2010). Robotic Motion Using Harmonic Functions and Finite Elements. *J. Intell. Robot Syst.* 59, 57–73. doi:10.1007/s10846-009-9381-3

Haddadin, S., Belder, R., and Albu-Schäffer, A. (2011). Dynamic Motion Planning for Robots in Partially Unknown Environments. *IFAC World Congr* 18. doi:10.3182/20110828-6-it-1002.02500

Halliday, D., Resnick, R., and Walker, J. (2013). *Fundamentals of Physics Extended*. 10th Edn. Wiley.

Justh, E. W., and Krishnaprasad, P. S. (2004). Equilibria and Steering Laws for Planar Formations. *Syst. Control. Lett.* 52, 25–38. doi:10.1016/j.sysconle.2003.10.004

Khatib, O. (1985). "Real-time Obstacle Avoidance for Manipulators and mobile Robots," in Proc. IEEE Int. Conf. Robot. Autom., St. Louis, MO, United States, March 25–28, 1985, 500–505. doi:10.1109/ROBOT.1985.10872472

La Valle, S. M. (2011). Motion Planning. *IEEE Robot. Automat. Mag.* 18, 108–118. doi:10.1109/MRA.2011.941635

Latombe, J.-C. (1991). *Robot Motion Planning*. Norwell, MA: Kluwer Academic Publishers.

LaValle, S. M. (2011). Motion Planning. *IEEE Robot. Automat. Mag.* 18, 79–89. doi:10.1109/MRA.2011.940276

Matveev, A. S., Hoy, M. C., and Savkin, A. V. (2015). A Globally Converging Algorithm for Reactive Robot Navigation Among Moving and Deforming Obstacles. *Automatica* 54, 292–304. doi:10.1016/j.automatica.2015.02.012

Matveev, A. S., Hoy, M. C., and Savkin, A. V. (2013). A Method for Reactive Navigation of Nonholonomic Under-actuated Robots in Maze-like Environments. *Automatica* 49, 1268–1274. doi:10.1016/j.automatica.2013.01.046

Matveev, A. S., Wang, C., and Savkin, A. V. (2012). Real-time Navigation of Mobile Robots in Problems of Border Patrolling and Avoiding Collisions with Moving and Deforming Obstacles. *Robotics Autonomous Syst.* 60, 769–788. doi:10.1016/j.robot.2012.02.006

Millman, R. S., and Parker, G. D. (1977). *Elements of Differential Geometry*. Englewood Cliffs, NJ: Prentice-Hall.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., et al. (2009). "ROS: an Open-Source Robot Operating System," in *ICRA Workshop on Open Source Software*, 3, 5.

Rizqi, A. A. A., Cahyadi, A. I., and Adji, T. B. (2014). "Path Planning and Formation Control via Potential Function for UAV Quadrotor," in Proc. Int. Conf. Adv. Robot. Intell. Syst., Taipei, Taiwan, June 6–8, 2014, 165–170. doi:10.1109/ARIS.2014.6871517

Sabattini, L., Secchi, C., and Fantuzzi, C. (2017). Collision Avoidance for Multiple Lagrangian Dynamical Systems with Gyroscopic Forces. *Int. J. Adv. Robotic Syst.* 14, 172988141668710. doi:10.1177/1729881416687109

Sabattini, L., Secchi, C., and Fantuzzi, C. (2013). "Collision Avoidance Using Gyroscopic Forces for Cooperative Lagrangian Dynamical Systems," in Proc. IEEE Int. Conf. Robot. Autom., Kalsruhe, Germany, May 6–10, 2013, 953–958. doi:10.1109/ICRA.2013.6630688

Savkin, A. V., and Wang, C. (2013). A Simple Biologically Inspired Algorithm for Collision-free Navigation of a Unicycle-like Robot in Dynamic Environments with Moving Obstacles. *Robotica* 31, 993–1001. doi:10.1017/s0263574713000313

Savkin, A. V., and Wang, C. (2014). Seeking a Path Through the Crowd: Robot Navigation in Unknown Dynamic Environments with Moving Obstacles Based on an Integrated Environment Representation. *Robotics Autonomous Syst.* 62, 1568–1580. doi:10.1016/j.robot.2014.05.006

Savkin, A., and Wang, C. (2015). *Safe Robot Navigation Among Moving and Steady Obstacles*. Elsevier Science & Technology Books.

Singh, L., Stephanou, H., and Wen, J. (1996). Real-time Robot Motion Control with Circulatory fields. *Proc. IEEE Int. Conf. Robot. Autom.* 3, 2737–27423. doi:10.1109/ROBOT.1996.506576

Singh, L., Wen, J., and Stephanou, H. (1997). "Motion Planning and Dynamic Control of a Linked Manipulator Using Modified Magnetic fields," in Proc. IEEE Int. Conf. Control Applicat., Hartford, CT, United States, October 5–7, 1997, 9–15. doi:10.1109/CCA.1997.627455

Valero-Gomez, A., Gomez, J. V., Garrido, S., and Moreno, L. (2013). "The Path to Efficiency: Fast Marching Method for Safer, More Efficient Mobile Robot Trajectories," *IEEE Robot. Automat. Mag.*, 20, 111–120. doi:10.1109/MRA.2013.2248309

Vannoy, J., and Jing Xiao, J. (2008). Real-Time Adaptive Motion Planning (RAMP) of Mobile Manipulators in Dynamic Environments with Unforeseen Changes. *IEEE Trans. Robot.* 24, 1199–1212. doi:10.1109/TRO.2008.2003277

Zhang, F., Justh, E., and Krishnaprasad, P. (2004a). Boundary Following Using Gyroscopic Control. *Proc. IEEE Conf. Decis. Control. (Ieee)* 5, 5204–5209. doi:10.1109/cdc.2004.1429634

Zhang, F., O'Connor, A., Luebke, D., and Krishnaprasad, P. (2004b). "Experimental Study of Curvature-Based Control Laws for Obstacle Avoidance," in IEEE Int. Conf. Robot. Autom. (IEEE), New Orleans, LA, United States, April 26–May 1, 2004, 3849–3854. doi:10.1109/robot.2004.1308868

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

APPENDIX: SOLVING $\int \frac{\sin\theta}{\cos\theta} \mathbf{d}\theta = \int (-\frac{\kappa}{1-\kappa \mathbf{r}} + \frac{\mathbf{c}}{\mathbf{mv^2r}})\mathbf{dr}$

Assuming u = cos θ, we have $du = -\sin\theta d\theta$. Hence, the left side of the equation can be transformed into

$$-\int \frac{1}{u} du = -\ln u + E,$$

Where E denotes an integration constant. Defining $v = (1 - \kappa r)$, *we* get $dv = -\kappa dr$. Thus, the right side of the equation can be simplified into

$$\int \frac{1}{v} dv + \frac{c}{mv^2}\int \frac{1}{r} dr = \ln v + \ln r^{\frac{c}{mv^2}} + F,$$

Where F denotes an integration constant. Combining the two terms together, we get

$$\ln\left(vur^{\frac{c}{mv^2}}\right) = G,$$
$$vur^{\frac{c}{mv^2}} = e^G = D,$$

Where G and D are constants. Substituting the value of u and v, we arrive at

$$(1 - \kappa r) r^{\frac{c}{mv^2}} \cos\theta = D,$$

And hence, conclude the derivation.