

# Dynamic Local Laplacian Potential Field for UAV Navigation in Unknown Environments

Xiaocheng Song, Xiaopei Liu and Jie Lu

**Abstract**—This paper develops a dynamic local potential field method for autonomous navigation of Unmanned Aerial Vehicles (UAVs) in unknown cluttered environments. The proposed navigation scheme allows a UAV to dynamically construct a local Laplacian potential field only based on its real-time obstacle detection and to keep track of the resulting reference velocity direction. Specifically, we propose a novel boundary configuration for the local potential field, which integrates the distances to nearby obstacles and a temporary goal for driving the UAV towards the destination or away from the local area with densely scattered obstacles. Under particular boundary conditions, we efficiently solve the Laplace’s equation via the Boundary Element Method, resulting in a harmonic potential field without local optima inside. We show that by following the negative gradient direction in this local Laplacian potential field, the UAV is guaranteed to proceed towards the temporary goal while assuring obstacle avoidance. To make the UAV follow such a reference velocity direction, we design a Linear-Quadratic-Gaussian controller which achieves optimal tracking and whose input can be converted to UAV attitude. With no empirical parameter selections, the UAV is able to safely approach the destination given properly placed temporary goals, and is robust to uncertainties. Simulations on AirSim simulator demonstrate the validity, efficiency, and robustness of the proposed method.

**Index Terms**—Autonomous navigation, online motion planning, online path planning, real-time obstacle avoidance, Laplacian potential field, boundary element method

## I. INTRODUCTION

UNMANNED Aerial Vehicles (UAVs) have a wide range of applications at low altitude, such as traffic monitoring, forest protection, electric power inspection, as well as rescue operations, etc. [1] Autonomous navigation, or path/motion planning, is critical for UAVs to accomplish most of these tasks, which generates a valid trajectory for UAVs to reach any given destination without colliding with obstacles.

To date, a large number of navigation schemes have been proposed for UAV/robot systems. One typical category of navigation schemes is global path planning [2], [3], where the UAV/robot maintains a global map of obstacles distributed in the entire environment so as to plan a collision-free path to the destination via various approaches. Traditional global path planning methods require the knowledge of all the obstacles in the global map so that the path can be computed offline, including the node-based methods such as A\* and Sparse A\* Search (SAS) [4], [5], the Probabilistic Roadmap (PRM) based methods [6], [7], and the Rapidly-exploring Random Tree (RRT) based methods [8]–[11]. Some global path planning

methods, such as the Artificial Potential Field (APF) methods [12], [13], allow the path to be computed online, yet still assume that all obstacles are known in advance. Recently, in order to handle unknown or dynamic environments, real-time global path planning methods were developed, including the variants of some well-known offline methods [14]–[17] and the Interfered Fluid Dynamical System (IFDS) method [18]. In these real-time methods, the UAV/robot keeps updating the global map of obstacles while moving and detecting, resulting in a dynamically generated path.

Although some state-of-the-art global path planning methods are applicable under unknown environments, they all require the UAV/robot to store a *global* map and all obstacle information that may be updated over time, which may cause memory issues. Moreover, the UAV/robot can only move within a limited, fixed-size area. Local path planning can overcome these drawbacks by only utilizing local information of obstacles from onboard sensors to form an online path. This means that there is no need to store a global map, which has a benefit of consuming only a very limited onboard memory. Typical local path planning methods include the APF method [19], the Virtual Force Field (VFF) method [20], the Vector Field Histogram (VFH) method [21], dynamic window approach [22], and the nearness diagram (ND) method [23]. In recent years, learning-based [24], [25] and vision-based [26]–[29] methods are also investigated. Although these local path planning methods only need limited obstacle information during a short period that covers a relatively small area compared to the entire environment, in general there is no guarantee on the validity of the generated paths. In addition to the aforementioned methods, there are a number of multi-layer navigation schemes [30]–[38], which combine online mapping, global and local path planning, state estimation, as well as localization together to form a hybrid method.

In this paper, we propose a novel dynamic local potential field for autonomous UAV navigation. The concept of artificial potential field (APF) is first brought to attention by [19], where attractive forces are defined from the destination and repulsive forces are defined from the obstacles, leading to a potential function for realizing real-time obstacle avoidance. Due to the simplicity and efficacy in practical implementations, this method is also integrated with other path planning methods [39]–[41]. However, the potential field in [19] often suffers from existence of local minima at which the UAV/robot may be trapped. Although escaping from local minima can be achieved through additional treatment [42], [43], it is more preferable to eliminate all local minima. In [44], the panel method is adopted to describe arbitrarily shaped obstacles by

X. Song, X. Liu and J. Lu are with the School of Information Science and Technology, ShanghaiTech University, 201210 Shanghai, China. Email:{songxch, liuxp, lujie}@shanghaitech.edu.cn

a sequence of panels, and a harmonic function is assigned to define the potential on each panel. Owing to the harmonicity of the overall potential function, there are no local minima inside the potential field. The negative gradient of the harmonic potential function is then chosen as the velocity. However, this method cannot ensure convergence to the destination, which may be influenced by empirical parameter tuning. Instead of directly assigning a harmonic function to form a potential field, [45], [46] solve the Laplace's equation with given boundary conditions by the Finite Difference Method (FDM), resulting in a harmonic Laplacian potential field free from local minima, which is guaranteed to drive the UAV to the destination.

Although the Laplacian potential field methods [45], [46] are simple to implement and have elegant theoretical guarantees, they require the environment to be static and known, since the global obstacle information is incorporated in computing the potential field. This paper takes advantage of the Laplacian potential field to design an autonomous UAV navigation scheme in the context of *unknown* obstacle distributions. Specifically, we assume that a UAV is placed in a completely unknown, cluttered environment, and is required to reach an arbitrarily given destination point. Also suppose that the UAV sensing range is limited compared with its initial distance to the destination. For simplicity, we let the UAV maintain a constant altitude, so that this paper focuses on two-dimensional Laplacian potential fields for two-dimensional navigation.

Our proposed navigation scheme has the following hierarchical structure consisting of a motion planning module and a motion control module: The motion planning module constructs a local Laplacian potential field that is dynamically updated via the latest detection of nearby obstacles, and then yields a reference velocity direction for the UAV to track when moving in such a local potential field. The motion control module enables the UAV to quickly reach the real-time reference velocity direction provided by the motion planning module. By repeating such a process, the UAV can eventually reach the destination while avoiding all the obstacles. The motion planning and control modules are designed as follows:

In the motion planning module, to construct a local Laplacian potential field, we first configure the potential field boundary via the distances between the UAV and the detected obstacles as well as a temporary goal that aims at guiding the UAV to proceed towards the destination or leave the neighborhood surrounded by obstacles. Then, to determine the potential function, we designate its function value or its directional derivatives on the boundary. Thus, the solution to the Laplace's equation with such boundary conditions gives a harmonic potential function. We show that the temporary goal is the unique global minimum of the potential function and there are no other critical points (i.e., local optima and saddle points) inside the potential field. Consequently, as long as the UAV follows the negative gradient direction in the local Laplacian potential field, it is guaranteed to approach the temporary goal without any collision. To solve for the Laplacian potential function at a low cost, we employ an efficient numerical method called the Boundary Element Method (BEM) [47]–[49], which uses only boundary samples during solution and leads to higher accuracy and improved smoothness compared

to the conventional FDM method [45], [46].

In the motion control module, the reference velocity direction for the UAV is set to be the negative gradient direction of the potential function computed in the planning module. To track such a reference, we develop a Linear-Quadratic-Gaussian (LQG) controller, which achieves fast tracking with low overshoots and is robust to disturbances/noises. We also convert the LQG control input in terms of the thrust acceleration to the UAV attitude, so that the control module can be adapted to most real UAV systems.

The advantages of the proposed navigation scheme are highlighted as follows:

- The dynamic local Laplacian potential field is established only from the UAV's real-time detection of obstacles. Unlike many existing methods, the proposed navigation scheme does not require the UAV to store any historical detection results or maintain a global map of the entire environment, so that it is memory-efficient and applicable to wide geographical regions without any prior knowledge of obstacle distributions.
- Given the local Laplacian potential field at any time instant, the streamlines throughout its negative gradient field are smooth, pointing to the temporary goal and opposite to the obstacles. Therefore, the resulting trajectory of the UAV following the negative gradient direction is smooth and safe, leveraging both path length and collision avoidance [50].
- Different from many alternative methods that provide an explicit path for the UAV to track, here the UAV only needs to track a reference velocity direction, which is robust to disturbances and path integral errors.
- The proposed scheme is computationally efficient and effective for online autonomous navigation under completely unknown environments, as is demonstrated via simulations on *AirSim* UAV simulator. Moreover, it is parameter-free in the sense that essentially no efforts are needed in empirically selecting the algorithm parameters.

The rest of this paper is organized as follows: Section II develops the motion planning module that constructs local Laplacian potential fields and provide theoretical guarantees. Section III presents the motion control module that tracks the negative gradient direction of the potential function. Section IV shows the simulation results. Finally, Section V concludes the paper and discusses possible future directions.

## II. MOTION PLANNING BY DYNAMIC LOCAL LAPLACIAN POTENTIAL FIELD

To plan the UAV motion, we develop a dynamic local Laplacian potential field, which is constantly updated using the UAV's real-time detection of nearby obstacles and indicates a reference velocity direction for the UAV. In this section, we investigate the construction of the local Laplacian potential field and show that its negative gradient direction is a legitimate velocity direction for the UAV.

To construct the Laplacian potential field, we first consider the Laplace's equation

$$\sum_{i=1}^n \frac{\partial^2 \phi}{\partial x_i^2} \equiv 0,$$

which is a second-order partial differential equation with  $n$  independent variables. Let  $\Omega$  be a domain (i.e., a non-empty open subset of  $\mathbb{R}^n$ ). If a twice continuously differentiable, real-valued function  $\phi$  satisfies the Laplace's equation on  $\Omega$ , then  $\phi$  is said to be *harmonic*. Harmonic functions are real analytic and smooth.

We assume the UAV remain a constant altitude, so that the local potential field to be constructed is two-dimensional. Let the domain  $\Omega \subset \mathbb{R}^2$  be bounded and simply-connected, and we will discuss how to shape  $\Omega$  shortly. Also, let  $\partial\Omega$  be the boundary of  $\Omega$  and  $\bar{\Omega} = \partial\Omega \cup \Omega$  be the closure of  $\Omega$ . To find a harmonic function  $\phi : \bar{\Omega} \rightarrow \mathbb{R}$ , we need to address the *Boundary Value Problem* (BVP), i.e., solve the two-dimensional Laplace's equation

$$\frac{\partial^2 \phi(\mathbf{x})}{\partial x_1^2} + \frac{\partial^2 \phi(\mathbf{x})}{\partial x_2^2} = 0, \quad \forall \mathbf{x} \in \Omega \quad (1)$$

with given boundary conditions. There are two classic boundary conditions for BVP. One is the Dirichlet boundary condition, which designates a value of  $\phi(\mathbf{x})$  at every point  $\mathbf{x}$  on the boundary  $\partial\Omega$ . The other is the Neumann boundary condition, which specifies the directional derivatives of  $\phi$  at each  $\mathbf{x} \in \partial\Omega$  along the outward normal. Mixtures of the Dirichlet condition and the Neumann condition are also commonly used. Thus, we may choose  $\Gamma_1, \Gamma_2 \subset \partial\Omega$  with  $\Gamma_1 \cap \Gamma_2 = \emptyset$  and  $\Gamma_1 \cup \Gamma_2 = \partial\Omega$ , and let the Dirichlet boundary condition hold on  $\Gamma_1$  and the Neumann condition hold on  $\Gamma_2$ . Then, a nonconstant harmonic solution  $\phi$  to such BVP is used as the potential function for the local potential field.

### A. Boundary Configuration

To address the above BVP problem with a mixture of the Dirichlet condition and the Neumann condition, our first step is to form the boundary of the local potential field based on the UAV's real-time detection of nearby obstacles.

In this paper, we do not focus on the detection/sensing problem, and simply assume that the UAV is able to obtain the radial distances to the obstacles within its sensing range, which can be realized by means of various sensors such as LiDARs and stereo-cameras. Suppose the UAV is located at point  $P$  when it performs sensing, and the sensing range is the disk  $\mathcal{B}$  centered at  $P$  with radius  $r > 0$ . Suppose the UAV performs sensing by shooting rays along sampled directions from the entire  $2\pi$ . We terminate each ray if it hits an obstacle within  $\mathcal{B}$ ; otherwise, the ray stops when it reaches the boundary of  $\mathcal{B}$ . Then, by connecting all the endpoints of the rays, a local visible region  $\mathcal{F} \subseteq \mathcal{B}$  for the UAV is formed, in which no obstacle exists. Fig. 1a illustrates an example where there are three obstacles in the sensing range  $\mathcal{B}$ , and the local visible region  $\mathcal{F}$  is the blue area in the figure.

In addition to obstacle avoidance, we also need to drive the UAV to the destination  $D$ . Ideally, we would like to

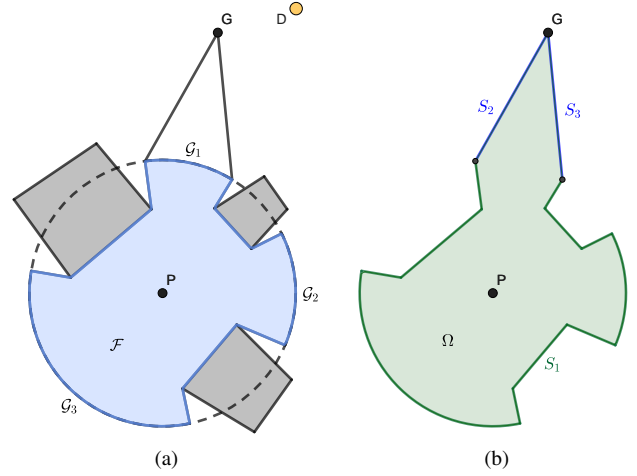


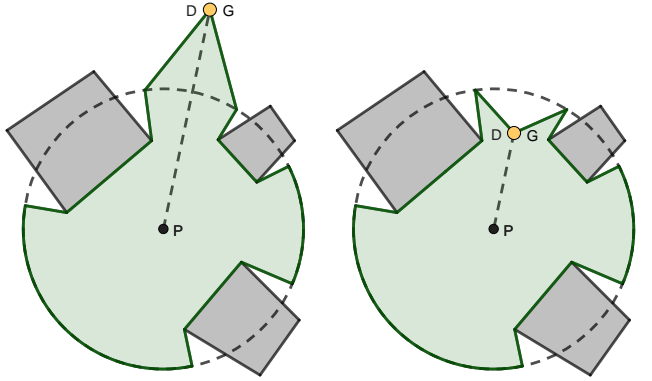
Fig. 1. Boundary configuration of local potential field. The gray objects are the obstacles, the blue area is the local visible region  $\mathcal{F}$ , the green area is the potential field region  $\Omega$ , and the yellow dot is the destination  $D$ .

connect  $\mathcal{F}$  to the destination  $D$  and generate a potential field on the resulting region. However, the destination  $D$  may be very far from the UAV and may also be hidden behind the detected obstacles, which could cause the shape of  $\Omega$  to be pathological so that the effect of the resulting potential field would be hampered. To address such issues, we artificially set a *temporary goal*  $G$  to be connected with  $\mathcal{F}$  and aim at driving the UAV towards  $G$ . The temporary goal  $G$  intends to help the UAV approach the destination direction or leave the local area in which the obstacles are too dense. It will be updated with the local Laplacian potential field after some time interval, and should successively move towards the destination  $D$ . We will discuss the strategy of placing  $G$  shortly.

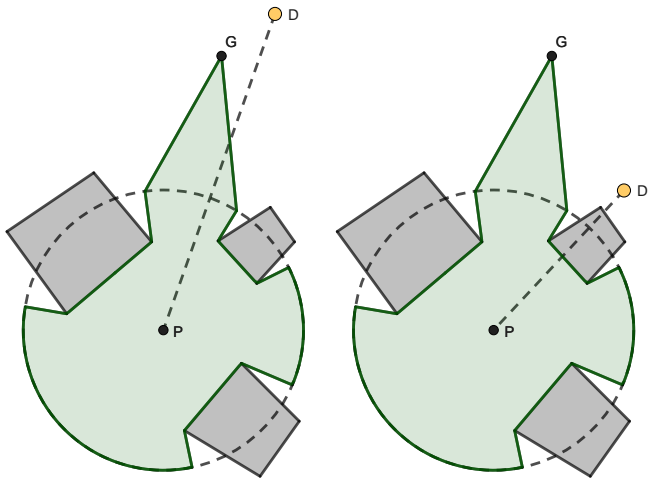
To connect the local visible region  $\mathcal{F}$  and the temporary goal  $G$ , note that the boundaries of  $\mathcal{F}$  and  $\mathcal{B}$ , i.e.,  $\partial\mathcal{F}$  and  $\partial\mathcal{B}$ , intersect at one or more arcs. We call each of these arcs a *gap*  $\mathcal{G}_i$ . In the example in Fig. 1a, there are three gaps  $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3$ . Then, we select one gap, say,  $\mathcal{G}_1$ , and connect each of its endpoints to the temporary goal  $G$  via a line segment. These two line segments, denoted by  $S_2$  and  $S_3$ , together with  $\partial\mathcal{F}$  excluding the selected gap  $\mathcal{G}_1$ , form the boundary of  $\Omega$ . Namely,  $\partial\Omega$  is the union of  $S_1, S_2, S_3$ , where  $S_1 = \partial\mathcal{F} \setminus \mathcal{G}_1$ . Figure 1b exemplifies such an  $\Omega$  determined by  $S_1, S_2, S_3$ .

It remains to discuss the selections of the temporary goal  $G$  and the gap through which the local visible region  $\mathcal{F}$  is connected to  $G$ . There are various ways of placing the temporary goal, and we propose the following strategy:

- If (a) the distance between the sensing position  $P$  and the destination  $D$  is less than or equal to some  $d > r$ , where  $r$  is the sensing radius, and (b) the selected gap covers the destination direction, i.e., the ray starting from  $P$  and passing through  $D$  intersects the selected gap, we set the temporary goal  $G$  to  $D$ .
- If neither of the above two conditions holds, we place  $G$  on the ray that starts from  $P$  and passes through the middle point of the selected gap such that the distance between  $G$  and  $P$  is  $d$ .



(a) The selected gap covers the destination direction and the destination is close  
 (b) The selected gap covers the destination direction and the destination is inside the sensing range



(c) The destination is far away  
 (d) The selected gap cannot cover the destination direction

Fig. 2. Temporary goal placement.

With this strategy, the temporary goal  $G$  is set to be the final destination  $D$  if and only if  $D$  is not far away and can be “seen” by the UAV through the selected gap. Fig. 2 illustrates our strategy of placing  $G$  in different scenarios.

To select a proper gap, we may choose the gap with  $\max \left\{ \left\langle \overrightarrow{PD}, \mathbf{v}_1 \right\rangle, \left\langle \overrightarrow{PD}, \mathbf{v}_2 \right\rangle \right\}$ , where  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are the two-dimensional vectors from  $P$  to the two endpoints of the gap. This guarantees that the selected gap either covers the destination direction or the directions towards  $G$  and  $D$  are as close as possible.

**Boundary Condition:** With the above boundary configuration, we impose the Dirichlet condition  $\phi(\mathbf{x}) = M$ ,  $M > 0$  for the points on  $S_1$  (including the endpoints). At the temporary goal  $G$ , another Dirichlet condition  $\phi(\mathbf{x}) = m$  with  $0 \leq m < M$  is imposed. For each point  $\mathbf{x}$  on  $S_2$  and  $S_3$  except the endpoints, the Neumann condition is used, which requires the directional derivative  $\frac{\partial \phi(\mathbf{x})}{\partial \mathbf{n}_x}$  of  $\phi$  at  $\mathbf{x}$  along the outward normal  $\mathbf{n}_x$  to be zero.

**Remark 1.** Further improvements on the above boundary configuration can be made to obtain more effective local

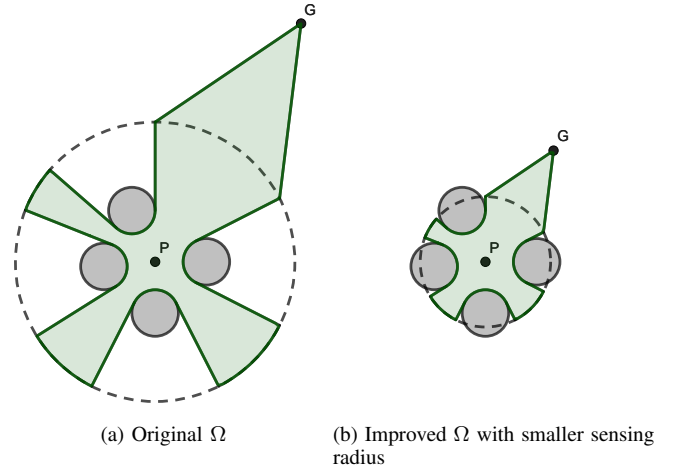


Fig. 3. Shrinkage of sensing radius. The gray objects are the obstacles and the green area is  $\Omega$ .

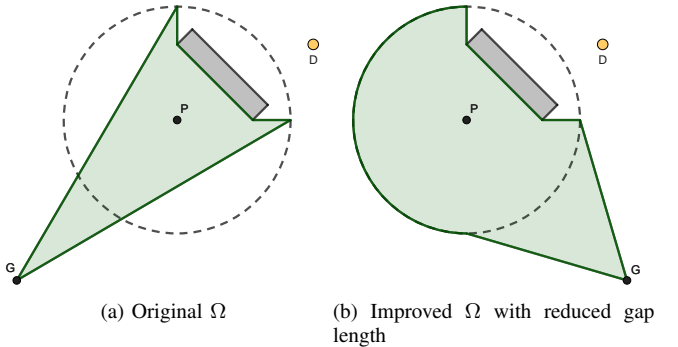


Fig. 4. Shrinkage of gap length. The gray objects are the obstacles, the green area is  $\Omega$ , and the yellow dot is the destination  $D$ .

*Laplacian potential fields for motion planning.* For instance, when the obstacles are densely scattered, there may be a lot of long narrow areas in the local visible region  $\mathcal{F}$ , as shown in Fig. 3a, which may cause an ill-conditioned local Laplacian potential field. We can overcome this by artificially decreasing the sensing radius  $r$ , so that the original local visible region shrinks as shown in Fig. 3b. By doing so, the boundary of the resulting potential field is more smooth. On the other hand, when there are few surrounding obstacles, the length of the selected gap may be very large. According to the above rule in placing the temporary goal  $G$ , the resulting  $\Omega$  may be narrow and  $G$  may even be unnecessarily deviated from  $D$ , like the case shown in Fig. 4a. Thus, we may improve the position of  $G$  by reducing the gap length as shown in Fig. 4b. Furthermore, when a gap is too narrow, we can remove it for safety.

## B. Properties of Laplacian Potential Field

Now suppose we obtain a non-constant harmonic function  $\phi$  on  $\Omega$  by solving (1) with the mixed boundary conditions in Section II-A, and we use  $\phi$  as the potential function. The computation of  $\phi$  will be discussed later in Section II-C. In this subsection, we analyze several favorable properties of such a

Laplacian potential field built upon  $\bar{\Omega}$ , which shed some lights on determining a legitimate reference direction for the UAV.

First of all, consider the following universal properties of harmonic functions:

**Proposition 1** ([51], [52]). *Let  $\Omega'$  be a bounded open connected set and  $\bar{\Omega}'$  be its closure. Let  $\phi'$  be a non-constant harmonic function defined on  $\bar{\Omega}'$ .*

- (i) [Maximum/Minimum Principle] *For any  $\mathbf{x} \in \Omega'$ ,  $\min_{\mathbf{y} \in \partial\Omega'} \phi'(\mathbf{y}) < \phi'(\mathbf{x}) < \max_{\mathbf{y} \in \partial\Omega'} \phi'(\mathbf{y})$ , where  $\partial\Omega'$  is the boundary of  $\Omega'$ .*
- (ii) *Every critical point (i.e., point at which the gradient is zero) of  $\phi'$  in  $\Omega'$  is a saddle point.*

Proposition 1(i) indicates that the maximum and minimum values of our potential function  $\phi$  on  $\bar{\Omega}$  can never be achieved in the interior of the potential field (i.e.,  $\Omega$ ). Proposition 1(ii) states that there is no local extremum (i.e., maximum/minimum) of  $\phi$  in  $\Omega$ .

Owing to the particular boundary configuration in Section II-A, we can say more about the potential function  $\phi$ . Recall that our boundary condition requires  $\phi$  to take the value of  $m$  at the temporary goal  $G$  and take the value of  $M$  on  $S_1$ . The following theorem shows that  $\phi$  is indeed bounded between  $m$  and  $M$  throughout the entire potential field.

**Theorem 1.** *Under the boundary configuration in Section II-A,  $m \leq \phi(\mathbf{x}) \leq M, \forall \mathbf{x} \in \bar{\Omega}$ .*

*Proof.* We first show that  $\phi(\mathbf{x}) \leq M, \forall \mathbf{x} \in \partial\Omega$ . To this end, we extend the harmonic function  $\phi$  to a bounded open connected set  $\hat{\Omega}$  containing  $\bar{\Omega}$ , which is also harmonic on  $\hat{\Omega}$ . Then, assume to the contrary that there exists  $\mathbf{x}_0 \in \partial\Omega$  such that  $\phi(\mathbf{x}_0) = \max_{\mathbf{x} \in \partial\Omega} \phi(\mathbf{x}) > M$ . Due to the boundary condition in Section II-A, we know that  $\mathbf{x}_0$  belongs to  $S_2 \cup S_3$  excluding the endpoints. Thus, the directional derivative of  $\phi$  at  $\mathbf{x}_0$  along the tangent to  $\partial\Omega$  is zero. In addition, recall that the boundary condition requires the directional derivative of  $\phi$  at  $\mathbf{x}_0$  along the normal to be zero. As a result, the gradient of  $\phi$  at  $\mathbf{x}_0$  is zero and, thus,  $\mathbf{x}_0$  is a critical point of  $\phi$  in  $\hat{\Omega}$ . Due to Proposition 1(ii) (with  $\phi' = \phi$  and  $\Omega' = \hat{\Omega}$ ) and because each critical point of a non-constant analytic function is isolated [53],  $\mathbf{x}_0$  is an isolated saddle point. This implies that there exists  $\mathbf{x}_1 \in \Omega$  in a neighborhood of  $\mathbf{x}_0$  such that  $\phi(\mathbf{x}_1) > \phi(\mathbf{x}_0)$ . This contradicts Proposition 1(i) (with  $\phi' = \phi$  and  $\Omega' = \hat{\Omega}$ ). Therefore,  $\phi(\mathbf{x}) \leq M, \forall \mathbf{x} \in \partial\Omega$ . Likewise, it can be shown that  $\phi(\mathbf{x}) \geq m, \forall \mathbf{x} \in \partial\Omega$ . Finally, due to Proposition 1(i) again,  $m = \min_{\mathbf{y} \in \partial\Omega} \phi(\mathbf{y}) \leq \phi(\mathbf{x}) \leq \max_{\mathbf{y} \in \partial\Omega} \phi(\mathbf{y}) = M, \forall \mathbf{x} \in \bar{\Omega}$ .  $\square$

Theorem 1, along with Proposition 1, implies that the temporary goal  $G$  is a global minimum of  $\phi$  on  $\bar{\Omega}$ , the points on  $S_1$  are global maxima, and all the interior points are neither local minima nor local maxima. Nevertheless, except the range  $[m, M]$ , it does not disclose the potential value on  $S_2$  and  $S_3$ . The theorem below further characterizes  $\phi$  and its level curves.

**Theorem 2.** *Consider the boundary configuration in Section II-A.*

- (i) *Each level curve given by  $\phi(\mathbf{x}) = c, m < c < M$  intersects  $S_2$  exactly once and  $S_3$  exactly once;*

- (ii) *For any  $c_1, c_2 \in [m, M]$  with  $c_1 < c_2$ ,  $\{\mathbf{x} \in \bar{\Omega} : \phi(\mathbf{x}) \leq c_1\} \subsetneq \{\mathbf{x} \in \bar{\Omega} : \phi(\mathbf{x}) \leq c_2\}$ ;*
- (iii)  *$\phi$  is monotonically decreasing along  $S_2$  and  $S_3$ , respectively, pointing towards  $G$ .*

*Proof.* We first introduce the following notations in the proof: For any two points  $A, B \in \bar{\Omega}$ ,  $\overline{AB}$  represents the line segment connecting  $A$  and  $B$ . For any  $c \in [m, M]$ , we use  $L(c)$  to denote the level curve given by  $\phi(\mathbf{x}) = c$  and  $S(c)$  to denote the level set  $\{\mathbf{x} \in \bar{\Omega} : \phi(\mathbf{x}) \leq c\}$ .

(i) Note that for any  $\mathbf{x} = [x_1 \ x_2]^T$  on  $S_2$  (including the endpoints), we can express  $\mathbf{x}$  as  $x_1 = g_1(t)$  and  $x_2 = g_2(t)$ , where  $g_1, g_2$  are two analytic scalar functions. Since a composition of analytic functions is analytic,  $f(t) := \phi(g_1(t), g_2(t))$ , the expression of  $\phi$  restricted to  $S_2$  is an analytic scalar function. Since the values of  $\phi$  at the two endpoints of  $S_2$  are  $m$  and  $M$ ,  $f(t)$  is not a constant function on  $S_2$ . Because the zeros of a non-constant analytic scalar function are guaranteed to be isolated and because adding a constant to any analytic function yields an analytic function, there does not exist a continuum of points on  $S_2$  with the same value of  $\phi$ . The same conclusion can be made for  $S_3$ .

On the other hand, consider the level curve  $L(c), c \in (m, M)$ . Clearly,  $L(c)$  does not intersect  $S_1$ . Assume to the contrary that  $L(c)$  intersects  $S_2$  at two distinct points  $A$  and  $B$ . Let  $V \subset \bar{\Omega}$  be the compact set enclosed by  $\overline{AB}$  and  $L(c) \cap \bar{\Omega}$ . Then, using the same rationale as the proof of Theorem 1 (with  $\bar{\Omega}$  replaced by  $V$ ), there cannot be any  $\mathbf{x}$  on the line segment  $\overline{AB}$  such that  $\phi(\mathbf{x}) > c$  or  $\phi(\mathbf{x}) < c$ . In other words,  $\phi(\mathbf{x}) = c, \forall \mathbf{x} \in \overline{AB}$ , which cannot happen due to the conclusion in the last paragraph. Therefore,  $L(c)$  cannot intersect  $S_2$  (and  $S_3$  likewise) more than once. Furthermore, it is shown in [54] that the level curves of a non-constant harmonic function are regular (i.e., locally homeomorphic to parallel lines) except at the critical points and that the level curves are not closed and never terminate inside the domain. This implies that  $C$  has to intersect  $S_2$  exactly once and  $S_3$  exactly once.

(ii) First, consider  $m < c_1 < c_2 < M$ . From (i), the level curve  $L(c_1)$  intersects  $S_2$  at a unique point  $A'$  and intersects  $S_3$  at a unique point  $B'$ . Let the bounded open set enclosed by  $\overline{A'G}, \overline{B'G}$ , and  $L(c_1)$  be denoted by  $\hat{\Omega} \subsetneq \Omega$ . Due to Proposition 1 and Theorem 1,

$$m < \phi(\mathbf{x}) < c_1, \forall \mathbf{x} \in \hat{\Omega}. \quad (2)$$

Note that  $L(c_2)$  and  $S_2$  cannot intersect at any point on  $\overline{A'G}$ , and  $L(c_2)$  and  $S_3$  cannot intersect at any point on  $\overline{B'G}$ —otherwise, part of  $L(c_2)$  would lie in  $\hat{\Omega}$ , which contradicts (2). Hence, the closure of  $\hat{\Omega}$  is exactly the level set  $S(c_1)$ , which is a subset of the level set  $S(c_2)$ . Moreover, due to Theorem 1,  $S(M) = \bar{\Omega}$ , and due to the regularity of the level curves [54],  $S(m) = \{G\}$ . Consequently,  $S(c_1) \subsetneq S(c_2), \forall c_1, c_2 \in [m, M], c_1 < c_2$ .

(iii) This is a straightforward consequence from (ii).  $\square$

The proposition below results from Theorem 2, which says that there cannot be any saddle point in the the potential field.

**Proposition 2.** *There is no saddle point of  $\phi$  in  $\bar{\Omega}$ .*

*Proof.* We first show that there is no saddle point in  $\partial\Omega$ . Since all the points on  $S_1$  are global maxima and the temporary goal  $G$  is a global minimum, they cannot be saddle points. For every point on  $S_2$  or  $S_3$  except the endpoints, due to Theorem 2(iii), its gradient cannot be zero and thus it cannot be a saddle point. Next, we assume to the contrary that there exists a saddle point  $\tilde{x}$  in  $\Omega$ . Due to Theorem 1, the level curves passing a sufficiently small neighborhood of  $\tilde{x}$  cannot intersect  $S_1$ . According to the portrait of the level curves near a saddle point and the property that level curves are not closed and do not terminate in  $\bar{\Omega}$  [54], there exists a level curve that passes the neighborhood of  $\tilde{x}$  and intersects  $S_2$  or  $S_3$  twice, which contradicts Theorem 2(i). Therefore,  $\Omega$  also does not include any saddle point.  $\square$

Based on all the above results, we make the following conclusion on the local Laplacian potential field described by the potential function  $\phi$  on  $\bar{\Omega}$ : All the global maxima of  $\phi$  on  $\bar{\Omega}$  lie in  $S_1$ , and the temporary goal  $G$  is the unique global minimum. In addition, there is no critical point (local maximum, local minimum, saddle point) in the interior of the potential field as well as  $S_2$  and  $S_3$  excluding the endpoints. Consequently, if the UAV follows the *negative gradient direction* of  $\phi$  at its position  $\mathbf{x}$ , the potential value  $\phi(\mathbf{x})$  keeps decreasing, so that the UAV behaves as follows:

- The UAV, upon establishing the current local potential field, keeps away from  $S_1$  and thus avoids colliding with the detected obstacles.
- The UAV is never trapped at any point in the potential field except the temporary goal  $G$ .
- The UAV is moving towards the temporary goal  $G$ .

**Remark 2.** *The negative gradient direction of the potential function at the UAV's current position gives a real-time reference velocity direction for the UAV to track, while many existing path planning methods require the UAV to follow explicit paths instead of velocity directions. When the UAV is deviated from its desired position due to disturbances or path integral errors, those path-following methods are forced to re-plan a path frequently. In contrast, as long as the UAV stays in the current local potential field, our method allows the UAV to effortlessly find a new reference velocity direction and the UAV will still be directed towards the temporary goal from the new position.*

Due to the unknown environment and the limited sensing range, the UAV has to repeatedly conduct sensing and update its local Laplacian potential field until it reaches the final destination. The UAV may choose to update the potential field at a fixed frequency, or only if its distance to the final destination is sufficiently decreased in order to reduce the computational load. Also, there might be obstacles in the area that is between  $S_2$  and  $S_3$  and out of the sensing range. Thus, the potential field must be updated before the UAV enters that area. Note that we do not force the UAV to reach the temporary goal each time. The temporary goal is only for the purpose of directing the UAV to a legitimate direction for both navigation and obstacle avoidance.

In most real scenarios, the UAV is able to reach the final destination if it keeps following the negative gradient direction in the dynamically updated local Laplacian potential field. Admittedly, in some pathological scenarios (e.g., a long corridor with an obstacle at one end and the destination behind that obstacle), the UAV may keep moving back and forth and be trapped in a small region, which, in fact, is a common issue for the existing local planning methods. However, if we allow the UAV to maintain a small amount of certain historical information such as the temporary goal positions during a short period in the past, we can always figure out strategies for the UAV to escape from such traps by selecting a temporary goal that is sufficiently different from the previous ones.

### C. Numerical Computation of Potential Function

This subsection provides an efficient numerical method to compute the potential function  $\phi$  as well as its gradient given any position on  $\Omega$ . To solve (1) with the aforementioned mixed boundary conditions, we may equivalently solve the *Boundary Integral Equation* (BIE) [48], [49], which is formulated as follows: For any  $\mathbf{x} \in \bar{\Omega}$ ,

$$\epsilon(\mathbf{x})\phi(\mathbf{x}) = \int_{\partial\Omega} [G(\mathbf{x}, \mathbf{y})q(\mathbf{y}) - H(\mathbf{x}, \mathbf{y})\phi(\mathbf{y})] dS(\mathbf{y}), \quad (3)$$

where  $q(\mathbf{y}) := \frac{\partial\phi(\mathbf{y})}{\partial\mathbf{n}_y}$  represents the directional derivatives along the outward normal, and  $S(\cdot)$  represents the boundary curve,

$$\begin{aligned} \epsilon(\mathbf{x}) &= \begin{cases} 1, & \mathbf{x} \in \Omega \\ \frac{1}{2}, & \mathbf{x} \in \partial\Omega \end{cases}, \\ G(\mathbf{x}, \mathbf{y}) &= \frac{1}{2\pi} \ln\left(\frac{1}{r}\right), \\ H(\mathbf{x}, \mathbf{y}) &= \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial\mathbf{n}_y} = -\frac{1}{2\pi r} \frac{\partial r}{\partial\mathbf{n}_y}, \end{aligned}$$

and  $r$  is the Euclidean distance between  $\mathbf{x} \in \bar{\Omega}$  and  $\mathbf{y} \in \partial\Omega$ . Note that  $G(\mathbf{x}, \mathbf{y})$  is also known as the *Green's function* and  $H(\mathbf{x}, \mathbf{y})$  is the directional derivative of  $G(\mathbf{x}, \mathbf{y})$  along the outward normal at  $\mathbf{y}$ .

Since there is no general analytic solution to BIE with irregular boundaries, we solve BIE numerically via a discretization approach called *Boundary Element Method* (BEM) [47]–[49]. Since  $\partial\Omega = S_1 \cup S_2 \cup S_3$ , we first discretize  $S_1$ ,  $S_2$ ,  $S_3$ , respectively, into a finite number of elements (i.e., line segments)  $\Delta S_1, \dots, \Delta S_N$ . Then, the above BIE can be approximated by

$$\frac{1}{2}\phi_i = \sum_{j=1}^N [G_{ij}q_j - H_{ij}\phi_j], \quad \forall i = 1, \dots, N, \quad (4)$$

where  $\phi_i = \phi(\mathbf{p}_i)$ ,  $q_i = \frac{\partial\phi(\mathbf{x})}{\partial\mathbf{n}_x}|_{\mathbf{x}=\mathbf{p}_i}$ ,  $\mathbf{p}_i$  is the center point of  $\Delta S_i$ , and

$$G_{ij} = \int_{\Delta S_j} G(\mathbf{p}_i, \mathbf{y}) dS(\mathbf{y}), \quad (5)$$

$$H_{ij} = \int_{\Delta S_j} H(\mathbf{p}_i, \mathbf{y}) dS(\mathbf{y}). \quad (6)$$

Note from the mixed boundary conditions in Section II-A that for each  $i = 1, \dots, N$ , only one of  $\phi_i$  and  $q_i$  has a designated value and the other one is an unknown. To include the requirement that the value of  $\phi$  is  $m$  at the temporary goal  $G$ , if  $\Delta S_i$  includes  $G$  as an endpoint, we set  $\phi_i$  to be  $m$  and set  $q_i$  to be unknown. In addition, the line integrals in (5) and (6) can be simply calculated using either Gauss quadrature or analytical solutions [48].

Let  $\phi = (\phi_1, \dots, \phi_N)$ ,  $\mathbf{q} = (q_1, \dots, q_N)$ ,  $\mathbf{G}$  be the matrix whose  $(i, j)$ -entry is  $G_{ij}$ , and  $\mathbf{H}$  be the matrix whose  $(i, j)$ -entry is  $H_{ij}$ . Then, we can write (4) as

$$\frac{1}{2}\phi + \mathbf{H}\phi = \mathbf{G}\mathbf{q}, \quad (7)$$

in which the number of the unknown entries in  $\phi$  and  $\mathbf{q}$  is exactly  $N$ . Note that (7) can be further transformed into a standard linear equation via elementary column operations, whose solution provides all the unknown  $\phi_i$ 's and  $q_i$ 's.

Once all the  $\phi_i$ 's and  $q_i$ 's are obtained, we can approximate the gradient of  $\phi$  at any  $\mathbf{x} = [x_1 \ x_2]^T \in \bar{\Omega}$ , i.e.,  $[\frac{\partial\phi(\mathbf{x})}{\partial x_1} \ \frac{\partial\phi(\mathbf{x})}{\partial x_2}]^T$ , by means of

$$\begin{aligned} \frac{\partial\phi(\mathbf{x})}{\partial x_k} \approx & \sum_{j=1}^N \left[ \left( \int_{\Delta S_j} \frac{\partial G(\mathbf{x}, \mathbf{y})}{\partial x_k} dS(\mathbf{y}) \right) q_j \right. \\ & \left. - \left( \int_{\Delta S_j} \frac{\partial H(\mathbf{x}, \mathbf{y})}{\partial x_k} dS(\mathbf{y}) \right) \phi_j \right], \forall k \in \{1, 2\}, \end{aligned}$$

where the integrals are computed analytically based on [48].

### III. REAL-TIME MOTION CONTROL

With the local Laplacian potential field constructed in Section II, the UAV chooses the negative gradient direction of the potential function at its current position as the reference velocity direction. This section provides a controller for tracking such a reference velocity direction.

Without loss of generality, we adopt the ground coordinate systems (GCS) that follows the North-East-Down (NED) convention, where the  $x$ -axis points to the north, the  $y$ -axis points to the east, and the  $z$ -axis points downwards. Again, we assume that the altitude of the UAV is constant, so that  $(x, y)$ -coordinates are sufficient for describing the UAV direction. Our first step is to model the UAV dynamics as the following linear dynamical system:

$$\begin{aligned} \dot{\mathbf{v}} &= \mathbf{A}\mathbf{v} + \mathbf{u} + \mathbf{w} \\ \rho &= \mathbf{v} + \tilde{\mathbf{w}}, \end{aligned} \quad (8)$$

where

$$\mathbf{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_x \\ u_y \end{bmatrix}, \quad \rho = \begin{bmatrix} \rho_x \\ \rho_y \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} -\frac{k}{m} & 0 \\ 0 & -\frac{k}{m} \end{bmatrix}$$

and  $v_x, v_y, u_x, u_y, \rho_x, \rho_y, k, m, \mathbf{w}$ , and  $\tilde{\mathbf{w}}$  are defined in Table I. Here, we assume the existence of a linear wind resistance  $k\mathbf{v}$ ,  $k \geq 0$ , and that  $\mathbf{w}$  and  $\tilde{\mathbf{w}}$  be zero-mean Gaussian white noises such that

$$E(\mathbf{w}\mathbf{w}^\top) = \mathbf{W}, \quad E(\tilde{\mathbf{w}}\tilde{\mathbf{w}}^\top) = \tilde{\mathbf{W}}, \quad E(\mathbf{w}\tilde{\mathbf{w}}^\top) = \mathbf{0},$$

TABLE I  
THE SYMBOL LIST

|                      |   |
|----------------------|---|
| $v_x$                | $x$ -component of Linear velocity             |
| $v_y$                | $y$ -component of Linear velocity             |
| $u_x$                | $x$ -component of thrust acceleration (input) |
| $u_y$                | $y$ -component of thrust acceleration (input) |
| $\rho_x$             | Measurement of $v_x$ (output)                 |
| $\rho_y$             | Measurement of $v_y$ (output)                 |
| $k$                  | Wind resistance coefficient                   |
| $m$                  | Total mass of UAV                             |
| $\mathbf{w}$         | Disturbance on UAV                            |
| $\tilde{\mathbf{w}}$ | Measurement noise                             |

where  $\mathbf{W}, \tilde{\mathbf{W}} \in \mathbb{R}^{2 \times 2}$  are symmetric positive semidefinite.

Next, let  $\mathbf{r}(t)$  be the reference velocity that the UAV attempts to track, whose magnitude can be arbitrarily assigned and whose direction is set as the negative gradient direction of  $\phi$  at the UAV's position at time  $t$ . To make  $\mathbf{v}(t)$  approach  $\mathbf{r}(t)$ , define  $\mathbf{e}(t) = \int_0^t (\mathbf{r}(\tau) - \rho(\tau)) d\tau$  as the accumulative error between the reference velocity and the noisy measurement of velocity. Thus, from (8) and the expression of  $\mathbf{e}(t)$ , we derive the following dynamical system with an augmented state  $[\mathbf{v}^\top \ \mathbf{e}^\top]^\top$ :

$$\begin{bmatrix} \dot{\mathbf{v}} \\ \dot{\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ -\mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{e} \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix} \mathbf{u} + \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix} \mathbf{r} + \begin{bmatrix} \mathbf{w} \\ -\tilde{\mathbf{w}} \end{bmatrix}, \quad (9)$$

where  $\mathbf{I}$  and  $\mathbf{0}$  represent the identity matrix and the zero matrix of proper size, respectively. We set our control goal as minimizing the performance measure below:

$$E \left\{ \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau [e^\top \ \mathbf{u}^\top \ \mathbf{v}^\top] \begin{bmatrix} \mathbf{Q}_e & & \\ & \mathbf{R} & \\ & & \mathbf{Q} \end{bmatrix} \begin{bmatrix} e \\ \mathbf{u} \\ \mathbf{v} \end{bmatrix} dt \right\},$$

where  $\mathbf{Q}_e, \mathbf{R} \in \mathbb{R}^{2 \times 2}$  are symmetric positive definite matrices and  $\mathbf{Q} \in \mathbb{R}^{2 \times 2}$  is symmetric positive semidefinite matrices. Such a performance measurement quantifies the expected control accuracy, control effort, and velocity magnitude while tracking the reference velocity. Note that  $\mathbf{Q}$  is allowed to be a zero matrix. However, a nonzero  $\mathbf{Q}$  suggests that the velocity magnitude be as small as possible before the reference value is reached, which plays a role in reducing the overshoot in velocity magnitude.

To achieve the control goal, we employ a Linear-Quadratic-Gaussian (LQG) controller [55], [56], whose block diagram is given in Fig. 5. The input  $\mathbf{u}$  to the UAV is given by

$$\mathbf{u} = -\mathbf{K} [\hat{\mathbf{v}}^\top \ \mathbf{e}^\top]^\top,$$

where  $\mathbf{K} \in \mathbb{R}^{2 \times 4}$  is the feedback gain and  $\hat{\mathbf{v}}$  is an estimate of the velocity  $\mathbf{v}$  generated by the Kalman filter. The feedback gain  $\mathbf{K}$  can be determined by solving the Linear Quadratic Regulator (LQR) problem [55] that requires the system (9) in the absence of  $\mathbf{w}$  and  $\tilde{\mathbf{w}}$  to minimize:

$$\int_0^\infty [e^\top \ \mathbf{u}^\top \ \mathbf{v}^\top] \begin{bmatrix} \mathbf{Q}_e & & \\ & \mathbf{R} & \\ & & \mathbf{Q} \end{bmatrix} \begin{bmatrix} e \\ \mathbf{u} \\ \mathbf{v} \end{bmatrix} dt.$$

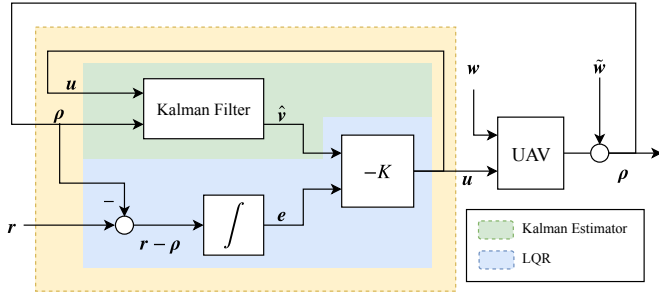


Fig. 5. LQG controller

This gives  $K = R^{-1}[I \ 0]X$ , where  $X$  is the symmetric positive definite solution to the algebraic Riccati equation with respect to (9) given by

$$\begin{bmatrix} A^T & -I \\ 0 & 0 \end{bmatrix} X + X \begin{bmatrix} A & 0 \\ -I & 0 \end{bmatrix} - X \begin{bmatrix} I \\ 0 \end{bmatrix} R^{-1} [I \ 0] X + \begin{bmatrix} Q \\ Q_e \end{bmatrix} = 0.$$

The Kalman filter, which generates  $\hat{v}$  to estimate  $v$ , is in the form of

$$\dot{\hat{v}} = A\hat{v} + u + K_f(\rho - \hat{v}).$$

We set the Kalman filter gain  $K_f \in \mathbb{R}^{2 \times 2}$  such that  $\lim_{t \rightarrow \infty} E\{(v - \hat{v})^T(v - \hat{v})\}$  is minimized. This gives  $K_f = P\tilde{W}^{-1}$ , where  $P$  is a symmetric positive definite matrix satisfying the filter algebraic Riccati equation

$$AP + PA^T - P\tilde{W}^{-1}P + W = 0.$$

Such an LQG controller is guaranteed to minimize the performance measure in (10) [55].

Finally, we convert the input  $u$  provided by the LQG controller to the UAV attitude (i.e., orientation) in the form of quaternion for the sake of practical implementation. To do so, let  $x = [1 \ 0 \ 0]^T$ ,  $y = [0 \ 1 \ 0]^T$ ,  $z = [0 \ 0 \ 1]^T$  be the standard unit basis vectors of the original GCS, and we develop a new coordinate system  $(x', y', z')$  in the following way: As is shown in Fig. 6a, the  $z'$ -axis of the new coordinate system is the rotor axis of the UAV pointing to the ground, and the  $(x', y')$ -plane is the plane perpendicular to the  $z'$ -axis. In addition, as in Fig. 6b, we let the UAV's nose point to the  $x'$ -axis direction, so that the horizontal component of  $x'$  follows the direction of the velocity  $v$ . Note that the horizontal component of the thrust force  $F_T$  is  $F = mu$  and the vertical component of  $F_T$  is equal to  $mg$ , where  $g$  is the gravitational acceleration. Hence, the new coordinate system can be described by the basis

$$x' = \begin{bmatrix} v_x \\ v_y \\ \frac{(v_x u_x + v_y u_y)}{g} \end{bmatrix}, \quad y' = z' \times x', \quad z' = \begin{bmatrix} -u_x \\ -u_y \\ g \end{bmatrix},$$

where  $\times$  denotes the cross product, and  $g = \|g\|$ . The corresponding rotation matrix is thus given by  $\begin{bmatrix} x' & y' & z' \\ \|x'\| & \|y'\| & \|z'\| \end{bmatrix}$ , and we can use the method described in [57] to convert  $u$  to the quaternion. The quaternion can then be realized by the bottom-level attitude controller of the UAV. Since this paper does not focus on any particular type of UAVs and

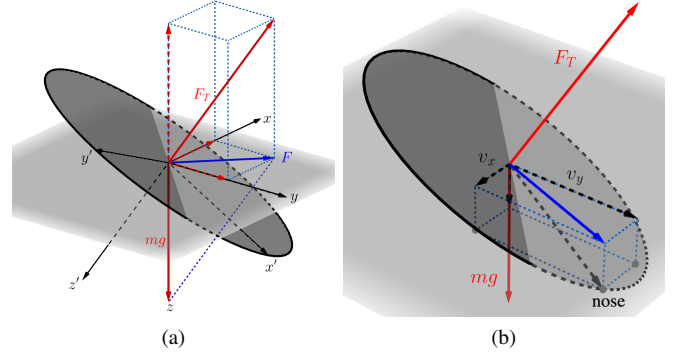


Fig. 6. Coordinate Transformation

most commercial UAVs are equipped with a reliable attitude controller, we omit the discussions of the attitude controller.

## IV. SIMULATION

### A. Architecture and Implementation

We implement our autonomous navigation algorithm into a simulation system using C++, which consists of two separate modules: the motion planning module and the motion control module. The motion planning module cyclically receives distance sensing data, and then immediately constructs a local Laplacian potential field on the fly, from which the negative gradient of any point inside the field can be easily computed. The module constantly receives new distance data and updates the potential field thereafter. The motion control module obtains the negative gradient of the potential field at the current position and takes its direction as the desired flight direction (i.e., the reference velocity direction) to control the UAV's attitude. The two modules work concurrently by two threads, which execute in different updating frequencies. Note that it is not necessary to update the potential field too frequently, while attitude control should be done with sufficient rate. Thus, in this paper, we update the potential field by 10 Hz whereas the velocity tracking is updated by 100 Hz. Fig. 7 shows the whole system architecture, and the implementation details are explained below.

1) *Motion Planning Module*: In order to simulate the real distance sensing process, we use ray casting in *AirSim* to perform ray-geometry intersection in the simulator and record the distance up to an upper bound, e.g., 20 meters, with noise added for more realism. When shooting distance sensing rays in the simulator, the angles of  $2\pi$  is sampled on a horizontal plane, and we used 120 equal angle intervals to generate the rays. With the distance data obtained (note that we give the maximum distance value to rays with no geometry intersection), we can connect these points to form an enclosed polygon as the domain boundary for BEM to compute the Laplacian potential field. However, since the boundary samples may not be uniformly distributed with respect to their mutual distances, large numerical errors may occur to introduce strong instability of our algorithm. Thus, we re-sample the boundary into elements with equal length (and in this paper, we re-sample the boundary into 200 such elements). In addition,

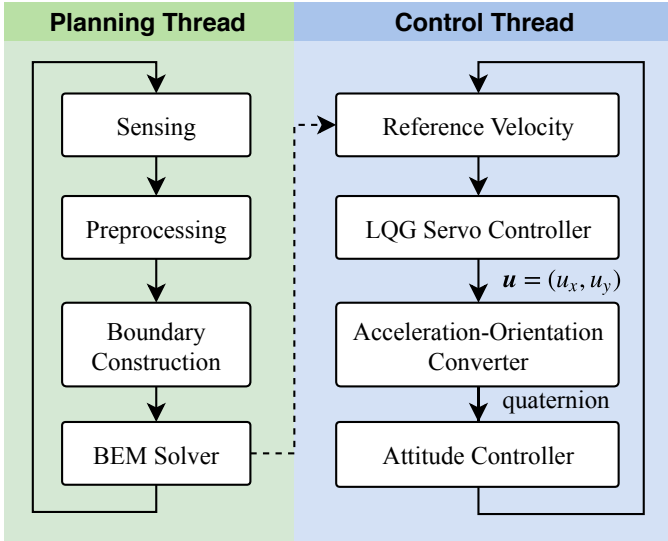


Fig. 7. Implementation architecture: We implement our simulation involving two modules with two threads. A motion planning model solves the Laplacian potential field with a much lower frequency than the motion control module in order to balance the computational efficiency.

the boundary condition should also be specified on the re-sampled elements: the elements on the sensing boundary  $S_1$  are given the condition of  $\phi = 100$ , and the two nearest elements on both sides to the temporary goal are given the condition of  $\phi = 0$ ; for other elements in  $S_2$  and  $S_3$ , they are given the homogeneous Neumann condition  $\frac{\partial \phi(\mathbf{x})}{\partial \mathbf{n}_x} = 0$ . By specifying these boundary conditions, we can finally determine the Laplacian potential field based on BEM to calculate a proper flight direction for the motion control module.

2) *Motion Control Module*: Once the desired flight direction is obtained from the motion planning module, the motion control module controls the attitude of the UAV using the LQG controller to track that direction. Note that for the entire navigation process, we adopt fixed flight speed and altitude which can be pre-specified before any flight, and in between two updates from the motion planning module, the flight direction is calculated from the same potential field at that instant for velocity tracking. When UAV is sufficiently close to the destination, the navigation terminates.

### B. Simulation Setup

Our whole simulation is based on the *AirSim* platform [58] on top of the Unreal real-time graphics engine [59], which is an open-source autonomous vehicle simulator developed by Microsoft AI & Research. Our test experiments are conducted with “Simple Flight” vehicle given by default in the simulator, where the body parameters are all default values: the total mass is 1 kg and the linear drag coefficient is 0.325. The parameters for the LQG controller are:  $\mathbf{Q} = \mathbf{I}$ ,  $\mathbf{Q}_e = 100\mathbf{I}$ ,  $\mathbf{R} = \mathbf{I}$ ,  $\mathbf{W} = 0.01\mathbf{I}$ ,  $\mathbf{\bar{W}} = 0.1\mathbf{I}$ , respectively. Since we only consider UAV navigation with constant speed and altitude, the UAV is set to always fly with a speed of 2 m/s and at an altitude of 7.5 m. Thus, we ignore the altitude (the  $z$ -coordinate value) when specifying UAV positions in our later discussions. Note that we use the NED coordinate system where the starting point is

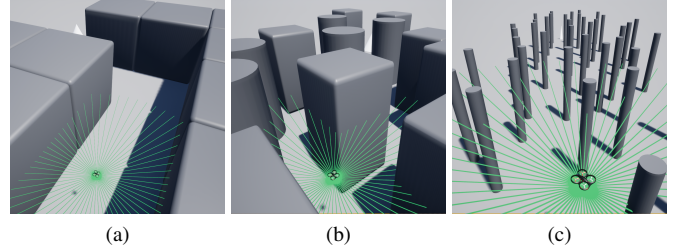


Fig. 8. Scene setups: We construct three different scenes in (a), (b), and (c) to test UAV navigation algorithms, from simple to more difficult scenarios.

taken as its origin. However, the flight speed and altitude can be adjusted at any time during the whole navigation process.

In our simulation, three different scene setups are used for experiments (see Fig. 8). In Fig. 8(a), the cube obstacles (with equal side length of 20 m) are placed next to each other, forming a corridor-like environment, and the destination point is located at (40 m, 60 m). In Fig. 8(b), randomly placed columnar obstacles create multiple paths where the obstacles are still large: each quadrangular prism is 10 m  $\times$  10 m in cross-sectional area, and the cylinder has a cross-sectional diameter of 10 m. The destination point is located at (100 m, 100 m). In Fig. 8(c), thin cylinders with equal cross-sectional diameter of 1 m are placed randomly, creating a even more complex obstacle environment, and the destination point is located at (50 m, 50 m). Note that the green lines shown in Fig. 8 indicate distance sensing rays emitted from the center of UAV, with a maximum sensing radius of 20 m.

### C. Simulation Results

1) *Potential Field Result*: To analyze the computed potential field, we take the scene setup in Fig. 8(c), where one of the distance sensing results is shown in Fig. 9(a), with the center of UAV shown as the black point. Obstacle points detected by the distance rays are shown as red points, while green points indicate no obstacles within the sensing range, which are all clamped to the maximum sensing radius. The uniformly sampled boundary elements with a temporary goal are shown in Fig. 9(b). By configuring our boundary condition and using BEM, we obtain the Laplacian potential field as visualized by isocoutours and color coding in Fig. 9(c) as well as streamlines in Fig. 9(d). It is apparent that the potential field is very smooth without local minima, and the streamlines, which are generated by integrals based on reference velocity directions, are always directed towards the temporary goal.

2) *Control Result*: To demonstrate the velocity tracking performance of the control module, Fig. 10 plots the change of state variables (the velocity components  $v_x$  and  $v_y$ , the speed  $\|v\|$ , and the yaw angle  $\psi$ ) versus time, where the reference and measured values are colored in red and blue, respectively. Observe that the LQG controller tracks the desired state very quickly, and the speed variation is relatively small.

3) *Comparison with VFH Method*: We argue that our dynamic Laplacian potential field helps generate an “optimal” path on the fly that maintains as-far-as-possible distances to the surrounding obstacles while directing the UAV towards

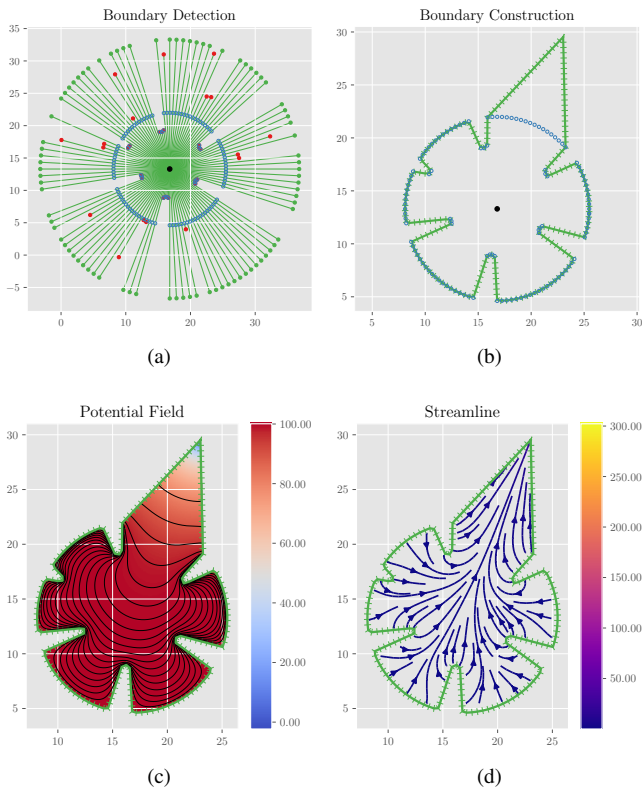


Fig. 9. An example of Laplacian potential field based on the detected distance data: (a) the discrete distance detection based on the sampled directions; (b) the constructed linear boundary elements involving the temporary goal; (c) the visualization by contour curves and color coding (with the potential function values shown in the color bar) of the computed Laplacian potential field; (d) the visualization of the Laplacian potential field by streamlines representing the negative gradient directions.

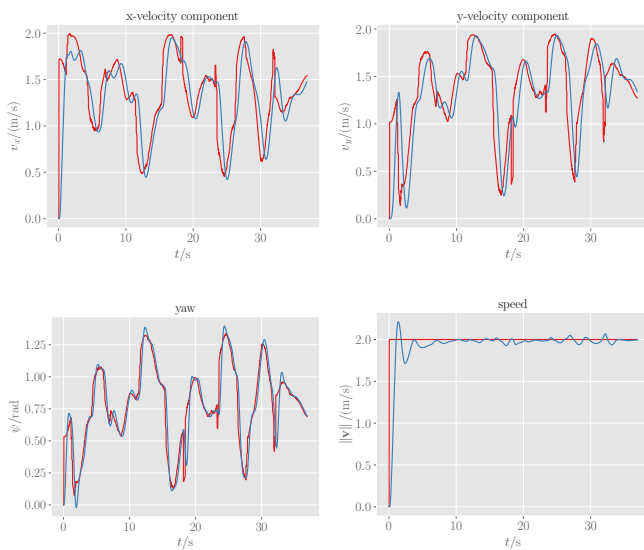


Fig. 10. Velocity tracking: The red curves are the reference state values to track, and the blue curves are the measurement values from the simulator.

the final target destination. To illustrate this advantage, we ran simulations with our method and one of the most popular local planning methods VFH [21]. For each detection, VFH fills a local grid map with detected obstacle points and calculates a smoothed polar histogram. Values of its sectors represent polar obstacle densities (PODs) and a group of consecutive sectors with POD below a threshold forms a “valley”. There may be several valleys and the valley which matches the goal direction best is selected. Finally, a suitable sector is selected as the best direction from the valley. To make VFH method compatible with our setting, we made two little changes to the original algorithm. First, the flight speed is kept constant instead of being calculated by the histogram. Second, we do not update the active window based on the accumulated detection to obtain the histogram but use only a single local detection. Note that the performance of VFH method depends on the selection of parameters, which can be chosen following some existing guidelines. However, the threshold aforementioned can only be tuned empirically, which is essential to the final performance, leading to different quality of locally planned paths. A lower threshold may miss potentially good directions, preventing the UAV from passing through narrow spaces, while a higher threshold may make the UAV unaware of some obstacles, bringing serious safety issues during flight. Since there is no explicit guidance on selecting the threshold, the VFH method may have difficulty in obtaining the “optimal” path. Fig. 11 (a) to (c) show the results for different test scenes in Fig. 8 with different threshold values (note that we only show the top view). On the contrary, our dynamic Laplacian potential field method does not need to empirically tune parameters, leading to more reliable UAV trajectories as shown in Fig. 11 (d).

4) *Discussions on Robustness*: In addition to the benefits mentioned above, another significant advantage of our method is its resistance to system disturbances, such as the measurement errors or path integral errors by velocity tracking as usually experienced by the traditional local planning method. To verify and demonstrate this advantage, we added Gaussian-like noise to the distance measurement, where the standard deviation  $\sigma$  in Gaussian distribution is set to 5% of the detected distance via a distance ray. Due to the smoothness of the Laplacian potential field, small disturbance of the boundary does not introduce obvious variation of the estimate velocity directions, and thus our method can still produce reasonably smooth collision-free paths, as shown in Fig. 12a and Fig. 12b as two examples. Moreover, the whole system may encounter unexpected external disturbances, such as a sudden strong wind, transient signal interference, or path integral errors due to inaccurate velocity/position measurements. In such cases, our Laplacian potential field naturally returns velocity directions that drag the UAV back to its desired optimal paths, avoiding potential collisions with obstacles due to these errors. Fig. 12c and Fig. 12d show two tests of navigation under sudden position disturbances of 2 m, indicating that our method can faithfully direct the UAV to the target without colliding with obstacles or flying too close to obstacles which may be unsafe in practical scenarios.

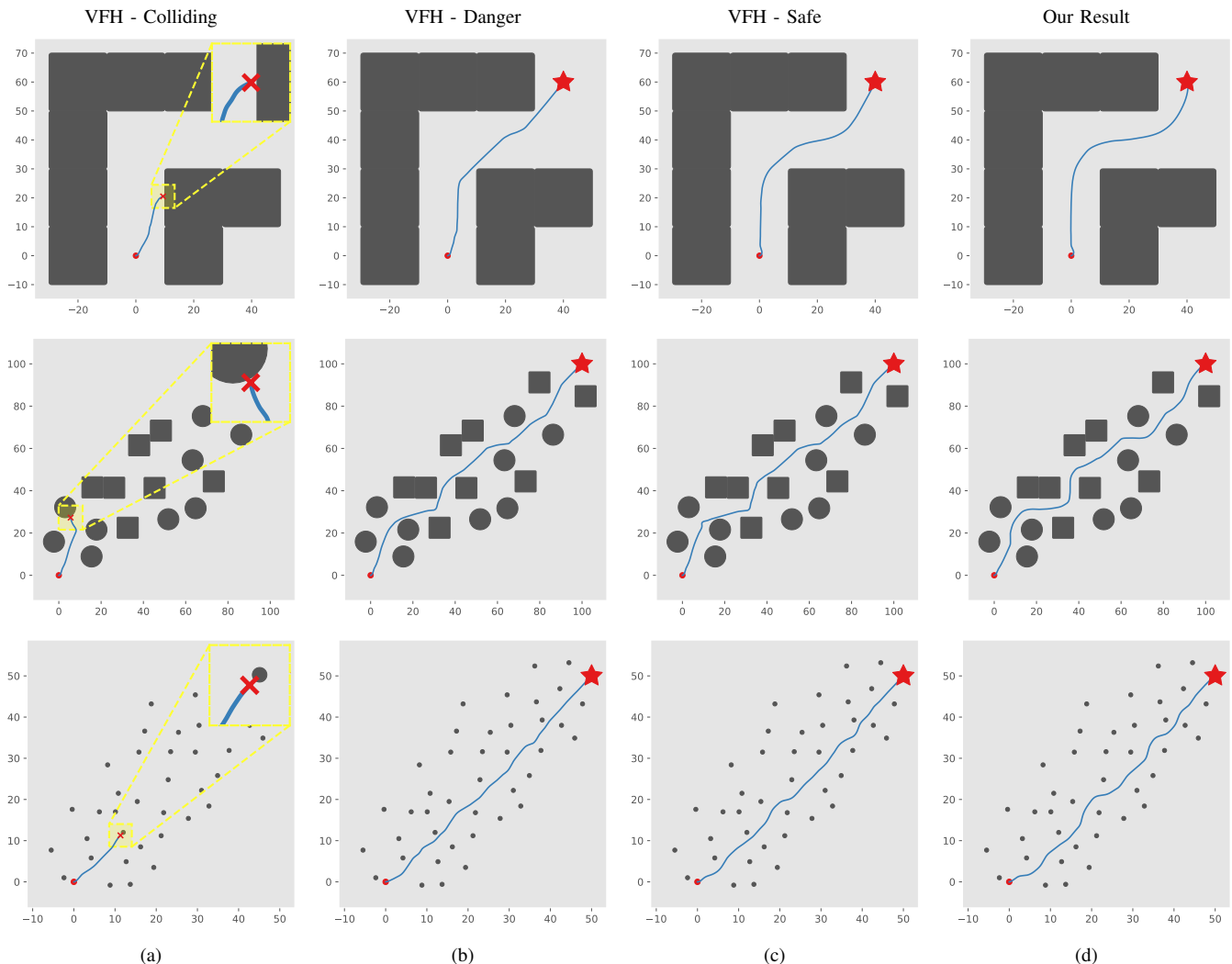


Fig. 11. Comparison with VFH with different threshold values on the three test scenes shown in Fig. 8. From top to bottom, the threshold values are: (a) 10, 16 and 18; (b) 3, 2 and 4; (c) 7, 4 and 10. Observe that while the navigation results generated by VFH depend heavily on the parameter tuning, our method shown in (d) is parameter-free and always yields desirable paths, indicating the advantages of our method.

## V. CONCLUSION

In this paper, we develop a novel UAV navigation scheme, which allows a UAV to autonomously, efficiently, and safely reach a given destination without prior knowledge of the environment. In the navigation scheme, a dynamic local Laplacian potential field is constructed, which only requires the UAV's real-time obstacle detection within a limited sensing range, and thus is memory saving and computationally efficient. We show that the local Laplacian potential field does not contain interior local optima and has a unique global minimum at the artificially set temporary goal. This indicates that if the UAV's velocity follows the negative gradient direction in the local Laplacian potential field, it is guaranteed to proceed towards the temporary goal and can go all the way to the final destination with properly selected temporary goals. We also design an optimal controller that enables the UAV to quickly track the negative gradient direction. Moreover, the navigation scheme is simple for practical implementations, as the parameters can be easily set and essentially no empirical tuning is needed. From simulations on various scenes with

different complexities of obstacles in the *AirSim* simulator, the UAV is capable of flying to the destination through a safe, smooth, and short trajectory, and is robust to uncertain disturbances. In future, this work can be extended to three-dimensional navigation where the altitude of the UAV is also involved so that three-dimensional local Laplacian potential fields should be constructed to enable autonomous navigation with altitude changes.

## REFERENCES

- [1] K. P. Valavanis and G. J. Vachtsevanos, *UAV Applications: Introduction*. Dordrecht: Springer Netherlands, 2015, pp. 2639–2641.
- [2] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1, p. 65, Nov 2009.
- [3] L. Yang, J. Qi, D. Song, J. Xiao, J. Han, and Y. Xia, "Survey of robot 3d path planning algorithms," *Journal of Control Science and Engineering*, vol. 2016, p. 22, 2016.
- [4] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, July 1968.

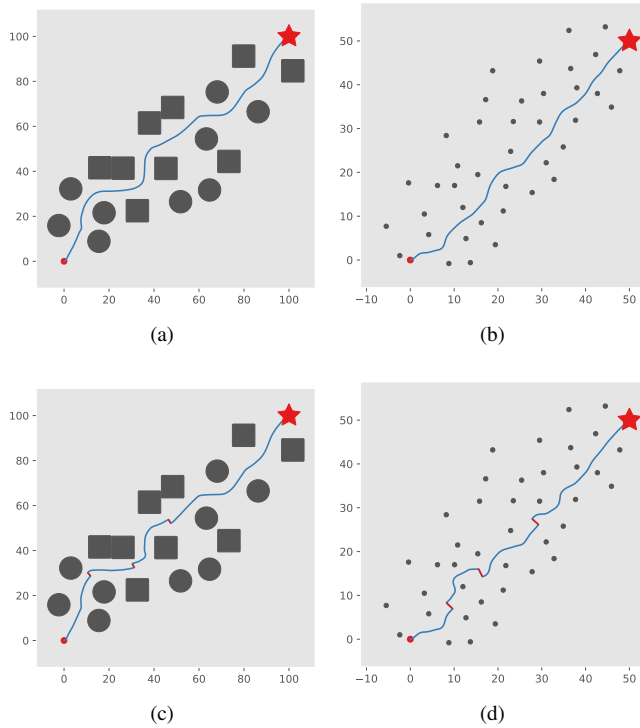


Fig. 12. Disturbance tests: In (a) and (b), we add distance measurement noises, while in (c) and (d), we add sudden position changes, all indicating the robustness of our method.

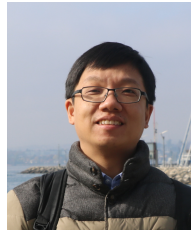
- [5] B. Meng and X. Gao, "UAV path planning based on bidirectional sparse A\* search algorithm," in *2010 International Conference on Intelligent Computation Technology and Automation*, vol. 3, May 2010, pp. 1106–1109.
- [6] L. E. Kavradi, P. Svestka, J. Latombe, and M. H. Overmars, "Probabilistic roadmap for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, Aug 1996.
- [7] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.
- [8] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.
- [9] A. Yershova, L. Jaillet, T. Simeon, and S. M. LaValle, "Dynamic-domain RRTs: Efficient exploration by controlling the sampling domain," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 3856–3861.
- [10] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *49th IEEE Conference on Decision and Control (CDC)*, Dec 2010, pp. 7681–7687.
- [11] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT\*," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 1478–1483.
- [12] C. W. Warren, "Global path planning using artificial potential fields," 1989, pp. 316–321 vol.1.
- [13] P. Shi and Y. Zhao, "Global path planning for mobile robot based on improved artificial potential function," in *2009 IEEE International Conference on Automation and Logistics*, Aug 2009, pp. 1900–1904.
- [14] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, May 1994, pp. 3310–3317 vol.4.
- [15] B. Chandler and M. A. Goodrich, "Online RRT\* and online FMT\*: Rapid replanning with dynamic cost," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2017, pp. 6313–6318.
- [16] M. Otte and E. Frazzoli, "RRTX: Asymptotically optimal single-query sampling-based motion planning with quick replanning," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 797–822, 2016.
- [17] J. L. Sanchez-Lopez, J. Pestana, and P. Campoy, "A robust real-time path planner for the collision-free navigation of multirotor aerial robots in dynamic environments," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, June 2017, pp. 316–325.
- [18] P. Yao, H. Wang, and Z. Su, "Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment," *Aerospace Science and Technology*, vol. 47, pp. 269 – 279, 2015.
- [19] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.
- [20] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, Sep. 1989.
- [21] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, June 1991.
- [22] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [23] J. Minguez and L. Montano, "Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 45–59, 2004.
- [24] B. Hamner, S. Singh, and S. Scherer, "Learning obstacle avoidance parameters from operator behavior," *Journal of Field Robotics*, vol. 23, no. 11/12, pp. 1037–1058, 2007.
- [25] O. Motlagh, D. Nakhacinia, S. H. Tang, B. Karasfi, and W. Khaksar, "Automatic navigation of mobile robots in unknown environments," *Neural Computing and Applications*, vol. 24, no. 7, pp. 1569–1581, Jun 2014.
- [26] L. Ma, "Vision-based avoidance of obstacles with unknown constant velocity," in *Proceedings of the 2010 American Control Conference*, June 2010, pp. 5550–5555.
- [27] A. Eresen, N. İmamoğlu, and M. Önder Efe, "Autonomous quadrotor flight with vision-based obstacle avoidance in virtual environment," *Expert Systems with Applications*, vol. 39, no. 1, pp. 894 – 905, 2012.
- [28] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss, "Stereo vision-based obstacle avoidance for micro air vehicles using disparity space," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 3242–3249.
- [29] J. R. Serres and F. Ruffier, "Optic flow-based collision-free strategies: From insects to robots," *Arthropod Structure & Development*, vol. 46, no. 5, pp. 703 – 717, 2017.
- [30] X. Jin, S. Gupta, J. M. Luff, and A. Ray, "Multi-resolution navigation of mobile robots with complete coverage of unknown and complex environments," in *2012 American Control Conference (ACC)*, June 2012, pp. 4867–4872.
- [31] M. Nieuwenhuisen, D. Droschel, M. Beul, and S. Behnke, "Autonomous navigation for micro aerial vehicles in complex GNSS-denied environments," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1, pp. 199–216, Dec 2016.
- [32] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *2011 IEEE International Conference on Robotics and Automation*, May 2011, pp. 20–25.
- [33] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1688–1695, July 2017.
- [34] K. Mohta, K. Sun, S. Liu, M. Watterson, B. Pfrommer, J. Svacha, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Experiments in fast, autonomous, GPS-denied quadrotor flight," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 7832–7839.
- [35] R. G. Valenti, I. Dryanovski, C. Jaramillo, D. P. Ström, and J. Xiao, "Autonomous quadrotor flight using onboard RGB-D visual odometry," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 5233–5238.
- [36] K. Schmid, P. Lutz, T. Tomić, E. Mair, and H. Hirschi, "Autonomous vision-based micro air vehicle for indoor and outdoor navigation," *Journal of Field Robotics*, vol. 31, no. 4, pp. 537–570, 2014.
- [37] Z. Fang, S. Yang, S. Jain, G. Dubey, S. Roth, S. Maeta, S. Nuske, Y. Zhang, and S. Scherer, "Robust autonomous flight in constrained and visually degraded shipboard environments," *Journal of Field Robotics*, vol. 34, no. 1, pp. 25–52, 2017.
- [38] K. Wu, T. Xi, and H. Wang, "Real-time three-dimensional smooth path planning for unmanned aerial vehicles in completely unknown cluttered

environments,” in *TENCON 2017 - 2017 IEEE Region 10 Conference*, Nov 2017, pp. 2017–2022.

- [39] M. G. Park, J. H. Jeon, and M. C. Lee, “Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing,” in *ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570)*, vol. 3, June 2001, pp. 1530–1535 vol.3.
- [40] U. Orozco-Rosas, O. Montiel, and R. Sep̄Åşlveda, “Pseudo-bacterial potential field based path planner for autonomous mobile robot navigation,” *International Journal of Advanced Robotic Systems*, vol. 12, no. 7, p. 81, 2015.
- [41] A. Melingui, R. Merzouki, J. B. Mbede, and T. Chettibi, “A novel approach to integrate artificial potential field and fuzzy logic into a common framework for robots autonomous navigation,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 228, no. 10, pp. 787–801, 2014.
- [42] M. G. Park and M. C. Lee, “Real-time path planning in unknown environment and a virtual hill concept to escape local minima,” in *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, vol. 3, Nov 2004, pp. 2223–2228 Vol. 3.
- [43] Y. Chen, G. Luo, Y. Mei, J. Yu, and X. Su, “UAV path planning using artificial potential field method updated by optimal control theory,” *International Journal of Systems Science*, vol. 47, no. 6, pp. 1407–1420, 2016.
- [44] J. Kim and P. K. Khosla, “Real-time obstacle avoidance using harmonic potential functions,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, June 1992.
- [45] C. I. Connolly, J. B. Burns, and R. Weiss, “Path planning using Laplace’s equation,” in *Proceedings., IEEE International Conference on Robotics and Automation*, May 1990, pp. 2102–2106 vol.3.
- [46] K. Sato, “Deadlock-free motion planning using the Laplace potential field,” *Advanced Robotics*, vol. 7, no. 5, pp. 449–461, 1992.
- [47] C. Brebbia and J. Dominguez, *Boundary Elements: An Introductory Course*. Sydney Grammar School Press, 1994.
- [48] Y. Liu, *Fast Multipole Boundary Element Method: Theory and Applications in Engineering*. Cambridge University Press, 2009.
- [49] J. T. Katsikadelis, “Chapter three - the BEM for potential problems in two dimensions,” in *The Boundary Element Method for Engineers and Scientists (Second Edition)*, second edition ed., J. T. Katsikadelis, Ed. Oxford: Academic Press, 2016, pp. 35–57.
- [50] J. S. Zelek, “Complete real-time path planning during sensor-based discovery,” in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 3, Oct 1998, pp. 1399–1404 vol.3.
- [51] F. Flanigan, *Complex Variables: Harmonic and Analytic Functions*, ser. Dover Books on Mathematics Series. Dover Publications, 1983.
- [52] S. Axler, P. Bourdon, and W. Ramey, *Basic Properties of Harmonic Functions*. New York, NY: Springer New York, 2001, pp. 1–29.
- [53] J. R. Muir, *Complex analysis: a modern first course in function theory*. John Wiley & Sons, 2015.
- [54] W. M. Boothby, “The topology of regular curve families with multiple saddle points,” *American Journal of Mathematics*, vol. 73, no. 2, pp. 405–438, 1951.
- [55] P. Dorato, V. Cerone, and C. Abdallah, *Linear Quadratic Control: An Introduction*. Melbourne, FL, USA: Krieger Publishing Co., Inc., 2000.
- [56] A. Locatelli, *Optimal Control: An Introduction*. Birkhuser Verlag, 2001.
- [57] P. Berner, R. Toms, K. Trott, F. Mamaghani, D. Shen, C. Rollins, and E. Powell, “Technical concepts: Orientation, rotation, velocity and acceleration, and the SRM,” 2008.
- [58] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and Service Robotics*, M. Hutter and R. Siegwart, Eds. Cham: Springer International Publishing, 2018, pp. 621–635.
- [59] Epic Games, “Unreal Engine.” [Online]. Available: <https://www.unrealengine.com>



**Xiaocheng Song** received the B.S. degree from Hefei University of Technology, China, in 2017. He is currently pursuing the M.S. degree in computer science and technology at ShanghaiTech University, China. His current research interests include autonomous UAV navigation and control.



computing techniques, with their applications in different domains.

**Xiaopei Liu** is currently an assistant professor at School of Information Science and Technology, ShanghaiTech University. He received his Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, and then worked as a Research Fellow at Nanyang Technological University in Singapore. His current research interests include computational fluid dynamics, computer graphics and visualization, machine learning and robotics, the design and autonomous navigation of small unmanned aerial vehicles, as well as parallel



Shanghai, China. Her research interests include distributed optimization and control, networked dynamical systems, and UAV navigation and control.

**Jie Lu** (SM’08-M’13) received the B.S. degree in Information Engineering from Shanghai Jiao Tong University, China, in 2007, and the Ph.D. degree in Electrical and Computer Engineering from the University of Oklahoma, USA, in 2011. From 2012 to 2015 she was a postdoctoral researcher with KTH Royal Institute of Technology, Stockholm, Sweden, and with Chalmers University of Technology, Gothenburg, Sweden. Since 2015, she has been an assistant professor in the School of Information Science and Technology at ShanghaiTech University,