Abstract—Wireless sensor networks are usually deployed in scenarios that are too hostile for human personnel to perform maintenance tasks. Wireless sensor nodes usually exchange information in a multi-hop manner. Connectivity is crucial to the performance of a wireless sensor network. In case a network is partitioned due to node failures, it is possible to re-connect the fragments by setting up bridges using mobile platforms. Given the landscape of a terrain, the mobile platforms should be able reach the target position using a desirable path. In this paper, an off-line robot path planner is proposed to find desirable paths between arbitrary points in a given terrain. The proposed path planner is based on ACO algorithms. Unlike ordinary ACO-based path planner, where artificial ants are only making route decisions at intersection points, the ACO algorithm proposed in [11] allows ants to make decisions at several lines, namely control point lines. Comparing with a limited number of outgoing paths at each intersection, the introduction of the control point lines allows ants to have virtually an infinite number of routes to choose from, which provide the path planner with greater flexibility. This paper is an extension to the ACO-based path planner proposed in [11]. The aim of this work is to further enhance the flexibility of an ACO-based path planner by allowing artificial ants to make routing decisions at any place within a terrain under consideration.

The rest of the paper is organized as follows. Section II elaborates the formulations of the terrain landscapes, path evaluation factors, and the objective function of the optimization problem. The proposed 2-D ACO-based path planner is explained in Section III. The proposed path planner was evaluated using computer simulations. Simulation settings and results are presented in Section IV. Performance of the proposed algorithm is analyzed and compared with the original ACO-based algorithm [11] in Section V. Concluding remarks are given in Section VI.

II. PROBLEM FORMULATION

In this section, the formulations of the terrain landscapes, the path evaluation factors, and the objective function will be elaborated.

A. Terrain Landscapes

The terrain considered in this paper is a three-dimensional (3-D) hilly landscape. The base of the 3-
to find a desirable path from an arbitrary starting point. The normalizing factor such that the desirability of a path is evaluated based on the following factors.

1) Path Length: The length of a path is often proportional to the fuel consumption of a robot, and thus it is desirable to have shorter paths. The first path desirability factor \( D_1 \) is expressed as follows

\[
D_1(U) = d_{proj}(P_s, P_t).
\]  

Here, \( d_{proj}(P_s, P_t) \) is the length of the projection of the path \( U \) on the \( x-y \) plane that is connecting \( P_s \) and \( P_t \). An illustration of \( d_{proj}(P_s, P_t) \) is shown in Fig. 2.

2) Turning Angle: A path with sharp turning points is undesirable as turnings often associated with decelerations, which are both time and energy consuming. In this paper, a path is represented using a B–Spline curve, which was invented by Schoenberg [12]. A B–Spline curve consists of multiple piecewise polynomial segments. B–Spline curves are often used to represent robots’ trajectories due to their continuity nature. The shape of a B–Spline curve can be adjusted by moving a set of control points [13]. As the control points are populated along a B–Spline curve, an angle formed by three consecutive control points can be used as an estimate of a turning angle. Given a path \( U \) with \( n \) control points \( (e_1, \ldots, e_n) \), its second desirability factor \( D_2 \) is expressed as follows

\[
D_2(U) = 180^\circ - \min_{i=2, \ldots, n-1} \angle c_{i-1} e_i c_{i+1}.
\]  

An illustration of a B–Spline curve with \( n = 6 \) control points is shown in Fig. 2.

3) Climbing and Descending Ratio: Steep climbing and descending should be avoided whenever possible as it is fuel inefficient to a robot. It can also be dangerous as a robot may flip or slide while maneuvering on slopes. Therefore, the resultant path should avoid having rapid changes on \( z \)-dimension. After the control points of a B–Spline curve are defined, a fixed number of evaluation points \( (e_i) \) are used to represent the piecewise polynomial curves between adjacent control points. Let \( z(e_i) \) be the \( z \)-dimension of the terrain at the position of evaluation point \( e_i \). Given a path \( U \) with \( k \) evaluation points \( (e_1, \ldots, e_k) \), its third desirability factor \( D_3 \) is expressed as follows

\[
D_3(U) = \max_{i=1, \ldots, n-1} \left| \frac{\Delta z_i}{\Delta d_i} \right| = \max_{i=1, \ldots, n-1} \left| \frac{z(e_{i+1}) - z(e_i)}{d(e_i, e_{i+1})} \right|.
\]  

Here, \( d(e_i, e_{i+1}) \) is the separation between consecutive evaluation points \( e_i \) and \( e_{i+1} \) on the \( x-y \) plane. An illustration on \( \Delta z_i \) and \( \Delta d_i \) are shown in Fig. 2.
C. An Objective Function

A desirable path should have all the aforementioned desirability factors being minimizing. Therefore, the objective function \( f \) used in this paper is a weighted-sum function of \( D_1, D_2, \) and \( D_3, \) which is expressed as follows

\[
 f_{\text{obj}}(U) = \frac{w_1 D_1(U) + w_2 D_2(U) + w_3 D_3(U)}{w_1 + w_2 + w_3}.
\]  

(5)

Here, \( w_1, w_2, \) and \( w_3 \) are the weights. Given (5), the optimization problem is to find an optimum path \( U_{\text{opt}} \) that has a minimum \( f_{\text{obj}} \), such that

\[
 U_{\text{opt}} = \arg \min_U f_{\text{obj}}(U).
\]  

(6)

III. THE PROPOSED 2-D ACO-BASED PATH PLANNER

In this section, the basic mechanisms of ordinary ACO algorithms are revisited. Afterward, the proposed ACO-based path planner will be introduced.

A. Background of ACO Algorithms

The first ACO algorithm was proposed by Dorigo in [14], who was inspired by the foraging behaviors in ant colonies. Suppose an ACO algorithm is used to find the shortest path between two arbitrary points in a network. In the first iteration, artificial ants will be released at the starting point and make random move until they reach the target point. Artificial pheromone will be injected to their paths, the concentration will be inversely proportional to the corresponding path lengths. Ants in the following iterations will make routing decisions based on the pheromone concentration on each path left by their predecessors. Ordinary ACO algorithms work well for network-based routing problems that a network is given and the number of routing options (i.e. connection degree) at each intersection is finite. An illustration of an ACO algorithm used to find the shortest path between vertices in a network is shown in Fig. 3.

However, the routing network may impose extra constraint to a path planning problem as it limits the possible directions an ant can visit. The ACO-based path planner with Gaussian functions (ACO-Gauss) in [11] tries to relax such limitation with the introduction of the control point lines (CPLs). CPLs are distributed evenly along the \( x \)-axis and parallel to the \( y \)-axis in the terrain. Instead of moving from one vertex to another in a routing network, an ant is moving from one CPL to another. An ant is allowed to select any point on the next CPL as its next hop. At the end of an iteration, the ants will leave pheromone, which is in form of Gaussian functions, onto the CPLs. An illustration of the CPLs is shown in Fig. 3. The CPLs in ACO-Gauss provide the ants with more routing options. However, the flexibility of the path planner is still limited by the number and the orientations of the CPLs in the terrain.

B. ACO-based Path Planner with 2-D Gaussian Functions

In the proposed ACO-based Path Planner with 2–Dimensional Gaussian Functions (ACO-2-Gauss), an ant is allowed to select any point on the circumference of a circle on the \( x-y \) plane with a radius of \( R \) centered at the current location as its next hop. Such circle is defined as the control point circle (CPC). An illustration of a CPC is shown in Fig. 4. An ant will decide its moving direction based on the distribution of the pheromone concentration on the circumference of the current CPC. An ant will select \( n-2 \) CPCs from the terrain and move toward \( P_i \). The last CPC will hop to \( P_i \) directly. Including \( P_s \) and \( P_t \), a path will consists of \( n \) control points. Once an ant has reach \( P_t \), its control points are used to construct a trajectory using B–Spline. The path will be evaluated based on (5). Artificial pheromone will then be distributed at the centers of the CPCs as 2–dimensional (2-D) Gaussian functions. Ants in the following iterations will decide their moving direction based on the pheromone residue in the terrain. Detailed procedures of ACO-2-Gauss are elaborated as follows

Step 1 Given \( P_s \) and \( P_t \), initialize the optimization process by distributing pheromone at \( P_t \). Set itera-
In Step 1, the pheromone distributed at $P_i$ is to develop a virtual potential field that attract ants toward the target [15]. Such practice is essential at the initial phase as the terrain has no pheromone residue. In later iterations, the pheromone distributed at $P_i$ acts as a biasing factor to favor directions toward $P_i$. The pheromone concentration due to $P_i = (x_i, y_i)$ measured at an arbitrary location $(x, y)$ is expressed as the following 2-D Gaussian function

$$g_i(x, y) = \phi_i \exp \left( - \frac{(x-x_i)^2}{2\sigma_{x}^2} - \frac{(y-y_i)^2}{2\sigma_{y}^2}\right).$$

(7)

Here, $\phi_i$ is representing the magnitude of the pheromone concentration at $P_i$, which is a tuning parameter. Variables $\sigma_x$ and $\sigma_y$ are the variances of the blob along $x$-axis and $y$-axis, respectively. In the proposed path planner, only symmetric 2-D Gaussian functions are considered and thus $\sigma_x = \sigma_y$.

In Step 5, for a path $U_i$, the initial magnitude of its pheromone is inversely proportional to its $f_{obj}(U_i)$, such that

$$\phi(U_i) = \frac{\phi_c}{f_{obj}(U_i)}.$$  

(8)

Here, $\phi_c$ is a tuning parameter. Suppose $U_i$ consists of $m$ CPCs. The pheromone concentration due to the $j$th CPC measured at an arbitrary location $(x, y)$ is expressed as the following 2-D Gaussian function

$$g_{i,j}(x, y) = \phi_i(U_i) \exp \left( - \frac{(x-x_{i,j})^2}{2\sigma_x^2} - \frac{(y-y_{i,j})^2}{2\sigma_y^2}\right).$$

(9)

Here, $x_{i,j}$ and $y_{i,j}$ are the coordinates of the $j$th CPC on the path $U_i$. Variables $\sigma_x$ and $\sigma_y$ are the variances of the blob along $x$-axis and $y$-axis, respectively. By the end of an iteration, the magnitudes of the pheromone are updated in Step 9 as follows

$$\phi(U_i) = \frac{\phi(U_i)}{\epsilon}.$$  

(10)

Here, $\epsilon > 1$ represents the evaporation rate of the pheromone. Steps 3 and 7, the pheromone concentration at the circumference of a CPC is the sum of all the Gaussian functions within the terrain.

IV. SIMULATIONS

The performance of the proposed ACO-2-Gauss path planner is evaluated against the ACO-Gauss path planner in [11] using computer simulations. In this section, simulation settings and simulation results are presented.
### Table I

<table>
<thead>
<tr>
<th>Parameters</th>
<th>ACO-Gauss</th>
<th>ACO-2-Gauss</th>
</tr>
</thead>
<tbody>
<tr>
<td>terrain constants ((a, b, c, d, e))</td>
<td>(-0.3, 3, 1.3, 0.5, -2)</td>
<td>(-0.3, 3, 1.3, 0.5, -2)</td>
</tr>
<tr>
<td>terrain width</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>terrain height</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>variances used in (7) ((\sigma^2_{tx}, \sigma^2_{ty}))</td>
<td>N/A, 0.0005</td>
<td>0.0125, 0.0025</td>
</tr>
<tr>
<td>variances used in (9) ((\sigma^2_{x}, \sigma^2_{y}))</td>
<td>N/A, 0.0005</td>
<td>0.125, 0.025</td>
</tr>
<tr>
<td>weighting ((\omega_1))</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>weighting ((\omega_2))</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>weighting ((\omega_3))</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>maximum iteration number ((T_{max}))</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>no. of ants per iteration ((q))</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>no. of CPLs in the terrain</td>
<td>5</td>
<td>N/A</td>
</tr>
<tr>
<td>no. of CPCs per path ((n - 2))</td>
<td>N/A</td>
<td>5</td>
</tr>
<tr>
<td>pheromone evaporating rate ((\epsilon))</td>
<td>1.75</td>
<td>1.75</td>
</tr>
<tr>
<td>order of B-spline curves</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Evaluations</th>
<th>ACO-Gauss</th>
<th>ACO-2-Gauss</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R = 0.09</td>
<td>R = 0.10</td>
</tr>
<tr>
<td>Averaged path length (D_1) (unit)</td>
<td>1.4020</td>
<td>1.1220</td>
</tr>
<tr>
<td>Averaged turning angle (D_2) (degree)</td>
<td>119.84</td>
<td>102.1743</td>
</tr>
<tr>
<td>Averaged climbing and descending ratio (D_3)</td>
<td>1.6297</td>
<td>1.8630</td>
</tr>
</tbody>
</table>

### A. Simulation Settings

Simulations are conducted in Matlab. In the simulations, the path planners under test are provided with the terrain introduced in Section II-A. The desirability of a path is evaluated based on the factors discussed in Section II-B. The weight of the each factor is kept equal for both the path planners for a fair comparison. Optimum values for the variances are decided based on the experimental results. Variables and tuning parameters used in the simulations are shown in Table I.

### B. Simulation Results

Based upon the parameters given in Table I, both ACO-based path planners are tested. Coordinates of the starting point \(P_s\) and the target point \(P_t\) are located at (0.0208, 0.3000) and (0.9792, 0.8000), respectively. In each simulation, the path planners are allowed to operate for 5 iterations. In each iteration, both ACO-based path planners release 5 artificial ants into the terrain. An ant picks 5 locations inside the terrain and uses them as control points to construct their B–Spline curves. In ACO-Gauss, artificial ants can only select locations on the CPCs as their control points. Simulations for ACO-2-Gauss were carried out with 7 different \(R\) values of the CPCs. In this paper, all the results are the averaged values obtained from 50 individual simulations. Simulation results are shown in Table II.

### V. Performance Analysis

According to the simulation results, the proposed ACO-2-Gauss path planner shows greatly improved performance over the original ACO-Gauss path planner, in terms of path length, turning angle, and climbing and descending ratio for all different \(R\) values under test. According to the simulation results for ACO-2-Gauss, the average path length has increased with \(R\). A good balance among all three evaluation factors can be found when \(R = 0.12\). When \(R = 0.12\), the proposed ACO-2-Gauss outperforms the original ACO-Gauss by reducing the path length by more than 18%, the turning angle by more than 20%, and the climbing and descending ratio by nearly 5%. However, due to the extra degree of freedom, the computational complexity of the ACO-2-Gauss is slightly higher than that of the ACO-Gauss. Nevertheless, the computational complexity
can be neglected as both the path planners are utilized in off-line path planning.

VI. CONCLUSIONS

Mobile wireless platforms can be used for reconnecting several fragments of a wireless sensor network by setting up bridges between them. In order to bridge fragments and shorten the down-time of a network, mobile robots needed to reach the target location quickly. The proposed ACO-2-Gauss can be used for off-line path planning of mobile wireless platforms. Comparing with its preceding version ACO-Gauss, the current version can considerably improve path quality due to the extra freedom provided by the 2-D Gaussian functions. In the proposed path planner, artificial ants have higher flexibility in making routing decisions. The simulation results verify the improved performance of the proposed ACO-2-Gauss path planner over the original ACO-Gauss path planner.

REFERENCES


