2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Whisker-Based Tactile Navigation Algorithm For Underground Robots

Tanel Kossas[1], Walid Remmas[2], Roza Gkliva[2], Asko Ristolainen[2], and Maarja Kruusmaa[2]

***Abstract*— This work explores the use of artificial whiskers as tactile sensors for enhancing the perception and navigation capabilities of mobile robots in challenging settings such as caves and underground mines. These environments exhibit inconsistent lighting conditions, locally self-similar textures, and general poor visibility conditions, that can cause the performance of state-of-the-art vision-based methods to decline. In order to evaluate the efficacy of tactile sensing in this context, three algorithms were developed and tested with simulated and physical experiments: a wall-follower, a navigation algorithm based on Theta*, and a hybrid approach that combines the two. The obtained results highlight the efficacy of tactile sensing for wall-following in intricate environments. When paired with an external method for pose estimation, it further aids in navigating unknown environments. Moreover, by integrating navigation with wall-following, the third, hybrid algorithm enhanced the map traversal speed by roughly $26-43\%$ compared to standard navigation methods without wall-following.**

I. Introduction

State-of-the-art perception techniques for robot navigation commonly rely on data from cameras [1], LiDAR [2], [3], or combinations of those [4], [5], and have demonstrated excellent performance in a variety of applications and environmental settings. However, the performance of these traditionally successful approaches diminishes in environments such as underground mines and caves, that are often dusty, muddy, dark, GPS-denied, partially or fully submerged, and locally self-similar. To mitigate these limitations, the scientific community often leverages biological inspiration [6]. Our work explores artificial whiskers as an alternative tactile sensing mechanism for navigation.

Previous implementations of whisker-like sensors rely on a variety of sensing modalities, such as magnetic [7], [8], [9], piezoelectric [10], [11], triboelectric [12], [13], piezoresistive [14], [15], capacitive [16], [17], strain gauges [18], [19] and other [20], [21], [22], [23]. Many of these methods serve as proof-of-concept, not considering the cost, scalability or robustness in harsh conditions. Whisker-based perception methods have shown good performance for a variety of applications, including flow sensing in air [24] and water [25], [26], [11], as well as for object identification [27], [20], obstacle detection [12], and various navigation-related applications [9], [28], [29].

Specific to navigation, in [28], an omnidirectional mobile robot using a 3 DoF arm with whisker sensors was used to demonstrate Simultaneous Localisation and Mapping (SLAM). In [29] a small whiskered robot was used to demonstrate SLAM for mapping simple small arenas with only whiskers and prior knowledge about the world structure. [30] demonstrated indoor wall following, using one artificial cockroach-inspired antenna, that combines an array of flex sensors. In [31], a vehicle with bumper-like tactile sensors on its periphery was used to demonstrate wall following in simple simulated environments.

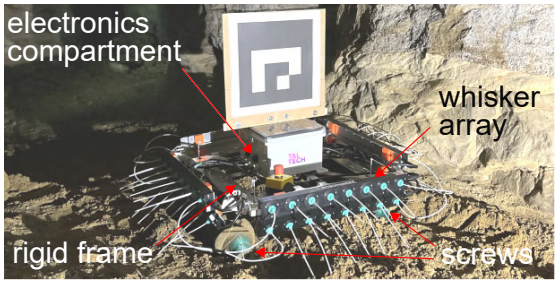

Fig. 1: RM3 robot with whiskers

Most of the strategies mentioned above include mechanically complex sensing systems, or implementations that have been tested only in simulated or indoor environments. In this work we propose a navigation algorithm for the RM3 robot platform (Fig. 1), that relies on low-cost passive whisker sensors. The contributions of this work include:

- two separate navigation algorithms that use whisker sensor data: wall-following and Theta*-based pathfinding;
- a hybrid navigation algorithm combining Theta* and wall-following;
- experimental validation of algorithms in simulated and real-world scenarios.

II. Platform

A. RM3 robot

The RM3 prototype (size: 0.79 by $0.74\mathrm{m}$), is a small testing platform, for investigating mining-related technologies in the ROBOMINERS project [32]. The robot consists of: a rigid frame, a main electronics compartment and two locomotion modules. Two counteracting Archimedean screws per module enable holonomic control on the x-y plane, similar to robots with mechanum wheels.

B. Whisker sensors

The whisker sensors (Fig. 2a) comprise a 3D Hall effect sensor (*Infineon TLV493D-A1B6*), a magnet mounted on the base of the stem, a flexible silicone membrane (*Elite Double* 22 by *Zhermack*), and a whisker stem made of 0.3m long polyamide M4 threaded rod [33]. When the whisker stem

This work was funded by European Union's Research and Innovation programme Horizon 2020 project ROBOMINERS (grant agreement n°820971).
[1] tanel.kossas@gmail.com
[2] Centre for Biorobotics, Tallinn University of Technology, Estonia firstname.surname@taltech.ee

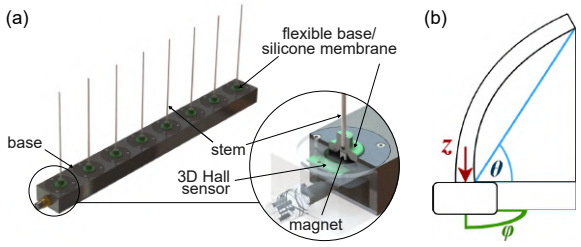

Fig. 2: Whisker sensor components (a) and axes (b)

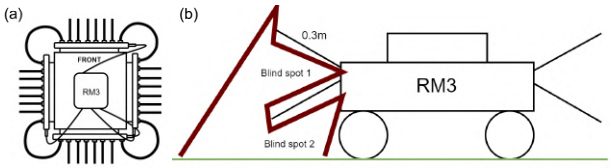

Fig. 3: Whiskers top-down view (a) and blindspots (b)

touches an object, the silicone joint allows it to be deflected from its original position. This causes the magnet to be displaced relative to the sensor, which is reflected in the magnetic field measurements. The silicone joint provides a small force for returning the whisker back to its resting state when not in contact with an object. The whiskers are mounted on all 4 sides of the robot in arrays of 8.

For navigation, minimising potential blind spots is essential. For turning smoothly and without hard collisions, a connected whisker setup was created in [34] to cover the corner blind spots (Fig. 3a), by connecting the ends of a long whisker into neighboring whisker sensors. This configuration requires no change in the algorithm due to the way whisker bias is accounted for. To cover blind spots vertically, a whisker setup is proposed (Fig. 3b), that uses two arrays per side. The initial configuration with whisker arrays at $-30°$ and $30°$ was replaced with $-30°$ and $0°$, as the walls used for final testing were shorter.

III. Algorithm

A. Whisker preprocessing

Each whisker sensor measures the deflection of its compliant joint across three axes, converting magnetic field data to spherical coordinates, namely azimuth ϕ and polar angle θ (see Fig. 2b). Since some deflection is absorbed by the stem, and the azimuth doesn't aid in determining object distance, the azimuth ϕ was not considered in this paper. The third axis (z) stands for perpendicularly applied pressure to the sensor. Various ratios for θ and z measurements were used, with weights assigned as w_θ and w_z respectively.

B. Recalibration

During real-world tests an issue of both initial and accumulating whisker bias was observed. The initial bias refers to the whiskers not being in an exact orientation to give a reading of exactly zero when not in contact with any object. Accumulating bias refers to whiskers changing the bias during operation, especially when higher deflection values occur. Both of these issues were mitigated by recalibrating the whiskers periodically.

The values of the whiskers are composed of three components: the actual whisker values $\boldsymbol{W_{in}}$, the bias matrix $\boldsymbol{B}$, the sensor noise at a given time $\boldsymbol{R}$ and the sensor output $\boldsymbol{W}$:

$$\boldsymbol{W} = \boldsymbol{W_{in}} + \boldsymbol{B} + \boldsymbol{R} \tag{1}$$

The matrices are three-dimensional, with m whisker arrays, n whiskers per array, and (θ, z) data for deflection and perpendicular pressure.

A key problem of using whiskers is that at a given point in time it cannot be known whether the whisker is displaced due to bias or due to contact. Thus a solution considered in this work is recalibration consisting of two parts for each of the robot's four sides. First, the robot is navigated to a neutral state, away from the whisker array's direction until the whisker array's pressure sample amount $s_m = 10$ has a standard deviation lower than $\xi_{cal} = 0.015$. Then, the robot is kept moving in the same direction for bias sample amount $s = 10$, after which the bias compensation matrix $B^{-'}$ is updated for that specific side (in this example the second of four sides, i.e. $j = 2$) with:

$$\boldsymbol{b^-_{i,j}} = \left[\frac{\sum_{t=0}^{s-1}\theta_{i,j}(t)}{s} \quad \frac{\sum_{t=0}^{s-1} z_{i,j}(t)}{s}\right]$$

$$\boldsymbol{B^-} = \begin{bmatrix} \boldsymbol{b_{0,0}} & \boldsymbol{b_{0,1}} & \cdots & \boldsymbol{b_{0,n-1}} \\ \boldsymbol{b_{1,0}} & \boldsymbol{b_{1,1}} & \cdots & \boldsymbol{b_{1,n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{b_{m-1,0}} & \boldsymbol{b_{m-1,1}} & \cdots & \boldsymbol{b_{m-1,n-1}} \end{bmatrix} \tag{2}$$

$$\boldsymbol{B^{-'}} = \begin{bmatrix}1 & 0 & 1 & 1\end{bmatrix}\boldsymbol{B^-_{old}} + \begin{bmatrix}0 & 1 & 0 & 0\end{bmatrix}\boldsymbol{B^-_{new}} \tag{3}$$

In (2), $\theta_{i,j}(t)$ and $z_{i,j}(t)$ represent the θ and z values of the j'th whisker in i'th whisker array respectively at time t, where the robot is also in a neutral state. This process is repeated for all four sides every time calibration is triggered — for this work, every 50 seconds (i.e., 1000 samples at 20 Hz sampling rate). In order to filter out noise not resulting from bias, a simple method of averaging by a constant $r = 3$ was sufficient due to the relatively small amount of noise in the whisker sensor readings. Thus the filtered three-dimensional output matrix of whiskers W'_k can be obtained with:

$$\begin{cases} \boldsymbol{W'_0} = [0]_{m\times n\times 2} \\ \boldsymbol{W'_k} = \boldsymbol{W'_{k-1}} + \dfrac{\boldsymbol{W_k} - \boldsymbol{B^-} - \boldsymbol{W'_{k-1}}}{r} \end{cases} \tag{4}$$

C. Errors calculated from whisker pressures

A whisker is always mapped to a single value, pressure p, which is normalized to $0 \le p \le 1$:

$$\begin{cases} p_z = max(1, abs(z/z_{max})) \\ p_\theta = max(1, (\pi/2 - abs(\theta))/(\pi/2 - \theta_{max})) \\ w_\theta \ge 0, w_z \ge 0, w_\theta + w_z = 1 \\ p = w_\theta * p_\theta + w_z * p_z \end{cases} \tag{5}$$

A p value 0 means no contact for the whisker and a p value 1 means that the whisker effectively cannot bend any more

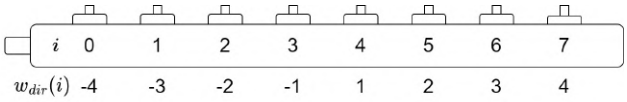

Fig. 4: Calculation of the directional whisker weight w_{dir} for $n = 8$

without likely damage to the whisker. The normalization values $\theta_{max} = 0.7$ and $z_{max} = 25$ were used in this work. Weights w_θ and w_z can be varied to prioritise the robot to use more of the polar angle θ or the perpendicular force z. As follows, p_{ij} will be used to refer to the pressure calculated for the j'th whisker in whisker array i.

1) Generalised whisker array errors: For calculating the errors to be used in higher-level algorithms, both average and maximum p values are calculated for each whisker array i, with n total number of whiskers in a specific whisker array.

$$P_{avg}(i) = \frac{\sum_{j=0}^{n-1} p_{ij}}{n}, \qquad P_{max}(i) = \max_{j \in \{0,1,2,\ldots,n\}} p_{ij} \quad (6)$$

2) Directional error: The purpose of the directional error is to align the whisker array with the wall. In addition, including front whisker pressures in the directional error allows the robot to rotate when encountering an obstacle in the front. For calculating the directional error, the concept of a directional whisker weight w_{dir} was introduced and generalised for any whisker array with equidistant whiskers, as seen on Fig. 4:

$$w_{dir} = \begin{cases} \lfloor j - n/2 \rfloor + 1, & \text{if } n \text{ is odd or } j \geq n/2 \\ \lfloor j - n/2 \rfloor, & \text{if } n \text{ is even and } j < n/2 \end{cases} \quad (7)$$

The directional error normalised to $[0, 1]$ for a given whisker array i is calculated with:

$$Ec_{dir}(i) = \frac{\sum_{j=0}^{n-1} p_{ij} w_{dir}(j)}{2 \cdot \sum_{k=0}^{\lfloor n/2 \rfloor} k} \quad (8)$$

For upcoming equations, we set helper definitions: $D_{tracked}$ as the direction of the tracked wall and $D_{opposite}$ as the direction opposite of the tracked wall. Similarly, D_{left}, D_{right}, D_{front} and D_{rear} (Fig. 3) are used to denote a specific side of the robot instead of a certain whisker array i value.

The whisker arrays were mounted on the robot such that the 0'th whisker was towards the rear of the robot for left and right arrays, and towards the right side of the robot for the front and rear arrays. We also chose positive E_{dir} values to turn the robot left. The final directional error is thus calculated with:

$$\begin{cases} w_{avg} \geq 0, w_{max} \geq 0, w_{avg} + w_{max} = 1 \\ sign_{tracked} = \begin{cases} 1, \text{if } D_{tracked} = D_{right} \\ -1, \text{if } D_{tracked} = D_{left} \end{cases} \\ E_{dir} = Ec_{dir}(D_{right}) - Ec_{dir}(D_{left}) \\ \qquad + sign_{tracked}(w_{avg} P_{avg}(D_{front}) \\ \qquad + w_{max} P_{max}(D_{front})) \end{cases} \quad (9)$$

TABLE I: Collision angle parameters based on whisker array.

Direction of whisker over the threshold	Collision angle α_{array}	Whisker angle multiplier $\lambda_{whisker}(i)$
D_{left}	90°	-1
D_{right}	-90°	1
D_{front}	0°	1
D_{rear}	180°	-1

TABLE II: Impact of v_x, v_y and $\dot{\psi}$ on motion.

Parameter	Positive values	Negative values
v_x	Forward motion	Backwards motion
v_y	Movement left	Movement right
$\dot{\psi}$	Turn right	Turn left

D. Path planning

Theta* [35] was used for path planning as it can create shorter paths in open spaces. The grid size was set to half the size of the robot's length. The collision marking for the purpose of this paper was simplistic, taking into account just the whisker's approximate angle from the robot location s' to destination point p'_0 compared to the center of the robot and using values from Table I:

$$\alpha_{collision} = \alpha_{robot} + \alpha_{array} + \lambda_{whisker}(i) w_{dir}(i) \cdot 10° \quad (10)$$

In order to rotate the robot towards the direction of the next node of Theta*, a path following angle error E_{path} was defined:

$$\begin{cases} s' = (x'_s, y'_s) \qquad p'_0 = (x'_0, y'_0) \\ \alpha_{target} = \frac{180}{\pi} atan2(y'_0 - y'_s, x'_0 - x'_s) \\ \alpha_{temp} = (\alpha_{robot} - \alpha_{target}) \mod 360 \end{cases} \quad (11)$$

$$E_{path} = \begin{cases} \alpha_{temp} - 360, \text{ if } \alpha_{temp} > 180 \\ \alpha_{temp}, \text{ if } \alpha_{temp} \leq 180 \end{cases} \quad (12)$$

E. Weight-based movement system

In order to avoid complex decision trees and the need to smooth their movements, but retain traceability of the robot's movement, the algorithm uses weight factors. A weight factor ϕ_i consists of a scalar *weight* w_i and a *movement vector* $\vec{v}_i$:

$$\begin{cases} w_i \geq 0, \quad -1 \leq v_x, v_y, \dot{\psi} \leq 1 \\ \vec{v}_i = \begin{bmatrix} v_x & v_y & \dot{\psi} \end{bmatrix} \\ \phi_i = (w_i, \vec{v}_i) \end{cases} \quad (13)$$

The way the parts of movement vector $\vec{v}_i$ impact movement is described in Table II. To determine the robot's final movement vector $\vec{v'}$ that is used by a separate module for calculating the motor speeds, the sum-normalised weights are multiplied with the respective movement vectors:

$$\vec{v'} = \frac{\sum_{i=1}^{n} w_i \vec{v}_i}{\sum_{i=1}^{n} w_i} \quad (14)$$

F. Weight factors

In this section all the weight factors created for the navigation and wall-following algorithms are presented. In addition, errors E_y, E_{dir} and E_{path} each utilised a PID-controller, with the input being the actual calculated value of the error and the output being used for calculating weight factors. This enabled simple tuning of the algorithm. For E_y, a non-zero setpoint could be used for keeping the robot in contact with the wall. For tuning PID controller parameters, the Ziegler-Nichols' method based on frequency response [36] was used, but with the improved starting parameters taken from [37, 160].

1) Base normalisation weight factor ϕ_{base}: Used to keep small weights small after normalization.

$$\begin{cases} w_{base} = 1 \\ \vec{v}_{base} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{15}$$

2) Wall distance weight factor ϕ_{wall_dist}: Used for keeping the robot at a fixed distance from the wall.

$$\begin{aligned} E_y =& w_{avg}P_{avg}(D_{tracked}) + w_{max}P_{max}(D_{tracked}) \\ & - w_{avg}P_{avg}(D_{opposite}) - w_{max}P_{max}(D_{opposite}) \end{aligned} \tag{16}$$

$$\begin{cases} \vec{v}_{tracked} = \begin{cases} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, \text{ if } D_{tracked} = D_{left} \\ \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}, \text{ if } D_{tracked} = D_{right} \end{cases} \\ w_{wall\_dist} = |E_y| \\ \vec{v}_{wall\_dist} = sgn(E_y)\vec{v}_{tracked} \end{cases} \tag{17}$$

3) Wall tracking direction weight factor ϕ_{dir}: Used for keeping the robot as parallel to the wall as possible.

$$\begin{cases} w_{dir} = |E_{dir}| \\ \vec{v}_{dir} = \begin{bmatrix} 0 & 0 & -sgn(E_{dir}) \end{bmatrix} \end{cases} \tag{18}$$

4) Path following angle weight factor ϕ_{path}: Used for following the calculated path during navigation.

$$\begin{cases} w_{path} = |E_{path}| \\ \vec{v}_{path} = \begin{bmatrix} 0 & 0 & sgn(E_{path}) \end{bmatrix} \end{cases} \tag{19}$$

5) Forward movement weight factors $\phi_{forward\_wall}$ and $\phi_{forward\_path}$: Used for moving forward when in a suitable position near the wall.

$$\begin{cases} \gamma_{wall\_dist} = max(1, min(0, 1 - 2|E_y|)) \\ \gamma_{dir} = max(1, min(0, 1 - 1.25|E_{dir}|)) \\ w_{forward\_wall} = \gamma_{wall\_dist}\gamma_{dir} \\ \vec{v}_{forward} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{cases} \tag{20}$$

For navigation, as E_y and E_{dir} are not used, the maximum pressure in the front was chosen.

$$\begin{cases} w_{forward\_path} = max(1, min(0, 1 - 1.25P_{max}(D_{front}))) \\ \vec{v}_{forward} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \end{cases} \tag{21}$$

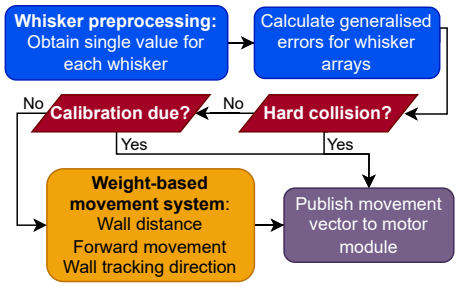

Fig. 5: High-level working principle for the wall-following algorithm, with decision points marked red and whisker processing marked blue.

G. Hard collision avoidance

To avoid situations with the algorithm that could potentially cause damage to the robot's components, a hard collision avoidance system was implemented. The system uses predefined thresholds $\xi_{avg} = 0.4$ for $P_{avg}(i)$ and $\xi_{max} = 0.8$ for $P_{max}(i)$ to invoke hard collision avoidance, upon which the robot stops and slowly moves back from the side with the highest pressure. To avoid perpetual oscillations between either front-rear or left-right directions in tight spaces, a threshold reduction $\xi_r = 0.3$ was also defined.

If perpetual oscillation is detected and wall-following is being performed ($D_{tracked}$ is set), then the movement direction is the opposite of the tracked direction, as we can be sure there is a wall on the other side. However, if $D_{tracked}$ is not set and perpetual oscillations are detected, e.g. between the left and right sides, then the directional error $E_c(i)$ is calculated for the sides between which the robot is oscillating in order to determine whether the front or rear side is more free to move towards.

H. Wall-following algorithm

The wall-following algorithm (Fig. 5) is repeated every time an update for the orientation of the whiskers is received (at 20 Hz). To obtain a movement vector for the robot when wall-following:

1) Check for hard collision using the previously described algorithm.
2) If calibration has been initiated, then use the previously described calibration method.
3) If $P_{avg}(D_{tracked}) < \xi_{track}$, i.e., no contact is detected on the tracked wall with a given threshold $\xi_{track} = 0.05$, then move slowly towards the $D_{tracked}$ direction.
4) Calculate the movement vector using (14) with set $S_{wall} = \{\phi_{base}, \phi_{wall\_dist}, \phi_{forward\_wall}, \phi_{dir}\}$.

In addition, if weight-based movement was not used in the previous cycle, then the integral values on the PID controllers should be reset to 0 to prevent integral windup.

I. Navigation algorithm

For the navigation algorithm (Fig. 6), the robot's pose is also taken into account, obtained using external sensors in this work. The algorithm, intended for navigating unknown environments, is the following:

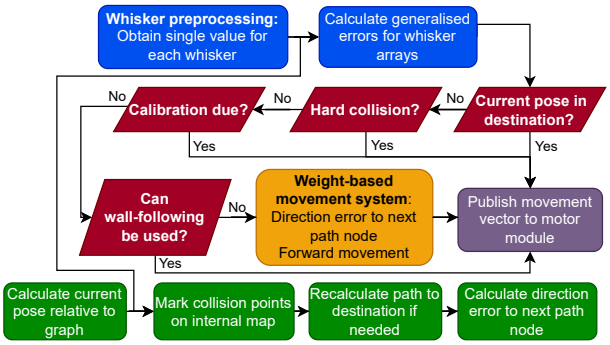

Fig. 6: High-level working principle for the navigation algorithm, with decision points marked red, whisker processing marked blue and path calculation marked green.

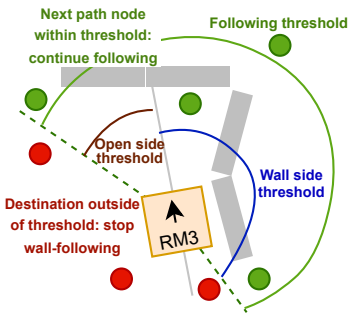

Fig. 7: Use of right wall-following during navigation

1) If the current position is the destination or no destination has been supplied, hold the robot still.
2) Check for hard collision using the previously described algorithm.
3) If calibration has been initiated, then use the previously described calibration method.
4) If it is possible to use wall-following during navigation (see Section III-J) and it is enabled, then use the given movement.
5) Calculate the movement vector using (14) with set $S_{path} = \{\phi_{base}, \phi_{forward\_path}, \phi_{path}\}$.

In addition, if weight-based movement was not used in the previous cycle, then the integral values on the PID controllers should be reset to 0 to prevent integral windup.

J. Utilising wall-following in navigation

In order to improve navigation speed in unknown areas, wall-following can be utilised (Fig. 7) — if a wall is detected and the next path node is within the thresholds, wall-following can be used. Here, the open side threshold was set to $\xi_{open} = 40°$ and the wall side threshold to $\xi_{wall} = 150°$.

Wall-following can also be utilised as backup when no path to a destination can be calculated. When disengaging wall-following and rotating towards the path, there also needs to be a threshold for E_{path} to reach when rotating towards the next path node before continuing the same algorithm, as otherwise the robot will try to start wall-following again, leading to a perpetual loop. In addition, if there is contact

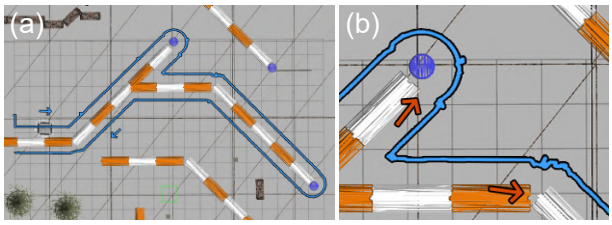

Fig. 8: Wall-following in simulation (a) and gaps in wall (b).

for only the front of the robot, then an attempt to rotate the robot towards its side to attempt wall-following is beneficial.

IV. Experiments and Discussion

The proposed algorithms were tested both in simulations and real-world environments. In simulations, obstacles or walls were represented using various Gazebo models, including road barriers, trees and barrels. For real-world testing (see accompanying video), we constructed uneven walls by merging polystyrene and bricks to replicate the irregularities of a cave wall. The robot confronted a series of challenges, ranging from narrow corridors, sharp turns, to dead-ends. Its position was continually monitored using a $0.54\,\mathrm{m}$ ArUco marker [38], and an *Intel Realsense D455* camera measured its pose in the Earth-fixed frame at a $10\,\mathrm{Hz}$ rate.

As the whisker tips were sharp, plastic end caps were glued onto them to reduce friction. In addition, to provide more reliable feedback, corner whiskers were vertically centered by tying them to both a high and a low point on the robot using strings.

For performance assessment, we compare the recorded traversal time and distance to a 'theoretical fastest time' (TFT). This benchmark is derived by summing the linear and angular distances divided by their respective maximum speeds. These speeds were $0.18\ m/s$ and $22.50°/s$ in simulations and $0.053\ m/s$ and $6.45°/s$ in real-world experiments. Examining the results in Table III, we observe the performance of the wall-following algorithm in both the simulated and real-world environments compared to the TFT for the chosen path.

In simulations, wall-following took 39.22% longer than the TFT to traverse the desired path. The distance covered was consistent with the expected distance, showing the algorithm's efficacy in navigating in an unknown environment. The wall-following trajectory in simulation (Fig. 8a) showcases an almost optimal route, with deviations mainly due to the simulated calibration procedure. A closer examination of sharp turns and wall gaps is provided in Fig. 8b.

The real-world scenario presents more variability. As reported in Table III, wall-following (WF) took over twice the time compared to the TFT. This can be attributed to real-world challenges not present in simulations. The algorithm had to handle intricate gaps and acute corners (see Fig. 9) along the path which led to the time discrepancy. Furthermore, the screw-based locomotion system of the robot demonstrated some difficulty in managing the heavily coupled degrees of freedom, resulting in minor navigation

TABLE III: Performance of wall-following and navigation algorithms compared to the theoretical best times. T: Time, D: Distance.

		Simulation				Real-world			
	Description	**T (s)**	**T (%)**	**D (m)**	**D (%)**	**T (s)**	**T (%)**	**D (m)**	**D (%)**
Wall-Following	TFT for path	363.04	100.00%	59.71	100.00%	531.49	100.00%	19.42	100.00%
	Wall-follower	505.40	139.22%	59.71	100.00%	1304.60	245.46%	21.40	110.19%
Navigation	TFT for explored map	149.53	64.87%	24.67	72.55%	116.35	55.20%	5.87	58.37%
	TFT for N-WF path	230.52	100.00%	34.00	100.00%	210.76	100.00%	10.06	100.00%
	N-WF	420.85	182.57%	34.00	100.00%	810.40	384.51%	10.06	100.00%
	N	571.70	248.01%	36.21	106.50%	-	-	-	-

Fig. 9: **Wall-following on a real-world grass surface.** Due to noise, the pink line signifies the path with measurements averaged over 10 readings, and the green for 100 readings.

delays. This is further emphasized by the slightly higher distance value of 110.19%, suggesting occasional deviation from the optimal path.

Table III compares the performance of two studied navigation algorithms: Theta* with wall-following (N-WF) and the regular Theta* (N) which does not leverage wall-following. In simulations, N-WF achieved 182.57% of the TFT outpacing the N algorithm, which took 248.01% longer. This suggests that integrating wall-following into the Theta* algorithm enhances time-efficiency in pathfinding. Fig. 10a and Fig. 10b depict the travel paths for N-WF and N, respectively. Though N-WF's time exceeded the TFT, it maintained a consistent path distance. Given N-WF's superior simulation performance, the N algorithm was not tested in real-world experiments. In the real-world tests, the N-WF algorithm demonstrated its efficacy by completing its path in 384.51% of the theoretical fastest time. The specific path taken by the proposed N-WF algorithm can be visualized in Fig. 11.

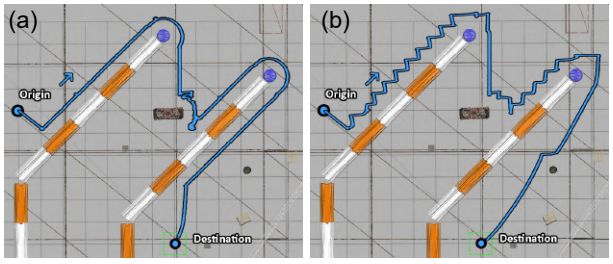

Fig. 10: Navigation in simulation. (a) N-WF and (b) N

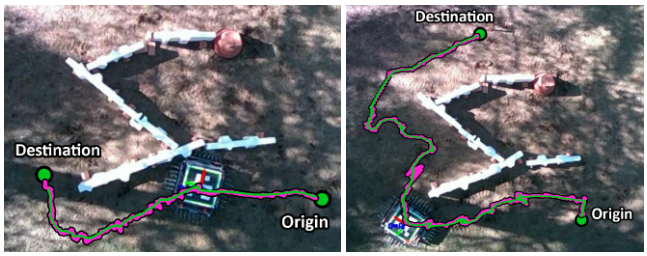

Fig. 11: Real-world N-WF paths

In sum, the results highlight the advantages of integrating tactile sensors within a wall-following navigation algorithm, especially when tailored for the demanding and unpredictable environments of underground mining.

V. Conclusions

This work demonstrated the ability to perform navigation using whisker sensors. While separately, the wall-following and Theta* navigation algorithms are not new concepts, their fusion in the context of tactile feedback from whiskers presents an innovative approach with great potential in sensory-deprived environments. While this method can not compete with state-of-the-art LiDAR and camera-based techniques under perfect conditions, it could be used to augment or even replace them when those methods' performance declines. Compared to previous implementations of whisker-based navigation that were tested in simulations or simplified physical conditions with flat non-slippery floors and straight walls with easily defined edges, our prototype and algorithms demonstrate good performance in outdoor conditions.

The work presented here opens up many avenues for further research, to improve the locomotion system, whisker configurations, the navigation algorithm itself and also to develop more realistic simulation environments. Future research can also look into extending the algorithm to other inexpensive sensing technologies and combining the work with SLAM algorithms to cover localization needs. An additional avenue of research is unifying the robot with other robots and interfaces for use in mine and cave environments, for industrial and research applications.

REFERENCES

[1] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," *SpringerPlus*, vol. 5, pp. 1–26, 2016.

[2] M. Labbé and F. Michaud, "Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of field robotics*, vol. 36, no. 2, pp. 416–446, 2019.

[3] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.

[4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[5] P. Kolar, P. Benavidez, and M. Jamshidi, "Survey of datafusion techniques for laser and vision based sensor integration for autonomous navigation," *Sensors*, vol. 20, no. 8, p. 2180, 2020.

[6] M.-A. Sayegh, H. Daraghma, S. Mekid, and S. Bashmal, "Review of recent bio-inspired design and manufacturing of whisker tactile sensors," *Sensors*, vol. 22, no. 7, p. 2705, 2022.

[7] S. Kim, C. Velez, D. K. Patel, and S. Bergbreiter, "A magnetically transduced whisker for angular displacement and moment sensing," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 665–671, 2019.

[8] J. C. Sullivan, B. Mitchinson, M. J. Pearson, M. Evans, N. F. Lepora, C. W. Fox, C. Melhuish, and T. J. Prescott, "Tactile discrimination using active whisker sensors," *IEEE Sensors Journal*, vol. 12, no. 2, pp. 350–362, 2011.

[9] O. Struckmeier, K. Tiwari, M. Salman, M. J. Pearson, and V. Kyrki, "ViTa-SLAM: A Bio-inspired Visuo-Tactile SLAM for Navigation while Interacting with Aliased Environments," *2019 IEEE International Conference on Cyborg and Bionic Systems, CBS 2019*, pp. 97–103, 9 2019.

[10] M. Asadnia, A. G. P. Kottapalli, J. Miao, M. E. Warkiani, and M. S. Triantafyllou, "Artificial fish skin of self-powered micro-electromechanical systems hair cells for sensing hydrodynamic flow phenomena," *Journal of the Royal Society Interface*, vol. 12, no. 111, p. 20150322, 2015.

[11] X. Zhang, X. Shan, T. Xie, J. Miao, H. Du, and R. Song, "Harbor seal whisker inspired self-powered piezoelectric sensor for detecting the underwater flow angle of attack and velocity," *Measurement*, vol. 172, p. 108866, 2021.

[12] P. Xu, X. Wang, S. Wang, T. Chen, J. Liu, J. Zheng, W. Li, M. Xu, J. Tao, and G. Xie, "A triboelectric-based artificial whisker for reactive obstacle avoidance and local mapping," *Research*, vol. 2021, 2021.

[13] J. Liu, P. Xu, J. Zheng, X. Liu, X. Wang, S. Wang, T. Guan, G. Xie, and M. Xu, "Whisker-inspired and self-powered triboelectric sensor for underwater obstacle detection and collision avoidance," *Nano Energy*, vol. 101, p. 107633, 2022.

[14] J. H. Solomon and M. J. Hartmann, "Robotic whiskers used to sense features," *Nature 2006 443:7111*, vol. 443, no. 7111, pp. 525–525, 10 2006. [Online]. Available: https://www.nature.com/articles/443525a

[15] G. Liu, Y. Jiang, P. Wu, Z. Ma, H. Chen, and D. Zhang, "Artificial whisker sensor with undulated morphology and self-spread piezoresistors for diverse flow analyses," *Soft Robotics*, vol. 10, no. 1, pp. 97–105, 2023.

[16] J. Stocking, W. Eberhardt, Y. Shakhsheer, B. Calhoun, J. Paulus, and M. Appleby, "A capacitance-based whisker-like artificial sensor for fluid motion sensing," in *SENSORS, 2010 IEEE*. IEEE, 2010, pp. 2224–2229.

[17] A. Dagamseh, C. Bruinink, H. Droogendijk, R. J. Wiegerink, T. S. Lammerink, and G. J. Krijnen, "Engineering of biomimetic hair-flow sensor arrays dedicated to high-resolution flow field measurements," in *SENSORS, 2010 IEEE*. IEEE, 2010, pp. 2251–2254.

[18] B. Eijking, R. Sanders, and G. Krijnen, "Development of whisker inspired 3d multi-material printed flexible tactile sensors," in *2017 IEEE SENSORS*. IEEE, 2017, pp. 1–3.

[19] N. H. Nguyen and V. A. Ho, "Mechanics and morphological compensation strategy for trimmed soft whisker sensor," *Soft Robotics*, vol. 9, no. 1, pp. 135–153, 2022.

[20] C. Xiao, S. Xu, W. Wu, and J. Wachs, "Active Multiobject Exploration and Recognition via Tactile Whiskers," *IEEE Transactions on Robotics*, 12 2022.

[21] J. T. Reeder, T. Kang, S. Rains, and W. Voit, "3d, reconfigurable, multimodal electronic whiskers via directed air assembly," *Advanced Materials*, vol. 30, no. 11, p. 1706733, 2018.

[22] J. Z. Gul, K. Y. Su, and K. H. Choi, "Fully 3d printed multi-material soft bio-inspired whisker sensor for underwater-induced vortex detection," *Soft robotics*, vol. 5, no. 2, pp. 122–132, 2018.

[23] B. J. Wolf, J. A. Morton, W. N. MacPherson, and S. M. Van Netten, "Bio-inspired all-optical artificial neuromast for 2d flow sensing," *Bioinspiration & biomimetics*, vol. 13, no. 2, p. 026013, 2018.

[24] T. A. Kent, S. Kim, G. Kornilowicz, W. Yuan, M. J. Hartmann, and S. Bergbreiter, "WhiskSight: A Reconfigurable, Vision-Based, Optical Whisker Sensing Array for Simultaneous Contact, Airflow, and Inertia Stimulus Detection," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3357–3364, 4 2021.

[25] A. Ristolainen, J. A. Tuhtan, and M. Kruusmaa, "Continuous, near-bed current velocity estimation using pressure and inertial sensing," *IEEE Sensors Journal*, vol. 19, no. 24, pp. 12 398–12 406, 2019.

[26] W. C. Eberhardt, B. F. Wakefield, C. T. Murphy, C. Casey, Y. Shakhsheer, B. H. Calhoun, and C. Reichmuth, "Development of an artificial sensor for hydrodynamic detection inspired by a seal's whisker array," *Bioinspiration & biomimetics*, vol. 11, no. 5, p. 056011, 2016.

[27] M. J. Pearson and M. Salman, "Active Whisker Placement and Exploration For Rapid Object Recognition," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 672–677.

[28] M. J. Pearson, C. Fox, J. C. Sullivan, T. J. Prescott, T. Pipe, and B. Mitchinson, "Simultaneous localisation and mapping on a multi-degree of freedom biomimetic whiskered robot," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 586–592, 2013.

[29] C. Fox, M. Evans, M. Pearson, and T. Prescott, "Tactile SLAM with a biomimetic whiskered robot," in *Proceedings - IEEE International Conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., 2012, pp. 4925–4930.

[30] J. Lee, S. N. Sponberg, O. Y. Loh, A. G. Lamperski, R. J. Full, and N. J. Cowan, "Templates and anchors for antenna-based wall following in cockroaches and robots," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 130–143, 2 2008.

[31] Xianchao Long, "Tactile-Based Mobile Robot Navigation," Master's thesis, Worcester Polytechnic Institute, Worcester, 5 2013. [Online]. Available: https://web.wpi.edu/Pubs/ETD/Available/etd-061313-134710/unrestricted/xlong.pdf

[32] "Robominers - resilient Bio-inspired Modular Robotic Miners." [Online]. Available: https://robominers.eu

[33] A. Nagel, "Blind Mapping and Localisation for Small-Scale Mining Robots," Master's thesis, Tallinn University of Technology, Tallinn, 2021.

[34] Tanel Kossas, "Tactile Navigation Algorithm For Autonomous Underground Mining Robots," Master's thesis, Tallinn University of Technology, Tallinn, 6 2023. [Online]. Available: https://digikogu.taltech.ee/en/Download/7f60430f-a1f8-4182-a0c1-136272a1b71f

[35] K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," *J. Artif. Intell. Res. (JAIR)*, vol. 39, 01 2014.

[36] B. J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Journal of Fluids Engineering*, 1942. [Online]. Available: https://api.semanticscholar.org/CorpusID:41336178

[37] K. J. Åström and T. Hägglund, *Advanced PID Control*. International Society of Automation (ISA), 2006. [Online]. Available: https://app.knovel.com/hotlink/toc/id:kpAPIDC001/advanced-pid-control/advanced-pid-control

[38] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.