Development of a fuzzy logic controller for autonomous navigation of building inspection robots in unknown environments

Siwei Chang[1], Ming-Fung Francis Siu[2], Heng Li[3]

[1]Ph.D. Candidate, Department of Building and Real Estate, Faculty of Construction and Environment, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (E-mail: siwei.chang@connect.polyu.hk).
[2]Assistant Professor, Department of Building and Real Estate, Faculty of Construction and Environment, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (E-mail: francis.siu@polyu.edu.hk).
[3]Chair Professor, Department of Building and Real Estate, Faculty of Construction and Environment, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (E-mail: heng.li@polyu.edu.hk).

Abstract

Robotic building inspection is gaining popularity as a way to increase the security, productivity, and cost-effectiveness of traditional inspection tasks. Despite the development of numerous building inspection robotic platforms, their motions still require manual control. To facilitate full automation, there is a need to explore autonomous navigation strategies for building inspection robots. Although various autonomous navigation strategies have been developed in the robotics field, few of them are suitable for building structural inspection behavior. In accordance with the responsibilities of professional inspectors, the robot is required to follow the structural components within a desired distance and dynamically avoid obstacles to conduct in-depth scanning. This navigation task becomes more difficult when providing smooth following path in special building scenarios, such as narrow corners. Motivated by this need, the present study aimed to explore autonomous navigation for building inspection robots. To save the cost of map construction, the local navigation strategies, which control the robots' travel in unknown environments, were targeted. Specifically, the objective is to develop a robust fuzzy logic controller (FLC) for wall-following behavior. The inputs are the distances within the designed interval ranges, which were measured with a 360-degree laser. The membership functions and the decision-making rules were designed based on robot and camera configurations, building designs, and structural inspection criteria. The outputs are the real-time angular and linear velocities. Tested in both simulation and real-world environments, the novelty of the designed FLC is: 1) enabling "finding wall," "wall-following," "turning," and "obstacle avoidance" behaviors in various unknown building scenarios; 2) preventing wavy motions; and 3) preventing path deviations for arbitrary surfaces. The results can be employed to perform daily structural inspections, and they are dedicated to automating the building inspection tasks. However, the FLC is sensitive to the reflective components because of the limitations of the position sensors.

Keywords: Autonomous navigation, mobile robot, fuzzy logic, path planning, building inspection

1. Introduction

Building inspection is essential to ensure the quality of building structures in compliance with legal regulations and construction codes. Effective and routine building inspection contributes to a smooth construction process and high-quality construction deliverables.

Building inspections are necessary for both new and aging buildings. For new buildings, including newly constructed buildings that have green building designs, quality inspection takes place during the whole construction stage, from project commencement to completion. Authorized site inspectors are hired to work on-site and are responsible for the regular inspection tasks, such as witnessing tests, monitoring construction progress, assessing defects, and providing quality reports. For existing buildings, quality inspection is mostly required for aged buildings or before buying a new home. Professional inspectors are hired to fully examine quality defects that exist in the building's interior and exterior. All the building components, such as foundations, structural and non-structural components, and plumbing, should be inspected.

In the inspection process, visual examination is applied most frequently to inspect defects such as cracks, stains, and spalling. However, traditional visual inspection has several drawbacks, including 1) Unsafe. Sometimes, visual inspection is dangerous because inspectors need to reach hazardous locations, such as high roofs or narrow pipelines, to see whether defects exist in building surfaces. 2) Costly. Considering the professional and risky nature of building inspections, the cost of hiring inspectors is relatively high. For example, the average annual salary of site inspectors, last updated on Feb. 20, 2021, is HK $408k, according to statistics from Payscale, Inc (PayScale, 2022). 3) Lack of efficiency. Manual inspection is inefficient since there are various types of defects that can appear in any building component, such as walls and columns, as well as at any construction stage, such as in the structural or decorative stage.

To solve the problems, inspection robotics is constantly being developed and employed. Using robotic systems to autonomously complete inspection works contributes to 1) improving productivity and reducing reliance on manual labor. Constant or 24/7 work on a repetitive cycle can be autonomously demonstrated. 2) Increasing precision. By executing tasks with mathematical programming and electronic sensors, the precise inspection can be guaranteed. 3) Reducing safety hazards. Robotic systems can be utilized to conduct hazard inspections without human intervention, such as examining defects in high-rise buildings' external walls.

Therefore, various inspection robotic systems have been invented in response to the trend of technological advancement. For example, snake-shaped robotic mechanisms for sewer pipe inspection (Baba et al., 2010, Selevarajan et al., 2019, Wakimoto et al., 2003), novel adhesion mechanisms of wall-climbing robots for high-rise building inspection (Faruq and Sattar 2015, Hillenbrand et al., 2008, Nguyen and La, 2021), and mobile robots for tunnel defect inspection (Yu et al., 2007). Cameras installed on inspection robotic systems are used to capture images or videos of buildings. By computing the captured photographs with the advanced computer vision algorithms (Chen and Jahanshahi 2017, Cho et al., 2014, Chen and Gupta 2017, Redmon et al, 2016), automated and remote demonstrations of building defects can be received in real-time instead of inspecting them manually on-site.

While a lot of research has been devoted to developing computer-vision algorithms for automated defect recognition, little research has been given to suggesting navigation approaches for building inspection robots. As a result, the robotic systems are required to be manually controlled to the inspection spots before inspecting building components. The motion of the teleoperated building inspection robot, for example, is controlled by operators using joysticks and virtual reality (Tang and Yamada 2011). Therefore, this study aims to develop a navigation strategy for building inspection robots to move autonomously in unknown environments.

In the robotic field, various autonomous navigation algorithms have been developed. Among them, this study focused on the wall-following algorithm (Yata and Yuta, 1998), a significant robotic control method, since it aids in controlling the robot to follow the inspection items and then perform detailed inspection scanning. While the other algorithms, such as the Kalman filtering approaches (Ayache and Faugeras 1989, Crowley 1989) and the Occupancy grids approach (Moravec 1985), focus more on robots' navigation from the start locations to the target locations, which may cause inaccurate inspection. However, because of the simple designs, the existing wall-following algorithms are not efficient enough for the building inspections, especially for buildings components with arbitrary shapes, such as curved interior walls. The wavy motions are also difficult to avoid, which affects the accuracy of the computer vision-based defect recognition.

Motivated by this research gap, the present study designed a fuzzy logic controller (FLC) to enhance the wall-following navigation of building inspection robots. Particularly, we focused on the structural inspection of both new and aging buildings because: 1) structural inspection provides vital proof for building safety (Hoskere et al., 2018); and 2) the repetitive and tedious behaviors of structural examination can be easily imitated by a mobile robot equipped with cameras. Automating structural inspections with robots increases efficiency and effectiveness. Meanwhile, the designed navigation strategy is feasible for a wide range of building types, including residential, commercial, and industrial buildings. Validated in both simulation and on-site building environments, the designed FLC system successfully controls the robot to conduct “finding wall”, “turning”, “wall-following”, and “obstacle avoidance” behaviors in various unknown building scenarios, including irregular regions, such as concave and convex walls and narrow aisles. In addition, the FLC controls the robot to follow walls straight within a desired distance and the path deviation problem can be effectively avoided. The novelty of the present study is the robust FLC design for the wall-following behavior of building inspection robots. The outcomes can be easily coded into the robotic systems for their autonomous navigation therefore contribute the industry with a fully automated robotic inspection system for daily building inspection work. The paper demonstrates the following contributions: 1) A literature review on navigation strategies for building inspection robots, wall-following algorithms, and FLCs for wall-following navigation. 2) A novel FLC design to improve the wall following algorithm. 3) Testing the designed FLC in simulation and on-site building environments.

2. Literature review

The basic theory, advantages, and limitations of the robot’ autonomous navigation, wall-following behavior, and designing FLCs for the wall-following behavior of building inspection robots are discussed to highlight research significance.

2.1 Navigation method for building inspection robot

One of the key research fields for achieving fully automated robot control is autonomous navigation. Various strategies have been explored so far to achieve autonomous navigation (Mantha et al., 2022, Demiral et al., 2021). For example, the visual marker-based indoor navigation developed for user-centered interactive applications (Naheem et al., 2022). By continuously acquiring and computing the position data stored in the markers, such as RFID and fiducial markers, autonomous navigation is achieved. Although autonomous navigation strategies have been widely developed in the robotics field, few of them are suitable to control the motions for building inspection behaviors, such as following the walls to conduct detailed quality scanning. As a result, human interventions are still required for the majority of inspection robots to control their movements and speeds (Bui et al., 2020). For example, operators remotely control the movement and speed of the building inspection robots using joysticks (Özaslan, T et al., 2017, Kaiwart et al., 2022). Specifically, operators employ joysticks to control the robots to move forwards, backwards, and turn after observing the visions of the surrounding environments, which are provided by cameras or virtual reality models.

Even so, some of the popular algorithms have begun to be integrated into the building inspection robots to gradually achieve autonomous navigation. Among them, the simultaneous localization and mapping (SLAM) technique (Durrant and Bailey 2006) has been frequently implemented for building inspection robots to conduct on-site quality inspections (Asadi et al., 2021, McLaughlin and Narasimhan 2020). In SLAM, an environmental map is needed in advance to obtain the positions of surrounding objects and the robots to plan their moving paths. For that reason, it is not convenient to employ SLAM for continuous building inspection work, although it contributes to autonomous and accurate navigation. For example, it is common to place or move objects in buildings, such as wardrobes, while the robot is operating. Because of the high computation cost of map construction, it is easy for the SLAM algorithm to fail to generate new path plans when objects change places, and a collision may occur.

To save the computation capacity of map construction, local navigation strategies, which directly process real-time sensor-provided position information (Gul and Nazli 2019), have also been employed to navigate mobile robots (Lee et al., 2021) in unknown building environments. For example, the navigation method, developed based on feedback control, was used to control the mobile robot to negotiate building corridors (Shi et al., 2006). The fiducial marker-based strategy was developed to navigate the mobile robot to travel from the initial position to the goal position in buildings (Mantha et al., 2018). For a structural inspection robot, it is worthwhile to investigate autonomous navigation in unknown environments because: 1) it is better suited to changing building environments; and 2) it is challenging to create maps of specialized building environments, such as aging nuclear power plants. Although the mentioned strategies realize the autonomous navigation in an unknown indoor building environment, they were not planned for structural inspection behaviors, which require the robot to follow the structural components within the desired distance to conduct detailed quality scanning. Motivated by the gaps, the present study aimed to investigate appropriate autonomous navigation strategies for building structural inspection robots in unknown environments.

2.2 Wall following algorithm

The present study focused on the wall-following algorithm (Saman and Abdramane, 2013), one of the significant ideas of robotics (Che et al., 2022, Wu et al., 2021), among various local navigation strategies because it is an appropriate local navigation solution for autonomous inspection tasks and has been continuously implemented in the inspection robotic platforms (Wei et al., 2017). Other local navigation strategies, on the other hand, place a greater emphasis on generating a moving path from the start points to the goal points (Altman 1992, Katoch et al., 2021), or obstacle avoidance (Poli and Blackwell 2007). Controlled by the wall-following behavior, the robots can constantly follow building components within a desired distance to conduct detailed inspection scanning.

Initially, wall-following behavior was developed as an efficient way to achieve autonomous navigation in maze environments (Saman, 2013). By computing the position data, obtained by sensors, with decision-making loops, the wall-following behavior guides the robot to turn right, left, and straight to follow the maze walls until it reaches the goal location. The principle is depicted in **Fig. 1**.

However, because of the simple logic, the initial wall-following behavior is ineffective for real-life building environments. Especially for the building components with arbitrary shapes, such as curved columns, concave and convex corners. For example, when the robot follows the rounded columns, the second rule "IF front is open, Then go_forward" may be triggered to control the robot to go forward instead of following the rounded columns, which generates path deviations. In addition, the traditional wall-following algorithm fails to keep the robot following walls in a straight line and within a desired distance, which makes it hard to capture distinct pictures for computer-vision based defect recognition. Therefore, the present study intends to improve the robustness of the wall-following logic for building inspection robots. The purpose is to ensure the building inspection robots to smoothly follow building components in various unknown building environments within a desired distance.

2.3 Fuzzy logic controller

The significance of wall-following behavior has made it a worthwhile topic to discuss its optimization strategies (Xue et al., 2020). Several ways have been proposed to enhance wall-following behavior, including machine learning algorithms (Hammad et al., 2019, Teng et al., 2020) and the fuzzy logic controller (FLC) (Omrane et al., 2016, Fatmi et al., 2006, Malhotra and Sarkar 2005, Faisal et al., 2013). The present study employs the fuzzy logic controller (FLC) for the following reasons: 1) FLC has been proven as an effective tool and is widely used for improving wall-following behavior because of its outstanding ability to deal with complex uncertainty, such as various decision-making rules (Suwoyo et al., 2020). 2) FLC is more applicable because it can be simply coded and computed. It is an efficient navigation strategy for most robotic systems that are controlled with CPUs, such as the Raspberry PI. Powerful GPUs are needed to compute a hundred million parameters in the machine learning process.

Similarly to the wall-following algorithm, the FLC generates navigation commands for robots based on input data and linguistic decision-making rules. Differently, the output of the FLC is more precise by computing precise input data with comprehensive decision-making rules and the membership functions (Novák et al., 2012). Instead of directly employing a developed FLC, we designed a new one for the wall-follow navigation of building inspection robots because: 1) it is hard to find a suitable one for building inspection robots. Majority FLCs were designed for the navigation problem of traveling from the start points to the goal points with obstacle avoidance instead of the wall-following behavior (Aouf et al., 2019, Singh and Thongam, 2018). 2) Because of the inappropriate design of the ranges of input data, the ranges and types of membership functions, and the decision-making rules, existing FLCs designed for wall-following behavior are not accurate enough for building environments. For example, path deviation happens in concave regions and wavy motion occurs in the FLC designed by (Muthugala et al., 2020).

In summary, the objective is to design a robust FLC for the wall-following behavior of building inspection robots. The ranges of the input data, the membership functions, and the decision-making rules were detailed based on camera and robot configurations, building designs, and inspection criteria. The outcomes contribute to achieving autonomous building inspection behaviors, including “finding wall,” “wall following,” “turning,” and “obstacle avoidance,” in various unknown building scenarios, especially for components with arbitrary shapes. Meanwhile, keep the robot following walls at a desired distance and avoid path deviation and wavy motions.

3. Research methodology

3.1 Robot kinematic model

In this study, the Turtlebot3 burger was employed as the testing robot. The Turtlebot3 burger is a three-floor octagon-shaped platform. The distance between the left and right wheels is 160mm (L), and its radius is 33mm (R). A 360-degree laser is installed on the top floor to obtain position information. The scanning distance ranges from 120mm to 3,500mm. A simplified depiction of the robotic platform is shown in **Fig. 2**.

Here, the established world frame and body frame are presented as $\{W\}$ and $\{B\}$, respectively. x_b, y_b refer to the heading directions of the robot, α refers to its orientations. V_l and V_r refer to the velocities of the left and right wheels, respectively. V and ω, referring to the linear and angular velocity, are directly used to control robot’s movement. The kinematic dynamics can be explained using the mentioned variables as equation 1:

$$\begin{cases} \dot{x}_b = \frac{R}{2}\left(V_r + V_l\right)\cos\alpha \\ \dot{y}_b = \frac{R}{2}\left(V_r + V_l\right)\sin\alpha \\ \dot{\partial} = \frac{R}{L}\left(V_r - V_l\right) \end{cases} \qquad \text{Equation (1)}$$

Here V_r and V_l can be expressed using V and ω as equation 2:

$$\begin{cases} \dot{V}_r = \dfrac{2V + \omega L}{2R} \\ \dot{V}_l = \dfrac{2V - \omega L}{2R} \end{cases} \qquad \text{Equation (2)}$$

Based on the right-hand rule (Widnall and Peraire 2008), both linear velocity V and angular velocity ω have three dimensions x, y, z . Because the robot is expected to conduct 2-dimension navigation, only the x dimension of V , and z dimension of ω for both right and left wheels are considered, namely *linear.velocity.x*, *angular.velocity,z*.

3.2 Fuzzy logic controller design

The research framework is shown in **Fig. 3** based on the operation flow of FLC. Four groups of parameters should be designed rationally to meet the requirements of building inspection work: input data, membership functions, decision-making rules, and defuzzification functions. Based on the mentioned kinetic model, linear velocity and angular velocity are used as the FLC outputs to control the speed and direction, respectively.

The designs were detailed based on the following requirements. It is expected that, controlled by the FLC designs, the robot can successfully perform "finding wall," "wall following," "turning," and "obstacle avoidance" behaviors in various building scenarios, as well as avoid wavy motions and path deviations. As mentioned, the building inspection robots employ computer-vision based object recognition techniques to realize automated defect inspection. Therefore, in the "wall following" process, the robot is also expected to move straight while keeping a desired distance from the building components to provide distinct views.

Because a behavior-based strategy is the principle for designing FLC (Cai et al., 2008), the requirements were established based on manual inspection behaviors. In a real-world inspection case, site inspectors basically walk along the edges of buildings within an appropriate distance and examine whether defects exist. The inspectors will come to a stop until all of the building elements have been fully inspected. In most cases, inspectors will naturally avoid frontal obstacles such as sand piles and hydrants.

3.2.1 Design of input data

Based on the principle of wall-following behavior, the left, front, and right distances between the robot and the nearby objects are employed as the input data. We did several experiments to define the most appropriate ranges of the input distances. According to (Muthugala et al., 2020) and (Schiffer et al., 2012), we first tested the separate (left: -90°, front: 0°, right: 90°) and continuous ranges (left: -45°~-135°, front: -45°~45°, right: -225°~-315°). The results showed that the robot fails to respond timely or responds too frequently when using the separate and continuous ranges, respectively. For example, when using the separate ranges, it is possible for the robot to collide with obstacles that are located obliquely ahead, such as those located at 15° instead of 0°. When using continuous ranges, it is possible to constantly alternate motion commands. For objects at the dividing lines of ranges, such as -315°, the robot is ambiguous between the "turning" command, to avoid forehead obstacles, or the "wall-following" command, to move straight, because distances of -315° belong to the "front" and "right" ranges at the same time.

Therefore, instead of separate and continuous ranges, we defined the interval ranges for the left, front, and right distances: [150°-180°], referring to the left distance, [60°-120°], referring to the front distance, [0°-30°], referring to the right distance, which yielded optimal commands after testing. The defined interval ranges are in line with the findings in (Cherroun et al., 2019). A 360-degree laser mounted on the top floor of the testing robot was used to acquire distance data within a particular range.

3.2.2 Design of membership functions

To link the inputs to the decision-making rules, membership functions (MFs) were employed to transform the crisp inputs to their respective fuzzy degrees of each linguistic decision-making rule. Fuzzy sets (computing variables), types of MFs (computing functions), and linguistic labels (computing limits) of MFs were detailed to design the membership functions.

(1) Design of fuzzy sets

Based on the inputs and outputs, three fuzzy sets were determined as the computing variables for the designed MFs: 1) the distance set, contains the left, front, and right distances as well as their fuzzy degrees; the speed set, contains the linear velocity as well as its fuzzy degree; and the rotation set, contains the angular velocity as well as its fuzzy degree.

(2) Design of MF types

The triangular MFs, trapezoidal MFs, and singleton MFs, which are the most extensively used MF types, were employed (Ali et al., 2015). The fuzzy degree (μ_{A} , A refers to the fuzzy set) in triangular MFs, trapezoidal MFs, and singleton MFs are calculated using equations 3 to 5, respectively.

$$\mu_A(x) = \max(\min(\frac{x-a}{i-a},\ \frac{b-x}{b-i}),0) \qquad \text{Equation (3)}$$

Here, x is the input value, a and b are the lower and upper limit of the triangle, i is the average of a and b, $a < i < b$.

$$\mu_A(x) = \begin{cases} 0 & ,(x < a)\ or\ (x > d) \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1 & ,b \le x \le c \\ \frac{d-x}{d-c}, & c \le x \le d \end{cases} \qquad \text{Equation (4)}$$

Here, x is the input value, a and d are the lower and upper limit of the trapezoid, b and c are the lower and upper support limit of the trapezoid, $a < b < c < d$.

$$\mu_A(x) = \begin{cases} 1, & if\ x = a \\ 0, & otherwise \end{cases} \qquad \text{Equation (5)}$$

Here, x is the input value, a is the limit.

(3) Design of linguistic labels and their limits and ranges

(i) MFs for distance fuzzy set

The linguistic labels and their limits and ranges for the distance fuzzy set is shown in **Fig. 4.**

The linguistic labels for the distance fuzzy set were designed to report the status of robot positions. Referring to Antonelli et al., 2007, linguistic labels were defined as N: “Near,” A: “Appropriate,” F: “Far,” VF: “Very Far” (as shown in Fig. 3). We additionally designed a “VF” label to achieve the “finding wall” behavior, which enables the building inspection robots to locate inspection items at a fast speed and, therefore, to save time and energy.

As the building inspection robots employ cameras to receive defect vision, the lower limit was determined as 20 cm according to the minimum focus distance of cameras (Witt et al., 2022). If the shooting distance is closer than the minimum focus distance, blurred images may be shown because cameras are unable to focus on the subject properly. The minimum focus distance is varied for different focal lengths. In this study, we used a USB camera with a 4mm focal length as it is one of the most common camera types and has been frequently used in object recognition studies (Sagawa et al., 2004, Okazaki et al., 2008, Ren et al., 2021). The minimum focus distance for a lens with a 4mm focal length is typically around 20 cm.

According to the inspection regulations, for example, the mandatory building inspection scheme (MBIS), site inspectors are obliged to inspect defects from a close distance (Chan et al., 2014). If inspectors work beyond a close distance, they may miss minor defects (such as cracks). Therefore, the required close distance was employed as the upper limit. According to previous studies (Brown et al., 2001), the least distance of distinct vision of human eyes is

25 cm (Maclnnes and Smith 2010). The vision of objects' details gets blurred when the distance increases. Therefore, 25 cm was determined as the upper limit. This rule can be deemed feasible because images captured by cameras have a similar resolution to images perceived by human eyes when gazing at close objects (Skorka and Joseph 2011). Image resolution has a direct impact on how clear images are displayed (Patti et al., 1994).

For the Turtlebot 3, the inspection camera and laser were mounted at the right edge and in the centre of the robot's top floor, respectively. Because the distance MF is designed for the inspection camera (employed as a human eye) and the distance is measured starting from the laser, a supplemented distance of 8 cm, between the centre of the laser and the camera, was also considered. For different robots, the supplementary distance, the distance between the inspection cameras and the laser, can be varied. In summary, the desired distance between the robot and inspection elements is 28 to 33 cm, with a 1 cm margin of error. The ranges were fine-tuned in several experiments to realize the expected movements.

(ii) MFs for speed and rotation fuzzy sets

The designed linguistic labels and their limits and ranges for the speed and rotation fuzzy sets are shown in **Fig. 5.**

The linguistic labels for the speed and rotation fuzzy sets were designed based on the behavior-based strategy (Seraji and Howard 2022), seen as a design principle for membership functions. To realize the expected behaviors detailed in section 3.2.3, the linguistic speed labels were designed as Z: "Zero," L: "Low," M: "Middle," H: "High" (as shown in **Fig. 5 (a)**). The rotation labels were defined as TRF: "Turn right far," TRN: "Turn right near," GS: "Go straight," TLN: "Turn left near," TLF: "Turn left far" (as shown in **Fig. 5 (b)**). The performances of the "L", "M", "H", "TRN", "TRF", "GS", "TLF", "TLN" labels have been proven in (Hagras 2004) and (Lee et al., 2017), respectively. We additionally designed the "Z" label to realize the safely travel of the building inspection robots in special building environments, such as "turning in ground" in narrow building corners.

The limits of the speed and rotation fuzzy sets were determined based on the robot's configurations. The ranges were determined based on the relationship between the robot's turning radius, speed, and rotation (shown in equation 6). To control the robot to slightly adjust movements (using "L" or "M" with "TRN" or "TLN") when it is near to the following objects, the turning radius is required to be within the designed distance limits (0.28m~0.34m). To achieve the "finding wall" behavior at a fast speed (using 100% "H" with 100% "TRF" or 100% "TLF") when the robot is far from the following objects, the turning radius is required to be around 2m, half of the average size of building rooms

(Lead C, 2022), to make sure the robot can locate a building component instead of turning in the ground. It should be noted that the designed linguistic labels, limits, and ranges are feasible for most of the building inspection robots because the Turtlebot3 configurations are set at an average level. Whereas they can be adjusted for particular inspection tasks and robot specifications.

$$R = \frac{v}{\omega} \quad \text{Equation (6)}$$

Here: R is the turning radius, v is the linear velocity, ω is the angular velocity.3.2.3 Design of decision-making rules

Fuzzy rules are accountable for establishing decision-making logic. Modus ponens, the most essential expression of fuzzy rules (McGee 1985), was employed to design fuzzy rules. The form of a modus ponens rule is: ***IF*** *x* ***is*** *A* ***THEN*** *y* ***is*** *B*. Specifically, *x* and *y* refer to the variables in the distance fuzzy set and the speed and rotation fuzzy sets, respectively. *A* and *B* refer to the linguistic labels of the three fuzzy sets. The t-norms (Gupta and Qi 1991), used as an ***AND*** connector, were employed to connect the multiple conditions.

Based on the design principle: fuzzy rules are defined based on both the sensor input and the robot's launching scenarios (Dias et al., 2018), we defined the specific connections of *x*, *y*, *A*, *B* based on the possible launching scenarios. **Fig. 6.** shows the representative sensing and launching scenarios for building inspection robots according to the architectural layout designs (Rahbar et al., 2022). A typical building layout was used as a showcase. In **Fig. 6**, the dotted lines on the robot split the obtained distance data into left, front, and right distance; the arrows refer to the expected moving directions (suppose the camera is mounted on the right side).

The eight representative situations can be interpreted as **Table 1**. Avoiding forehead obstacles can be considered sub-scenarios of Scenario F, in which the robot can detect front elements. Examples are shown as F' and F''.

Table 1. Interpretation of representative launching scenarios

Interpretations / Scenarios	Building element is located on			Next direction
	Left	Front	Right	
Scenario A	√	×	×	Turn right in place
Scenario B	×	×	×	Turn right
Scenario C	×	×	√	Go straight
Scenario D	√	√	×	Turn left in place
Scenario E	×	√	√	Turn left
Scenario F	×	√	×	Turn left in place
Scenario G	√	×	√	Go straight
Scenario H	√	√	√	Turn left in place

As shown in **Table 2**, to realize the expected behaviors, the fuzzy rules were determined by considering every possible combination of the eight representative scenarios with the designed membership functions in section 3.2.2.

The modus ponens rule and t-norms were used to combine and connect the fuzzified variables. For example: ***IF*** Left ***is*** N ***AND*** Front ***is*** VF ***AND*** Right ***is*** VF, ***THEN*** Linear velocity ***is*** H ***AND*** Angular velocity ***is*** TRN. The minimum operator was used to decide rules' the entries (Hellman 2001).

Table 2. Fuzzy rules

Scenarios	Rules	Left	Front	Right	Linear velocity	Angular velocity
	A1	N			Z	TRN
A	A2	A	VF	VF	Z	TRN
	A3	F			Z	TRN
B	B	VF	VF	VF	H	TRF
	C1			N	L	TLN
C	C2	VF	VF	A	M	GS
	C3			F	L	TRN
	D1		N		Z	TLN
	D2	N	A	VF	Z	TLN
	D3		F		Z	TLN
	D4		VF		Z	TLN
	D5		N		Z	TLN
D	D6	A	A	VF	Z	TLN
	D7		F		Z	TLN
	D8		VF		Z	TLN
	D9		N		Z	TLN
	D10	F	A	VF	Z	TLN
	D11		F		Z	TLN
	D12		VF		Z	TLN
	E1			N	Z	TLN
	E2	VF	N	A	Z	TLN
	E3			F	Z	TLN
	E4			N	Z	TLN
E	E5	VF	A	A	Z	TLN
	E6			F	Z	TLN
	E7			N	Z	TLN
	E8	VF	F	A	Z	TLN
	E9			F	Z	TLN
	F1		N		L	TLN
F	F2	VF	A	VF	L	TLN
	F3		F		L	TLN
	G1			N	M	GS
	G2	N	VF	A	M	GS
	G3			F	L	TRN
	G4			N	M	GS
G	G5	A	VF	A	M	GS
	G6			F	L	TRN
	G7			N	M	GS
	G8	F	VF	A	M	GS
	G9			F	L	TRN
	H1	N	F	N	M	GS
	H2			A	M	GS
H	H3	N	N	N	Z	TLN
	H4	A	N	A	Z	TLN
	H5	A	N	F	Z	TLN

3.2.4 Design of defuzzification method

After the fuzzification and decision-making process, linguistic labels of output linear and angular velocities are obtained. The defuzzification process contributes to converting the linguistic labels to crisp numbers of outputs. The most commonly used defuzzification method, the centroid method (Chakraverty et al., 2019), was employed. As shown in equation 7 and equation 8.

$$v^* = \frac{\sum_{i=1}^{4} \mu(v_i) \times \bar{v}_i}{\sum_{i=1}^{4} \mu(v_i)} \qquad \text{Equation (7)}$$

Here: v^* is the crisp linear velocity, $\mu(v_i)$ is the fuzzy degree of the $i-th$ membership function in the speed fuzzy set, $\bar{v}_i$ is the centroid position of the $i-th$ membership function.

$$\omega^* = \frac{\sum_{i=1}^{3} \mu(\omega_i) \times \bar{\omega}_i}{\sum_{i=1}^{3} \mu(\omega_i)} \qquad \text{Equation (8)}$$

Here: ω^* is the crisp linear velocity, $\mu(\omega_i)$ is the fuzzy degree of the $i-th$ membership function in the rotation fuzzy set, $\bar{\omega}_i$ is the centroid position of the $i-th$ membership function.

3.2.5 Heading adjust algorithm

After several tests in Robot Operating System (ROS) simulation (Mittler 2017), we observed that when following the wall, the robot still moved in an S-curve rather than a straight path (scenario C), which causes difficulties for defect recognition. This may happen because the robot needed to adjust its heading timely in order to stay within the desired distance. Therefore, a heading adjust (HA) algorithm was designed to optimize the designed FLC by sending correct commands to control the robot to follow walls in a straight line.

The objective of the proposed HA algorithm is to keep the robot's heading parallel to the following elements. As shown in **Fig. 7**, right-angled triangles with the hypotenuse side a (distance from $30°$) and the other two sides b, c (distance from $0°$, and the following elements) are established in real-time. The HA algorithm requires the distance from $0°$ (side b) and $30°$ (side a) maintain a ration of $\sqrt{3}/2$ ($\cos 30°$), which keeps the robot's heading parallel to the following elements according to the Pythagorean theorem (Agarwall 2020). To avoid noise, a range of -0.07 to +0.09 is adopted.

In summary, if the ratio remains between [0.80, 0.97], the robot could follow a straight path by keeping its heading parallel to the following elements. The main concept of the proposed HA algorithm is presented below:

Algorithm 1 Heading adjust algorithm

Result: Heading (*H*)
Input: Distance from 0° *D0*, distance from 30° *D30*, right distance *Dr*
Initialization
Randomly initialize *D0, D30, Dr*
1: *dc*: 0.28, *df*: 0.33 ← lower and upper limits of right distance
2: **if** $dr \leqslant dc$ **then**

H is Turn left
if $dc \leqslant dr \leqslant df$ **then**
  if $0.80 \leqslant D0/D30 \leqslant 0.95$ **then**
    H **is** Go straight
  else
    H **is** Adjust heading slowly
  end if
if $dr \geqslant df$ **then**
  H **is** Turn right
end if
return H

3.2.6 Behavior distinguish algorithm

Another problem for the designed FLC is that the path deviation problem may happen during the "obstacle avoidance" behavior. Because of the designed fuzzy rule B, when the robot reached the end of an obstacle's side, the "finding wall" behavior was triggered because the left, front, and right distances all belong to "VF" instead of "turning" and "following" the obstacle. Therefore, a behavior distinguish (BD) algorithm was developed to improve the FLC system by preventing path deviation.

The behind-right distance from [-135°-0°], and the behind-left distance from [180°-225°] shown in **Fig. 8** were used to achieve this. When the left, front, and right distances are "VF", the robot is required to first consider the behind-left/right distance. If the behind left/right distance is within 0.36m, the robot is expected to turn right/left slowly for a short distance to keep following the obstacle and return to the initial path. On the other hand, the robot is expected to speed up to find new walls. The main concept of the proposed BD algorithm is shown as follows:

Algorithm 2 Behavior distinguish algorithm

Result: Moving state (M)
Input: Distance from 180° to 225° br, distance from -135° to 0° bl, left distance from 160° to 180° l, front distance from 60° to 120° f, right distance from 0° to 20° r.
Initialization
Randomly initialize br, bl, l, f, r
VF: very far, A: appropriate ← distance level
if l, f **is** VF **and** r **is** A **then**
  M **is** Following the obstacle
if l, f, r **is** VF **then**
  if br **or** bl <= 0.36 **then**
    M **is** Following the obstacle
  elif br **or** bl **is** VF **then**
    M **is** finding new elements
  end if
end if
return M

4. Results

4.1 Simulation in ROS

The feasibility of the designed FLC system was first validated in ROS simulation. The eight launching scenarios, typical and curved, square-shaped building layouts, are included in the simulation environments.

4.1.1 Performance in various building scenarios

(1) Performance of “finding wall”, “turning”, “wall following” behaviors in eight individual building scenarios

The navigation paths, crisp values of distance, and linear and angular velocities of robot navigation in eight individual scenarios are shown in **Fig. 9**.

It can be observed that in scenario A and D, the robot was located at the left corner with different towards. The robot properly recognized the position by computing the fuzzy degree of left, front, and right distance. In scenario A, the FLC system first output “VF”: 0.51m~ 3.5m for the front distance. “N”: 0.22m~0.27m, “A”: 0.29m~0.32m, and “F”: 0.32m~0.34m for the left distance, and “VF”:0.34m~3.5m for the right distance. In that case, the FLC system sent the angular velocity as “TRN”: -0.3rad/s and the linear velocity as “Z”: 0 m/s to control the robot to first slowly turn right in place. When the robot properly turned its direction, the FLC system output “VF”: 0.38m~3.5m for both front and left distance. “F”: 0.31m~0.33m and “A”: 0.29m~0.31m for the right distance. In that case, the FLC system output the angular velocity as “GS”: 0rad/s, and linear velocity as “M”: 0.1m/s to control the robot to follow the wall at a normal speed.

In scenario D, the FLC system output “N”: 0.28m~0.32m, “A”: 0.28m~0.31m, “F”: 0.31m~0.34m first for the front distance. “A”: 0.28m~0.32m, “F”:0.32m~0.34m for the left distance. “VF”: 0.39m~0.66m for the right distance. In that case, the FLC system output the same velocity command as in scenario A to control the robot to first turn its heading in place. When the front and left distance changed to “VF”: 0.34~3.5m, right distance changed to “N”: 0.15m~28m and “A”: 0.30m~0.32m, the FLC system output “GS” and “M” to command the robot to follow the wall within 0.30~0.33m.

Scenario B refers to the “finding wall” behavior. In that case, the robot was located far away from the inspection elements. The FLC system sent both the left, front, and right distance as “VF”: 0.36m~3.5m. In that case, the robot was expected to turn right quickly over a long distance to find the wall as soon as possible. To achieve this, the FLC system sent the angular velocity as “TRF”: -0.1rad/s, linear velocity as “H”: 0.25m/s. When the robot found the wall, the FLC system sent the angular velocity as “TLN”: 0.3rad/s, and linear velocity as “L” and “M”: 0.05m/s~0.1m/s to control the robot to adjust its position and follow the wall.

Scenario C refers to the “wall following” behavior. In that case, the FLC system sent the fuzzy degree of the front and left distance as “VF”: 0.58m~3.5m. “N”: 0.25m~0.28m, “F”: 0.31m~0.33m for the right distance. To conduct inspection work within a desired distance, the FLC system sent the angular velocity as “TLN” and “TRN”: -0.3rad/s~0.3rad/s and linear velocity as “L”: 0.05m/s to control the robot to adjust position by turning to the left and right slowly. When the right distance was changed and kept to “A”: 0.31m~0.32m and the robot’s heading was parallel to the wall, the angular velocity was turned to “GS”: 0rad/s and linear velocity to “M”: 0.1 m/s to control the robot to follow the wall straight at normal speed.

In scenario E, the robot was located in the right corner. The FLC system sent “VF”: 0.59m~3.5m for the left distance. “N”: 0.21m~0.27m for the front distance. “A”: 0.28m~0.31m for the right distance. In that case, the FLC system sent the angular velocity as “TLN”: 0.3 rad/s and linear velocity as “Z”:0 m/s to control the robot to first turn left in place to avoid collision with forehead walls. When the front and left distance changed to “F”: 0.32m~0.33m and “VF”: 0.36m~3.5m. The right distance changed to “N”: 0.25m~0.28m and “A”: 0.28m~0.32m, the “wall following” behavior was activated. The FLC system then output the angular and linear velocity in scenario C.

When the robot launched in scenario F, the FLC system sent the same velocity command as in scenario E. The variation of the left and front distance fuzzy degree in scenario F was similar to that in scenario E. Because there were no blocks on the robot’s right side in scenario F, the FLC sent the right distance as “VF”: 0.35m~0.5m first, then changed to “N”, “A” as in scenario E.

Scenario G and H usually represent the narrow spaces in buildings. In scenario G, the fuzzy degree of both the left and right distances was “N”:0.17m~0.27m and “A”:0.27m~0.28m, and the front distance was “VF”:0.37m~0.40m. In that case, the FLC system sent angular velocities as “GS”: 0rad/s, and “L”: 0.05m/s to control the robot to go straight slowly in narrow places. Similar navigation was conducted in scenario H. The only difference is that in scenario H, fuzzy degree of the front distance was “N”:0.23m~0.28m first, and then changed to “VF”: 0.36m~0.5m after the robot turned around. In that case, the robot was expected to first turn around slowly in place, with an angular velocity as “TLN”: 0.3 rad/s and linear velocity as “Z”: 0m/s. Then go straight slowly by changing the angular and linear velocity to “GS”: 0rad/s, and “L”: 0.05m/s. When the right or left distance was “F”: 0.31m~0.36m, the FLC system also output the angular velocity as -0.3rad/s~0.3 rad/s to control the robot to turn left or right slowly to keep following the wall within the desired distance.

The above-mentioned robot initial positions, expected behaviors, velocity commands, and changes of sensed distance in each scenario are briefly summarized in **Table 3**.

Table 3. Robot initial positions, expected behavior, velocity commands, changes of sensed distance in each scenario

Building scenario	Initial position	Expected behavior	Velocity command		Changes of sensed distance					
			Linear (m/s)	Angular (rad/s)	Left		Front		Right	
					Before(m)	After(m)	Before(m)	After(m)	Before(m)	After(m)
A	Left corner	Turn right in place	“Z”:0	“TRN”:-0.3	“N”:0.22~0.27 “A”:0.29~0.32 “F”:0.32~0.34	“VF”:0.38~3.5	“VF”:0.51~ 3.5	“VF” 0.38~3.5	“VF”: 0.34~3.5	“F”:0.31~0.33 “A”:0.29~0.31
B	Far away to the wall	Turn right quickly	“H”: 0.25	“TRF”:-0.1	“VF”:0.36~3.5	“VF” 0.38~3.5	“VF”:0.36~3.5	“VF” 0.38~3.5	“VF”:0.36~3.5	“A”: 0.29~0.31
C	Near to the wall	Slowly adjust and go straight	“L”: 0.05 Then “M”: 0.1	“TLN” and “TRN”: -0.3 ~0.3 Then “GS”:0	“VF”:0.58~3.5	“VF”:0.58~3.5	“VF”:0.58~3.5	“VF”:0.58~3.5	“F”:0.31~0.33 “N”:0.22~0.27	“A”: 0.31~0.32
D	Left corner	Turn right in place	“Z”:0	“TRN”:-0.3	“A”:0.28~0.32 “F”:0.32~0.34	“VF”: 0.34~3.5	“N”:0.28~0.3 “A”:0.28~0.31 “F”:0.31~0.34	“VF”:0.34~3.5	“VF”:0.39~0.66	“N”: 0.15~0.28 “A”:0.30~0.32
F	Right corner	Turn left in place	“Z”:0	“TLN”:0.3	“VF”:0.59~3.5	“F”:0.32~0.33 “VF”:0.36~3.5	“N”:0.21~0.27	“F”:0.32~0.33 “VF”:0.36~3.5	“A”:0.28~0.31	“N”:0.25~0.28 “A”: 0.28~0.32
F	Facing the wall	Turn left in place	“Z”:0	“TLN”:0.3	“VF”: 0.59~3.5	“F”:0.32~0.33 “VF”:0.36~3.5	“N”:0.21~0.27	“F”:0.32~0.33 “VF”:0.36~3.5	“VF”: 0.35~0.5	“N”:0.25~0.28 “A”: 0.28~0.32
G	In narrow places	Slowly go straight	“L”: 0.05	“GS”:0	“N”:0.17~0.27 “A”:0.27~0.28	“N”:0.17~0.27 “A”:0.27~0.28	“VF”:0.37~0.40	“VF”:0.36~0.5	“N”:0.17~0.27 “A”:0.27~0.28	“N”:0.17~0.27 “A”:0.27~0.28
H	In narrow places	Slowly turn around in place and adjust	“Z”:0	First 0.3, Then “GS”:0 and “TLN” and “TRN”: -0.3~0.3	“N”:0.17~0.27 “A”:0.27~0.28	“N”:0.17~0.27 “A”:0.27~0.28	“N”:0.23~0.28	“VF”:0.36~0.5	“N”:0.17~0.27 “A”:0.27~0.28	“N”:0.17~0.27 “A”:0.27~0.28

(2) Performance of “finding wall”, “turning”, “wall following” behaviors in integral building layers

As seen from **Fig. 10**, a typical building layer with four rooms, eight external walls, and four internal walls, as well as special building layers with curve-shaped and square-shaped walls, were established to see if the robot could achieve autonomous navigation in the integral unknown building layers without collisions. The lines in Fig.10(a) – Fig.10(c) present the travelling path.

As seen from **Fig. 10(a)**, the results revealed that the FLC system successfully controlled the robot to complete navigation in typical building layers. The navigation path followed a sequence of rooms A, D2, B, C, D1, and exterior walls, covering all interior and external walls. The distance fuzzy degree, linear, and angular velocities in scenarios B, F, C, and E were combined to accomplish this. Outputs in scenario B were first used to control the robot to find the wall at a fast speed. When the robot detected the forehead walls, the outputs from scenario F were utilized to command the robot to turn left and adjust its position. Outputs from scenario C were then used control the robot to maintain following walls within a certain distance. When the robot reached the left corner, the outputs from scenario E were utilized to command the robot to turn left first to avoid collision and then continue the “wall following” behavior.

As shown in **Fig. 10** (b) and (c), the FLC system can also control the robot to conduct inspection work in special-shaped building layers. In the same way, the outputs of the FLC system in scenarios B, F, C, and E were integrated. When the robot approached the corner of a curve or a square, the FLC system would sometimes report all of the left, front, and right distances as “VF.” Different from “finding wall” cases, the robot was still located near the inspection elements, and there was no need to turn right fast to find new walls. In that case, the proposed BD algorithm assisted the robot in turning right slowly for a short distance and continuing the same inspection path.

(3) Performance of “obstacles avoidance” behavior

As shown in **Fig. 11,** the robot successfully avoids both curved and square-shaped obstacles during the navigation process. The lines in **Fig. 11** present the travelling path. The FLC system output distance fuzzy degrees and crisp velocities in scenarios E or F and C, respectively, to control the robot to turn first and continue “wall following.” Similarly, when the robot reached the end of one side of the obstacles, the FLC system reported all of the left, front, and right distances as “VF.” Instead of “finding wall,” the robot was expected to keep following the barriers and return to the initial inspection path. In that case, the proposed BD algorithm also assisted in avoiding path deviations.

Specifically, when the robot moved to the end of one side of an obstacle, the distance from three directions was all rated as “VF,” as shown in **Fig. 12 (c)**. The “finding wall” behavior was then triggered, causing the path deviation.

Apart from reporting the left, front, and right distances as "VF," the FLC additionally reported the "right-behind" distance after applying the BD algorithm, as shown in **Fig.12 (d)**. According to the BD algorithm, the FLC system reported velocities as "TRN" and "L" instead of "TRF" and "H" when the "right-behind" distance was less than 0.36m. In that way, the FLC system controls the robot to keep the following behavior by constantly turning right slowly over a short distance rather than turning right fast over a large distance to find new walls. The green lines in **Fig. 11** present the travelling path.

4.1.2 Performance of designed HA and BD algorithm

It should be noted that the robot may turn too often during the "wall following" stage to maintain a certain distance, as seen in **Fig. 12 (a)**. The HA algorithm assists in providing the robot with a straight wall following path. Specifically, the FLC system sent the velocities "TRN" or "TLN" and "L" initially to command the robot's left or right turn. According to the HA algorithm, instead of continually adjusting orientations, the FLC system gives "GS" and "M" commands to tell the robot to go straight without turning at a normal speed when its heading is parallel to the following elements.

It's also worth noting that we attempted to have the FLC system output velocity as "GS" and "Z," to control the robot to turn straight while stationary. Although the robot's heading may be adjusted more precisely in that way, it tends to stop and move frequently, which also causes camera shake. Therefore, adjusting direction at a slow speed is a better option. After using the proposed HA algorithm to optimize the FLC system, it is clear that the robot could move in a relatively straight path in the "wall following" stage, as shown in **Fig. 12 (b)**. The lines in Fig. 12 present the travelling path.

4.2 On-site validation

On-site validation was conducted inside the Hong Kong Polytechnic University to validate the designed FLC in real-world environments.

4.2.1 Feasibility and efficiency in unknown environment

To validate the advantages of the local navigation strategy in unknown environments, discussed in the literature review section, the performance of the designed FLC and the SLAM algorithm (a global navigation strategy) was compared. **Fig. 13 (a)** shows the initial map prepared for SLAM navigation. As can be seen, by navigating using the initial map, the robot successfully reached the goal position (**Fig. 13 (b)**). However, when placing a box obstacle later, the robot was blocked (**Fig. 13 (d)**) because the environmental map was not updated in real-time and the SLAM algorithm failed to calculate new path (**Fig. 13 (c)**). Differently, when navigated using the designed FLC, the robot can successfully pass obstacles that are placed at any time and reach the goal position without collision (**Fig. 13 (e)**).

4.2.2 Feasibility and efficiency in various building scenarios

Special locations, including concave and convex regions, curve-shaped columns, and narrow aliases, were selected to highlight the robustness of the designed FLC.

As seen from **Fig.14**, it was validated that the designed FLC successfully controlled the robot to navigate in concave and convex regions without collision. As expected, the robot firstly conducted "wall following" behavior using the velocities in scenario C. Velocities in scenario E was then triggered to control the robot to turn left first and continue following the wall. When the robot moved to the convex region, velocities in scenario B were sent and the designed BD algorithm was activated to control the robot to adjust its heading without deviating from the path.

As seen from **Fig.15**, the FLC system was proven to be suitable for navigation in narrow regions and curve-shaped columns. When the robot was close to both the right and left walls, velocities in scenario G were delivered to control the robot to slowly move straight. When the robot reached the end of the narrow regions, velocities in scenario H were delivered to control the robot to turn around in place and then continue wall following. When the robot met curve-shaped columns, velocities in scenario C and B were sent alternately to control the robot's movement in a curving path. The BD algorithm helped to avoid path deviation.

As seen from **Fig. 16**, it is validated that the FLC system successfully controlled the robot to pass through the forehead obstacles. The robot started with the "wall following" behavior first. The designed HA algorithm helped to control the robot's movement straight and keep a desired distance. Similarly, when the robot met forehead obstacles, the navigation strategies in scenario E, B, and C were activated respectively to make the robot avoid the obstacles and keep following the wall.

In summary, the designed FLC system is validated as feasible for the wall-following navigation of building inspection robots in various unknown building environments. To achieve this, the basic "finding wall", "wall following," "turning," and "obstacle avoidance" behaviors can be realized, which are correlated to the findings in (BraunstingI et al., 1995, Nadour et al., 2019).

This study outperforms the existing algorithms from: 1) The designed FLC is suitable for various unknown launch situations. The robot can navigate properly in eight different launch scenarios, including narrow spaces and building corners. If the robot is launched far from the inspection elements, the designed "finding wall" behavior enables the robot to quickly locate the inspection region. 2) Optimized by the proposed HA algorithm, the designed FLC ensures a relatively straight wall-following trajectory and keeps the robot following within a desired distance. 3) Optimized by the proposed BD algorithm, the path deviation problem can be effectively avoided for complex environments, such as concave and convex regions, curved or square-shaped building elements.

5. Conclusions and Limitations

Employing robotics for automated building inspection is becoming a new trend. Navigation strategies are essential for autonomous movements of building inspection robots but still lack development. Although autonomous navigation strategies have been widely proposed in the robotics field, fewer of them are suitable for the building structural inspection behavior, which requires the inspectors to follow the building components within the desired distance and dynamically avoid obstacles. Therefore, this study aimed to explore an autonomous navigation algorithm for the following behavior. To achieve this, the objective was to design a novel FLC for the wall-following behavior. The FLC empowered wall-following algorithm are the local navigation strategy, which enables autonomous navigation in unknown environments. We focused on exploring autonomous navigation in unknown environments because 1) it is better suited to changing building environments; and 2) it is challenging to create maps of specialized building environments, such as aging nuclear power plants.

The designed FLC enables robots to conduct basic inspection behaviors: “finding wall”, “wall following”, “turning” and “obstacle avoidance” without referring to prepared maps. The designed FLC is robust, it ensures safe travel in various building environments. Inserted with the proposed optimization algorithms, the FLC provides straight following path within a desired distance. The path-deviation problem can be effectively addressed.

In the FLC system, the inputs are the left, front, and right distances within the designed interval ranges: [150°-180°], referring to the left distance, [60°-120°], referring to the front distance, [0°-30°], referring to the right distance. The outputs are the angular and linear velocity. Three fuzzy sets (distance, speed fuzzy, and rotation fuzzy set) and membership functions were established based on robot configuration, camera configuration, building designs, and building inspection criteria to transform crisp distance data to linguistic fuzzy degrees and crisp velocity data in the fuzzification and defuzzification process. 45 fuzzy rules are defined for the robot’s decision-making based on every possible sensing and launching situation. Two optimization algorithms were also proposed based on the Pythagorean theorem and the distances between the behind-right and left ranges.

The FLC system was validated in both simulation and real-world environments using the Turtlebot3 burger robot. It is validated that the designed FLC realizes the autonomous navigation of building inspection robots in unknown environments. It is feasible to control robots to conduct inspection work in eight different building scenarios, such as, building corners, narrow alias. By integrating the output velocities of the eight individual scenarios, the robot successfully navigated in integral typical, curved, and square-shaped building layers and avoided collision with forehead obstacles. The proposed HA and BD algorithms effectively assisted in generating straight wall following paths within a desired distance and avoiding path deviation in complex regions, such as, concave and convex regions.

However, there are still some limitations: 1) Because of the shortcomings of the distance laser, the FLC tends to generate wrong commands when the robot meets transparent and reflective building materials, such as glass walls, and metal doors. This problem needs to be solved because the FLC is sensitive to the input distance data and glass walls, or metal doors are widespread in modern buildings. Therefore, developing multi-sensor-based path planning algorithms for more accurate robot navigation, such as, integrating lasers, lidar, or cameras, can be considered a potential research topic for our future study. 2) The designed FLC can be directly coded in any wheeled mobile robots to conduct building inspection works in various buildings, such as residential and public buildings. Because the fuzzy sets, types, ranges, and limits of the membership functions, and the fuzzy rules were designed based on robot and camera configurations, building designs, and inspection criteria, the designs are feasible for most of the building inspection robots. It should be noted that although the ranges of distance membership functions are feasible for different cameras, the supplementary distances may be various for different platforms. The specific number depends on the distance between the laser and the camera. Another concern is that the designed FLC is not feasible to control the robot to go up stairs. The simplest way to solve this problem is to enhance the mechanism designs, such as installing crawlers on wheels to enable the wheeled mobile robot to go up or down stairs. In that case, the designed parameters can still work, but the lasers need to be installed higher to distinguish forehead obstacles and stairs.

Data Availability Statement

All data or code that support the findings of this study are available from the corresponding author upon reasonable request.

References

Building Surveyor with Project Management Skills Salary in Hong Kong | PayScale. Payscale.com. https://www.payscale.com/research/HK/Job=Building_Surveyor/Salary/549fc858/Project-Management. Published 2022. Accessed September 8, 2022.

Baba, T., Kameyama, Y., Kamegawa, T., & Gofuku, A. 2010. "A snake robot propelling inside of a pipe with helical rolling motion." In *Proceedings of SICE Annual Conference.*, 2319-2325. Taiwan: IEEE.

Selvarajan, A., Kumar, A., Sethu, D., & bin Ramlan, M. A. 2019. "Design and development of a snake-robot for pipeline inspection." In *2019 IEEE Student Conference on Research and Development.*, 237-242. Malaysia: IEEE.

Wakimoto, S., Nakajima, J., Takata, M., Kanda, T., & Suzumori, K. 2003. "A micro snake-like robot for small pipe inspection." In *Proceedings of 2003 International Symposium on MiIRomechatronics and Human Science*., 303-308. Japan: IEEE.

Faruq Howlader, M. O., & Sattar, T. P. 2015. "Novel adhesion mechanism and design parameters for concrete wall-climbing robot." In *SAI Intelligent Systems Conference (IntelliSys).*, 267-273. UK: IEEE.

Hillenbrand C, Schmidt D, Berns K. 2008. "CROMSCI: development of a climbing robot with negative pressure adhesion for inspections." *Industrial Robot: An International Journal*., 35(3), 228-237. http://doi.org/10.1108/01439910810868552.

Nguyen, S. T., & La, H. M. (2021). A climbing robot for steel bridge inspection. Journal of Intelligent & Robotic Systems, 102(4), 1-21. http://doi.org/10.1007/s10846-020-01266-1.

Yu S, Jang J, Han C. 2007. "Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel." *Autom Constr*., 16(3), 255-261. http://doi.org/10.1016/j.autcon.2006.05.003.

Chen, F. C., & Jahanshahi, M. R. 2017. "NB-CNN: Deep learning-based crack detection using convolutional neural network and Naïve Bayes data fusion." *IEEE Transactions on Industrial Electronics*., 65(5), 4392-4400. http://doi.org/10.1109/tie.2017.2764844.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. 2014. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv.1406.1078.* https://doi.org/10.48550/arXiv.1406.1078.

Chen, X., & Gupta, A. 2017. "An implementation of faster RCNN with study for region sampling." *arXiv preprint arXiv:1702.02138.* https://doi.org/10.48550/arXiv.1702.02138.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition.*, 779-788. USA: IEEE

Tang, X., & Yamada, H. 2011. "Tele-operation construction robot control system with virtual reality technology." *Procedia Engineering*., 15, 1071-1076. https://doi.org/10.1016/j.proeng.2011.08.198.

Ayache, N., & Faugeras, O. 1989. "Maintaining representations of the environment of a mobile robot." *IEEE transactions on Robotics and Automation*, 5(6), 804-819. https://doi.org/10.1109/70.88101.

Moravec, H., & Elfes, A. 1985. "High resolution maps from wide angle sonar." In *Proceedings. IEEE international conference on robotics and automation.*, Vol. 2, 116-121. USA: IEEE.

Hoskere, V., Narazaki, Y., Hoang, T., & Spencer Jr, B. 2018. "Vision-based structural inspection using multiscale deep convolutional neural networks." *arXiv preprint arXiv*, 1805.01055. https://doi.org/10.48550/arXiv.1805.01055.

Mantha, B. R., & Garcia de Soto, B. 2022. "Investigating the Fiducial Marker Network Characteristics for Autonomous Mobile Indoor Robot Navigation Using ROS and Gazebo." Journal of Construction Engineering and Management, 148(10), 04022115. https://doi.org/10.1061/(ASCE)CO.1943-7862.0002378.

Demiral, E., Karas, I. R., Karakaya, Y., & Kozlenko, M. 2021. "Design of indoor robot prototype guided by RFID based positioning and navigation system." International Society for Photogrammetry and Remote Sensing.

Naheem, K., Elsharkawy, A., Koo, D., Lee, Y., & Kim, M. 2022. "A UWB-Based Lighter-Than-Air Indoor Robot for User-Centered Interactive Applications." Sensors, 22(6), 2093. https://doi.org/10.3390/s22062093.

Lee, W. C., Lim, M. C., & Choi, H. L. 2021. "Extendable Navigation Network based Reinforcement Learning for Indoor Robot Exploration." In 2021 IEEE International Conference on Robotics and Automation (ICRA)., 11508-11514. China: IEEE.

Shi, W., & Samarabandu, J. 2006. "Corridor line detection for vision based indoor robot navigation." In 2006 Canadian Conference on Electrical and Computer Engineering., 1988-1991. Canada: IEEE.

Mantha, B. R., Menassa, C. C., & Kamat, V. R. 2018. Robotic data collection and simulation for evaluation of building retrofit performance. Automation in Construction, 92, 88-102. https://doi.org/10.1016/j.autcon.2018.03.026.

Bui, H. D., Nguyen, S., Billah, U. H., Le, C., Tavakkoli, A., & La, H. M. 2020. "Control framework for a hybrid-steel bridge inspection robot." In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).*, 2585-2591. USA: IEEE.

Özaslan, T., Loianno, G., Keller, J., Taylor, C. J., Kumar, V., Wozencraft, J. M., & Hood, T. 2017. "Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs." *IEEE Robotics and Automation Letters.*, 2(3), 1740-1747. https://doi.org/10.1109/lra.2017.2699790.

Kaiwart, A., Dubey, N. D., Naseer, F., Verma, A., & Pradhan, S. 2022. "Design of Adaptive Wheel Driven Pipeline Inspection Robot." In *Advances in Mechanical Engineering and Technology.*, 583-595. Singapore: Springer

Durrant-Whyte, H., & Bailey, T. 2006. "Simultaneous localization and mapping: part I." *IEEE robotics & automation magazine.*, 13(2), 99-110. https://doi.org/10.1109/MRA.2006.1638022.

Asadi, K., Haritsa, V. R., Han, K., & Ore, J. P. 2021. "Automated object manipulation using vision-based mobile robotic system for construction applications." *Journal of Computing in Civil Engineering.*, 35(1), 04020058. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000946.

McLaughlin, E., Charron, N., & Narasimhan, S. 2020. “Automated defect quantification in concrete bridges using robotics and deep learning.” *Journal of Computing in Civil Engineering.*, 34(5), 04020029. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000915.

Gul, F., Rahiman, W., & Nazli Alhady, S. S. 2019. A comprehensive study for robot navigation techniques. *Cogent Engineering.*, 6(1), 1632046. https://doi.org/10.1080/23311916.2019.1632046.

Che, X., Zhang, Z., Sun, Y., Li, Y., & Li, C. 2022. “A Wall-Following Navigation Method for Autonomous Driving Based on Lidar in Tunnel Scenes.” In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*., 594-598. China: IEEE.

Wu, P., Fang, M., & Ding, Z. 2021. “Wall-Following Navigation for Mobile Robot Based on Random Forest and Genetic Algorithm.” In *International Conference on Intelligent Computing*., 122-131. Cham: Springer.

Wei, X., Dong, E., Liu, C., Han, G., & Yang, J. 2017. A wall-following algorithm based on dynamic virtual walls for mobile robots navigation. In *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*., 46-51. Japan: IEEE.

Altman, N. S. 1992. “An introduction to kernel and nearest-neighbor nonparametric regression.” *The American* Statistician., 46(3), 175-185. https://doi.org/10.2307/2685209.

Katoch, S., Chauhan, S. S., & Kumar, V. 2021. “A review on genetic algorithm: past, present, and future.” *Multimedia Tools and Applications.*, 80(5), 8091-8126. https://doi.org/10.1007/s11042-020-10139-6.

Poli, R., Kennedy, J., & Blackwell, T. 2007. “Particle swarm optimization.” *Swarm intelligence*, 1(1), 33-57. https://doi.org/10.1007/s11721-007-0002-0.

Saman, A. B. S., & Abdramane, I. 2013. “Solving a reconfigurable maze using hybrid wall follower algorithm.” *International Journal of Computer Applications*, 82(3), 22-26. https://doi.org/ 10.5120/14097-2114.

Xue S, Jiang R, Wong S, Feliciani C, Shi X, Jia B. 2020. “Wall-following behaviour during evacuation under limited visibility: experiment and modelling.” *Transportmetrica A: transport science*., 16.3, 626-653. https://doi.org/ 10.1080/23249935.2020.1722281.

Hammad, Issam, Kamal El-Sankary, and Jason Gu. 2019. “A comparative study on machine learning algorithms for the control of a wall following robot.” *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. China: IEEE.

Teng T, Veerajagadheswar P, Ramalingam B, Yin J, Elara Mohan R, Gómez B. 2020. “Vision based wall following framework: a case study with hsr robot for cleaning application.” *Sensors*., 20(11), 3298. https://doi.org/10.3390/s20113298.

Novák, V., Perfilieva, I., & Mockor, J. 2012. “Mathematical principles of fuzzy logic”. Vol. 517. *Springer Science & Business Media*.

Omrane, H., Masmoudi, M. S., & Masmoudi, M. 2016. “Fuzzy logic based control for autonomous mobile robot navigation.” *Computational intelligence and neuroscience.*, 1-10. https://doi.org/10.1155/2016/9548482.

Fatmi, A., Al Yahmadi, A., Khriji, L., & Masmoudi, N. 2006. “A fuzzy logic based navigation of a mobile robot.” In *Transactions on engineering, computing and technology*.

Malhotra, R., & Sarkar, A. 2005. “Development of a fuzzy logic based mobile robot for dynamic obstacle avoidance and goal acquisition in an unstructured environment.” In *Proceedings, 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics.*, 1198-1203. CA: IEEE.

Faisal, M., Hedjar, R., Al Sulaiman, M., & Al-Mutib, K. 2013. “Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment.” *International Journal of Advanced Robotic Systems.*, 10(1), 37. https://doi.org/10.5772/54427.

Suwoyo, H., Tian, Y., & Hajar, M. H. I. 2020. “Enhancing the Performance of the Wall-Following Robot Based on FLC-GA.” *Sinergi*., 24(2), 141-152. https://doi.org/10.22441/sinergi.2020.2.008.

Aouf, Awatef, Lotfi Boussaid, and Anis Sakly. 2019. “Same fuzzy logic controller for two-wheeled mobile robot navigation in strange environments.” *Journal of Robotics*., 2019, 1-11. https://doi.org/ 10.1155/2019/2465219.

Singh, N. H., & Thongam, K. 2018. “Mobile robot navigation using fuzzy logic in static environments.” *Procedia Computer Science*., 125, 11-17. https://doi.org/10.1016/j.procs.2017.12.004.

Muthugala, M. A., Samarakoon, S. M., Mohan Rayguru, M., Ramalingam, B., & Elara, M. R. 2020. “Wall-following behavior for a disinfection robot using type 1 and type 2 fuzzy logic systems.” *Sensors*, 20(16), 4445. https://doi.org/10.3390/s20164445.

Widnall, S., & Peraire, J. 2008. “Lecture L10-Angular Impulse and Momentum for a Particle.” *Dynamics*.

Cai, L., Liang, Z., Hou, Z. G., & Tan, M. 2008. “Fuzzy control of the inspection robot for obstacle-negotiation.” In *2008 IEEE International Conference on Networking, Sensing and Control*., 117-122. China: IEEE.

Schiffer, S., Ferrein, A., & Lakemeyer, G. 2012. “Reasoning with qualitative positional information for domestic domains in the situation calculus.” *Journal of Intelligent & Robotic Systems*., 66(1), 273-300. https://doi.org/10.1007/s10846-011-9606-0.

Cherroun, L., Nadour, M., & Kouzou, A. 2019. “Type-1 and Type-2 Fuzzy Logic Controllers for Autonomous Robotic Motion.” In *2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*., 1 (1-5). Turkey: IEEE.

Ali, O. A. M., Ali, A. Y., & Sumait, B. S. 2015. “Comparison between the effects of different types of membership functions on fuzzy logic controller performance.” *International Journal*., 76, 76-83.

Antonelli, G., Chiaverini, S., & Fusco, G. 2007. “A fuzzy-logic-based approach for mobile robot path tracking.” *IEEE transactions on fuzzy systems*., 15(2), 211-221. https://doi.org/10.1109/TFUZZ.2006.879998.

Witt, B. L., Wilbanks, J. J., Owens, B. C., & Rohe, D. P. 2022. “Stereophotogrammetry Camera Pose Optimization.” In *Rotating Machinery, Optical Methods & Scanning LDV Methods.*, Vol.6, 13-38). Switzerland: Springer, Cham.

Sagawa, R., Kurita, N., Echigo, T., & Yagi, Y. 2004. “Compound catadioptric stereo sensor for omnidirectional object detection.” In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*., Vol. 3, 2612-2617. Japan: IEEE.

Okazaki, S. Y., Tanaka, T., Kaneko, S. I., & Takauji, H. 2008. “Object recognition based on relationship between camera vibration and measurement error on stereo measurement.” In *Optomechatronic Technologies.*, Vol. 7266, 388-395. https://doi.org/10.1117/12.807265.

Ren, J., Ren, M., Liu, R., Sun, L., & Zhang, K. 2021. “An Effective Imaging System for 3D Detection of Occluded Objects.” In *The 4th International Conference on Image and Graphics Processing.*, 20-30. NY: ACM.

Chan, D. W., Hung, H. T., Chan, A. P., & Lo, T. K. 2014. “Overview of the development and implementation of the mandatory building inspection scheme (MBIS) in Hong Kong.” *Built environment project and asset management.*, Vol.4, 71-89, https://doi.org/10.1108/BEPAM-07-2012-0040.

Brown, M. M., Brown, G. C., Sharma, S., Busbee, B., & Brown, H. 2001. “Quality of life associated with unilateral and bilateral good vision.” *Ophthalmology.*, 108(4), 643-647. https://doi.org/10.1016/S0161-6420(00)00635-7.

MacInnes, I., & Smith, S. 2010. “A simple demonstration for estimating the persistence of vision.” *The Physics Teacher.*, 48(6), 394-394. https://doi.org/10.1119/1.3479718.

Skorka, O., & Joseph, D. 2011. “Toward a digital camera to rival the human eye.” *Journal of Electronic Imaging*, 20(3), 033009. https://doi.org/10.1117/1.3611015.

Patti, A. J., Sezan, M. I., & Tekalp, A. M. 1994. “High-resolution image reconstruction from a low-resolution image sequence in the presence of time-varying motion blur.” In *Proceedings of 1st International Conference on Image Processing.*, Vol. 1, 343-347. TX: IEEE.

Seraji, H., & Howard, A. 2002. “Behavior-based robot navigation on challenging terrain: A fuzzy logic approach.” *IEEE Transactions on Robotics and Automation.*, 18(3), 308-321. https://doi.org/10.1109/TRA.2002.1019461

Hagras, H. 2004. “A type-2 fuzzy logic controller for autonomous mobile robots.” In *IEEE International Conference on Fuzzy Systems.*, Vol. 2, 965-970. Hungary: IEEE.

Lee, Y. T., Chiu, C. S., & Kuo, I. T. 2017. “Fuzzy wall-following control of a wheelchair.” In *2017 Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS).*, 1-6. Japan: IEEE.

Lead C. Standard Room Sizes & Their Location In Residential Building |Standard Room Size In a House - Civil Lead. Civil Lead. https://www.civillead.com/standard-room-sizes/. Published 2022. Accessed September 14, 2022.

McGee, V. 1985. “A counterexample to modus ponens.” *The Journal of Philosophy*, 82(9), 462-471. https://doi.org/10.2307/2026276.

Gupta, M. M., & Qi, J. 1991. “Theory of T-norms and fuzzy inference methods.” *Fuzzy sets and systems.*, 40(3), 431-450. https://doi.org/10.1016/0165-0114(91)90171-L.

Dias, L. A., de Oliveira Silva, R. W., da Silva Emanuel, P. C., & Bento, R. T. 2018. “Application of the fuzzy logic for the development of automnomous robot with obstacles deviation.” *International Journal of Control, Automation and Systems*., 16(2), 823-833. https://doi.org/10.1007/s12555-017-0055-9.

Rahbar, M., Mahdavinejad, M., Markazi, A. H., & Bemanian, M. 2022. “Architectural layout design through deep learning and agent-based modeling: A hybrid approach.” *Journal of Building Engineering*., 47, 103822. https://doi.org/10.1016/j.jobe.2021.103822.

Hellmann, M. 2001. *Fuzzy logic introduction*. Rennes: Université de Rennes.

Chakraverty, S., Sahoo, D. M., & Mahato, N. R. 2019. “Defuzzification.” In *Concepts of Soft Computing*., 117-127. Singapore: Springer.

Mittler, R. 2017. “ROS are good.” *Trends in plant science.*, 22(1), 11-19. https://doi.org/10.1016/j.tplants.2016.08.002

Agarwal, R. P. 2020. “Pythagorean theorem before and after Pythagoras.” *Adv. Stud. Contemp.*, Math, 30, 357-389. http://dx.doi.org/10.17777/ascm2020.30.3.357.

Braunstingl, R., Sanz, P., & Ezkerra, J. M. 1995. “Fuzzy logic wall following of a mobile robot based on the concept of general perception.” In *Procs. of the 7th Int. Conf. on Advanced Robotics (ICAR'95)*., 367-376. Spain: IEEE.

Nadour, M., Boumehraz, M., Cherroun, L., & Puig Cayuela, V. 2019. “Hybrid type-2 fuzzy logic obstacle avoidance system based on horn-schunck method.” *Electrotehnica, Electronica, Automatica.*, 67(3), 45-51. http://hdl.handle.net/2117/178652.

List of figures:

Fig. 9. (e) Navigation in scenario E

Fig. 9. (f) Navigation in scenario F

Fig. 9. (g) Navigation in scenario G

Fig. 9. (h) Navigation in scenario H

Fig. 10. Navigation in integral building layouts

Fig. 10. (a) Navigation path (a)

Fig. 10. (b) Navigation path (b)

Fig. 10. (c) Navigation path (c)

Fig. 11. Obstacle avoidance

Fig. 12. Navigation path with and without HA and BD algorithm

Fig. 12. (a) Path without HA algorithm (b)

Fig. 12. (b) Path with HA algorithm (b)

Fig. 12. (c) Path without BD algorithm (c)

Fig. 12. (d) Path with BD algorithm (d)

Fig. 13. Comparison of SLAM and the designed FLC

Fig. 13. (a) (a)

Fig. 13. (b) (b)

Fig. 13. (c) (c)

Fig. 13. (d) (d)

Fig. 13. (e) (e)

Fig. 14. Navigation in concave and convex region

Fig. 15. Navigation in narrow regions and curved columns

Fig. 16. Passing forehead obstacles