Design and Analysis of a Welding Inspection Robot

pengyu zhang
Shanghai University of Engineering Science
Feng Zhang
Shanghai University of Engineering Science
Pei-Quan Xu (pqxu@sues.edu.cn)
Shanghai University of Engineering Science
Lei-Jun Li
University of Alberta

Article

Keywords: weld inspection, autonomous navigation, path planning, TEB algorithm

Posted Date: June 22nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1755435/v1

Design and Analysis of a Welding Inspection Robot

Peng-Yu Zhang[1], Feng Zhang[1], Pei-Quan Xu[1, *], Lei-Jun Li[2, **]

[1] College of Materials Engineering, Shanghai University of Engineering Science, Shanghai 201620, China.

[2] Department of Chemical & Materials Engineering, University of Alberta, T6G 2V4, Edmonton, Canada.

* Corresponding author: Tel: +86 (21) 6779-1204. E-mail: pqxu@sues.edu.cn.

** Corresponding author: Tel: +1 (780) 492-3472. E-mail: leijun@ualberta.ca.

Abstract: Periodic inspection of weld seam quality is an important step in assessing equipment safety. But this task requires personnel to perform inspection rounds. To save labor costs and improve efficiency, autonomous navigation and autonomous weld inspection robot are developed. The development process involves the design of chassis damping, target detection mechanism, and control system and algorithms. In particular, when performing weld inspection in complex outdoor environments, the robot is required to avoid any obstacles. The problem of planning the inspection route is solved by improving a timed elastic band (TEB) algorithm. The robot is capable of conducting inspection tasks in complex and dangerous environments efficiently.

Keywords: weld inspection, autonomous navigation, path planning, TEB algorithm.

1. Introduction

With the rapid development of robotics, robots can help people to perform simple or complex tasks in dangerous environments that are beyond people's reach. Today, the use of robots as work aids has become increasingly common in both industrial and consumer spaces. The benefits of robots are that they can reduce labor costs, save time, improve safety, and improve the quality of work [1]. Robots also play a significant role in weldings, such as spot welding, arc welding, and composite welding. Welding is widely used for metallic materials in the industry, from vessels and pipelines to bridges and railways[2]. Good weld quality can ensure the strength and toughness of the connection between metal parts. During service, welds often will deteriorate, including corrosion and cracking, to result in fracture failures. The most serious are welds in extreme conditions, where weld damage caused by corrosion leads to serious cases of cracking, leakage, or bursting of vessels. In large-scale infrastructures where the welds deteriorate over a large area, it is impractical and cumbersome to inspect the welds using human labor. The use of robots becomes necessary and feasible [3]. The design of weld inspection robots revolves around two questions: how can the robot accurately reach the place where the weld is located, and how can the robot accurately inspect the weld.

Different solutions have been proposed by researchers involved in studying inspection robotic systems to solve these essential questions. A robotic climber with multiple breathing chambers for inspection was designed for the inspection of concrete walls[4]. The propulsion system consisted of three omnidirectional drive wheels with great maneuverability. Combined with a vacuum system comprising seven controllable vacuum chambers and a large fluid reservoir operating system, the robot had pressure sensors and valves integrated for controls. Shang et al. [5] introduced a method that utilized neodymium permanent magnets for bonding, giving the robot payload carrying capacity. The arrangement of the magnets improved the ground clearance, enabling the robot to overcome obstacles. To be able to work on curved surfaces, a wheeled robot with two articulated segments was designed, which had the advantages of high speed and good maneuverability. A robot with magnetic wheels and vision sensors for defect

detection was designed. In addition to magnetic wheels fixed to metal surfaces to detect defects, the drive mechanisms of inspection robots included three forms: a track, a wheel, and a leg[6-7].

Inspection robots have recently evolved to become autonomous and semi-autonomous that are more efficient at saving inspection time and reducing labor costs. Nita [8] and other researchers have studied a semi-autonomous tracked inspection robot to detect defects in building ceilings. The developed inspection robots were equipped with wireless cameras and data processing functions, which could provide valuable information for the repair of the damaged structures. The inspection robot was able to assess the damage without the need for an engineer on site. Krenich [9] designed a six-axis robot that could move autonomously for inspection, while a human could also inspect around the weld seam with a camera carried by the robot. Bruzzone[10] introduced a mobile robot with a hybrid wheel and leg design, which featured wheels that could roll on flat grounds at a high speed, while the legs enabled the robot to avoid obstacles and climb hills[11].

The main damages to aged welds are corrosion and cracking. Detection algorithms have been developed for them. The use of advanced vision algorithms based on deep learning and machine learning made it possible to detect and recognize the defects. An algorithm based on an improved Gaussian mixture model for weld seam detection and classification was introduced in 12. It could classify the identified weld defects with high accuracy and in real-time. Li[13] et al. proposed a deep learning-based algorithm for weld seam image recognition, which accelerated the neural training on several thousand images of the welds. The disadvantage of this algorithm was that it was too computationally intensive and hardware demanding. In 14, Yang et al. proposed a method to improve the defect localization of U-net mesh to improve the automatic localization accuracy for weld detection. For low-cost robot development using low- to mid-cost main control boards, less computationally intensive algorithms need to be developed to achieve the identification and localization of weld defects.

This study design fabricates and tests a weld inspection robot shown in Figure 1.

The mechanical structure and overall frame of the automatic weld inspection robot are designed to have four-wheel independent suspension and individually controlled chassis to improve the flexibility of the robot's movement. A set of inspection information collection platforms is developed by using the robot operating system (ROS) to remotely collect defect information of the welds. The timed elastic band (TEB) algorithm, a control system for path planning of the robot, is improved for obstacle avoidance.

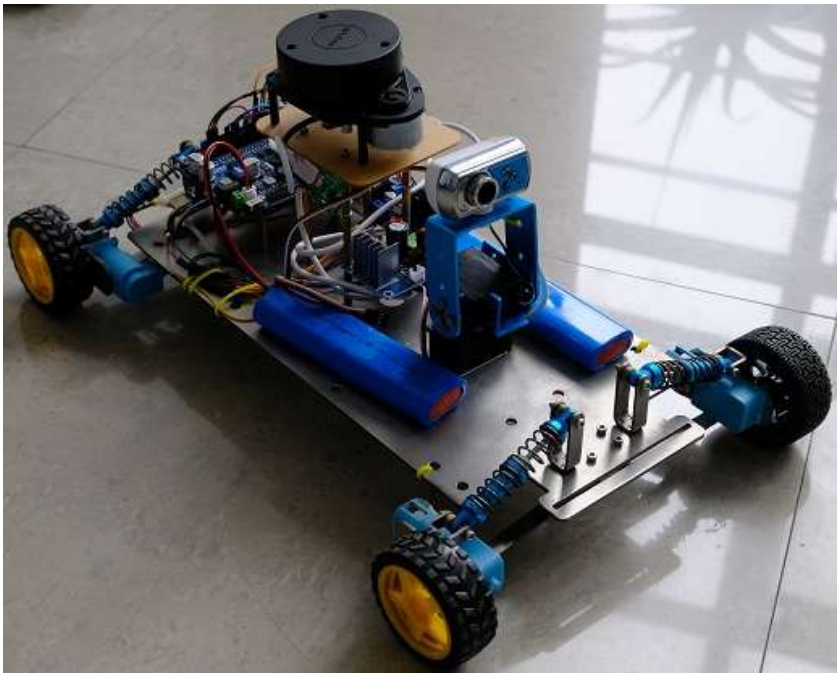


Figure 1. The overall configuration of the weld inspection robot

2. Robot Design

In this section, the design principles, motion dynamics, and primary control system of the inspection robot are described in detail.

2.1 Mechanical Design

2.1.1 Structural Design of Chassis

Figure 2 shows the CAD design of the vibration-damping chassis. Each motor is individually connected to the chassis through the motor mounting plate. The vibration dampers are placed between the motor mounting plates and the chassis. The chassis is

designed with many small holes for mounting other sensors and is made of aluminum alloy for strength and lightweight. Each wheel is controlled by a separate geared motor to achieve reasonable power distribution as well as flexible control.

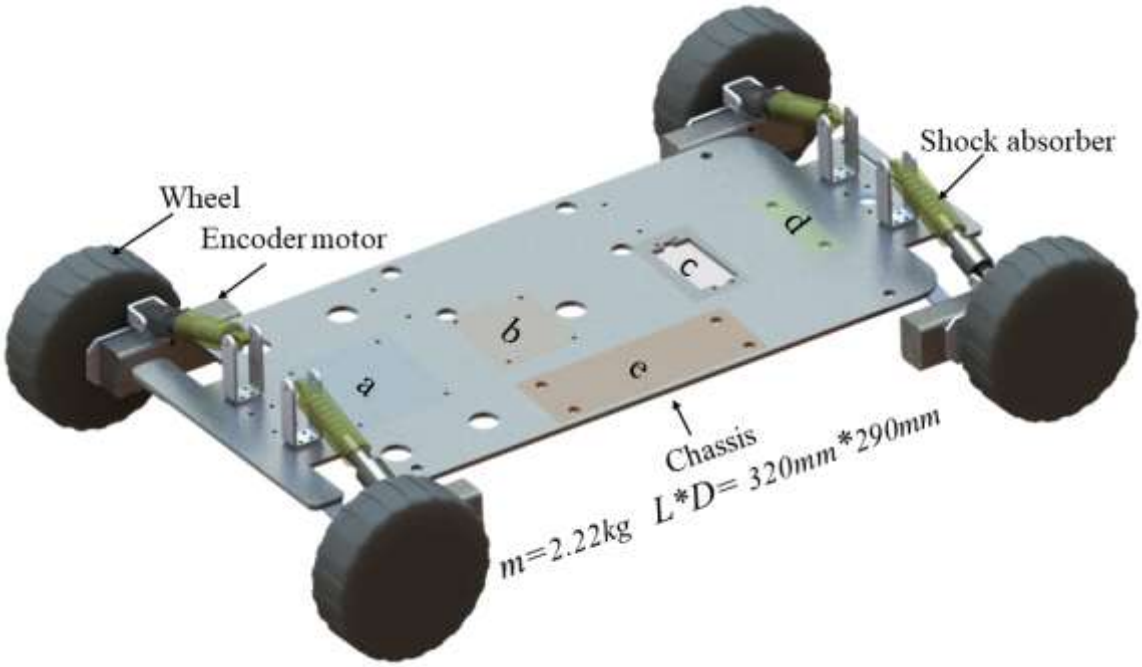


Figure 2. CAD design of the chassis, motors, and wheels.

The distribution of mounting holes in the chassis is shown in Figure 2. Area *a* is for the main control board. Area *b* is for the LIDAR. Area *c* is for the visual detection platform. Area *d* is for the video surveillance camera. Area *e* is for the power supply for the robot. The unmarked holes are used to install other auxiliary sensors, such as infrared sensors and motor drives.

2.1.2 Shock Absorber Design

The key part of the shock absorber is the spring. Figure 3 shows the force analysis of the spring, from which the spring is subjected to a partial force *Ft* of the chassis gravity, and the robot remains stable when the spring is not compressed. If the compressible range of the damper *s* is assumed as 15 *mm,* several equations can be derived from the force analysis of the spring mechanism.

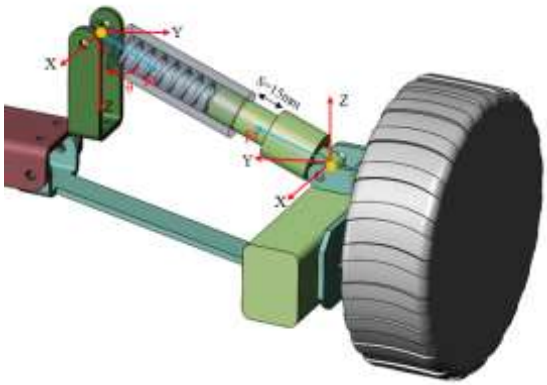


Figure 3. Force analysis of the vibration-damping structure.

To calculate the number of coils of the spring,

$$n = \frac{G \cdot d^4 \cdot s}{8 \cdot Ft \cdot D^3} \tag{1}$$

To determine the spring coefficient,

$$k = \frac{G \cdot d^4}{8 \cdot n \cdot D^3} \tag{2}$$

where b is wire width; d is wire diameter; D is spring coil center diameter; F is spring load; G is the shear modulus of elasticity; k is spring coefficient; n is an effective number of turns; s is spring deflection.

2.1.3 Simulation of shock absorbers

To verify the accuracy of the above shock absorber design parameters and to determine better design parameters, a model is built using SolidWorks motion simulation software and a simulation environment where the operation of the shock absorber could be dynamically observed. The spring setup parameters in the simulation environment are provided in Table 1.

Table 1 Spring setting parameters

scenes	spring coil diameter (mm)	number of laps	wire diameter (mm)	Load (N)
(a)	15	4	1.4	0.5
(b)	15	4	1.4	1.5

The robot is placed in the simulated environment as shown in Fig. 4. With the same set of obstacles and different applied loads, the linear displacement of the spring in the two states and the changes in the bearing forces are analyzed.

(a) (b)

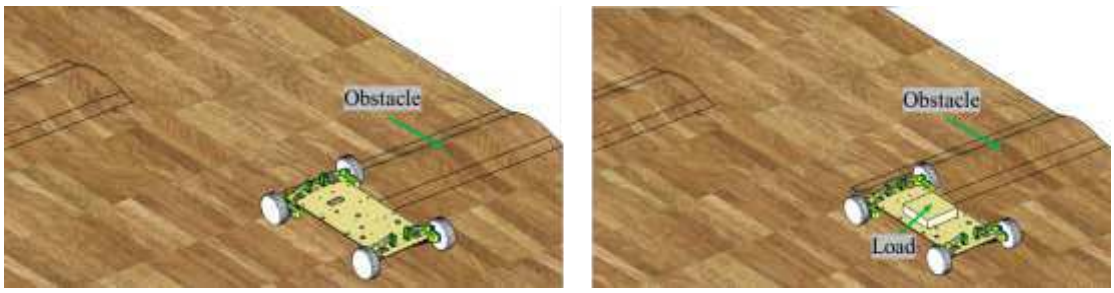
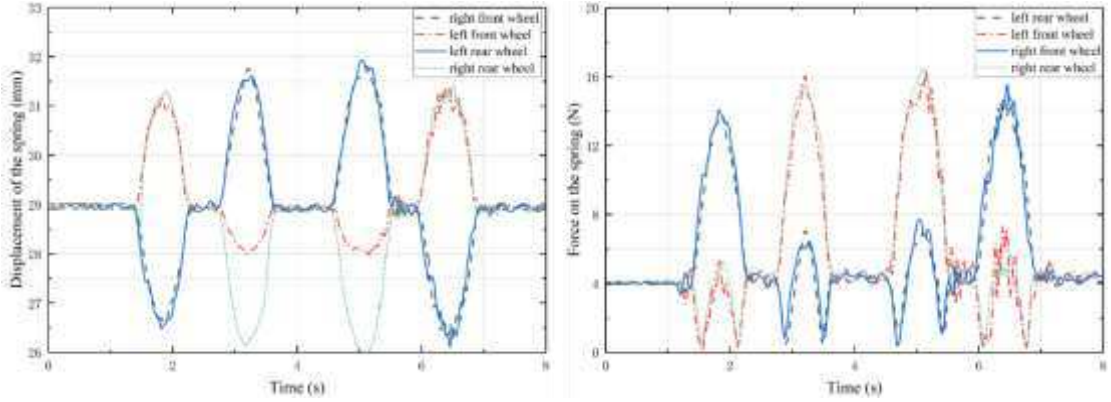


Figure 4. Simulation environment for the springs without load (a) and with load (b).

The simulation results are shown in Fig. 5. Without load, the spring displacement when the robot passes the obstacle is 2-3 mm (Fig. 5a). The maximum spring force on the spring is 16 N (Fig. 5b). When the robot is installed with the required sensors (i.e., with applied load), the spring can still meet the requirements of cushioning, and the displacement of the spring is kept between 0-8 mm, within the design range of 0-15 mm (Fig. 5c). With load, maximum the spring force is below 25 N (Fig. 5d). Since the position of the applied load is slightly closer to the front wheels, the initial load on the left front and right front wheels are larger than that of the rear wheels.

(a) displacement without load (b) spring force without payload



(c) displacement with load (d) spring force under the payload

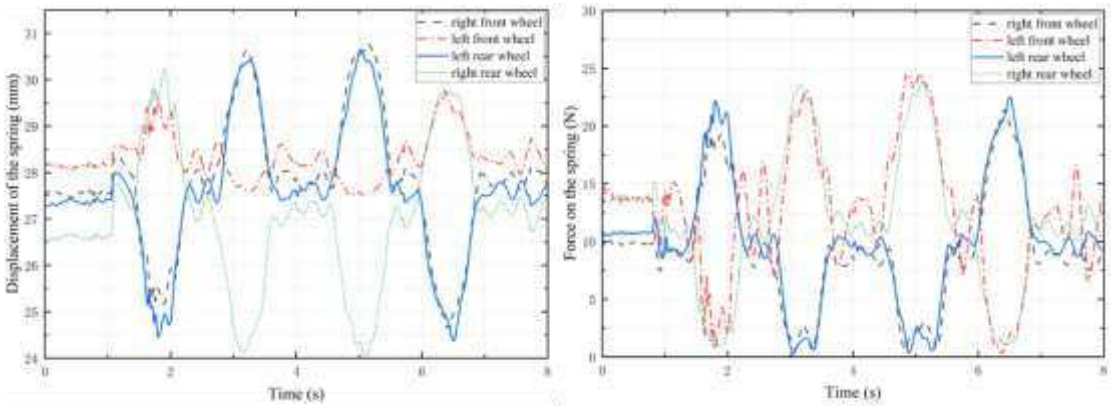


Figure 5. Displacement and spring force diagrams of springs under different chassis payloads.

2.1.4 The Overall Design of The Robot

The main control system is configured on the chassis, equipped with sensors for inspection functions. Fig. 6 shows the installation of each component of the robot. With an encoder and 1:90 gear ratio, a larger torque can be transferred to the wheels. High-accuracy motors can accurately feedback the speed and position information. Four-wheel drive enables fast turns and easier passage through complex roads.

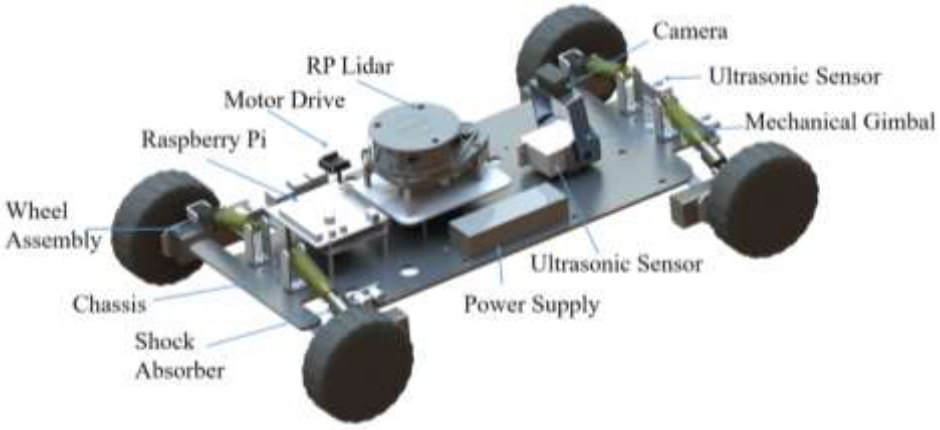


Figure 6. Mechanical components assembled on the chassis.

Figure 7 shows the control scheme of the inspection robot. Fig. 7(1) shows the actual connection of each sensing of the robot. Fig. 7(2) shows the main control board of the robot, a Raspberry Pi, equipped with a Linux-based ROS control system, which is the control system of the robot. Fig. 7(3) shows the motor driver board L298n

connected to the Raspberry Pi via pins. The camera in Fig. 7(5) and LIDAR in Fig. 7(6) are connected to the main control board through USB ports. The camera collects environmental information and feeds it back to the PC, as shown in Fig. 7(6). The LIDAR is A1M8, which gives the robot information for map building and navigation and allows the robot to sense obstacles in the surrounding environment.

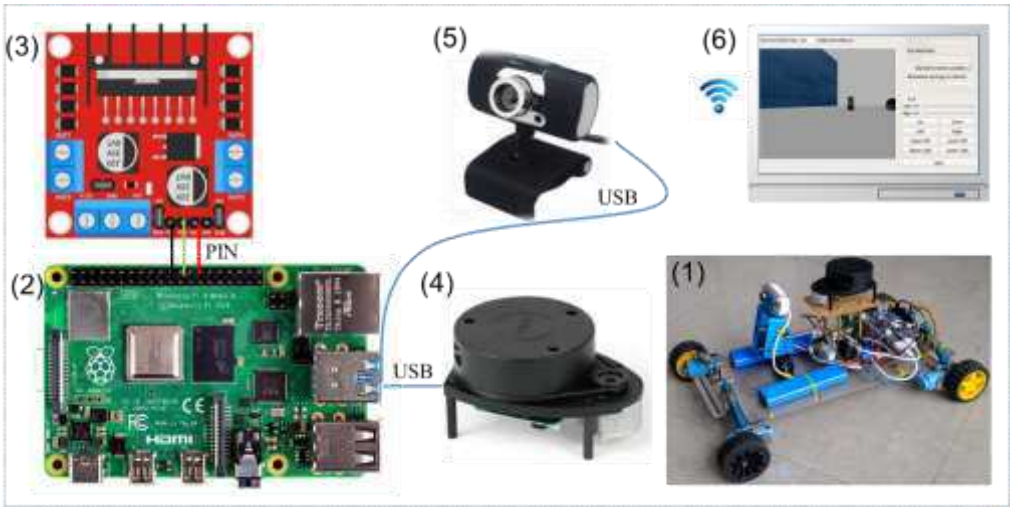


Figure 7. Mechatronics architecture of the control for the robot

2.1.5 Inspection system

For weld inspection, this robot carries a camera (which can also be replaced with other sensors as needed) as shown in Figure 8. The inspection system consists of a camera, two servos, two brackets, and several bolts, and adjusting the direction of the servos can achieve multi-directional detection and improve the area of detection. The information received by the detection system is transmitted to the PC, on which the detected information can be viewed and analyzed remotely.

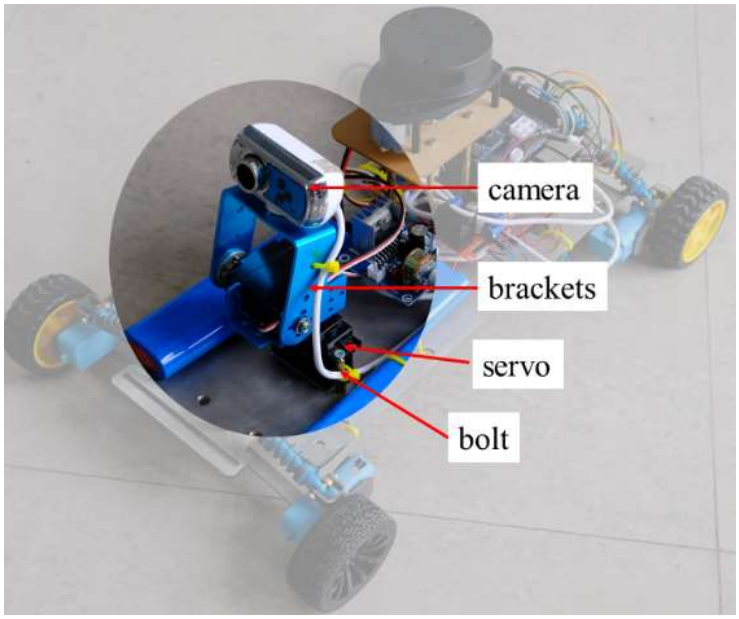
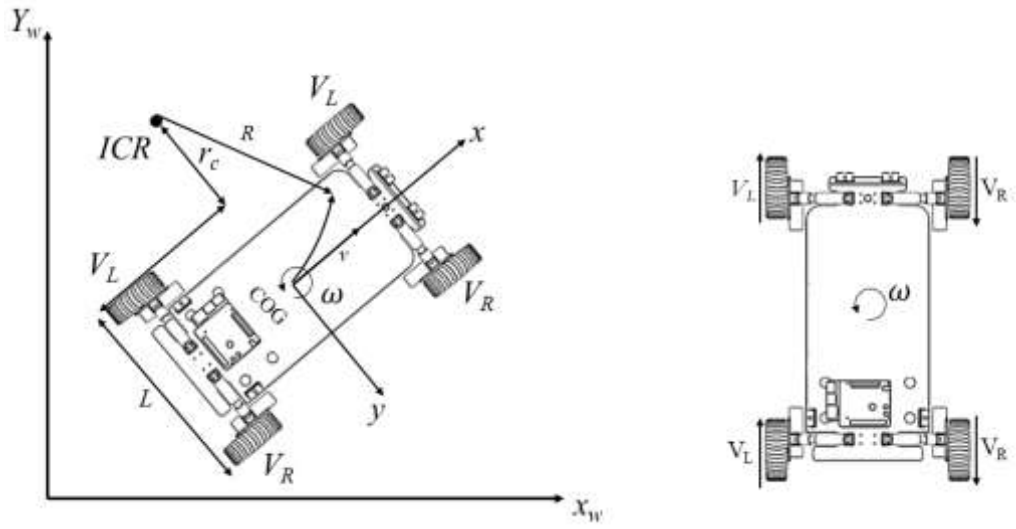


Figure 8. Detection device mounted on the chassis.

2.1.6 Kinematic Model of the Robot

Common models for chassis drive include the wheel differential model, Ackerman model, and omnidirectional model. The turning radius of the Ackerman model cannot be 0, which does not allow the robot to turn in narrow space. The omnidirectional model needs to use McNamee wheels instead of the common wheels. However, the gaps between each small wheel of the McNamee wheel are liable to be struck by foreign objects and affect the robot's movement. Therefore, the wheel differential model is selected, and optimized for the robot to turn in tight places and to reduce sliding friction.

To study the kinematics of the robot, we assume that a local coordinate frame, denoted as (x, y, z), is located at the center of gravity (COG) of the model. The motion of the robot is on the horizontal plane formed by the Xw and Yw axes of the world coordinate system, shown in Figure 9(a).



(a) (b)

Figure 9. (a) Motion model of the differential speed robot. (b) The steering model.

The instantaneous center of rotation of the robot is *ICR*, the linear velocity of the left wheel is V_L, the linear velocity of the right wheel is V_R, the angular velocity is ω, the distance between the two wheels is L, and the distance from the left wheel to the center of the circle is r_c.

The physical relationship between angular velocity, linear velocity (v), and radius of motion of the differential robot is as follows:

$$v = \omega \times R \tag{3}$$

The decomposition of the velocity of the left wheel and the right wheel can be found as:

$$\begin{cases} V_L = \omega \times \mathrm{r_c} = v - \dfrac{L}{2}\omega \\ V_\mathrm{R} = \omega \times r_c = v + \dfrac{L}{2}\omega \end{cases} \tag{4}$$

From this, the relationship between the overall linear velocity of the robot, the angular velocity, and the left and right wheels can be solved as follows：

$$\begin{cases} v = (V_L + V_R)/2 \\ \omega = (V_R - V_L)/L \end{cases} \tag{5}$$

As shown in Fig. 9(b), the left and right wheels of the robot designed are configured

in parallel, and robot turning is realized by the speed difference between the left wheel and the right wheel. The radius of curvature of the turn increases when the speed difference between the left and right wheels is larger. When the robot works in a narrow space, the robot turns around its middle vertical axis, i.e., the left and right wheels have the same speed but in opposite directions, and the turning radius is 0.

2.2. Control Architecture

2.2.1. Electronics and Control

The control architecture of the robot is shown in Fig. 10. The motion module is controlled by four DC motors with an encoder (gear ratio 1:90, size 25X25X80 mm), two motors are installed on each side of the chassis, and the motor on each side is connected to the L298n motor driver board through the Dupont cable. The motor driver board and the Raspberry Pi main control board are connected through the IO port. The Raspberry Pi subscribes to the data from the encoder through the IO port, and it processes the data and sends speed commands to control the speed of the motors. The sensing module of the robot consists of the LIDAR (Lidar A1M8), a camera head, an odometer, and an infrared sensor. The laser sensor, communicating with the Raspberry Pi through a USB port, is used for building a map of the surrounding environment, positioning, and avoiding obstacles. However, LIDAR has a blind scanning area, and there is a possibility that obstacles in the complex environment are not perceived, so the infrared sensor is used to supplement this shortcoming and improve the safety of the robot's operation. The odometer is used for robot positioning, and the LIDAR positioning is used to improve the positioning accuracy. The camera head is connected to the Raspberry Pi via a USB port for weld detection. The power supply module consists of two batteries (12V 5000 mAh; 12V 3000 mAh), the 5000 mAh battery supplies power to the motor driver board L298n. The 3000 mAh battery supplies power to the Raspberry Pi mainboard, and the rest of the sensors are powered through a USB or IO port.

The control of the motor does not use stm32 or the Arduino control method but directly connects to the motor driver board through the IO port on the Raspberry Pi,

which improves the convenience of operation and the sensitivity of control. The choice core controller of the robot is the Raspberry Pi 4B. The data processed on this robot is moderate, so the Raspberry Pi 4B is the best choice. The Raspberry Pi is installed with Ubuntu 18.04 Linux and the Robot Operating System (ROS), and the data collected by various sensors is directly transferred to the ROS system for processing. The various sensors, the main control board, the GUI, and the PC are integrated through the ROS framework. The robot works with the chassis drive, sensor, navigation, and map building nodes, and the collected data is transferred through WIFI to the PC for processing through the distributed framework of ROS to reduce the pressure on the Raspberry Pi to process the data.

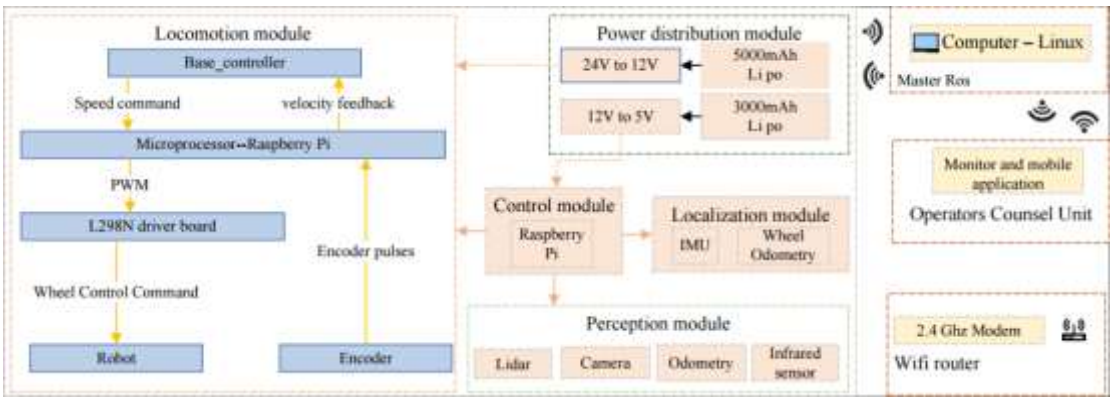


Figure 10. Schematic of the control architecture.

2.2.2. Graphical User Interface (GUI)

The robot and the PC side can be remotely monitored and controlled through ROS' distributed framework when they are connected to the same network. Communication between the robot and the control device is carried out through ROS' Message Queue Telemetry Transfer (MQTT). When the navigation control node is started on the mobile side, a message can be received on the PC side and a map building command can be executed on the PC side at the same time, controlling the robot to build a map and issue a point-to-point cruise operation. This operation reduces the computational pressure on the robot. A graphical user interface for monitoring and control is developed on the PC side using qt (spell out and provide a reference here). The robot can be viewed through

the GUI when it is working in an unknown environment and is matched with control buttons as well as video output as shown in Fig. 11.

Figure 11. Overview of the inspection system

3. Navigation and Control in Complex Environments

3.1. Selection of Local Path Planning

Autonomous navigation is the focus of this research, and robots require path planning algorithms that can find a collision-free path from the starting point to the target point in a complex environment and can avoid obstacles that appear in real-time. Path planning requires the cooperation of global path planning and local path planning. The weight of global path planning in the obstacle avoidance process is less than that of local path planning, therefore, in this paper, we mainly study the local path planning. The most important issue is that the robot must safely avoid static obstacles and dynamic obstacles in the inspection process. The local path planning methods in complex environments mainly include the artificial potential field method (APF) [15], the genetic algorithm[16], the dynamic window method (DWA) [17], neural network algorithm, and other intelligent algorithms. But the above algorithms have lower convergence speed or ability to avoid local extremes.

An improved TEB algorithm is selected to implement the local path planning. The TEB algorithm was proposed by Rösmann [18] and was developed based on the classical elastic band algorithm, which is an obstacle avoidance method by optimizing multi-objective trajectory optimization. Compared with the local path planning described above, the TEB algorithm can set multiple constraints as needed to ensure the applicability of the algorithm. The multi-objective optimization of the TEB algorithm

relies on only a few continuous states, thus optimizing for a sparse matrix model. Rösmann et al. proposed that the sparsity problem of the hypergraph-based TEB algorithm can be solved quickly and efficiently using the G2o framework to improve the computational speed. However, mobile robots equipped with the TEB algorithm can appear to be trapped in local minima and unable to cross obstacles in complex environment navigation. To solve this problem, Rösmann et al. 19 and 20 proposed an extension of the TEB technique by using parallel trajectory planning in a spatially unique topology. However, these approaches only consider the location of obstacles and do not consider potential collisions between the robot and surrounding obstacles. Lan et al. [21] proposed an active timed elastic band (PTEB) technique for autonomous mobile robot navigation systems in dynamic environments. Previous work to improve the effectiveness of TEB algorithms operating in complex environments has focused on obstacle avoidance. Moreover, most of the related research only pursued avoiding local minima and smoothing the planned paths in complex environments. It lacks the constraint of considering the shortest local path, and the planned local path may not be the optimal path[22]. Therefore, the improved TEB algorithm mentioned above still suffers from the robot backing up during turning, local detours, and the inability to enter narrow areas.

Based on the above analysis, the improved TEB algorithm proposed in this paper optimizes the behavior of local bypassing and reversing, adds the constraint of angular acceleration to the constraints of the multi-objective optimization problem, and considers the time consumption brought by the problem of excessive turning, and finally proves through experiments that the improved TEB experimental method can achieve fast turning and reduce the behavior of reversing, improve the detection range of the inspection robot, and reduce the time cost of the inspection.

3.2. Timed Elastic Band Algorithm (TEB) Model Construction

The TEB algorithm is based on the elastic band algorithm with the addition of temporal information between bit-pose sequences, as shown in equation (6), which considers the dynamic constraints of the robot and modifies the trajectory directly

instead of modifying the path. The operation principle of the TEB algorithm is to convert the position information of the searched initial path into the trajectory sequence with time information for the existing global path points, as shown in Fig. 12. The large-scale optimization algorithm of the sparse system in the "G2O framework" is solved to obtain the optimal control quantity that satisfies the constraints, and the robot drive system is directly commanded by calculating the control variables v and ω, as in equation (7).

$$\begin{aligned} Q &= \{X_i\}, i = 0,1,2...n \qquad n \in N \\ \tau &= \{T_i\}, i = 1,\ 2,\ \ldots,\ n\ -\ 1 \end{aligned} \tag{6}$$

$$B := (Q, \tau) = [X_1, \Delta T_1, X_2, \Delta T_2, \ldots, \Delta_{n-1}, X_n] \tag{7}$$

where X_i is the poses at the time I, and Q is the sequence of the poses; ΔT_i is the time interval between adjacent poses, and τ is the time interval sequence; the pose sequence and the time interval sequence are combined into a trajectory sequence B.

Because the objective function of the TEB algorithm depends on only a few continuous pose states, this leads to a sparse system matrix that represents these constraints as objectives according to a segmented continuous, differentiable cost function that penalizes the violation of the constraints that represent the boundaries, as in equation (8).

$$\lambda_\tau(x, x_r, \varepsilon, S, n) \approx \begin{cases} \left(\dfrac{x - (x_r - \varepsilon)}{S} \right)^n, & x > x_r - \varepsilon \\ 0, & x \le x_r - \varepsilon \end{cases} \tag{8}$$

where x_r is the critical value, S is the scaling factor, and n is the polynomial coefficient, which usually takes the value of 2; ε is a small section of displacement near the critical value.

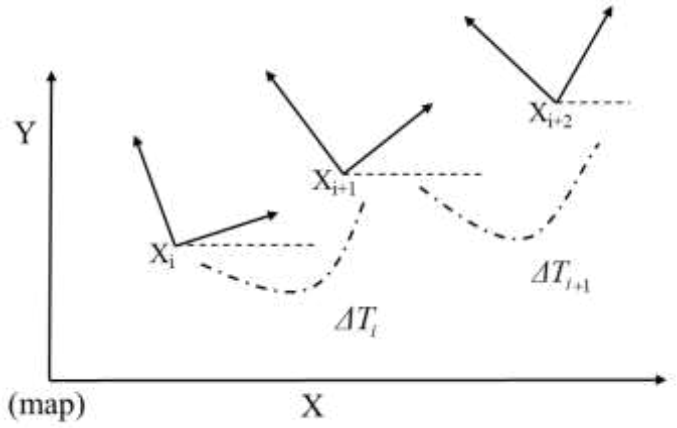


Figure 12. Pose and time interval of mobile robot in the world coordinate system.

The multi-objective optimization function is shown in equation (9).

$$f(B) = \sum_{k} \gamma_k f_k(B) \tag{9}$$

where $f_k(B)$ is a constraint function in Fig. 13, and γ_k is the weight corresponding to the constraint function.

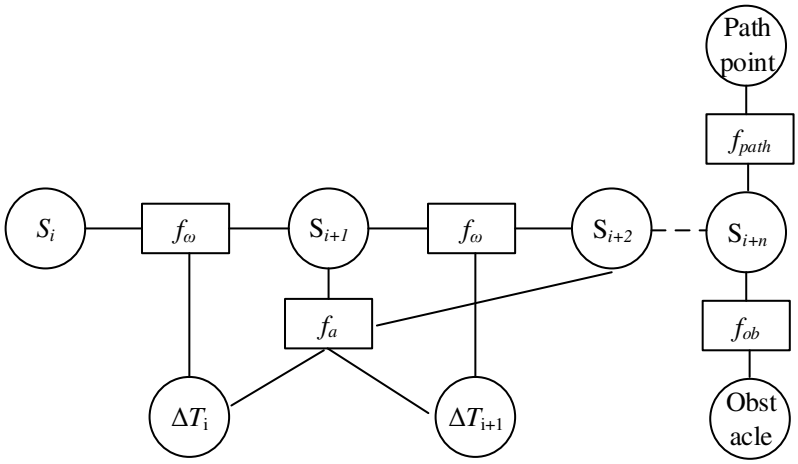


Figure 13. The improved hyper-graph.

The trajectory constraints of the TEB algorithm are divided into two parts. The first part is constrained by global path planning; the second part is constrained by velocity, acceleration, and its own kinematic model. In this paper, we focus on optimizing the velocity and acceleration constraints and the obstacle constraints.

The obstacle constraint is the most critical condition to ensure that the robot can avoid the obstacle completely. The minimum distance allowed between the robot and the obstacle is set to d_{min}, and the distance between the robot and the obstacle is set to D. The position information of the obstacle on the map is obtained by sensors such as LIDAR. To ensure the safety of the planned trajectory, each bit posed on the TEB trajectory is related to the obstacles appearing on the map, and the penalty function is triggered when the distance D between the robot and the obstacle is lower than d_{min}. The penalty function is expressed as equation (10):

$$f_{ob} = \sum_{i=0}^{n} \begin{cases} 0, & d_{min} > d_{imin} \\ d_{\min} - d_{imin}, & d_{min} \le d_{imin} \end{cases} \tag{10}$$

The velocity and acceleration constraints are described similarly to the geometrically constrained penalty functions. The linear and angular velocities are approximated by the Euclidean distance between adjacent poses and the amount of change in the directional angle, and can be expressed as (11).

$$\begin{cases} v_i \approx \dfrac{1}{\Delta T_i} \left\| \begin{pmatrix} x_{i+1} & -x_i \\ y_{i+1} & -y_i \end{pmatrix} \right. \\ \omega_i \approx \dfrac{\theta_{i+1} - \theta_i}{\Delta T_i} \end{cases} \tag{11}$$

The acceleration is related to two consecutive average velocities, so the average velocity corresponding to three consecutive poses needs to be acquired, and the equation can be expressed as (12).

$$\begin{cases} \alpha_{\mathrm{i}} = \dfrac{2(v_{i+1} - v_i)}{\Delta T_i + \Delta T_{i+1}} \\ \beta_i = \dfrac{2(\omega_i + 1 - \omega_i)}{\Delta T_i + \Delta T_{i+1}} \end{cases} \tag{12}$$

3.3. Constraints Based on Improved TEB Algorithm

The inspection robot will appear to avoid continuous obstacles and pass through narrow spaces in complex environments. The traditional TEB algorithm will appear to go around partially and get stuck in the process of avoiding obstacles, and the turning in narrow spaces will be limited. To improve the actual energy consumption, the control

algorithm of the turning angle speed of the TEB algorithm is considered next.

To realize the reversing and detour behavior of the robot in the process of avoiding obstacles, the control algorithm of the angular velocity is optimized. When the target point of the robot is given, the position point of the robot is set to $（x_i,y_j）$. The adjacent path points are $(x_i, y_i)、(x_{i+1},y_{i+1})$ the angle between the line connecting the two points and the robot's initial test pose θ_{i}. The robot will change during the moving process, so a minimum threshold θ_{imin} is set. When θ_{i} is greater than the minimum threshold, the angular velocity is set to the maximum. The robot accelerates turning to avoid the behavior of reversing, and as the θ_{i} becomes smaller, the angular velocity also decreases to achieve a smooth transition of the turn. The penalty function can be expressed as equation (10):

$$\omega=\begin{cases}\omega_{\max}, & \theta_{\mathrm{i}} \geq \theta_{imin} \\ \dfrac{\theta_i}{\theta_{i\min}} \times \omega_{max}, & \theta_{\mathrm{i}}<\theta_{imin}\end{cases} \tag{13}$$

The current angular velocity optimized control is used as an angular steering constraint and the angular velocity constrained edges are added to the hypermesh. A new hypergraph is thus constructed, as shown in Fig. 8. The optimized angular velocity constraint function is connected to two poses vertices S_i and S_{i+1} .The optimization problem is transformed into a hypergraph and solved using a large-scale algorithm for sparse systems in the G2O framework, and the robot is driven directly by computing control variables v and ω after verifying the optimized TEB algorithm.

4. Simulation Experiment and Real Robot Experiment

4.1. Simulation Experiment

The simulation experiments were conducted on the ROS platform, first buil ding the simulation environment in Gazebo and observing the motion of the ro bot equipped with the improved TEB algorithm using the Rviz visualization pl atform. The motion of the robot is modeled as a 4WD differential, with the le ft and right wheels controlled separately.

Table 1. Parameter configurations of the simulated experiment.

Constraint Parameters	Values
Maximum X linear velocity (m/s)	0.6
Maximum backward linear velocity (m/s)	0.4
Maximum angular velocity (rad/s)	0.5
Maximum X linear acceleration (m/s2)	0.5
Maximum angular acceleration (rad/s2)	0.5
Obstruction expansion radius (m)	0.6

4.2. Real Robot Experiment

The experimental platform of the automatic inspection robot has been built as shown in Fig. 14. This experimental platform is used to carry the improved TEB algorithm for controling the motion of the robot in real-time and test the feasibility of the algorithm in a multi-obstacle environment. It is to veriy whether the robot's reversal in avoiding obstacles and the robot's turning problem in a narrow environment is significantly improved.

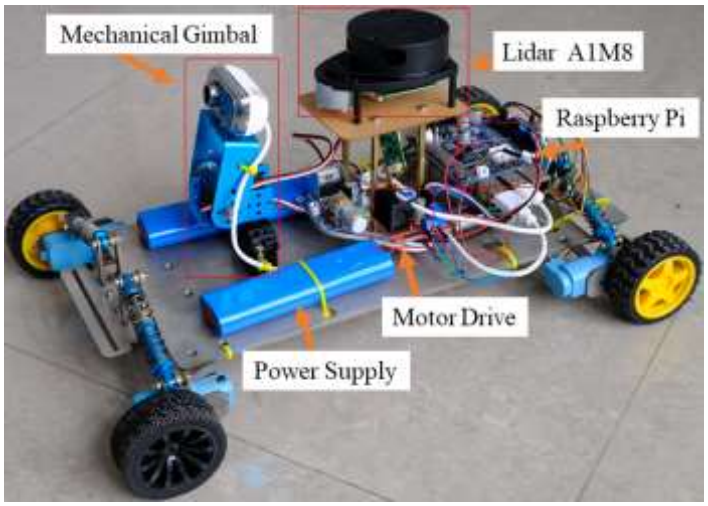


Figure 14. Robotic inspection platform

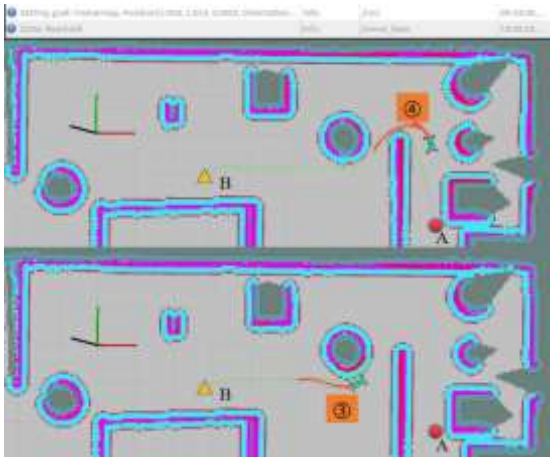
5. Results Analysis

5.1. Simulated Experiment Results Analysis

The motion of the inspection robot equipped with the conventional TEB algorithm and the improved TEB algorithm in a complex environment is shown in Fig. 15, which

shows the comparison of the planned path and the running time.

（a）The planning time after the improved TEB algorithm



b）The original planning time of the TEB algorithm

Global path planned by A* algorithm Local path planned by TEB algorithm. A start point. B end point.

Figure 15. Time comparison of TEB algorithm planning before and after improvement.

Fig. 15a shows the planning time for the improved TEB algorithm, where the robot turns at point ① and point ② closely to the global path to shorten the running time while ensuring the avoidance of obstacle. Fig. 15b shows the conventional TEB algorithm, where it is obvious that the robot goes around further at points ③ and ④, increasing the time cost for a given average velocity. From Table 2, it can be concluded that the running time of the improved algorithm is 49s while that of the traditional algorithm is 50.3s. The improved TEB algorithm shortens the running time by a small 3% in the process of ensuring smooth operation, which improves the efficiency, and at the same time, the robot moves close to the global path to avoid the energy loss caused

by excessive turning.

Table 2. Time comparison after improving TEB algorithm.

	Starting Point	Starting Time	End Time	Total Time Spent
Traditional TEB algorithm planning time	（11.1,1.07）	09:59:38.1	10:00:28.6	50.5s
Improved TEB algorithm planning time	（11.2,1.06）	10:14:08.7	10:14:56.8	49.1s

To verify the turning sensitivity of the inspection robot equipped with TEB algorithm and prevent reversing behavior in narrow spaces, Figure 16 shows the robot turning and reversing when encountering successive narrow road sections. The robot starts from point A through the narrow sections at points ①, ②, and ③ to reach point B. The speed profile generated at each stage is viewed to compare the reversing phenomenon before and after the improvement of the TEB algorithm.

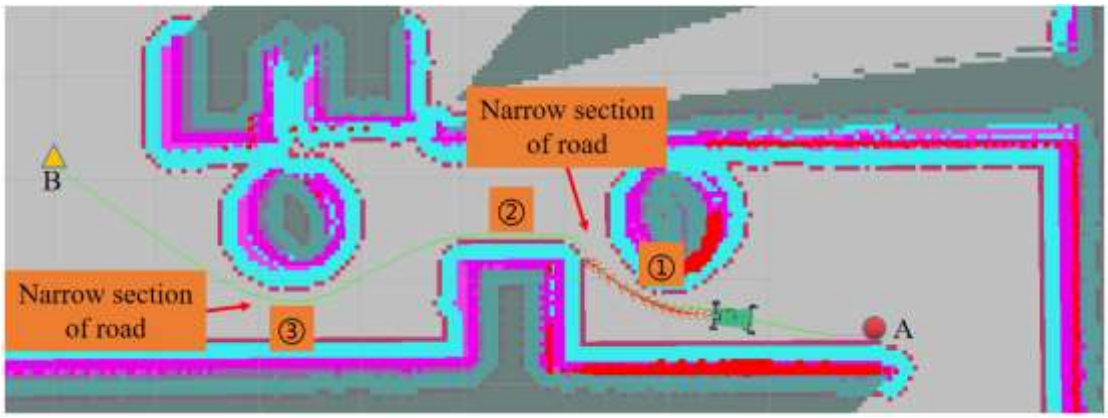


Figure 16. The robot passes through the narrow road.

The velocity output curve of the robot from the starting point A to the target point B, after a continuous narrow road section, is shown in Fig. 17. The traditional TEB algorithm shows a continuous reversing phenomenon when the robot passes the narrow road section at point ②, which is very easy to collide; while for the improved TEB algorithm reversing phenomenon is improved, and turning efficiency and safety and

smoothness are improved.

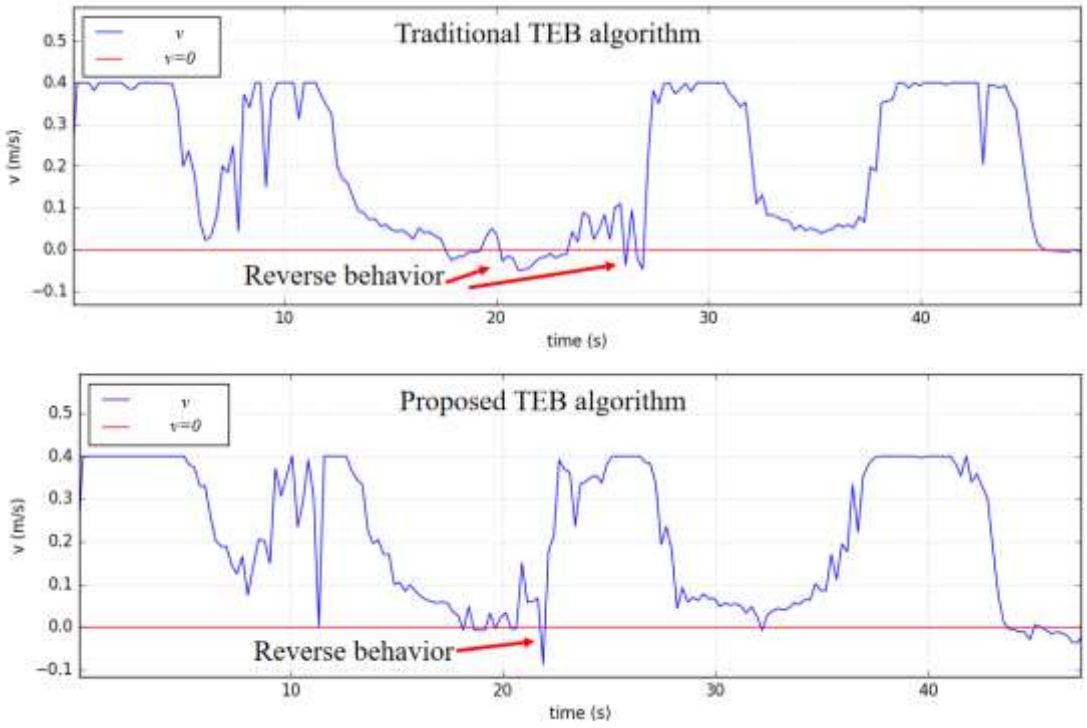


Figure 17. Robot speed output curve of the traditional TEB algorithm and the proposed TEB algorithm.

6. Conclusion

This paper presents the design of a novel and flexible inspection robot. The inspection robot is equipped with a four-wheel independent suspension to adapt to the undulating sections of the ground, a detection head that can flexibly detect all around, and a control algorithm that can detect in narrow environments, all of which improve the inspection efficiency of the robot. The inspection route planning is modeled by improving a timed elastic band (TEB) algorithm. Experiments on the path planning algorithm, the key problem of the robot, showed the modified planning algorithm can effectively control the robot to operate in narrow spaces, ensuring that the robot does not encounter other obstacles, and the running time is reduced by 3%, improving the efficiency of inspection.

Data availability statement

The data used in the manuscript are available from the corresponding author on reasonable request.

References

1. Salama, S., Hajjaj, H. & Khalid, I. bin. Design and Development of an Inspection Robot for Oil and Gas Applications. *IJET.* **7**, 5–10 (2018).
2. Feng, X. *et al.* Application of Wall Climbing Welding Robot in Automatic Weldi ng of Island Spherical Tank. *J COASTAL RES.* **107**, 1–4 (2020).
3. Nguyen, L. & Miro, J. V. Efficient Evaluation of Remaining Wall Thickness in C orroded Water Pipes Using Pulsed Eddy Current Data. *IEEE Sens.* **20**, 14465–144 73 (2020).
4. Hillenbrand, C., Schmidt, D. & Berns, K. CROMSCI: Development of a climbing robot with negative pressure adhesion for inspections. *IND ROBOT*. **35**, 228–237 (2008).
5. Shang, J., Bridge, B., Sattar, T., Mondal, S. & Brenner, A. Development of a cli mbing robot for inspection of long weld lines. *IND ROBOT*. **35**, 217–223 (2008).
6. Fischer, W. *et al.* Foldable magnetic wheeled climbing robot for the inspection of gas turbines and similar environments with very narrow access holes. *IND ROBO T*. **37**, 244–249 (2010).
7. Okamoto, J. *et al.* Development of an autonomous robot for gas storage spheres i nspection. *J Intell Robot Syst.* **66**, 23–35 (2012).
8. Nitta, Y. *et al.* Damage assessment methodology for nonstructural components wit h inspection robot. *KEM*. **558**, 297–304 (2013).
9. Krenich, S. & Urbanczyk, M. Six-legged walking robot for inspection tasks. *SOLI D STATE PHENOM*. **180**, 137–144 (2012).
10. Bruzzone, L. & Fanghella, P. Functional Redesign of Mantis 2.0, a Hybrid Leg-Wheel Robot for Surveillance and Inspection. *J Intell Robot Syst*. **81**, 215–230 (2 016).
11. Kim, S. H., Choi, H. H. & Yu, Y. S. Improvements in adhesion force and smart embedded programming of wall inspection robot. *J SUPERCOMPUT*. **72**, 2635–2 650 (2016).
12. Sun, J., Li, C., Wu, X. J., Palade, V. & Fang, W. An Effective Method of Weld Defect Detection and Classification Based on Machine Vision. *IEEE T IND INFO RM*. **15**, 6322–6333 (2019).
13. Li, Y., Hu, M. & Wang, T. Weld Image Recognition Algorithm Based on Deep L earning. *INT J PATTERN RECOGN*. **34**, doi: https://doi.org/10.1142/S02180014205 20047(2020).
14. Yang, L., Wang, H., Huo , B., Li, F. & Liu, Y. An automatic welding defect loca tion algorithm based on deep learning. *NDT and E International* **120**, doi: https:// doi.org/10.1016/j.ndteint.2021.102435 (2021).
15. Chen, W., Wu, X. & Lu, Y. An Improved Path Planning Method Based on Artific ial Potential Field for a Mobile Robot. *CIT*. **15**, doi: 10.1515/cait-2015-0037.(201 5).
16. Seddaoui, A. & Saaj, C. M. Collision-free optimal trajectory generation for a spac e robot using genetic algorithm. *ACTA ASTRONAUT* **179**, 311–321 (2021).

17. Saranrittichai, P., Niparnan, N. & Sudsang, A. Robust local obstacle avoidance for mobile robot based on Dynamic Window approach. *In 2013 10th International C onference on Electrical Engineering/Electronics, Computer, Telecommunications an d Information Technology,* Krabi, Thailand.1-4(2013)
18. Rösmann, C., Feiten, W., Wösch, T., Hoffmann, F. & Bertram, T. Trajectory modi fication considering dynamic constraints of autonomous robots. *In: ROBOTIK 201 2; 7th German Conference on Robotics,* Munich, Germany.1-6 (2012).
19. Rösmann, C., Hoffmann, F. & Bertram, T. Integrated online trajectory planning an d optimization in distinctive topologies. *ROBOT AUTON SYST* **88**, 142–153 (201 7).
20. Rösmann, C., Oeljeklaus, M., Hoffmann, F. & Bertram, T. Online trajectory predi ction and planning for social robot navigation. *In 2017 IEEE International Confer ence on Advanced Intelligent Mechatronics (AIM)* , Munich, Germany. 1255–1260 (2017) doi:10.1109/AIM.2017.8014190.
21. Nguyen, L. A., Pham, T. D., Ngo, T. D. & Truong, X. T. A Proactive Trajectory Planning Algorithm for Autonomous Mobile Robots in Dynamic Social Environm ents. In： *2020 17th International Conference on Ubiquitous Robots(UR)* Kyoto, J apan. 309–314 (2020) doi:10.1109/UR49135.2020.9144925.
22. Wu, J., Ma, X., Peng, T. & Wang, H. An Improved Timed Elastic Band (TEB) A lgorithm of Autonomous Ground Vehicle (AGV) in Complex Environment. *Sensor s.***21**, 8312 (2021).

Acknowledgements

This work was supported by the Natural Science Foundation of Shanghai [Grant number: 20ZR1422700]. Pei-Quan Xu has received research support from Science and Technology Commission of Shanghai Municipality (STCSM).

Author Contributions

P.X. was the leader of this project. P.Z. and P.X. designed the whole research plan and directed writing of the manuscript. P.Z. and F.Z. analyzed the simulation data and wrote the manuscript. L.L. and P.X. reviewed and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to P.X.