Glass Cleaning Robot Design

Boce Hu, Chongzhi Qiu, Jiaxing Fu, Zhiyuan Wang

1. Introduction

Nowadays, there are increasing numbers of skyscrapers with glass panel designs. However, this brings a serious problem in cleaning the outer surface of these buildings. Currently, glass panel cleaning is performed by workers hanging from the roof of the building and using simple tools. Their safety is threatened by windy weather conditions and worn-out equipment. Thus, we want to design a glass-cleaning robot to replace human workers who clean glass panels on skyscrapers.

A. Prior Work

A team from SSN College of Engineering came up with a design of Automatic Skyscraper Window Cleaning System (ASWCS) in 2017, with the design of a roller-based cleaning module to clean the vertical glass panel [1]. Also, a company called IPC Eagle has a product named Automated High Rise Window Cleaning System, which has a 7' cleaning path and two thruster fans to keep the system on glass while in operation [2].

B. Motivation and Novelty

Our main motivation is to protect the glass cleaning workers. Different from other working conditions, accidents for glass cleaning workers are more likely to lead to death. An analysis report compiled by OSHA lists 88 window cleaning accidents in the United States in the past 15 years, 62 of which resulted in fatalities, and there have been 22 cases in the past five years and 11 of them dead [3], [4]. If the risk of death is already reaching 70% in the United States, where those workers are well regulated and trained, it's hard to imagine how the number will be in some less developed country. The two robots in our prior research are both designed for cleaning large vertical panels, which means they are limited to a 2-D working space. Our idea of using a robotic arm outperforms them with a 3-D, reprogrammable working space, which gives it a relatively large cleaning range and allows the robot to clean surfaces with various contours and slopes. Another motivation is to incorporate the idea of artificial intelligence. With the redundant robot arm design and on-board computer, our robot can be constantly upgraded with more features in the future.

C. Problem we try to solve

The problem we are trying to solve is that we want to build a reliable robot which can replace humans in the glass cleaning process for skyscrapers. After figuring out our main problem, we brainstormed other small problems to guide us through the designs of all subsystems.

2. System Design

A. Design of Robot Arm

We chose to equip the robot with an arm with four degrees of freedom. The arm has three revolute joints, two links, and a rotational base. Since the arm only needs to perform the cleaning job on a single surface, we designed it to have two links and three joints to avoid redundancy. The length of the first link is 100 centimeters and the length of the second link is 80 centimeters. We designed the links to be quite long because the glass panels on high-rise buildings are usually very large, which means the robot arm's workspace needs to be large too. The robot arm has an end effector that can attach various cleaning tools.

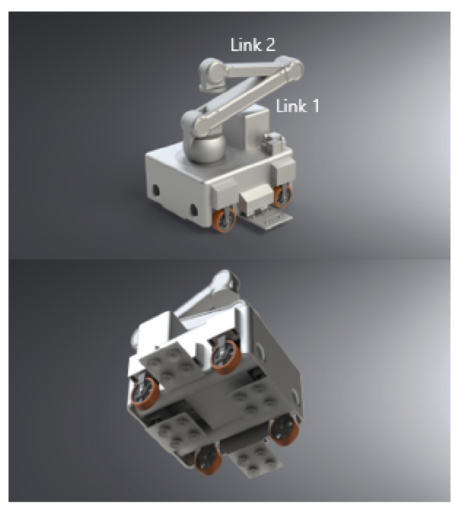

Fig. 1. Front and bottom view of the robot

Considering the glass surface will not always be flat, the joint 3 can adjust the angle of the end effector to align it with the tilted surface. To avoid scratching the glass, the arm will retract and fold when the robot is moving. Alternatively, the 4 wheels retract, the suction cups attach to the glass, and the arm unfolds if the robot reaches a certain location and starts to perform the cleaning work.

Through measuring in our CAD file, the link 1 has a rotation range of 135 degrees without collision. The link 2 has a rotation range of 165 degrees without collision. As for the rotational joint that connects the arm to the base, it has a 360 degrees of rotation range in certain arm positions.

B. Design of Robot Base

Our group designs the robot base to solve several problems that we developed during brainstorming. The overall look of our robot base is shown in *Fig. 1*. On the top, our group fit a toolbox for carrying more tools for various tasks. The communication module is also located on the top for better wireless transmission ability. From the bottom view, shown in *Fig. 1*, our group features four retractable suction cup units and retractable wheel units. As mentioned above, the retractable design allows our robot to have more clearance while moving without compromising the stability against wind while in work. All retractable mechanisms will be powered by step motors and the suction cup unit will be powered by two compressors onboard. The base also has locking and climbing mechanisms that allow the robot to adjust and fix itself at a certain altitude.

3. System Analysis

We utilize MATLAB to simulate the cleaning robotic arm. Firstly, we build the basic structure of our robotic arm as shown in *Fig. 2*. It is a 4 DOF cleaning arm and the D-H table of this arm is shown in *Table 1*. We add two constraints for the 2nd and the 3rd joints of the arm so that the arm will always elbow up. The second joint's angle is constrained to be between 0 to pi*3/4, and the third joint's angle is constrained to be between -pi*5/12 to pi/2. With this design, the links will not hit the base. Secondly, the workspace of this robotic arm is recalculated by iterating all joint angles which are not beyond their limits in MATLAB, and the result is shown in *Fig. 3*. Each point in this image is the position that the arm can reach. Although the result shows that the workspace's shape is a hemisphere with a radius of 1.3 meters, our robot arm will not reach most positions during its normal working time. It works just on a flat surface in most cases. Only under special circumstances can it be possible to reach these spaces to prevent jamming of the robotic arm. Because our robotic arm is a redundant robot, it has more than one way to reach a position, and this will also help to prevent the robotic arm from having problems with singularities.

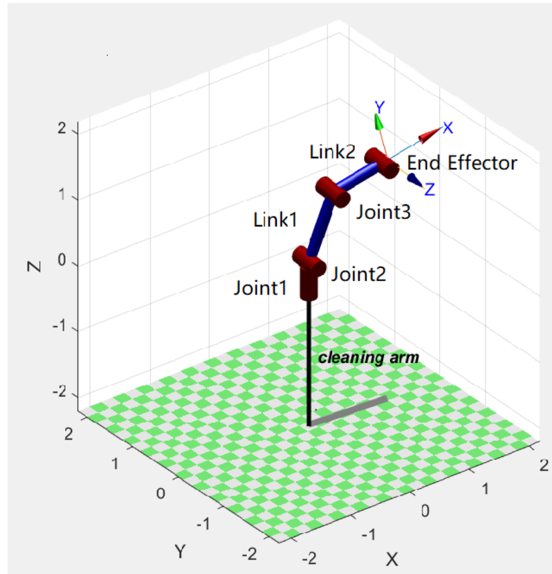

Fig. 2. Basic structure of the cleaning arm

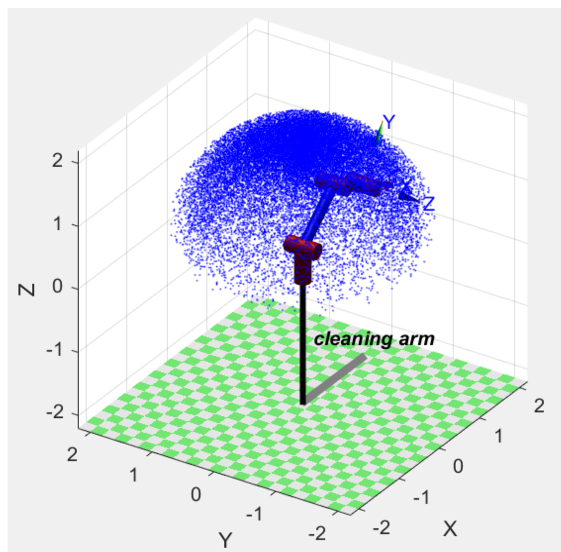

Fig. 3. Workspace of the cleaning arm

Table 1. D-H Table of the cleaning arm

	a	alpha	d	theta
1	0	pi/2	0.3	theta1
2	1	0	0	theta2
3	0.8	0	0	theta3
4	0.1	0	0	theta4

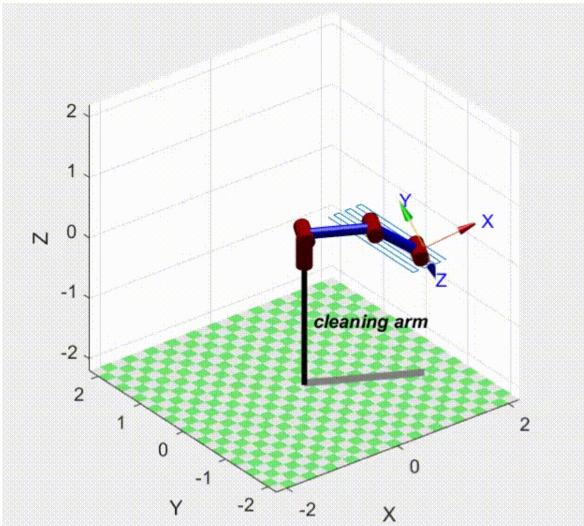

Fig. 4. Cleaning trajectory of the robotic arm

The cleaning trajectory is regulated by the specific cleaning mission and is also simulated in MATLAB as shown in *Fig. 4*. The trajectory of cleaning windows is from top to bottom and from right to left. After calculating and simulating, our robotic arm can wipe windows up to 2 meters by 1.5 meters and will not generate any singularities in this process. This coverage area is big enough for the daily cleaning mission. In conclusion, this robotic arm can handle daily cleaning tasks perfectly.

4. Discussion

In this project, our group has drawn a preliminary solution to the primary problem of replacing human workers with robotic devices in glass cleaning work. We designed a glass cleaning robot that could freely move outside the glass walls of high-rise buildings. The robot consists of two parts, a support base and a robotic arm, the former helps the robot to move and fix at the working position, the latter takes charge of cleaning glasses. In the market, most of the glass cleaning robot products are positioned at household works while very few large devices are used to clean the outer surface of glass curtain walls, our solution perfectly fits the blank of this field.

At the same time, we still have limitations in our design. Human operators are still needed to control the robot, where extra training will be needed. Human operators are also easier to make mistakes than autonomous systems, so in the future, implementing AI technology in this robot is inevitable. Additionally, the robot performs better on vertical surfaces while lots of high-rise buildings prefer to use curved surfaces.

5. Conclusion

Through this project, we successfully gave a potential solution for the problem of replacing human workers with glass cleaning robots. We designed the model for the robot in SolidWorks and utilized what we have learned in the classroom, including constructing DH table, transfer matrix, and inverse kinematics to simulate the robotics arm. In the future extension, we will add more tools to explore more functions of the robot and make it capable of switching functions while working. The design will be optimized to fit different curved surfaces. An autonomous control system will also be added for better efficiency and control. Based on this platform, we can further design multifunctional robot products such as hazard removal robots and moon cars.

Reference

[1] K. S, K. M.S., and A. K. C, "Automatic Skyscraper Window Cleaning System," IAES International Journal of Robotics and Automation (IJRA), vol. 6, no. 1, p. 15, 2017.

[2]"IPC Eagle Automated High Rise Window Cleaning System," *Window Cleaning Resource*. [Online]. Available: https://windowcleaner.com/ipc-eagle-automated-high-rise-window-cleaning-system?gclid=Cj0KCQiAnuGNBhCPARIsACbnLzrMe4iLZluIbrUenaGI31cNWFW0xmi6glX4PeWfR5t9vX1geP_7XpkaArzhEALw_wcB. [Accessed: 14-Dec-2021].

[3] H. D. News, "Window washers," *HealthDay*, 14-Dec-2021. [Online]. Available: https://consumer.healthday.com/encyclopedia/work-and-health-41/occupational-health-news-507/window-washers-646930.html. [Accessed: 14-Dec-2021].

[4]"United States Department of Labor," *Accident Search Results Page | Occupational Safety and Health Administration*. [Online]. Available: https://www.osha.gov/pls/imis/accidentsearch.search?sic=&sicgroup=&naics=&acc_description=&acc_abstract=&acc_keyword=%22Window+Washer%22&inspnr=&fatal=&officetype=&office=&startmonth=&startday=&startyear=&endmonth=&endday=&endyear=&keywo

rd_list=on&p_start=20&p_finish=40&p_sort=&p_desc=DESC&p_direction=Prev&p_show=20. [Accessed: 14-Dec-2021].

Appendix

Cleaning Trajectory:

```
clc;
clear;
L(1) = Link('d', 0.3, 'a', 0, 'alpha', pi/2);
L(2) = Link('d', 0, 'a', 1, 'alpha', 0);
L(3) = Link('d', 0, 'a', 0.8, 'alpha', 0);
L(4) = Link('d', 0, 'a', 0.1, 'alpha', 0);
L(2).qlim = [0,pi*3/4];
L(3).qlim = [-pi*5/12,pi/2]

robot = SerialLink([L(1) L(2) L(3) L(4)], 'name',
'cleaning arm');

T1=transl(1.8,-1,-0.3);T2=transl(1.8,1,-0.3);
T3=transl(1.7,1,-0.3);T4=transl(1.7,-1,-0.3);
T5=transl(1.6,-1,-0.3);T6=transl(1.6,1,-0.3);
T7=transl(1.5,1,-0.3);T8=transl(1.5,-1,-0.3);
T9=transl(1.4,-1,-0.3);T10=transl(1.4,1,-0.3);
T11=transl(1.3,1,-0.3);T12=transl(1.3,-1,-0.3);
T13=transl(1.2,-1,-0.3);T14=transl(1.2,1,-0.3);

t= [0:0.02:0.2]';
Ts1=ctraj(T1,T2,length(t));
Ts2=ctraj(T2,T3,length(t));
Ts3=ctraj(T3,T4,length(t));
Ts4=ctraj(T4,T5,length(t));
Ts5=ctraj(T5,T6,length(t));
Ts6=ctraj(T6,T7,length(t));
Ts7=ctraj(T7,T8,length(t))
;Ts8=ctraj(T8,T9,length(t));
Ts9=ctraj(T9,T10,length(t));
Ts10=ctraj(T10,T11,length(t));
Ts11=ctraj(T11,T12,length(t));
Ts12=ctraj(T12,T13,length(t));
Ts13=ctraj(T13,T14,length(t));


q_s1=robot.ikcon(Ts1);q_s2=robot.ikcon(Ts2);
q_s3=robot.ikcon(Ts3);q_s4=robot.ikcon(Ts4);
q_s5=robot.ikcon(Ts5);q_s6=robot.ikcon(Ts6);
q_s7=robot.ikcon(Ts7);q_s8=robot.ikcon(Ts8);
q_s9=robot.ikcon(Ts9);q_s10=robot.ikcon(Ts10);
q_s11=robot.ikcon(Ts11);q_s12=robot.ikcon(Ts12);
q_s13=robot.ikcon(Ts13);
q =
[q_s1;q_s2;q_s3;q_s4;q_s5;q_s6;q_s7;q_s8;q_s9;q_s10;
q_s11;q_s12;q_s13];

x=squeeze(Ts1(1,4,:)); y=squeeze(Ts1(2,4,:));
z=squeeze(Ts1(3,4,:));
x1=squeeze(Ts2(1,4,:)); y1=squeeze(Ts2(2,4,:));
z1=squeeze(Ts2(3,4,:));
x2=squeeze(Ts3(1,4,:)); y2=squeeze(Ts3(2,4,:));
z2=squeeze(Ts3(3,4,:));
x3=squeeze(Ts4(1,4,:)); y3=squeeze(Ts4(2,4,:));
z3=squeeze(Ts4(3,4,:));
x4=squeeze(Ts5(1,4,:)); y4=squeeze(Ts5(2,4,:));
z4=squeeze(Ts5(3,4,:));
x5=squeeze(Ts6(1,4,:)); y5=squeeze(Ts6(2,4,:));
z5=squeeze(Ts6(3,4,:));
x6=squeeze(Ts7(1,4,:)); y6=squeeze(Ts7(2,4,:));
z6=squeeze(Ts7(3,4,:));
x7=squeeze(Ts8(1,4,:)); y7=squeeze(Ts8(2,4,:));
z7=squeeze(Ts8(3,4,:));
x8=squeeze(Ts9(1,4,:)); y8=squeeze(Ts9(2,4,:));
z8=squeeze(Ts9(3,4,:));
x9=squeeze(Ts10(1,4,:)); y9=squeeze(Ts10(2,4,:));
z9=squeeze(Ts10(3,4,:));
x10=squeeze(Ts11(1,4,:)); y10=squeeze(Ts11(2,4,:));
z10=squeeze(Ts11(3,4,:));
x11=squeeze(Ts12(1,4,:)); y11=squeeze(Ts12(2,4,:));
z11=squeeze(Ts12(3,4,:));
x12=squeeze(Ts13(1,4,:)); y12=squeeze(Ts13(2,4,:));
z12=squeeze(Ts13(3,4,:));

x2=[x;x1;x2;x3;x4;x5;x6;x7;x8;x9;x10;x11;x12];
y2=[y;y1;y2;y3;y4;y5;y6;y7;y8;y9;y10;y11;y12];
z2=[z;z1;z2;z3;z4;z5;z6;z7;z8;z9;z10;z11;z12];
figure('Name','robotic arm trajectory');
plot3(x2,y2,z2);
robot.plot(q);
robot.fkine(theta)
robot.plot([0 pi/2 0 0]);
```