ISSN 2959-6157

Optimization of PID Control of a Snake Robot's Winding Crawling Motion

Junwei Hu, Yiran Yao, Tianhan Zhou, Runzhe Jiang

thomashu07@hotmail.com
Simulink Design PID and forward motion design
2329952470@qq.com
simulink simulation
win47821143@163.com
matlab code
2897526808@qq.com
modeling and paper collection

Abstract:

This paper explores the application of PID control to the winding crawling motion of a snack robot. Snack robots, mimicking the movement of real life limbless reptile, are designed with multiple connected segments equipped with actuators and sensors, enabling them to navigate complex environments. The study aims to optimize the ground crawling behavior and obstacle avoidance of the snake-like robot using MATLAB simulation software. Initially, the robot's joints were modeled in SolidWorks, and their degrees of freedom were programmed in MATLAB's Simulink to enable controlled rotation. The PID control system was added to manage the robot's motion, ensuring smooth and accurate crawling.

Keywords: PID control, snake-like robot, crawling motion, MATLAB simulation, Simulink, robot modeling, SolidWorks.

I. Introduction

Snake robots, also known as serpentine robots or snake-like robots, are a specialized category of robotic systems designed to emulate the locomotion capabilities of snakes. These robots typically consist of multiple connected segments, mimicking the flexible body structure of their biological counterparts. Each segment is equipped with actuators, sensors, and sometimes cameras, allowing the robot to move in a coordinated and controlled manner. The design of snake robots enables them to navigate through complex and confined spaces that are difficult for traditional wheeled or legged robots to access

The applications of snake robots are diverse and impactful across various industries. In search and rescue operations, they excel in exploring disaster zones, such as collapsed buildings or rubble, where human rescuers may face challenges. Their ability to slither through narrow gaps and uneven terrain makes them invaluable tools for locating and assisting survivors. In industrial settings, snake robots are used for inspection and maintenance tasks in pipelines, where their agility and ability to traverse intricate networks reduce the need for disruptive excavation or dismantling.

II.Motivation

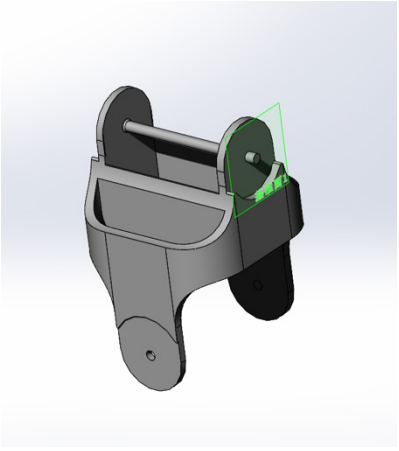
We intend to employ MATLAB simulation software to regulate the ground crawling behavior of a snake-like robot and to simulate its obstacle avoidance during crawling by placing obstacles at the path's turning points. Initially, we utilized SolidWorks to model the robot's joints, followed by limiting and programming their degrees of freedom in MATLAB's Simulink to enable rotation within specific angles. Subsequently, we designed the main body of the snake-like robot and also imported it into MATLAB for simulation, adding obstacles on the ground and at the turning points of its meandering motion to simulate its obstacle avoidance behavior. Throughout the simulation and design phases, we utilized PID control, successfully achieving the goal of simulating the serpentine crawling motion of the snake-like robot.

III.background

Through extensive research, we have learned that snake-like robots are widely used for rescue and search in large-scale disasters, underwater exploration, and more. Therefore, we have gained a certain understanding of snake-like robots. Currently, some studies showcase snake-like robots, such as underwater models, with joints designed for three degrees of freedom, allowing them to efficiently explore inaccessible areas like the ocean. Additionally, their resemblance to sea snakes prevents interference from other marine organisms, significantly facilitating seabed exploration. However, their limited resistance and anti-interference capabilities underwater result in high energy consumption. Moreover, there are earthquake relief snake robots that can navigate through the gaps in earthquake ruins more flexibly than humans, searching for and rescuing victims. However, their shortcomings are also evident. The irregular gaps in earthquake ruins make these robots susceptible to getting stuck, potentially hindering search and rescue efforts.

IV.Design Method and Theory

The snake-inspired robotic system comprises interconnected modular bodies with vertical axes that facilitate movement in three dimensions. Each modular unit of the snake robot possesses the capability to accommodate diverse tools and instruments, thereby enabling it to adapt to the specific equipment needs of different environments and tasks in prospective applications.



Advance

Progression is facilitated by a sinuous movement pattern. Our study drew upon the research of Professor Hirose to develop a mathematical representation of snake locomotion in natural environments, specifically utilizing a sinusoidal crawling function.

$$x(s) = \int_0^s \cos(a * \cos(bs) + cs)ds$$

$$y(s) = \int_0^s \sin(a * \cos(bs) + cs)ds$$

Merely integrating the sinuous sine function into the locomotion of the serpentine robot may lead to notable disruptions and variations stemming from error accumulation. Consequently, the implementation of Proportional-Integral-Derivative (PID) control is being considered as a potential solution to address this issue.

To facilitate simulation within the MATLAB environment, a three-link robot model has been adopted as a surrogate. In this model, the midpoint of the second link is designated as the robot's center, with the angles of each link denoted as theta1, theta2, and theta3, respectively. As a result, the coordinates of the endpoints of the second link can be expressed as follows.

```
x2 = omx0 - (l/2)*cos(th20);
y2 = omy0 - (l/2)*sin(th20);

x3 = omx0 + (l/2)*cos(th20);
y3 = omy0 + (l/2)*sin(th20);
```

Included in this group are the initial position coordinates of the center point, denoted as omx0 and omy0, the length of the link represented by l, and the initial angle of link2 at the commencement of the task, indicated as th20.

By recalculating the angles and lengths of the links starting from the initial and final points, the positions of the endpoints of link1 and link3 can be determined. This process can be expressed as:

```
x1 = x2 - l*cos(th10);
y1 = y2 - l*sin(th10);

x4 = x3 + l*cos(th30);
y4 = y3 + l*sin(th30);
```

The angles of link1 and link3 are th10 and th30 respectively.

```
figure('units','normalized','outerposition',[0 0 1 1]);
axesHandle = gca;
xlim(axesHandle, [(-5) (30)]);
ylim(axesHandle, [(-10) (10)]);

rectHandle1 = rectangle('Position',[x2 y2 .05 .05], 'Curvature',[1,1],'FaceColor','g');%link segment ends
rectHandle2 = rectangle('Position',[x3 y3 .05 .05], 'Cur-
```

```
vature',[1,1],'FaceColor','g');%link segment ends
hold on
lineHandle1 = line([x2 x1],[y2 y1]); %link segments
lineHandle2 = line([x3 x2],[y3 y2]); %link segments
lineHandle3 = line([x4 x3],[y4 y3]); %link segments

set(lineHandle1,'Color','r'); %line 1 color
set(lineHandle2,'Color','g'); %line 2 color
set(lineHandle3,'Color','b'); %line 3 color

hold off
```

This is to connect and draw lines between endpoints, and to differentiate each link with different colors.

```
for j=1:size_rowSS
drawnow; %Forces MATLAB to render the snake
%%recalculate positions -
x2 = -simwx(j,1) + (l/2)*cos(simth2(j,1));
y2 = simwy(j,1) + (l/2)*sin(simth2(j,1));

x3 = -simwx(j,1) - (l/2)*cos(simth2(j,1));
y3 = simwy(j,1) - (l/2)*sin(simth2(j,1));

x1 = x2 + l*cos(simth1(j,1));
y1 = y2 + l*sin(simth1(j,1));

x4 = x3 - l*cos(simth3(j,1));
y4 = y3 - l*sin(simth3(j,1));
%redraw links
set(rectHandle1,'Position',[x2 y2 .05 .05]);
set(rectHandle2,'Position',[x3 y3 .05 .05]);
%redraw snake
set(lineHandle1,'XData',[x2 x1],'YData',[y2 y1]);
set(lineHandle2,'XData',[x3 x2],'YData',[y3 y2]);
set(lineHandle3,'XData',[x4 x3],'YData',[y4 y3]);
pause(0.01)
end
```

This is an iteration of the simulation of the entire system, achieving the demonstration of the entire control process.

Avoid obstacles

During the actual movement of the snake-like robot, it may encounter obstacles such as steps that require the robot to climb over. Therefore, we hope to control the robot to lift the front half of its body to make contact with the surface of the step, and then complete the process by crawling in a winding manner to pull the rear half of its body onto the step.

V.RESULT

3-link snake robots

We first designed the two joints at 90 degrees to each other.As a result,we could control it rotate at three-dimensional space.However, each joints simply controls the rotation of the two-dimensional plane,so in one plane it become to a 2-link robot.

But when we use it to do simulation,it couldn't move. Through force analysis, we find that the friction generated by the front and rear links of the serpentine robot with only two links cancels each other out during swinging, preventing it from moving.

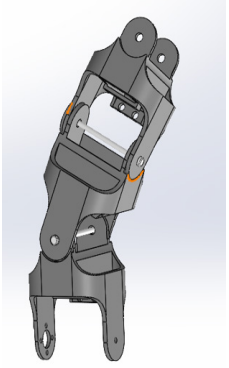


Fig.1 3-link snake robot

4-link snake robot

Then we extended the number of sections of the snake robot to four links and three joints,making it become a 3-link snake robot in one plane.

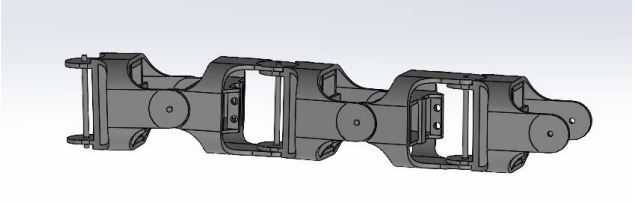


Fig.2 4-link snake robot

In order to perform better control simulation,we simplified the model at two-dimensional plane with 3 segments as 3 links and 2 dots as 2 joints.Besides,we added the ground with a thick black straight line.Then we began to simulate the two common gaits of the snake, serpentine and concertina,by using simulink software in MATLAB.We wrote PID programs and put PID control and SIN functions into our simulation to simulate the two gaits in the ground.

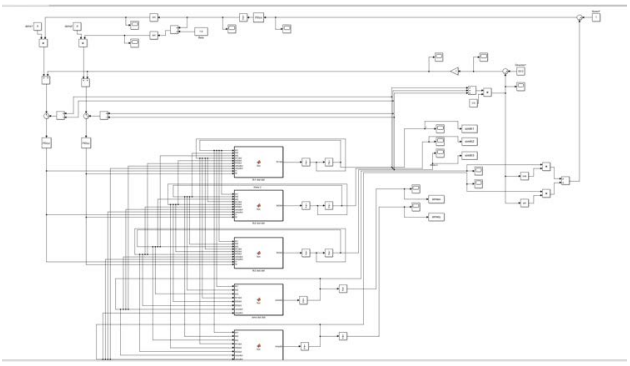


Fig.3 Simulink_ClosedLoop

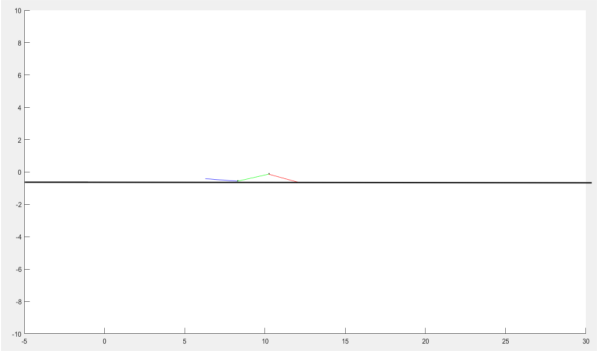


Fig.4 result of above simulation

VI. Summary and Future Work Development

After all the simulations and work were completed, we conducted a proper summary of our work and made a series of prospects for the future development direction of the project, as shown below Acknowledgment

The work summarry

In this simulation of the winding motion of the snake like robot, our team members actively communicated and completed their tasks well even under the pressure of exam week. We still encountered some problems during the project. After completing the joint modeling, we were not proficient in using Matlab, so we were unable to import the joint models built in SolidWorks into different versions of Matlab. Secondly, after solving the import problem, we encountered the problem of model penetration during the simulation of joint motion. Through continuous learning and searching for relevant literature and materials, we were able to finally solve the model penetration problem. Afterwards, after completing the modeling and PID programming, we encountered another problem: during the movement of the model, our model always penetrated the floor, so we had to rebuild a new model to simulate winding crawling. In the end, we successfully simulated the winding crawling process of the snake like robot on the ground and set obstacles reasonably at the turning points of its path to simulate obstacle avoidance function.

Outlook for future work

For this snake like robot, the design of our team members is not perfect and can even be said to have many problems. Firstly, we did not create a three degree of freedom snake like robot in the model to achieve three-dimensional motion. Our simulation of meandering crawling was limited to simulating its motion in a two-dimensional plane. Secondly, we did not truly achieve obstacle avoidance in the simulation of snake like robots in obstacle avoidance function because we did not set up a recognition block for obstacles. We only added obstacles to the set motion trajectory for simulation. In practical applications, a module that senses obstacles can be added to change the robot's motion from two-dimensional to three-dimensional, thus achieving true obstacle avoidance motion for snake like robots. Finally, regarding the control system, nowadays most snake like robots, whether underwater or on land, use ADCR control systems. This system can provide an additional feedback signal on top of the PID system to achieve more precise control of the robot. Therefore, our future work should focus on improving the control system, transforming the model from two-dimensional to three-dimensional, and adding obstacle sensing modules, in order to create a truly snake like obstacle avoidance robot.

References

[1] B. Tao, H. Sun, and J. Sun, "Dynamic Modeling and Control of Underwater Snake Robot." Data Driven Control and Learning Systems Conference, Emeishan, China, May. 7, 2022

[2] Mathworks-Robotics, "Example files for MATLAB and Simulink Robotics Arena Walking Robot Videos.," GitHub, https://github.com/mathworks-robotics/msra-walking-robot (accessed Jun. 29, 2024).

[3] Celia0731, "Snake_robot/file matlab at main · celia0731/snake_robot," GitHub, https://github.com/Celia0731/snake_robot/tree/main/File%20matlab (accessed Jun. 29, 2024).