Stair Climber Carrier: Design and Analysis using Simulation

Yutong Wu

I. Introduction

Robotics have been used to assist human tasks in various fields, such as environment exploration, cooking, and public security. One of its essential aspects is working as a carrier of goods or people in need when no elevator or escalator is available. In Texas, the robotic assistant helps police officers to take down shooters. Additionally, in the most recent winter Olympics in Beijing, the robotic cook served all kinds of food to guests from the entire world. Besides these applications, robots are also used as carriers of goods and people in need. Especially, robotic carriers are needed when the environment is more complex than flat ground. For example, stairs are one of the scenarios that might need for carrying forces when escalators or elevators are not available. To fulfill this purpose, many possible designs for stair climbers are proposed. This project explores and simulates the possible and better design of the stair climbing carrier robots in a virtual environment using leg-wheels design.

II. Backgrounds

At the beginning of the design of the stair climbers, the simplest and most direct method is to increase the friction between the connected surfaces between the wheels of the robots and the surface of the stair. Since the surface of the stair is not controllable, robots with large and sticky wheels are designed.

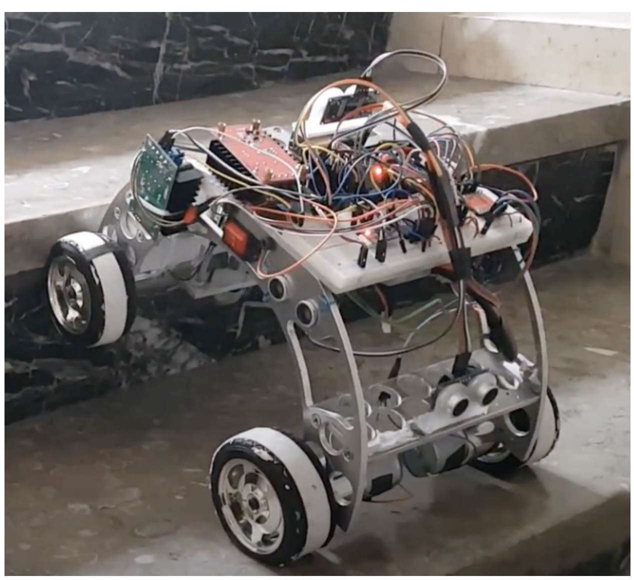

Figure 2.1 Robot with sticky materials on the wheels to increase friction

Another way to increase friction is using a track instead of wheels since tracks have a larger contact area than wheels with the surface of the stairs. More importantly, the frictional force between the tracks and the surface is sliding friction, which is relatively more stable than the rolling friction applied by the wheels to the surface.

.

Figure 2.2 Tracked wheelchair helps disabled people to climb stairs

The third approach is the design of “legged-wheels” (This is different from the leg-wheel design we talked about later in the paper.). These wheels typically have no tires that bound the shape of the wheels as circles. Instead, their shapes could be stars, triangles, and other shapes since they are made of some legs with one of their ends connected by axes. These types of robots are normally used for environment discovery since the versatile design of the wheels is adaptable in different terrains.

Figure 2.3 Robot Asgard

III. Related Works

All three approaches have their field of applications. However, in terms of the stability when carrying the goods over the stairs, since all three kinds of design normally have the body of the robots parallel to the rising trend of the stairs instead of the ground, additional protection is needed to prevent the sliding of the goods or people the robots carry. To further improve the stability of the carrying process A brand new leg-wheel hybrid design is proposed that is able to keep the body of the robot horizontally while carrying the target over the stairs. The robot is designed by a group of Japanese researchers and is named Zero Carrier, which consists of 8 legs, 8 wheels, and the main body.

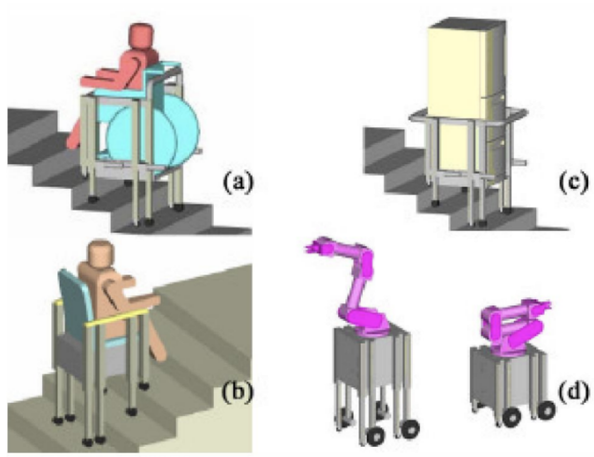

Figure 3.1 Usage of the “Zero Carrier”

Unlike the three approaches discussed above, the Zero Carrier is able to lift the legs two at a time with the other six legs remaining on the ground that keeping the stability of the robot, which has its main body parallel to the ground. This design significantly increases the stability of the carrying process, regardless of going up or downstairs.

IV. Simulation Approach

The main purpose of the simulation is to propose a simplified control flow of the sensors on the robot. Additionally, the casters on the original Zero Carrier are replaced with wheels that has less degree of freedom aiming to further decrease the instability of the carrying process.

A. Design of the Robot

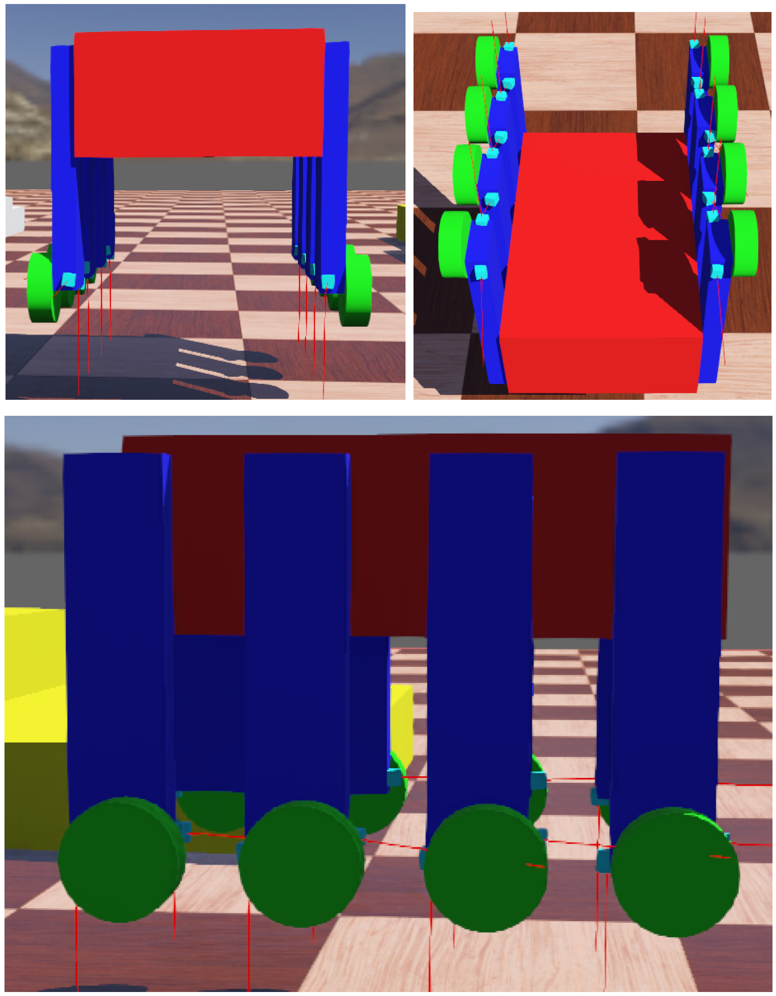

Figure 4.1 upper-left: front; upper-right: bottom; lower: side

The whole robot has a similar consistency as that of the “Zero Carrier”, including the main body (width = 0.2m, length=0.3m, height=0.1m), 8 legs (width = 0.02m, length=0.05m, height = 0.2m) and 8 wheels (radius=0.03m, height=0.02m). There are in total 24 sensors which are 24 distance sensors and 12 position sensors. As the figure shows, the red lines are the sensor ray of the distance sensors (blue cubes). There are 12 distance sensors facing the front and the other 12 facing the ground. The sensors that face the ground are at relative positions (x: -0,02, y:-0,0007, z:-0.1) of the leg and the ones facing the front are at (x:0.0278, y:-0.00416, z:-0.09). Position sensors do not have the actual objects in the simulation. We can regard them as embedded into the centers of the legs. All wheels are controlled by rotational motors and all legs are controlled by linear motors. The position sensors are used to record the positions of the legs on z axes.

B. Control Flow

The main algorithms can be divided into upstairs and downstairs.

When the robot is going upstairs, only the sensors that face the front and position sensors will be activated. The distance sensors on the first pair of legs will be activated first to detect whether the stairs are approaching. As the first pair of legs are close enough to the stairs, they will be lifted by the linear motor at a given distance by the user (around the height of the stairs). The sensors on the next leg will be activated when the position sensor on the previous leg detects that the leg is lifted. When the position sensors on the last pair of legs detect that both legs are lifted and the floor-facing distance sensors on them detect that legs land on a new surface, all the legs will be re-set to the original position preparing for the next move. When all legs are lifted to the stair, the legs will be set to the original position by linear motors. Going downstairs has the same structure as the control flow of going upstairs. The only difference is that now the distance sensors that face the ground will be activated instead. The position sensors are always activated regardless of the robot's direction.

```
        leg[7].setPosition(iposition-0.04)
if ps[2].getValue() < -0.05 or ps[3].getValue() < -0.05:
    # ds[2].enable(TIME_STEP)
    # ds[3].enable(TIME_STEP)
    if dsf[4].getValue() > 900:
        print("leg 4 activate")
        leg[4].setPosition(-0.04)
    elif dsf[5].getValue() > 900:
        print("leg 5 activate")
```

Figure 4.1 An example of the identical unit in the control flow: “ps” is a list of position sensors, “dsf” is the list of the floor-facing distance sensors. “Leg” is a list of linear motors corresponding with each movable legs

V. Results and Analysis

I mainly test the simulated bot with two different sets of settings: Stair length > body length of the robot (5 stairs, in white); Stair length < body length of the robot (2 stairs, in yellow)

The results show that the first set of the experiment requires that only the movable distance is able to afford the accumulation of height of the entire stairs, i.e. number of stairs * stair height. The reason is that the robot is unable to set to the original position (i.e. re-set the legs) since there is no situation where all four legs are lifted to the same stair. The stability of the robot climbing and going down the stair with a short length is significantly affected by the input height of the stair height by the user since there is not much space for the robot to adjust its position if there is instability caused by the clashing of the wheels with the edge of the stair because of insufficient lifting height. Conversely, if the stair is long enough to hold the entire robot, the moveable leg length only needs to afford the height of a single stair since every time the legs will be reset to the original position as the robot is on the ground each time all legs arrive the new stair.

Another problem found during the simulation on the stairs with enough length to hold all four pairs of robot legs is the drifting of the heading direction of the robot caused by the displacement of the center of mass of the whole robot when the legs at the front are lifted or dropped. The drifting of the heading accumulates as the robot proceeds forward. In the previous trials, if we set the heading of the robot to align with the x-axis, the drifting caused the robot to turn right and go down the last stair at the right corner. This issue caused the wrong reading of the bottom-facing distance sensor, which makes the robot legs unable to return to the original position. To tackle this problem, I intentionally make the initial heading direction of the robot around 1.83 rad in terms of the x-axis, and the robot finally went downstairs smoothly. However, this problem might occur again if we increase the level of the stairs, which allows the error drifting to accumulate more.

VI. Conclusion & Future Work

Compared to the actual Zero Carrier, my simulated bot has a half-closed loop control flow since it required the user to hard-code the height into the brain of the robot instead of detecting itself. This might be a limitation when bringing this design into reality since there exist errors in the height of each stair even on the same stairs. Another limitation caused by the virtual simulation itself is that there is an unexpected movement of the legs when the first pair of legs is activated. This problem even exists when the distance sensors on the remaining legs are not enabled for sure. It could be caused by the complex physical feature of the simulated environment, which might not be the problem in the real world. This problem is solved by giving mass to the robot body, though causing the head-drifting issue.

A potential next step will be to further study the torque applied by the legs to the robot body and find ways to make it more stabilized during the vertical movement of the legs. Additionally, detecting the height of each stair itself instead of directly assigning it to linear motors will be an important step to make the robot better adapted to the new environment automatically.

Reference

[1] Jianjun Yuan and S. Hirose, "Research on leg-wheel hybrid stair-climbing robot, Zero Carrier," *2004 IEEE International Conference on Robotics and Biomimetics*, 2004, pp. 654-659, DOI: 10.1109/ROBIO.2004.1521858.

[2] M. Eich, F. Grimminger and F. Kirchner, "A Versatile Stair-Climbing Robot for Search and Rescue Applications," *2008 IEEE International Workshop on Safety, Security and Rescue Robotics*, 2008, pp. 35-40, DOI: 10.1109/SSRR.2008.4745874.

[3] Chun-Ta Chen and Hoang-Vuong Pham, "Enhanced development and stability analysis of a new stair-climbing robotic wheelchair," *2008 IEEE Workshop on Advanced Robotics and Its Social Impacts*, 2008, pp. 1-6, DOI: 10.1109/ARSO.2008.4653582.