VR Tank: A First-Person View Self-Propelled Artillery with Automatic and Manual Modes

Bo-Hsun Huang[1], PoYu Wu[2], Ta-Yo Mu[1], Chih-An Tsao[2], and Li-Chen Fu[3]

[1]*Discrete Algorithm and Wireless Network Laboratory, National Taiwan University, Taipei, Taiwan*
[2]*Communications and Multimedia Laboratory, National Taiwan University, Taipei, Taiwan*
[3]*Intelligent Robot Laboratory, National Taiwan University, Taipei, Taiwan*
{*r07922015,r08922a13,r08922132,r08922070,lichen*}*@ntu.edu.tw*

***Abstract*—In this paper, we show how to build up a virtual reality tank based on Robot Operating System(ROS). There are two modes of our tank, auto mode and manual mode. One can search for the target by the robot itself and shoot automatically, and the other let a real human control the robot to find and shoot at the target remotely.**

We provide our critical source codes on GitHub[1], and the demo video on YouTube[2].

***Index Terms*—first-person view, remote control, target searching, robotics**

I. Introduction

Recently, the virtual reality is compelling drastically all over the world. The experience brought by virtual reality is much more powerful and realistic than the normal monitor. Lots of different applications have been invented along with such a trend. In medical, military, entertainment and so on, we can see that the virtual reality is changing our livings.

Due to the trend described above, we come up with a thought to build up tank based on virtual reality. There are some reasons. Firstly, it is impossible for almost all of us to drive a real tank during our lifetime. Secondly, those who are the drivers of the normal tank have to take risk on their lives while driving. Lastly, if we can reduce the space of the cockpit in the tank, not only the cost would be more economical but also the design would be more flexible. Therefore, we build up a device called VR tank.

We implemented the VR tank based on Robot Operating System(ROS). This device let us control the tank to destroy the target remotely and safely. In addition to manual control, we also implement the mode of auto control for our VR tank. The auto mode makes sure that even we somehow lost the control of our tank accidentally, it can still complete the mission automatically.

II. Related Work

A. First-Person View and Remote Control Robot

The previous work [1] builds a first-person view and remote control ball-picking robot with LEGO Mindstorms EV3. It captures the stereo images from two web camera and streams them to the smartphone via Bluetooth. The user can control the robot's movement and gripper by a joystick also via Bluetooth. Inspired by this work, we use similar way to remotely control the robot.

B. Table Tennis Serve Machine

In the previous works and commercial products, the table tennis serve machine generally use DC motors with wheels to accelerate balls. [3] use only one motor to make the ball spin, while [2] and [4] use a pair of motors so it can shoot straight and easy to predict the balls trajectory. As we want to shoot at the target accurately, we use 2 DC motor as accelerator. [4] placed the motor pair horizontally, while [2] placed them vertically. For this we have conduct a pilot study and found that the machine can shoot farther when the motors are placed vertically.

III. Methodology

A. Hardware Device

1) Pioneer 3-DX: Pioneer 3-DX is a rugged general purpose differential-drive mobile robot platform with several options including sensors, onboard computer, autonomous navigation software, gripper, and more. We use ROSARIA as an interface between ROS and Adept MobileRobots' open source ARIA library.

Fig. 1. Pioneer 3-DX.

2) Hokuyo URG-04LX-UG01: Hokuyo URG-04LX-UG01 is a laser range finder for autonomous robot. With wide-range and decent accuracy, it's an ideal sensor for mapping and localization. We create a node called urg_node to start up laser range finder

[1]https://github.com/frozenburst/2019NTU-Robotics
[2]https://youtu.be/7lwAKmkuUFs

Fig. 2. Hokuyo URG-04LX-UG01.

3) C920 HD PRO WEBCAM: C920 HD PRO WEBCAM is a RGB camera with the Full HD 1030p at 30 frames per second and clear detail color imformation. We use it to get the location of our target and determine the distance between the cannon and target according its size in the image.

Fig. 3. c920-pro-hd-webcam-refresh.

4) Cannon:

a) Components: Fig. 4 shows the cannon that we designed and made by ourselves. We use a servo motor to actuate the trigger and 2 DC motors (RS-540, 22000 rpm with 12V power supply) to accelerate the bullets. The DC motors are powered by 6 AA batteries (i.e. 9V) through a relay. Finally, the servo motor and the relay are controlled by an Arduino MEGA board. The body of the cannon is made of laser-cut acrylic board.

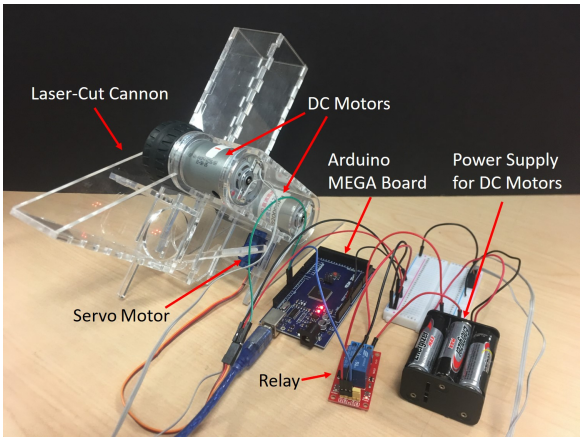

Fig. 4. Components of the cannon.

b) Design Concerns:

- Placement of DC Motors: For the placement of DC motors, most of the designs we have seen in previous works or commercial products are either on the top and bottom of the track, or on the left and right. To decide which design to use, we made a prototype shows in Fig. 5 to try different placement methods. In this preliminary study, we found that when the motor is placed on the top and bottom, the bullets are fired farther than on the left and right.

Fig. 5. Prototype of the cannon for testing different placement of DC motors.

- Tilt Angle: For the shooting angle, we want to design a mechanism such that we can control its tilt angle. However, due to lack of time and experience, we fix the tilt angle to 45 degrees, which is should provide maximum shooting range. Because of the high tilt angle and the high power supply, if the cannon is placed on the top deck of the robot, it may shoot towards the ceiling. So we put the cannon on the bottom deck of the robot.

c) Control: To control the cannon, we write an Arduino program that controls the switch of relay and the turning angle of the servo by command. To communicate with other nodes on ROS, we run a rosserial node to transmit command through serial port. When firing, the trigger pushes the bullet in the slot towards the DC motors, and the bullet is shoot out along the track. At the same time, a blocking rod on the trigger prevents the next ball from falling into the slot. After a second, the trigger retrieves and the ball falls into the slot. The slot fits only one ball so the next one can not get in. However, if there are too many balls in the storage, the ball in the slot may be pushed towards the DC motors by others balls. This can be fixed by redesign the storage.

B. Control Mode

There are two modes in the system, auto mode and manual mode. In auto mode, VR Tank will automatically search for the target and attack when found. In manual mode, user can receive its first-person view and control its movement.

We decide to choose ArUco marker as the target. An ArUco marker is a synthetic square marker composed by black border

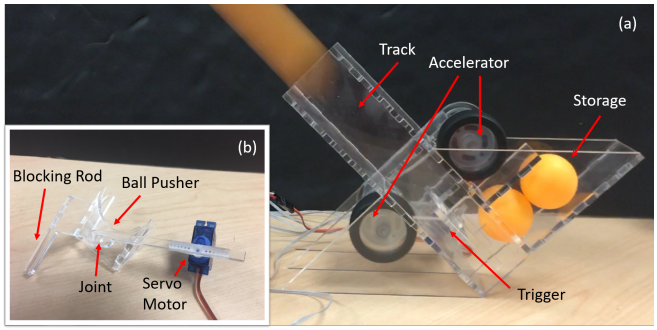

Fig. 6. (a) The structure of the cannon. (b) The structure of the trigger.

and an inner binary matrix which determines its identifier (id). The reason we choose ArUco marker is because the black border facilitates its fast detection in the image and the binary matrix allows its identification and the application of error detection and correction techniques. Therefore, we can detect and identify a target in real time. Moreover, a target can be found rotated. The size of our target is 6x6. Fig. 7 and Fig. 8 illustrates an ArUco marker of id 1 and id 3.

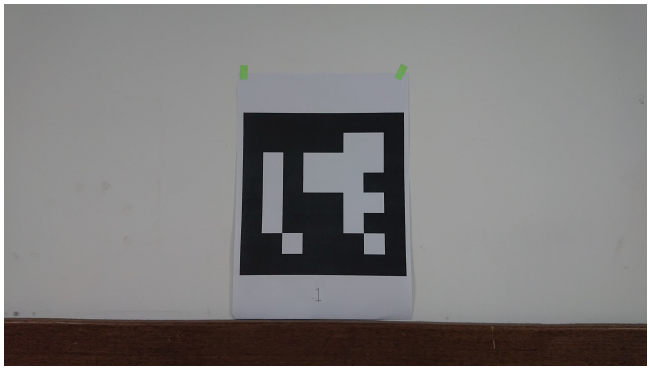

Fig. 7. ArUco marker id 1.

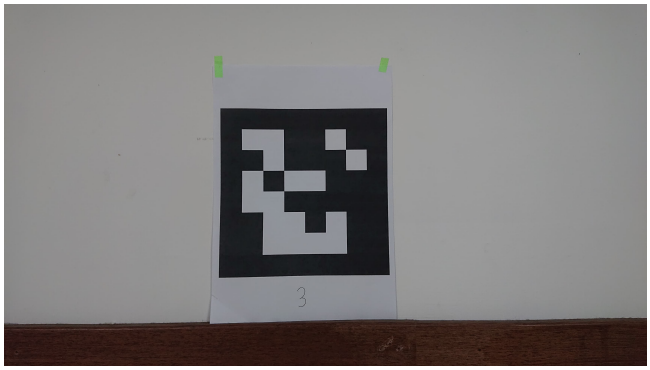

Fig. 8. ArUco marker id 3.

1) Auto Mode:

a) Mapping: For mapping, we use an open source called gmapping. The gmapping package provides laser-based SLAM(Simultaneous Localization and Mapping), as a ROS node called slam_gmapping. Gmapping is Filter-based approach which is the classical approach that performs prediction and update steps recursively. It stores the robot pose and the environment feature positions in one state vector and uses an error covariance matrix to store the uncertainties of these state estimates along with cross correlation terms between features and poses. With the help of slam_gmapping, we can manually control Pioneer 3-DX and create a 2-D grid map (like a building floorplan) from laser and pose data collected by Hokuyo URG-04LX-UG01. We build a map of National Taiwan University Ming-Da Hall 2F, which is shown in Fig. 9.

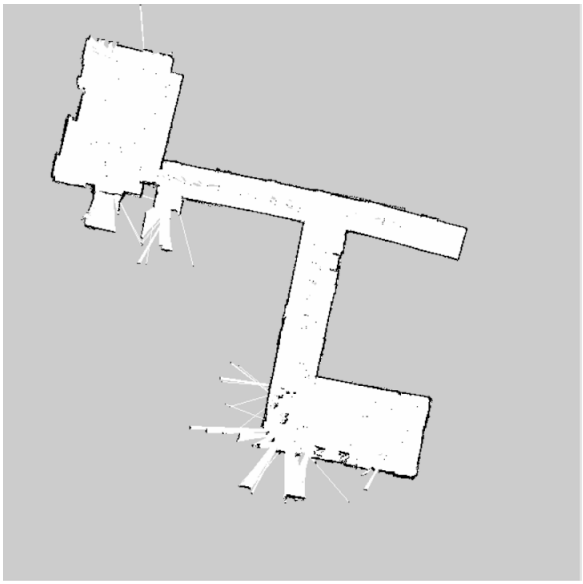

Fig. 9. Map of NTU Ming-Da Hall 2F.

b) Localization: After mapping, we need to get the exact location of our robot, that is, localization. Our method is AMCL(adaptive Monte Carlo localization). AMCL is an upgraded version of Monte Carlo localization, which uses a particle filter to track the pose of a robot on a known map. The idea is to spread some particles uniformly. We then manually move the robot until all particles converge to the environment.

c) Navigation: Now we have the location of robot in our test environment. The next step is to move the robot to the destination. We create ROS node called simple_navigation_goals and move_base to navigate and also implement obstacle avoidance. Moreover, we set some checkpoint, which is shown in Fig. 10, and measure the absolute coordinates. When the robot reached a checkpoint, it will start to rotate clockwise and search for target simultaneously.

d) Rotation: Now we arrive the coordinate which we set in navigation. The robot would start turning around until the camera get the target that we paste on the wall before. If target is not found, the robot will restart spinning until next target be found.

e) Aiming: The target is appear in the image of camera now. Generally speaking, the target would not locate in the middle of our image, which means our cannon front sight.

Fig. 10. Checkpoint.

For example, Fig. 11 shows that the target is appear in the left-center, so the robot would turn left slightly to aim.

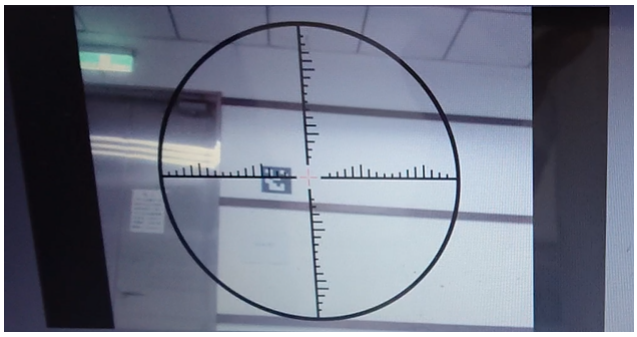

Fig. 11. Aiming at the target.

f) Moving: The target is located in the middle of image now. Limited by the power of our cannon, our cannonball could not shoot too far. So, the robot should move to the appropriate distance to shoot the target accurately. Because our target is in the same size, we calculate the size of our target in the image to get the appropriate distance to shoot. If the size of target in image is too small, means the distance between cannon and target is too long, the robot should move forward slightly to shoot accurately. In the similar reason, if the size of target in image is too large, means the distance between cannon and target is too short, the robot should move backward slightly to shoot accurately.

Fig. 12 illustrates the relation of each ROS node in auto mode.

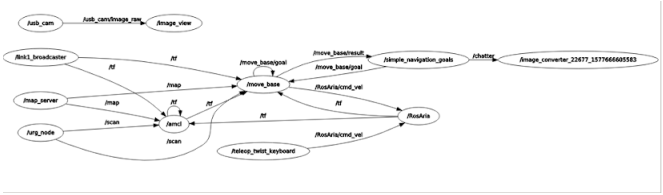

Fig. 12. ROS states of auto mode.

2) Manual Mode: In manual mode, we aim to control the robot to shot on the target like driving a real tank. According to Fig. 13, the first person vision displayed through the headset, and the remote control is done by joystick with Bluetooth.

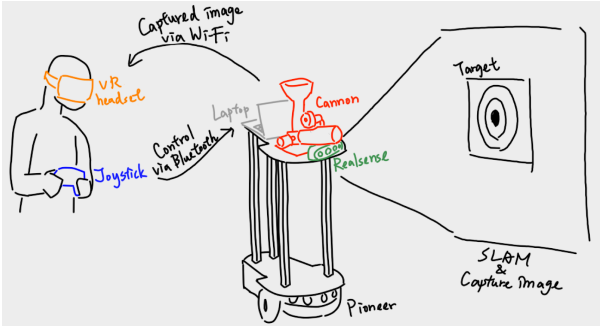

Fig. 13. Concept of the manual mode.

a) Virtual Reality: We first set up two webcams on the robot, streaming the video on the Internet, and then display the 3D-like view on the smart phone.

b) Joystick Control: Fig. 14 shows the concept of the remote control. We use the joystick, Xbox One s, to control the robot via the Bluetooth. Then set the signals of the joystick to trigger the cannon and control the robot base on the ROS environment Fig. 15. The robot is Pioneer 3-DX and the controlling system is RosAria.

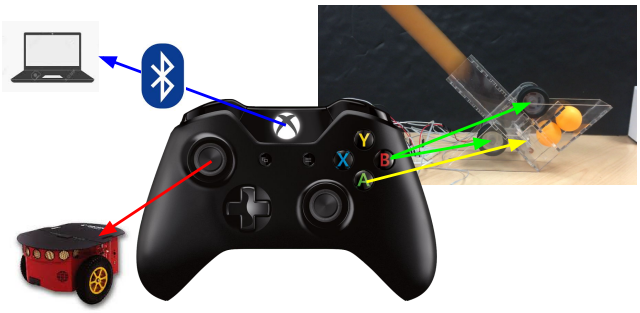

Fig. 14. Concept of the Joystick.

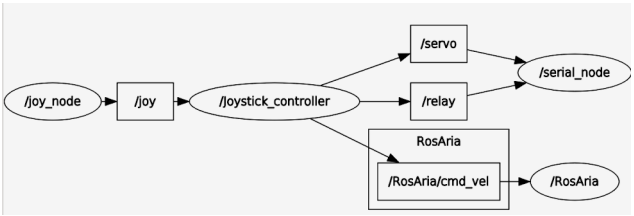

Fig. 15. ROS states of joystick control.

IV. Experimental Setup

This section shows how we build up the system of the VR tank and how we evaluate the performance.

The base of our robot is Pioneer 3-DX, with Hokuyo URG-04L equipped on it. Our homemade cannon is placed right

behind the laser range finder, and the webcam is placed about 50 cm above Pioneer 3-DX. Fig. 16 illustrates the structure of the robot.

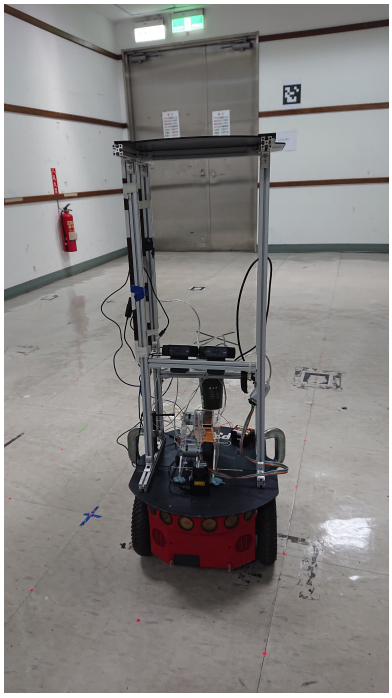

Fig. 16. VR Tank.

The ROS environment runs on a laptop. The webcams positioned on the robot connect to the laptop via USB port. The streaming video upload to Twitch[3]. The Joystick connect to the laptop via Bluetooth. And the cannon built on the Arduino connect to the laptop through USB port as well.

We evaluate our robot at Ming-Da Hall 2F, which is windless, so the bullets will not be affected by the environmental condition.

A. Auto Mode

To evaluate the performance of our robot in auto mode, we setup a scenario for it to go through. The robot is initially place at the corridor, and is asked to enter the room by reaching the checkpoint. When the robot reach a checkpoint, it will start to search for target. There are 2 target sheet stick on different walls. Once a target is destroyed, we covered the target so that the robot will start to search for another target. The robot is required to shoot at all the targets.

B. Manual Mode

To evaluate the performance of the manual mode, we trial and error by a real human. Make sure that the vision in headset display appropriately and the cannon could actually shoot at the target aimed by the cross hair.

[3]https://www.twitch.tv

V. Results

A. Cannon

In both evaluation of auto mode and manual mode, we can shoot around the target most of the time. The average displacement error in horizontal is small, while the average displacement error in height is large. This is due to following reasons:

- Unstable Accelerator: In the early design of the cannon, we have tried different DC motors to accelerate bullets. It is convenient and material-saving to laser cut shaft-replaceable wheels which can adapt to different motors. However, the wheel and the shaft are not tightly coupled, so the wheel is slightly off-center. This leads to different friction and contact time between wheels and bullets, so the initial velocity of the bullet is different, which results in different shoot height. This may be improved by replace the shaft-replaceable wheels with single-piece wheels, e.g. 3D printed wheels.

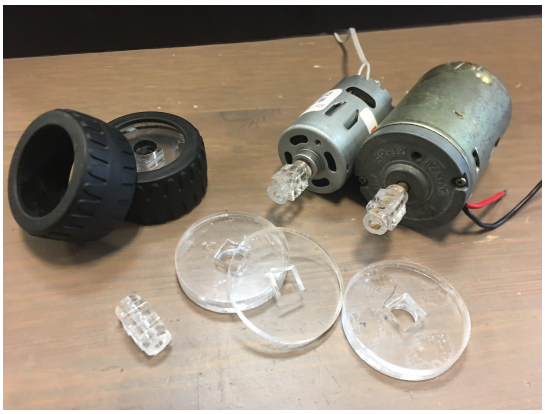

Fig. 17. Shaft-replaceable wheel.

- Unstable Power Supply: To drive the DC motors, we use 6 AA batteries for power supply. However, the power consumption of the motors is high. These disposable batteries can only run the motors at full speed for about 5 minutes. As the battery voltage decreases, the speed of the motor drops. To provide more stable and durable power supply, also not to waste so much money, we use rechargeable batteries instead. To better maintain the power supply, we may use voltage stabilizer.

Another problem of the cannon is that if there are many balls in the storage, the ball in the slot will be pushed towards the DC motors by others when loading. This will cause undesired firing or even jam the trigger. To solve this problem, we may have another mechanism to block the second ball in the storage when loading.

B. Auto Mode

We use laser to build the map and coordinate in the space. After that, we set the coordinate close to our targets and the robot will arrive here. When the robot arrive the coordinate that we set before, it start to spin and find target. After the

target appear in the image of camera, the robot will aim and move to the appropriate distance to shoot. After shooting, the target would be gone and the robot would start to find next target to shoot.

C. Manual Mode

We successfully control the robot like driving a real tank through the manual mode setup. Both the VR view in headset and the control of joystick work well. Nevertheless, due to the streaming restriction right now, the vision of the webcams lag for five seconds as displaying on the smart phone.

VI. Conclusion

Motivated by the growing of the virtual reality, we implemented a device called VR tank. This invention let us control the tank remotely and safely, moreover, the driver has no risk on his or her lives while training or attacking. In addition to manual control, we also build up the auto control mode for the possibility of accidental disconnection. Although we have not achieved the real-time 3D view from tank to the driver due to the streaming restriction, this problem should be solved after the 5G revolution.

References

[1] PABR Technology, "First-person view and remote control with LEGO Mindstorms EV3," January 5, 2015, [Online] Available: http://www.pabr.org/bricks/brickfpv/brickfpv.en.html.

[2] Yu-Xi Huang, Yuan-Ze Zhang, Guan-Xiong Wang, Wei-Huai Qiu, Qing-Long Gao, "Table tennis robot," NTUSTR 2015, [Online] Available: http://ir.lib.ntust.edu.tw/handle/987654321/49488.

[3] YA.Thamemul Ansari, K.Thivakaran, M.Hareesh Kumar, R.Gopalakrishnan "Design of a smart ping pong robot," IJIRST, March 2015, [Online] Available: http://www.ijirst.org/articles/IJIRSTV1I10008.pdf.

[4] Cody Hsu "Arduino-based homemade table tennis robot," Youtube, June 9, 2019, [Online] Available: https://www.youtube.com/watch?v=IIPMFiulIzE.

VII. Division of Work

- Bo-Hsun Huang (R07922015): Auto mode
- Ta-Yo Mu (R08922132): Auto Mode
- PoYu Wu (R08922A13): Manual Mode
- Chih-An Tsao (R08922070): Cannon