1. ABSTRACT **(Andria)**
2. INTRODUCTION **(Andria)**
3. MECHANICAL SYSTEMS **(Priyanka)**
   1. Chassis and Component Mounts
   2. Trapdoor Mechanism
   3. Dispensing Mechanism
   4. Container
4. ELECTRONICS AND POWER SYSTEMS
   1. Microcontroller **(De-Wei)**

We used Arduino Mega 2560 as our controller on the robot because there are lots of pins on it to accommodate the pixy cam, three DC motors, a servo motor, two IR sensors, and infrared emitter. For the container, we use Arduino Uno as the controller due to just one DC motor and an infrared receiver.

* 1. Actuators **(De-Wei)**
     1. *Drive Motors*

For the drive system, two 12V DC motors with encoders were used. These motor have no load speed of 251 rpm and 43.7:1 metal gearbox, which make it rotate fast and powerful. With L298N motor drive board, these motors can be easily controlled.

* + 1. *Dispensing and Container Lead Screw Motor*

For the lead screw driving motor, we used a 12V DC worm geared motor, which has huge torque to push heavy tennis balls. The only problem of this motor is that the rotational speed is not enough. Therefore, we couldn’t push the balls fast enough to have sufficient momentum to roll into the container by itself. As a result, we had no choice but to change the way we dispense the tennis balls.

* + 1. *Trapdoor Motor*

In order to easily control the door to be the specified positions, a servo motor was used to move the trapdoor. Moreover, there was no huge load on the door. Therefore, a servo motor was our best option.

* 1. Sensors **(Tong)**
     1. *Ball Detection*

The ball detection mechanism is the core sensing part of our robot. Basically, we use the Pixy camera detection system to complete this part. First, we need to find the balls. To do this, we made our robot rotate itself if no ball was detected. And it also moved straight for a short distance in every 3 seconds to search in a different location. If a ball was detected, we need to detect the location of the ball as well as the distance between the ball and the vehicle. First, we need to know the relative position of the ball to navigate our vehicle. At very beginning, we used pan servo to rotate pixy to keep the ball in the center of the pixy camera and read the rotating angle of pan servo to compensate the move direction of our vehicle. In this way, we could successfully collect the ball but it was not fast enough and the precision and stability was not easy to control. So, we removed the pan servo and install the camera directly to our vehicle and read the relative position of the ball blocks in the image to compensate the moving direction of our vehicle. In this way, we could move our vehicle efficiently to the balls.

If the distance between the ball and the vehicle is small enough, we need to get into the collection state. First, we use the ball’s block size in the image to determine this distance. It turned out that the algorithm of the block size is unreliable in some corner cases. And then we found that y coordinate of the ball block in the image could also represent this distance. So, we changed our method and let the robot get into collection state after the ball block position in the image was low enough. In this way, we could make sure the vehicle would get into collection state every time at certain distance in front of the ball.

* + 1. *Lead screw switches*

To prevent the lead screw from moving beyond the limit to damage other components, we added to two switches in both side of the lead screw to regulate its moving range. Once the moving part touched the switch, it would stop or move to opposite direction immediately.

* + 1. *Container Detection*

It was easy to detect the container but the critical part was how to keep the alignment of the vehicle and the container to make sure the ball was dispensed in the right direction. We tested different method like the color code or mechanical structure guidance. But these method was not robust enough and couldn’t perfectly work in different situations. And then, we thought out of box and decided to change the shape of our container. We made a rectangle container and used a big color bar as a symbol. In this way, we didn’t need to worry about the vehicle direction in common cases. Our vehicle stopped in front of the container and got into dispensing state.

* + 1. *Ultrasonic detection*

In order to prevent our vehicle from hitting the wall, we introduced two ultrasonic sensor to our robot. In the searching mode, if any ultrasonic detected something, we could make some changes based on this information. We also developed an FSM for this to make sure our robot could go to the right way.

*Task Completion - Container Door Close*

For the door open and close, we used infrared signal as remote control to operate the container door. We installed three infrared emitter in the front and both sides of our robot and placed one receiver at the center of the container door. In this way, we could use our robot to open and close the container door.

* 1. Power System

1. SOFTWARE **(Ze An)**
   1. Find Ball State
   2. Move to Ball State
   3. Obtain Ball State
   4. Dispense Ball State
   5. Container Close State
2. RESULTS
   1. Experiments and Testing
      1. *Drive Motor Torque Testing* ***(Ze)***
      2. *Ball Detection Testing* ***(De Wei)***

At the beginning, we used Pixy with pan and tilt servos mounted so that it can move as eyes on human's neck in order to get more vision. We first used the center point of the pixy camera to minus the x and y position of the block of the object to get the error, and use it to control the movement of the servo motors. After that, we took the angles the servos move from center as the error. With this error, we then built the differential speed for each sides of the motors to track the object. However, with feedback loop, it increased the difficulties to control the robot. As a result, we could successfully track tennis balls, but we failed to align with the balls perfectly. In other words, sometimes we might stop next to the ball, rather than in front of the ball, which made us unable to collect the ball. Therefore, we then tried to simplified the design and got rid of the pan and tilt servos, which sacrificed the broader range of vision. Without the servos for pan and tilt, we directly used the x axis error of pixy to build the differential speed for each motors. In this method, our robot would rotate without forward speed to search for the ball. Once aligned, it would go straight and collect it.

* + 1. *Ball Collection Performance Testing* ***(De Wei)***

For the collection, we initially used size as the threshold to tell Arduino when to go into the collect state. However, after several tests, we found out that it was not stable as the size might vary with the light condition. Thus, we modified our method and used y position of the object as the threshold. We first used the serial monitor to find out the desired value as the robot approaching the tennis ball. After getting the desired value, we put it into the program and tested.

* + 1. *Ball Dispensing Performance Testing* ***(Andria)***

The initial concept was to dispense the balls was to use a push plate driven by a lead screw mechanism. The plate however did not push the balls paste the plain of the vehicle. There were concerns that

* + 1. *Container Door Performance Testing* ***(Priyanka)***
  1. Budget **(Andria)**
  2. Competition Performance **(Andria)**

1. DISCUSSION **(Andria)**
2. CONCLUSIONS **(Andria)**
3. REFERENCES
4. APPENDIX