

DOT. 2022 Team Description Paper

I. Ustinov, B. Viktorov, A. Sviridov, Presidential Lyceum of Physics and Mathematics #239



Figure 1. v2022 model (left) and v2021 model (right)

Abstract — Our team DOT. is passionate about robotics. Robotic soccer is a very interesting and complex area with its unique problems and challenges. So our team participates in RoboCupJunior Soccer since 2018.

This year we developed new version of robots and software using all our experience in mechanics, electronics, programming and robotics. In this Team Description Paper we briefly describe all main aspects of the development process.

I. MAIN CHANGES FROM 2021

Due to changes in rules for 2022 our team had to rebuild robots and remove some main features from the 2021 version (Fig. 1, right). Previous version of our robots was bigger, heavier and had two dribblers placed on opposite sides of the robot with a multi-directional kicker. This year it was impossible to fit all these complex systems into the smaller body, so v2022 robots (Fig. 1, left) have only one dribbler and simplified kicker system. Also, because of big data transferring delays, we decided to remove the single board computer Khadas VIM3 and use OpenMV camera for machine vision.

This year we spent a lot of time developing and debugging advanced gameplay tactics to minimize time traveling to the ball and grabbing it from every direction and maximize our chances of scoring a goal after that.

II. CONSTRUCTION

Robot is based on an omni-directional platform with 4 drive units. Drive units mounts between aluminum levels. This assembly creates a simple but rigid frame. All other components mount to it. For better outlook and protection of internal components we also made a plastic body on a 3D printer.

I. Ustinov - captain, mechanical engineer, electronics engineer, Presidential Lyceum of Physics and Mathematics #239, (e-mail: ilust.d@gmail.com).

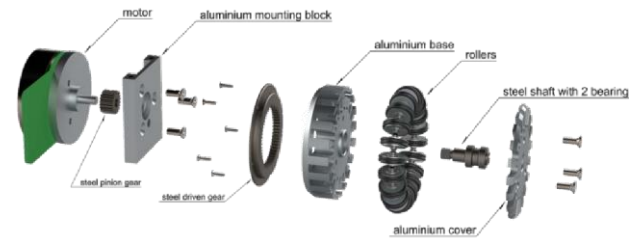


Figure 2. Drive unit exploded view

A. Drive units (Fig. 2)

Self-developed drive units greatly improve the quality and precision of robots moves. Each drive unit consists of an omni-wheel, bearing module, gearbox and Maxon flat EC45 motor with encoder 4620004+200142.

B. Dribbler (Fig. 3)

For ball dribbling we made a special dribbler. It spins the ball with two silicone rollers. We use two separate rollers instead of one, because based on our experience this configuration works better at high angular accelerations. Rollers are driven by Maxon 283856 motor through gears. This year we decided to move the pivot point of the dribbler as low as possible. We thought it would reduce ball bouncing, but it seems to have no direct effect on it.

C. Kicker

Ball kicking is achieved by 2 solenoids placed with a 30-degree angle to the forward direction of the robot. Thus for straight kick both solenoids are used simultaneously and for sideways kick only one solenoid can be used.

D. Vision block (Fig. 5)

Vision block is placed on top of the robot. It contains OpenMV camera pointed at conical mirror, transparent tube and LIDAR. Mirror is placed as high as possible to get a better view of the field. Mirror is made and polished manually on a



Figure 3. Dribbler

B. Viktorov - programmer, Presidential Lyceum of Physics and Mathematics #239, (e-mail: boris@victorov.com).

A. Sviridov - electronics engineer, Presidential Lyceum of Physics and Mathematics #239, (e-mail: an.sviridov.spb@gmail.com).

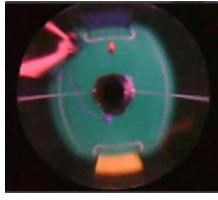


Figure 4. Image from camera Figure 5. Vision block

small lathe. Its shape is very similar to a cone with rounded tip, but due to the manual manufacturing process there might be some differences. Example of image from robot's camera is shown on Fig. 4.

III. ELECTRONICS

All electronics inside robots are placed on self-developed PCBs. There are three main PCBs: Motherboard, Voltage boost module and Line sensors board (Fig. 6).

A. Motherboard

Motherboard includes the main microcontroller (STM32F407), motor drivers, power switch, voltage stabilizers, inertial measurement unit MPU9250, Bluetooth module HC-05, OLED 0.96-inch screen, buttons and connectors for all peripheral units. For communication with OpenMV camera and Line sensors board we use SPI interface.

B. Line sensors board

Line sensors board has its own microcontroller (STM32F103). It reads 30 infrared reflective sensors TCRT5000 through two ADG706BRUZ multiplexers. Data is transferred to Motherboard through the SPI interface.

C. Voltage boost module

Voltage boost module charges two capacitors 680uF each up to 150V for kicking the ball. Boost module itself is made on the LT3751 chip. Our team spent a lot of time finding good voltage boosts schematics, and LT3751 fits all our requirements.

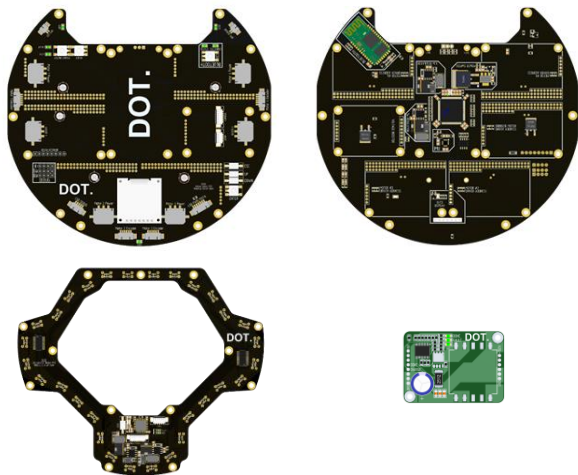


Figure 6. Motherboard (top), Line sensors board (bottom left), Voltage booster (bottom right)



Figure 7. Battery

D. Battery (Fig. 7)

For powering robots we use a self-made battery. It consists of a frame, 4 18650 LG HG2 Li-ion cells and PCB with power connector. For fast replacement we use special battery connectors. Nominal voltage is 14.4V. Battery inserts in the middle of the robot and locks in with built in spring loaded locks.

IV. SOFTWARE

A. Navigation

Robot's navigation is done in global coordinates. Coordinates are obtained by an OpenMV camera pointing vertically upwards into a conical mirror. Special shape of the mirror allows the camera to see all objects at the entire field. Robots find goals and the ball using machine vision algorithms written in MicroPython. Then all distances are linearized and all necessary coordinates are calculated. From this data the robot obtains its own coordinates on the field that are used to make precise moves.

In the future, LIDAR can be used for navigation in addition to the camera. Lidar will find the location of the walls that limit the field, which will allow us to find other objects on the field, such as enemy robots. This will give new opportunities when implementing various strategies for attack and defense. For example, a goalkeeper can find the position of the attacking opponent in cases when it hides the ball from our view.

B. Program architecture

The low and high-level code is implemented on STM32 controllers and written in C++ using the CMSIS standard. The Keil uVision 5 development environment is used to write the program and debug robots.

Gameplay tactics are shown at the diagrams at appendix.

C. Ball following

We wanted to make ball following and approach as quick and precise as possible. So we selected and tested two different algorithms:

- Vector field (Fig. 8). For this algorithm a special pregenerated vector field is used. It is written to the controller's memory. This field is an array that matches a vector to a point with integer coordinates. One of the main advantages of using a vector field is the ability to adjust the trajectory of movement to bend around some points (for example, enemy robots). The trajectory is corrected by adding a small vector field of the detour point to the main vector field. Thus, by

adding vector fields, it is possible to obtain the necessary trajectory for the robot. However, the vector field has some disadvantages due to the fact that the vector field takes up a lot of space in the controller's memory, and besides, to change vector field, we need to recalculate it, which excludes the possibility of quickly changing the robot's motion algorithm.

- Parabola (Fig. 9). The parabola is a very convenient trajectory to approach the ball, since it is very close to the shortest trajectory and also provides smooth movement. In comparison to vector fields this method uses much less memory. At the current code version we use this algorithm.

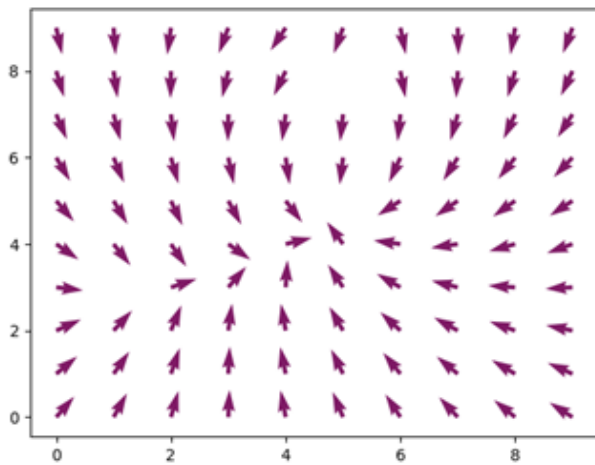


Figure 8. Vector field

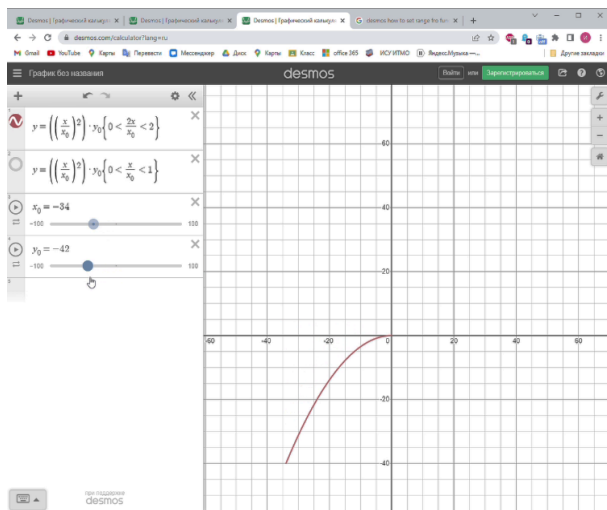


Figure 9. Parabola trajectory

Appendix

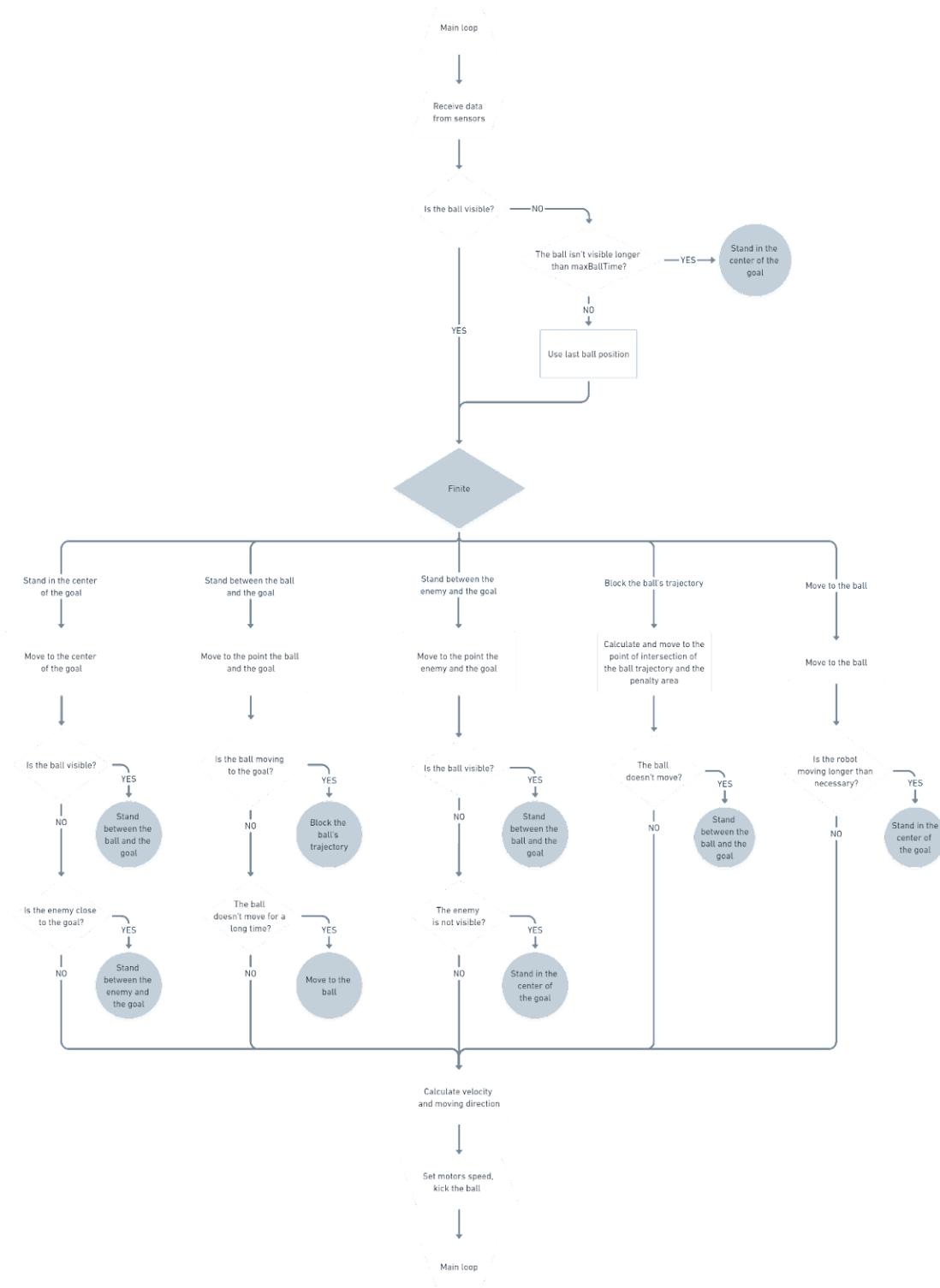


Figure 10. Goalkeeper tactic diagram

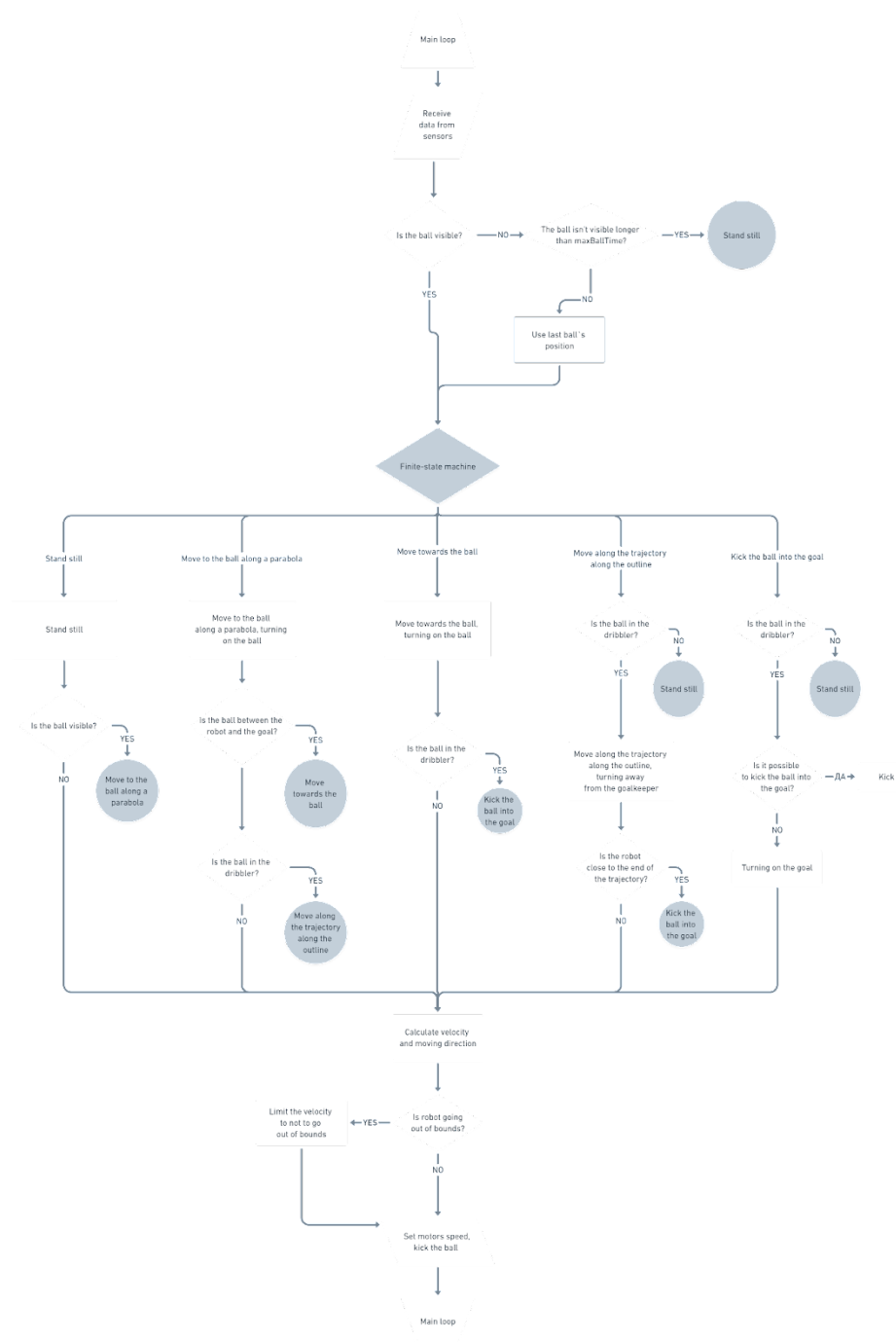


Figure 11. Striker tactic diagram