OPEN ACCESS
The Journal of Engineering and Exact Sciences – jCEC, Vol. 09 N. 10 (2023)
journal homepage: https://periodicos.ufv.br/jcec
eISSN: 2527-1075
ISSN: 2446-9416

Initial development of IEEE Very Small Size Soccer robots: Electromechanics, Computer Vision System and Navigation Strategy

Desenvolvimento inicial para robôs da IEEE Very Small Size Soccer: Eletromecânica, Sistema de Visão Computacional e Estratégia de Navegação

Article Info:
Article history: Received 2023-11-01 / Accepted 2023-12-15 / Available online 2023-12-30
doi: 10.18540/jcecvl9iss10pp18148

Ryan Henrique Xavier Ferreira
ORCID: https://orcid.org/0009-0009-3273-2699
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Ibirité, Brasil
E-mail: ryan.h.xavier@gmail.com
Pedro Lucas da Silva
ORCID: https://orcid.org/0009-0006-9040-8238
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Ibirité, Brasil
E-mail: plucassilva4@gmail.com
Vinícius Caputo Resende de Oliveira
ORCID: https://orcid.org/0009-0004-1416-0141
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Ibirité, Brasil
E-mail: viniciuscaputoresende@outlook.com
Carlos Dias da Silva Junior
ORCID: https://orcid.org/0009-0008-9564-2706
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Ibirité, Brasil
E-mail: carlos.dias@ifmg.edu.br
Elias José de Rezende Freitas
ORCID: https://orcid.org/0000-0002-6986-0602
Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - Campus Ibirité, Brasil
E-mail: elias.freitas@ifmg.edu.br

Resumo
Este artigo apresenta o desenvolvimento inicial dos robôs da categoria IEEE Very Small Size Soccer (VSSS), incluindo a modelagem e impressão 3D de componentes mecânicos, a integração de uma eletrônica embarcada, um sistema de visão computacional e a simulação de uma estratégia de navegação. O sistema de visão computacional proposto é capaz de capturar e processar imagens durante a partida para fornecer as posições e velocidades dos robôs e da bola. A partir desses dados de odometria, a estratégia de navegação proposta, baseada em campos vetoriais, calcula comandos para os robôs, permitindo diferentes funções de robô na partida de futebol (atacante, defensor e goleiro). Essa estratégia, implementada em um simulador, sendo intuitiva e de baixo custo computacional.
Palavras-chave: Futebol de robôs. VSSS. Visão computacional. Campo vetorial.

Abstract
This paper presents the initial development of the IEEE Very Small Size Soccer (VSSS) robots, including the modeling and 3D printing of mechanical components, the integration of embedded electronics, a computer vision system, and the simulation of a navigation strategy. The proposed

This article is an extended English version of (FERREIRA *et al*., 2023) and (OLIVEIRA *et al*., 2022), presented in the III and IV CoBICET (Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia).

computer vision system can capture and process images during the VSSS match to provide the robots' and the ball's positions and velocities. From these odometry data, the proposed navigation strategy based on vector fields calculates commands to the robots, allowing different robot roles in the soccer match (striker, defender, and goalkeeper). This strategy, implemented in a simulator, is intuitive and has a low computational cost.
Keywords: Robot soccer. VSSS. Computer vision. Vector field.

1. Introduction

IEEE Very Small Size Soccer (VSSS) is a competition where two teams of three or five robots face each other to play a game similar to soccer but with their own rules. A computer controls these robots remotely, with no human interference while the match occurs (IEEE VSSS, 2023). Figure 1 illustrates a match's environment, showing the camera position above the pitch for each team. By capturing images of the match, a computer is responsible for processing the data and sending the necessary commands to each of the team's robots.

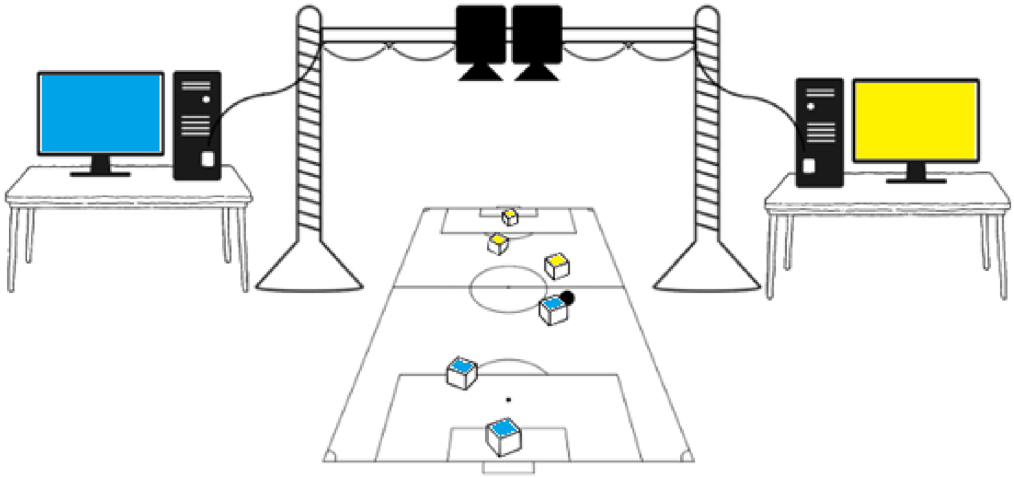


Figure 1 - Diagram of a match in the VSSS category.

The robots used in this category are differential robots, which are powered by two parallel wheels that, based on the difference in speed between them, make it possible to change the robot's linear and angular speed and its orientation in the field. For example, when the wheels' speeds are equal but in opposite directions, the robot would rotate around its axis and not move. However, if the speeds and directions are equal, the robot will follow a straight line, considering an ideal model (SIEGWART, R. *et al*, 2004).

Each robot has an embedded processor. Some works use FPGA (Field Programmable Gate Array) with parallel processing, as shown in (ÁGUAS, 2018). However, most robot projects in this category use Arduino, usually with an Xbee communication module (PINTO, 2017).

The computer vision system captures an image, which is then processed by a computer. The robots then receive the necessary calculations to execute their best strategy during the match. With this information, the robots navigate the field autonomously, calculating the best movement to reach their goal at each moment without human intervention. OpenCV, a popular library, is used for this processing. It can extract information from images, such as the location of objects like robots and balls. The library allows the use of the HSV color system, which has three measures of color perception, namely hue, saturation, and value, as shown in Figure 2. By independently controlling these measures, color detection algorithms can be fine-tuned to work effectively under different lighting conditions.

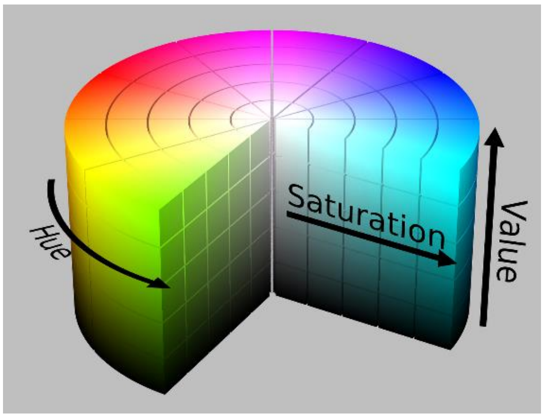


Figure 2 - HSV chromatic disk (Wikipédia, 2015).

Several studies have been carried out in the field of computer vision for robot soccer. For example, in the work of Cunha (2019), the OpenCV library was used to obtain the positions and velocities of the robots and the ball in the VSSS game using the HSV color system. Differently, Rosa (2015) used an algorithm with OpenCV functions to detect objects in the RGB color standard. Some studies in this area have aimed at finding ways to increase the effectiveness of the computer vision system. For example, in his work, Hofstatter (2016) used the extended Kalman filter technique to achieve greater accuracy in the computer vision system. The author used a more refined version of the filter to transform the kinematic model of robot soccer into a linear model. The result of this application was that the vision system captured positions much more consistent with the real positions of objects on the field.

Another example to mention is the work of Bianchi and Reali-Costa (2000), which highlights the system for calibrating the position of objects using color. The system implemented by the authors has a good response speed, making it possible to complete the calibration within the time required by the regulations. In addition, the way it is done is simple: just by clicking with the mouse on the objects of interest, identifying the colors of each one, even with the variations in brightness during the match.

Therefore, based on the position, orientation and speed information of the robots and the ball on the field, it is necessary to define a navigation strategy. One of the most widely used methods is based on Vector fields because these methods are an intuitive and flexible representation and have a low computational cost (DA ROSA AND ROSA, 2021), (DOS SANTOS *et al*., 2019) (LIM, Y. *et al*., 2008). Recently, there has been a search for solutions using Artificial Intelligence, as proposed by Lugli *et al*. (2019) and Freitas *et al*. (2023).

In this context, this paper presents the initial development of soccer robots, from the modeling and 3D printing of the team's mechanical model through the development of the embedded electronics to the simulation of a navigation strategy based on Vector fields. In addition, we present the initial implementation of a computer vision system for the VSSS category, starting from the treatments needed to correct the captured images to object recognition.

2. Methodology

The methodology can be summarized in the following stages: (2.1) Mechanical design of the prototype robots; (2.2) Embedded electronics, which deals with the electronic components used in the robots; (2.3) Computer vision system, which involves recognizing objects in the game, and (2.4) Navigation strategy.

2.1 Mechanical design

Computer-Aided Design (CAD) software was used to create the 3D design of the parts that make up the robot and to simulate its assembly. The software chosen was Onshape, which has a free license available to students and educators. Its features make it possible to develop complex 3D shapes by applying parametric design techniques, using constraints and geometric relationships in the parts. In addition, Onshape can be used through a web browser, eliminating the need to install it on the user's computer.

The actual component sizes were measured to accommodate all the necessary electronic parts inside the robot and digital copies were created in Onshape. This made it possible to define the structural requirements of the project, such as the maximum size of the robots, the diameter of the wheels, the ball guide attachment, and the assembly method.

After several assembly simulations and corrective modifications, a virtual prototype of the robot's mechanical part was completed. Figure 3 illustrates the structure of the robots after the simulations carried out in Onshape. It is also important to note that priority was given to avoiding corners at the junction of surfaces, as these concentrate stress, thus reducing the strength of the parts (HIBBELER, 2016).

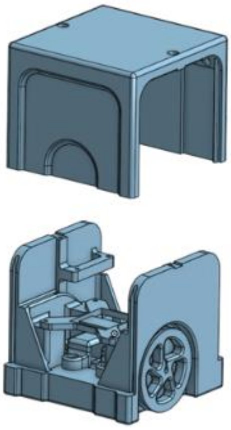


Figure 3 - Robot structure in 3D from Onshape simulation.

We chose to print all components in a 3D printer, as it allows for rapid and cost-effective testing and iteration of designs. For that, we use Creality's slicing software to prepare the file for printing, setting essential parameters such as nozzle and table temperature. Slicing is transforming the digital model into Computerized Numerical Control (CNC) instructions. The material chosen for printing was Polylactic Acid (PLA), a thermoplastic monomer derived from renewable, organic sources such as corn starch or sugar cane. The nozzle extrusion temperature was 200ºC to 210ºC.

2.2 Embedded electronics

The electronic part of each differential robot consists of one battery, two DC motors, one H-bridge module and one ESP32 microcontrolled platform. The batteries used in the soccer robots are two-cell LiPo batteries, totaling 7.4V. The DC motors used have an operating voltage between 3V and 9V, a maximum torque of 0.86 kgf.cm and an angular speed of 500 rpm.

An H-bridge was used to drive the motors. The working principle of this circuit is shown in Figure 4, which shows four electronic switch representations. Note that three different drive situations can occur: By activating S1 and S4, the current flows in one direction. Activating S2 and S3 causes the current to flow in the opposite direction, and activating S1 and S2 or S3 and S4 causes nothing to happen, allowing you to control the direction of rotation of the motor.

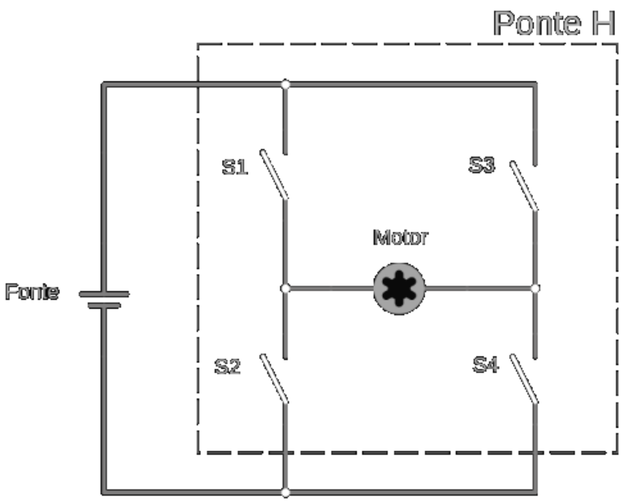


Figure 4 - H-bridge representation driving a DC motor.

It is worth noting that since the speed of the motor is proportional to the voltage applied to its terminals, it can be controlled by the time intervals in which the switches are open/closed. In addition, the switches can be transistors that operate in the cutoff and saturation regions, acting similarly to an open or closed switch. In either case, a control circuit is required to operate the switches. For the robots under development, the TB6612FNG H-bridge module was used.

The microcontroller selected for the project was the ESP32, which has built-in Bluetooth and Wi-Fi, eliminating the need to purchase additional modules to connect to the internet, as it happens with the Arduino. The ESP32 has a supply voltage of up to 3.3V and consumes/supplies a current of up to 80mA. It has 30 programmable input and output pins, which can be seen in Figure 5, with various functions, including pins with analog converters, capacitive touch sensors, I2C and serial connections.

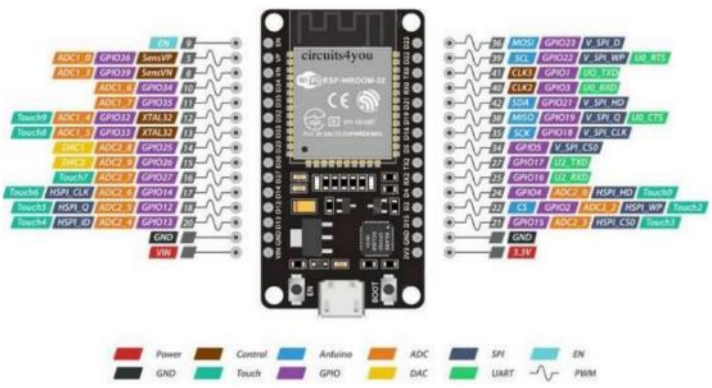


Figure 5 - ESP32 pinout (Circuits4you, 2018).

2.3 Computer Vision System

We use a Logitech camera with a resolution of up to 4K, 90 FPS (frames per second), and a 90º field of view to capture images from the soccer field. However, it is worth noting that, following the steps of Rosa (2015), the resolution chosen for the camera was 720p to reduce the processing time of the program and save storage space. Thus, the images were collected in real time and corrections were made to each frame to prepare them for later analysis using a code prepared in the Python programming language. According to Fortes (2013), preprocessing techniques are applied to improve the quality of images and specific parts, depending on the system's needs.

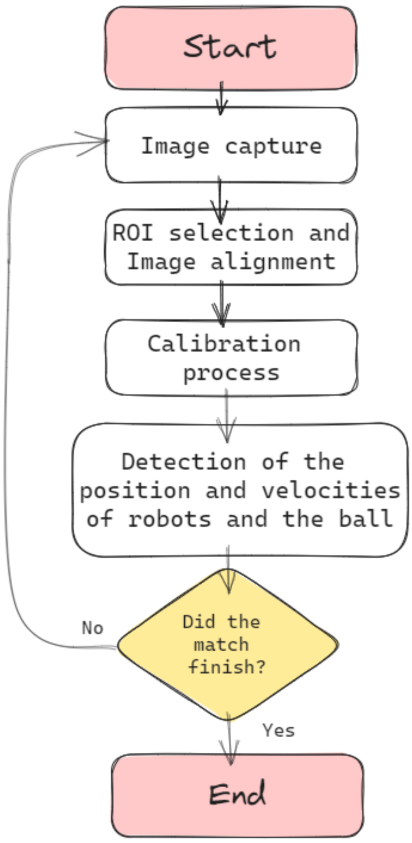


Figure 6 - Flowchart for detecting the position and velocities of robots and the ball.

The steps for detecting the position and velocities of robots and the ball can be summarized in Figure 6 and explained as follows:

- Image capture: a 720p image is captured by the camera.
- ROI (Region of interest) selection and Image alignment: It is typical for the original image obtained to capture areas beyond the soccer field, making it necessary to carry out the segmentation process. According to Fortes (2013), segmenting an image boils down to separating only its desired parts. In a computer vision system, this stage is considered critical, since it is precisely the regions of interest that will be worked on during image processing. Removing these areas helps to reduce noise when detecting objects, increase the speed of analysis by reducing the size of the analyzed area, and improve visualization and understanding of the desired location. At this stage, the corners of the soccer field are selected. Then, a geometric transformation is carried out to obtain an orthogonal view of the field, in other words, a plane perpendicular to its main axis.
- Calibration process: According to Klaser (2014), optical distortions from cameras negatively influence the process of image perception and, when not corrected, are a significant source of noise. According to Osowsky *et al*. (2006), images that suffer from optical distortions do not have their information lost. To correct the effects of distortion, according to the developers of OpenCV, it is necessary to provide some samples with a well-defined pattern, for example, images of a chessboard, in which the corners of the board form right angles, which serve as a reference for making corrections (OpenCV, 2023). Figure 7 shows the chessboard being used as a sample.

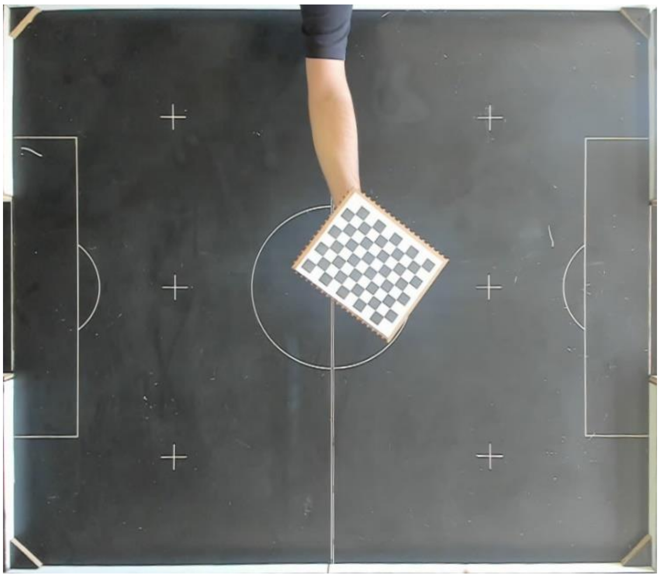


Figure 7 - Sample chessboard used in the calibration process.

To obtain information in real measurements, such as length (m), speed (m/s), angle (°), distance (m), and the position of objects (m), it is necessary to perform a calibration procedure based on known positions in the image. For example, Figure 8 shows the six markers of known positions on the soccer field. An average of the values found for each distance between the points was calculated for greater accuracy.

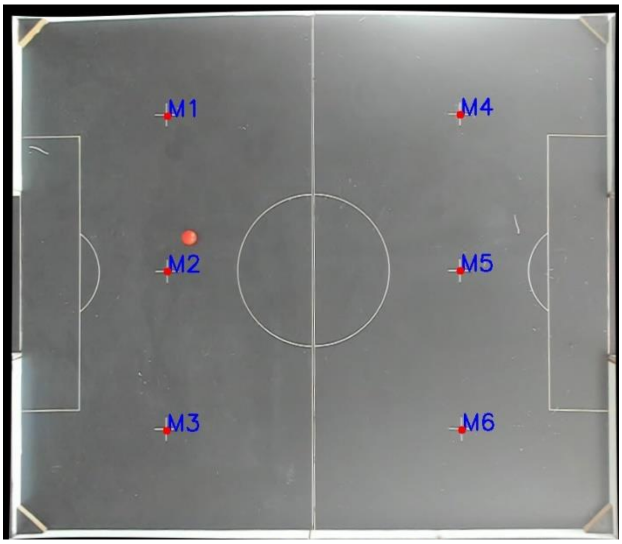


Figure 8 - Markings for measurement calibration.

The last calibration process used is to define the point of origin. This step is important so that, just like on a Cartesian plane, it is possible to identify the real position of objects using x and y coordinates. Figure 9 shows the bottom left corner of the field being defined as the point of origin (x = 0m, y = 0m).

Figure 9 - Defining the origin point.

- Detection process: After the preprocessing, the image could be analyzed to identify the odometry parameters for controlling the robots. For example, to detect the ball, a filter consisting of minimum and maximum values was used for each measure of the HSV chromaticity diagram. In addition, a range of area values was specified as a second filter to reduce detection noise. Implementing both filters, only pixels within the defined range of the chromaticity diagram and the area whose size corresponds to the defined range were detected. In this way, it is possible to obtain the approximate outline of the sphere, which is indicated by a green circle in the images. Then, based on the frame rate and the central position of the ball, it is possible to obtain the speed of the ball. Similarly, the process is done to obtain the information of all the robots in the soccer field.

2.4 Navigation strategy

The robot's navigation strategy was implemented in Python. The simulator used in the tests was FIRASim (FIRASim, 2022), developed with a focus on this game category and shown in Figure 10. The 3D virtual environment contains a communication interface with external systems that provide the positions of the ball and the robots and their speeds and orientation angles. A library provided by the ThundeRatz team (https://github.com/ThundeRatz /VSSSProtoComm/tree/main/lib) at the Polytechnic School of the University of São Paulo (EPUSP) was used to integrate the simulator and the algorithm.

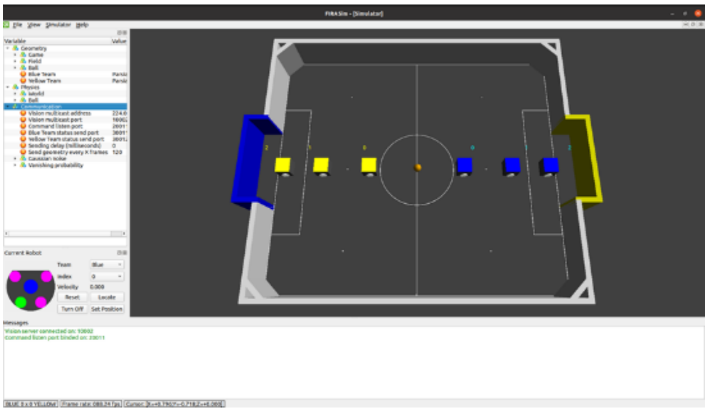


Figure 10 - FIRASim interface.

For a better understanding of the variables involved in the kinematics of the differential robot, they are shown schematically in Figure 11, where R is the radius of the wheel, L is the distance between the wheels, xr and yr are the positions of the center of the robot about the plane of the field, θr is the angular position of the robot, Vr is the linear velocity, and ω is the angular velocity. These variables must be manipulated to find the angular velocities that must be applied to the left wheel (ω_L) and the right wheel (ω_R) to control the direction in which the robot will move.

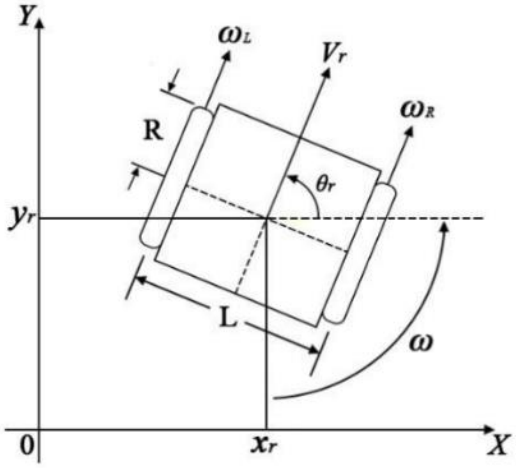


Figure 11 - Differential robot schematic.

Considering the field's inertial referential, the robot's position vector $\vec{p}_r$ can be defined in Equation 1.

$$\vec{p}_r = \begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{1}$$

and the position vector of the ball $\vec{p}_b$ in Equation 2.

$$\vec{p}_b = \begin{bmatrix} x_b \\ y_b \end{bmatrix} \tag{2}$$

At each instant of time, this information, together with the orientation (θ_r) of each robot, is provided by the computer vision system. In this work, it is provided directly by the simulator.

The kinematic model of the differential robot is given by Equation 3.

$$\begin{bmatrix} v_x \\ v_y \\ \dot{\theta}_r \end{bmatrix} - \begin{vmatrix} cos\theta_r & 0 \\ sen\theta_r & 0 \\ 0 & 1 \end{vmatrix} \cdot \begin{bmatrix} V_r \\ \omega \end{bmatrix} \tag{3}$$

A common way to control the robot is through the Feedback Linearization technique, as Grepl (2008) indicates and applied in our previous work for a semiautonomous robot (PEREIRA & Freitas, 2020). To do this, a coordinate transformation is performed considering a control point displaced from the center by d_r cm, representing the distance from the center of the robot to a point in front of it.

Then, our navigation strategy is based on the previous work from (SANTOS *et al*., 2019). This strategy is based on the vector fields approach, considering an attraction and a repulsive field. The attraction vector field is responsible for moving the robot towards the ball, according to Equation 4, =

$$\vec{F} = V_{máx} \cdot \frac{\vec{p}_b - \vec{p}_r}{||\vec{p}_b - \vec{p}_r||} \quad , \qquad (4)$$

where $Vmáx$ is the maximum robot's speed and $||\vec{p}_b - \vec{p}_r|||$ represents the norm of the vector.

Therefore, to determine the speeds that the robot must execute to follow the vector field, the following control law can be applied, as described in Equation 5.

$$\vec{V} = \begin{bmatrix} V_r \\ \omega \end{bmatrix} = \vec{F} \cdot \begin{bmatrix} \cos\theta_r & \operatorname{sen}\theta_r \\ \frac{-\operatorname{sen}\theta_r}{d_r} & \frac{\cos\theta_r}{d_r} \end{bmatrix} \qquad (5)$$

The angular velocities of the left and right wheels can then be obtained from Equation 6.

$$\begin{bmatrix} \omega_r \\ \omega_L \end{bmatrix} = \begin{bmatrix} \frac{1}{R} & \frac{L}{R} \\ \frac{1}{R} & \frac{-L}{R} \end{bmatrix} \cdot \vec{V} \qquad (6)$$

Based on the equations presented above, control strategies can be devised to develop algorithms responsible for moving the different positions that the robots can assume during the game: striker, defender and goalkeeper. With the help of FIRASim, it is possible to visualize and test how each of these strategies works.

In a similar way to that proposed in Santos *et al*. (2019), to allow the robot to reach the ball at a desired angle, for example, aiming at the opponent's goal, the possibility of using the attractive vector field described in Equation 7,

$$\overrightarrow{F_a} = (1+n) \cdot \angle\vec{p}_g - n \cdot \angle\vec{p}_r \qquad (7)$$

where $\angle\vec{p}_g$ represents the angle of the vector $\angle\vec{p}_g$ (vector originating from the robot's position and directed towards the ball's position), $\angle\vec{p}_r$ represents the angle of the vector $\angle\vec{p}_r$ (vector originating from the robot's position and directed towards a point between the ball and the goal) and n is a constant to be calibrated.

To make the robots able to avoid obstacles (robots of opposing players and of their own team), a repulsive vector field ($\overrightarrow{F_r}$) is defined in Equation 8,

$$\overrightarrow{F_r} = \angle\overrightarrow{o_v p}, \qquad (8)$$

where $\angle\overrightarrow{o_v p}$ is the angle of the vector $\angle\overrightarrow{o_v p}$, which represents the distance from any position $\vec{p}$ to a virtual obstacle. The fact that it is a virtual obstacle is important, as it will cause a smoother and more anticipated deviation movement, compared to using the real obstacle. Therefore, the virtual obstacle is calculated by Equation 9,

$$\overrightarrow{o_v} = \begin{cases} \vec{o} + \frac{d}{||\vec{p}_v||} \cdot \vec{p}_v \ , & se\ d < ||\vec{p}_v|| \\ \vec{o} + \vec{p}_v \ , & se\ d \geq ||\vec{p}_v|| \end{cases} \quad , \qquad (9)$$

where $\vec{o}$ is the position of the real obstacle, d is the distance between the robot and the real obstacle and $\vec{p}_v$ is a vector described by Equation 10.

$$\vec{p}_v = \mathrm{K}(\vec{V}_{obs} - \vec{V}_r), \qquad (10)$$

where K is a constant that calibrates the displacement of the virtual obstacle.

3. Results

Based on our simulation design, Figure 12 shows the prototype developed with the embedded electronics after validating the 3D design and starting the first tests with the physical robots.

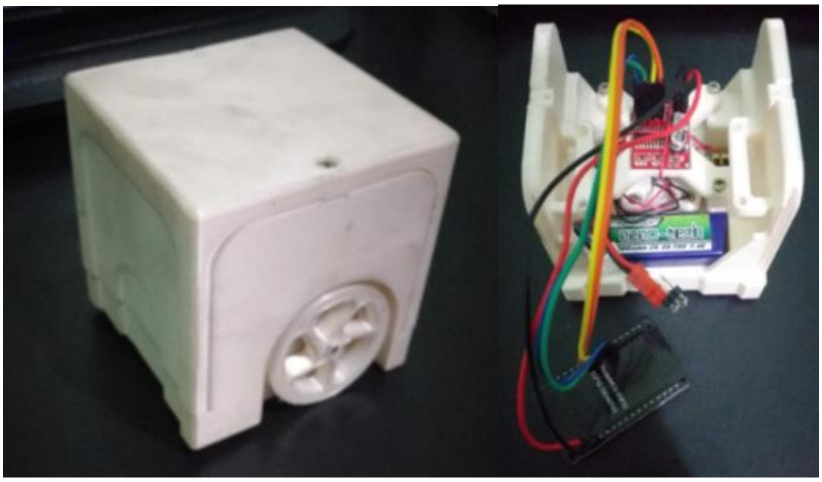
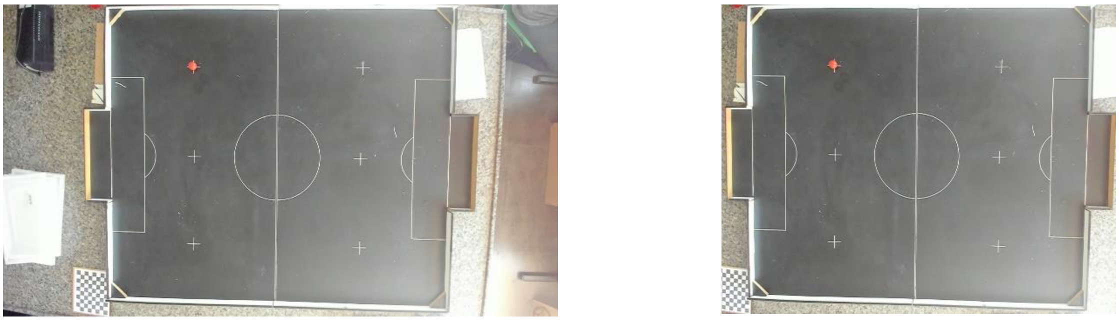


Figure 12 - Developed prototype of the soccer robot.

Concerning the first stages of image processing, Figure 13 shows the original image taken by the camera and the same frame after cropping. In this way, it is possible to process only the area of interest, which allows for an increase in processing speed since the area analyzed has been reduced, as well as reducing noise in the detection of objects.



(a) **(b)**

Figure 13 - Top view of the soccer field (a) before and (b) after ROI.

Figure 14(b) shows the result of the alignment stage from the previous situation in Figure 13(b). The image on the right shows the result obtained with the calibration of the image's optical distortion, where it is noticeable that the edges have been corrected by inserting black pixels in the areas with strong distortion effects.

(a) **(b)**

Figure 14 - Top view of the soccer field after alignment (a) and calibration of the camera's optical distortion (b).

Figure 15 shows the detection of the ball and the robot in the field. Note that it was possible to obtain the desired result using the methods implemented, as the green circle is clearly very accurate around the ball and the red square all over the top face of the robot, even when they were moving.

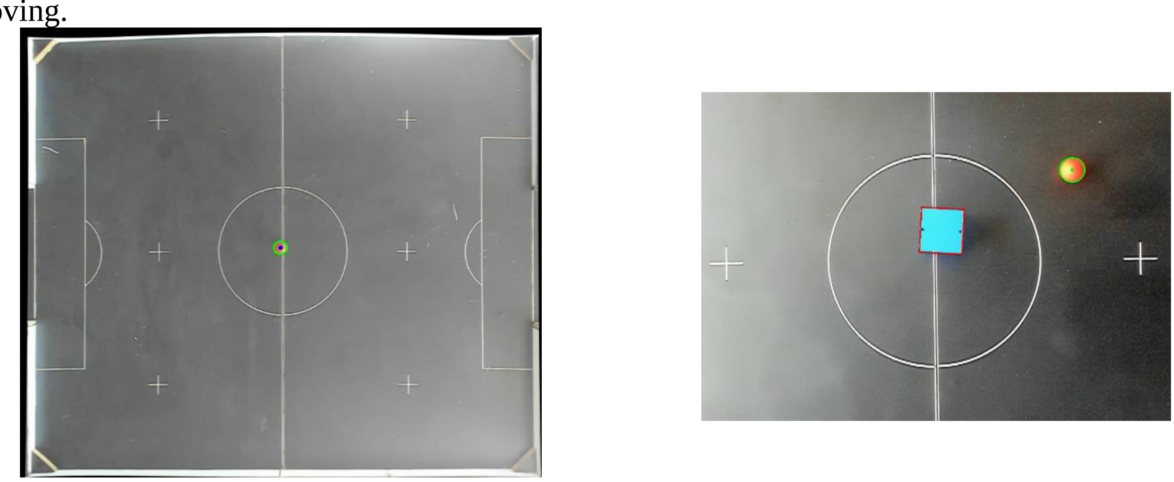


Figure 15 - Robot and ball detection.

Considering our navigation strategy in the simulation, we can see in Figure 16 the result of the attractive field using Equation 7. In it, the blue arrows represent the direction and orientation in which the robot must move to follow the field, the orange circle represents the ball, the green arrow represents a point between the ball and the target, and the gray rectangle represents the goal.

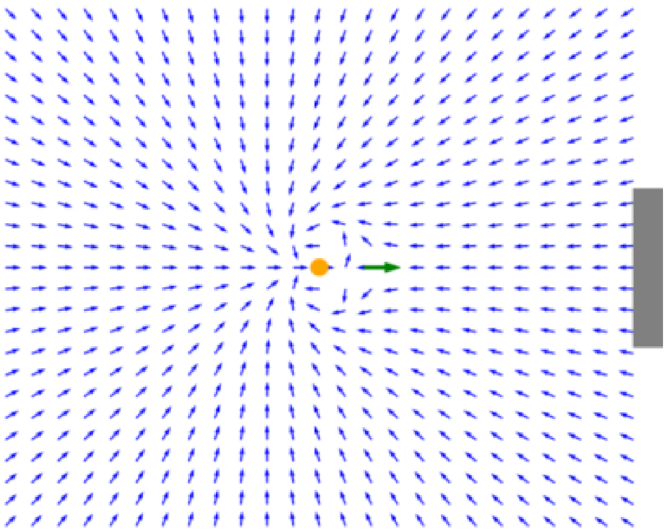


Figure 16 - Example of an attractive vector field.

The implementation of a navigation strategy for tracking the ball using vector fields can be seen in Figure 17 and in more detail in the experiment video in the simulator: https://www.youtube.com/watch?v=O5gZ2FCHrCM.

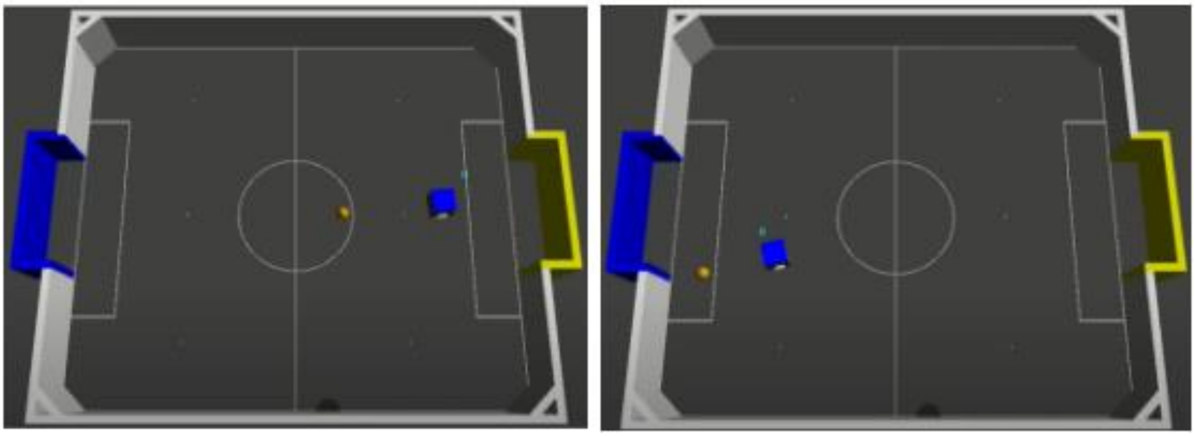


Figure 17 - Snapshots of the ball tracking in the experiment video: https://www.youtube.com/watch?v=O5gZ2FCHrCM.

Finally, the navigation strategy for the striker, defender and goalkeeper in the simulator can be seen in detail in the video: https://youtu.be/zRTtfBAgMZY. Figure 18 shows the goalkeeper's movement as he follows the position of the ball to make a save.

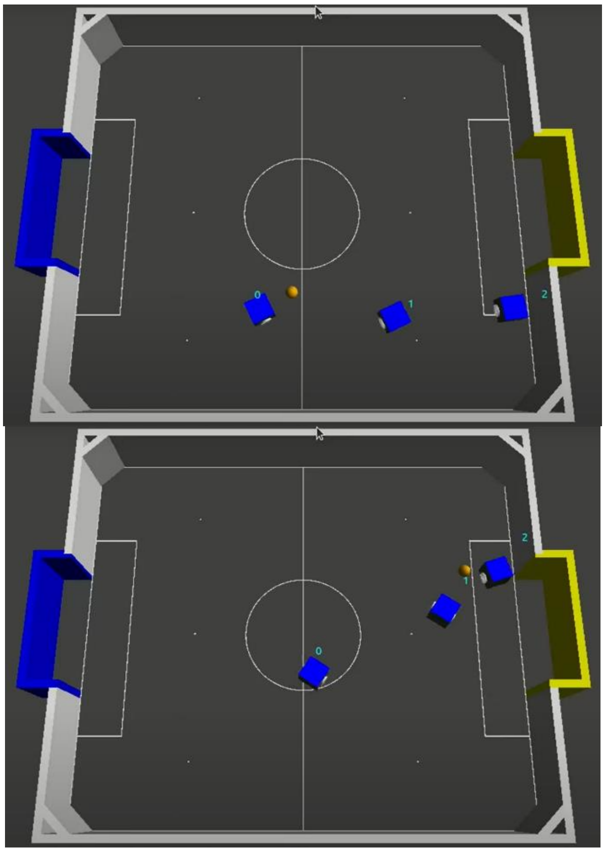


Figure 18 - Snapshots of the Goalkeeper's movement, where Robot 0 is the attacker, Robot 1 is the defender and Robot 2 is the goalkeeper.

4. Conclusion

This article presents the initial development of mobile robots applied to the IEEE Very Small Size Soccer category of robot soccer. The robots were designed and printed using 3D modeling software, and the basic electronics were integrated into the project.

For the computer vision system, Python algorithms were used, supported by the OpenCV library, to process the images. The images were processed according to the HSV color system, which includes hue, saturation, illumination and brightness parameters. It was possible to select the area of interest in the images and correct the field orientation using geometric transformations. The optical distortions caused by the camera were corrected by a checkerboard calibration process. Another process carried out during the work was the calibration of measurements, where known positions in the field were marked. After all the calibrations, a point of origin was marked to find the x and y positions of the game elements. Then, the position of the ball on the field and the position of the robot were determined.

Finally, using a simulator, we tested a navigation strategy based on vector fields. This strategy allows the tracking of the ball and the robots, allowing different robot roles in the soccer game (striker, defender and goalkeeper).

As future work, it is intended to continue testing with real robots and to develop navigation techniques based on both classical methods and techniques using artificial intelligence. In addition, the computer vision system will be integrated with the robot control system so that the robots can play games autonomously using the navigation strategy.

Acknowledgments

This work has been supported by the Brazilian National Council for Scientific and Technological Development (CNPq) and by the Federal Insitute of Minas Gerais (IFMG).

References

Águas, W. G. (2018). Aplicação de FPGA para controle de sistemas embarcados multiagentes: robôs jogadores de futebol da categoria IEEE very small size soccer (Dissertação de mestrado em Engenharia Elétrica e da Computação). Universidade Federal de Goiás, Goiânia.

Bianchi, R., & Reali-Costa, A. (2000). O Sistema de Visão Computacional do time FUTEPOLI de Futebol de Robôs. https://fei.edu.br/~rbianchi/publications/CBA2000.pdf

Circuits4you. (2018). ESP32 DevKit ESP32-WROOM GPIO Pinout. https://circuits4you.com/2018/12/31/esp32-devkit-esp32-wroom-gpio-pinout/

Cunha, M. M. (2019). Planejamento de caminho de robôs futebolistas utilizando busca em grafo (Monografia de graduação em Engenharia de Controle e Automação). Escola de Minas, Universidade Federal de Ouro Preto. http://www.monografias.ufop.br/handle/35400000/2571

Da Rosa, J. F., & Rosa, P. F. F. (2021). Planejamento de Trajetória de Múltiplos Robôs Terrestres Autônomos em Ambientes Dinâmicos . Revista Militar De Ciência E Tecnologia, 38(3), 27-36. http://www.ebrevistas.eb.mil.br/CT/article/view/8653

De Milano, D., & Honorato, L. B. (2010). Visão computacional. Faculdade de Tecnologia, Universidade Estadual de Campinas. https://www.academia.edu/9621896/VIS%C3%83O_COMPUTACIONAL_Palavras_Chaves.

Ferreira, R. H. X., Silva, P. L., Oliveira, V. C. R., Silva Junior, C. D., & Freitas, E. J. R. (2023). Sistema de visão computacional para a categoria IEEE Very Small Size Soccer: Implementação e desenvolvimento. In IV Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia (p. 1-6). Diamantina.

FIRASim. (2023). Simulador de VSSS. https://github.com/VSSSLeague/FIRASim. Accessed July 05, 2022.

Freitas, E. J. R., Cohen, M.W., & Guimarães, F. G. (2023). An Autonomous Mobile Robot Path Planner using Spline Curves and Differential Evolution. IEEE Latin American Conference on Computational Intelligence, (in the publication phase).

Fortes, A. (2013). Identificação e Rastreamento de robôs no Contexto de uma Partida de Futebol de Robôs. Alegrete. URL: https://dspace.unipampa.edu.br/bitstream/riu/1582/1/Identificação%20e%20rastreamento%20de%20robôs%20no%20contexto%20de%20uma%20partida%20de%20futebol%20de%20robôs.pdf

Grepl, R., Lee, B., Kratochvíl, C., Solc, F., & Hracbec, J. (2008). Exact linearization of nonholonomic system dynamics applied to control of differentially driven soccer robot. Engineering Mechanics, 15(5), 311-318.

Hibbeler, R. C. (2010). Resistência dos materiais. Pearson Educación

Hofstatter, G. (2016). Filtro de Kalman aplicado à localização de robôs móveis. Juiz de Fora.

IEEE VSSS. (2023). Liga Very Small Size Soccer. https://ieeevss.github.io/vss/index.html. Accessed February 28, 2023.

Klaser, R. L. (2014). Navegação de veículos autônomos em ambientes externos não estruturados baseada em visão computacional (Dissertação de mestrado, Universidade de São Paulo). https://www.teses.usp.br/teses/disponiveis/55/55134/tde-22092014-095209/publico/Rafael_L_klaser_diss_revisada.pdf.

Lim, Y., Choi, S., Kim, J., & Kim, D. (2008). Evolutionary univector field-based navigation with collision avoidance for mobile robot. IFAC Proceedings Volumes, 41(2), 12787-12792.

Lugli, A. B., Oppenheimer, D. R., Neto, E. R., Ribeiro, J. V. O. G., Nogueira, L. E., Lima, M. E. C., Oliveira, P. P. S., & Saldanha, W. E. (2019). Robotbulls Very Small Size Soccer (VSSS) Team. Retrieved from http://sistemaolimpo.org/midias/uploads/df63b9f477d1a91a68b3bc449dd4762f.pdf. Accessed July 12, 2022.

Martins, M. F., Tonidandel, F., & Bianchi, R. A. C. (2006). Reconhecimento de Objetos em Tempo Real para Futebol de Robôs. Anais do XXVI Congresso da SBC (173-182). São Paulo. https://fei.edu.br/~rbianchi/publications/EnRI2006-Murilo.pdf

Pereira, Guilherme A. S. ; FREITAS, Elias J. R. . Navigation of Semi-autonomous Service Robots Using Local Information and Anytime Motion Planners. ROBOTICA, v. 38, p. 2080-2098, 2020.

Pinto, A. H. M. (2022). Regras IEEE Very Small Size Soccer (VSSS) - Série B. https://www.cbrobotica.org/wp-content/uploads/2021/05/vssRules3x321.pdf

Oliveira, V. C. R., SILVA, P. L., Vicentini, G. A., Chagas, J. V. M., Silva, L. G. V., Silva Junior, C. D., & Freitas, E. J. R. (2022). Protótipo de robô para a categoria IEEE Very Small Size Soccer: design mecânico, eletrônica embarcada e simulação. In III Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia (p. 1-7). Diamantina (MG): Even3.

OpenCV. (2023). About - OpenCV. Retrieved from https://opencv.org/about/. Accessed March 2, 2023.

Osowsky, J., Dias, N., Borba, G.B., & Gamba, H. R. (2006). Correção da distorção radial causada por lentes ópticas. Em XX Congresso Brasileiro de Engenharia Biomédica. Curitiba.

Rosa, J. F. da. (2015). Construção de um time de futebol de robôs para a categoria IEEE Very Small Size Soccer. Petrópolis. http://sirlab.github.io/assets/pdfs/tcc-johnathan.pdf

Santos, P. H., Filho, A. M. S. T., & Freitas, E. J. R. (2019). Desenvolvimento de robôs diferenciais futebolistas: uma aplicação de campos vetoriais no planejamento de caminho. 14º Simpósio Brasileiro de Automação Inteligente (SBAI).

Siegwart, R., Nourbakhsh, I. R., & Scaramuzza, D. (2011). Introduction to autonomous mobile robots. MIT Press.

Simões, A. da S., & Costa, A. H. R. (2001). Classificação de cores por redes neurais artificiais: um estudo do uso de diferentes sistemas de representação de cores no futebol de robôs móveis autônomos. XXI Congresso da Sociedade Brasileira de Computação São Paulo.