

# Soccer Ball Position Tracking with a 3D RR Robot

Arturo Flores Alvarez  
afloresa@ucla.edu

Naravit Vicathorn  
anaravit@g.ucla.edu

Shuhao Yang  
shuhao24@g.ucla.edu

## Abstract

*This paper presents an innovative approach for accurately tracking the position of a soccer ball using a camera integrated into the RR robot's neck, which forms part of a humanoid robot competing in the Adult Size League of RoboCup 2023. The state-of-the-art object detection algorithm YOLOv5 is employed to precisely identify the ball, while stereo vision techniques are harnessed to estimate its depth information. Leveraging the obtained ball position, forward and inverse kinematics computations are utilized to determine the optimal orientation of the camera. To ensure precise, and fast computation of the tracking algorithms, a decentralized position controller is implemented. Experimental results showcase the camera's great ability to effectively track high-speed soccer balls, surpassing velocities of 6.81m/s, while maintaining a remarkably low mean tracking error of only 12.16°. This study highlights the remarkable efficacy of combining modern artificial intelligence techniques, computer vision advancements, and robotics to advance the field of visual servoing, particularly in dynamic environments such as humanoid robotic soccer competitions.*

visual input captured from cameras. While traditional sensor measurements, such as joint encoders or force sensors, suffice for precise positioning and manipulation in well-known and structured environments like factories and laboratories, they often face challenges in unknown object locations and constantly changing surroundings. For instance, the application of visual servoing to unknown scenarios was first demonstrated in early work on object grasping [1].

To address these challenges, visual servoing offers an ideal solution by utilizing exteroceptive sensors to extract valuable information, such as images and depth estimations, from the surrounding scene. By incorporating real-time visual feedback, this approach enables robots to adapt quickly, respond to unforeseen changes, expand their manipulation capabilities, and achieve enhanced control precision across a wide range of applications. The integration of visual servoing enables robots to effectively navigate and manipulate objects in dynamic and unstructured environments, where traditional sensing methods may fall short. This breakthrough technology opens up new possibilities for robotics in domains such as autonomous navigation, object tracking, and human-robot interaction, revolutionizing the way robots interact with and perceive their surroundings.

## 1. Introduction



Figure 1. Object detection using YOLOv5 medium size (106 layers) to the left an photo of upper body and the neck of Artemis

Visual servoing is an advanced technique that empowers robotic platforms to navigate and manipulate objects in highly dynamic and uncertain environments by leveraging

This project is dedicated to addressing a tracking challenge specifically tailored for a RR serial-link robot, which will function as the neck of Artemis (a humanoid robot developed by RoMeLa, see Fig.1) during the upcoming RoboCup 2023 [2] in Bordeaux, France. Our primary objective revolves around the accurate tracking of a soccer ball utilizing a ZED2 camera (Stereo labs), which is affixed to the distal end of an RR robotic arm equipped with two back-drivable electromechanical actuators known as Koala BEAR. To accomplish this, we will employ a real-time object detection algorithm YOLOv5 [3] alongside depth estimation data to extract crucial information regarding the ball's position and velocity relative to the camera's location. Other articles have used YOLO to track objects [4], however this application aimed to track a quasi-static object, which is much more slower than to the movements of a highly-dynamic soccer ball. Moreover, we will design an appropriate controller that consistently adjusts the camera's

orientation to align with the ball's position updates. The key objectives of this control algorithm encompass maintaining the ball's position at the center of the frame (or with an admissible tracking error), ensuring smooth and continuous tracking, and effectively mitigating the influence of measurement noise disturbances. This survey [5] describes the state-of-the-art learning-based algorithms, especially those algorithms that combine with model predictive control (MPC) and more advance control techniques.

In the short term, this module will be integrated into Artemis will significantly enhance its performance on the field, contributing to its competitiveness as a humanoid soccer player. In the long term, we believe that our module's contributions extend beyond the realm of robotics in sports. The insights gained from this project can potentially accelerate applications that rely on visual servoing, particularly in the context of fast pick-up tasks within industrial production settings [6]. By leveraging the robust tracking and control mechanisms developed for soccer ball tracking, these applications can experience improved productivity and efficiency, ultimately leading to cost savings and increased production rates.

## 2. Problem Formulation

The first step to solve the tracking problem is to solve the forward kinematics of the neck. Figure 2 depicts the convention used to establish the coordinate frames using the Denavit Hartenbeg convention, where the subscript "0", "1", "2", "d" represent the base frame, first frame second frame, and end effector frame respectively.

Our primary objective is to ensure that the ball remains at the center of the image. To accomplish this, we need to orient the y-axis of the end effector frame toward the center of the ball. Utilizing a computer vision system, we are able to determine the x, y, and z coordinates of the ball's center of mass (CoM). By computing  $\hat{u}_p$ , a unit vector originating from the end effector coordinate frame and pointing towards the center of the ball, we can obtain the necessary camera orientation to track the ball. The Euler angle of vector  $\hat{u}_p$  is calculated with respect to the end effector frame, providing us with the desired camera orientation.

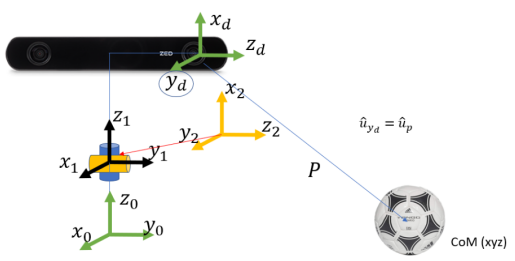


Figure 2. Frames Assignment Schematic

Subsequently, the inverse kinematics are employed to find the angles  $\theta_1$  and  $\theta_2$  for the two motors. These angles serve as the reference inputs for the controller. Through this pipeline, we can effectively solve the tracking problem, ensuring the ball remains consistently centered within the image and updates the position of the motors whenever the ball moves.

## 3. System

The system utilized in this study is derived from the upper body of Artemis, as depicted in the right image of Figure 1. It encompasses various components, including a base frame constructed using 80/20 bars and 3D printed parts, two KOLA BEAR from Westwood Robotics [7], a ZED 2 Camera, and an HP Elite mini computer GTX 3070, which is equipped with computer vision and the tracking algorithm. Figure 3 illustrates the implemented ROS2 network diagram, showcasing the communication between different nodes. It is important to note that the computer vision node publishes information at a slower rate due to the significant computational resources required for inference processes.

### 3.1. Mechanical System

The mechanical system consists of multiple 80/20 bars cut from a 3 feet 80/20 bar using a band saw. The desired shapes are manufactured using a Haas VF-2SS CNC machine. The 80/20 bars are assembled together using 3D-printed parts and screws. Then 2 KOLA BEAR actuators are attached to the surface as shown in (Figure 4). Then ZED2 camera is attached to the top actuator using a 3D-printed part.

According to the frame assignment in (Figure 2) the DH parameters are in the table below

Table 1. DH Table of Artemis' neck

i-1	i	a	$\alpha$	d	$\theta$
0	1	0	0	0.0375	$\theta_1$
1	2	0	$\pi/2$	0	$\theta_2 - \pi/2$
2	T	0.125	0	0.06	0

From the DH parameter the forward and inverse kinematic and be calculate, from equation 1 and equation 2 respectively.

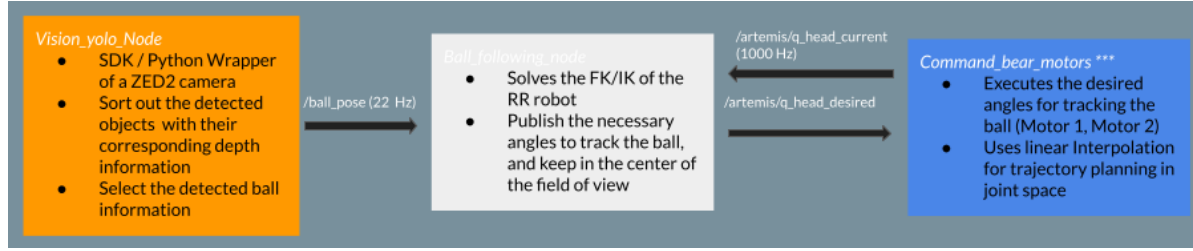


Figure 3. Communication Network in ROS2 foxy and the publishing rates of each topic

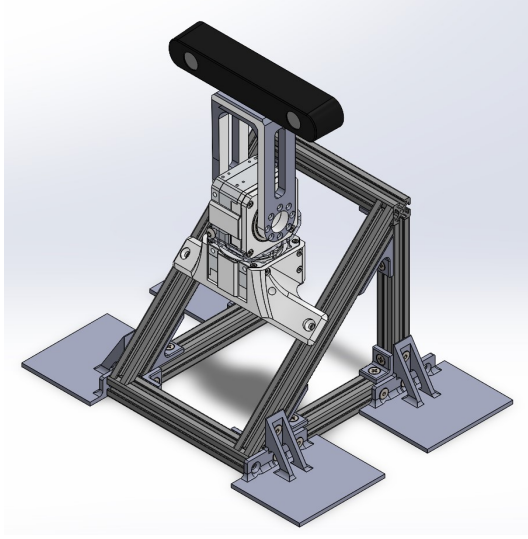


Figure 4. Mechanical Design

$$\begin{aligned}
 r_{11} &= \cos(t1)\cos(t2) \\
 r_{12} &= -\cos(t1)\sin(t2) \\
 r_{13} &= -\sin(t1) \\
 r_{14} &= (1347 * \cos(t1)\cos(t2))/20000 - (3\sin(t1))/50 \\
 r_{21} &= \cos(t2)\sin(t1) \\
 r_{22} &= -\sin(t1)\sin(t2) \\
 r_{23} &= \cos(t1) \\
 r_{24} &= (3 * \cos(t1))/50 + (1347 * \cos(t2)\sin(t1))/20000 \\
 r_{31} &= -\sin(t2) \\
 r_{32} &= -\cos(t2) \\
 r_{33} &= 0 \\
 r_{34} &= 7/200 - (1347\sin(t2))/20000 \\
 r_{41} &= 0 \\
 r_{42} &= 0 \\
 r_{43} &= 0 \\
 r_{44} &= 1
 \end{aligned}$$

(1)

$$\begin{aligned}
 \theta_1 &= \text{atan2}(-T(1,3), T(2,3)) \\
 \theta_2 &= \text{atan2}(-T(3,2), T(3,1))
 \end{aligned} \tag{2}$$

By considering the joint angle limits for each motor, the forward kinematics were employed to visualize the camera workspace. The resulting camera workspace is depicted in Fig.5

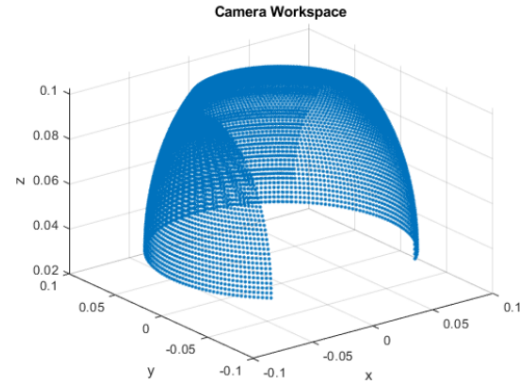


Figure 5. Camera Workspace

### 3.2. Computer Vision System

The computer vision system utilized YOLOv5 for object detection and Stereo vision for depth estimation.

#### 3.2.1 YOLOv5 (You Only Look Once)

YOLOv5, a real-time object detection AI model based on convolutional neural networks (CNN), was employed for accurate object detection. Given an input image, YOLOv5 generates bounding boxes and provides the likelihood that an object exists within each bounding box. For further refinement, the image within each bounding box is passed through a trained CNN for image classification. By training this CNN, we can effectively detect the soccer ball from an image.

*Note: It is important to highlight that the YOLOv5 repository is open-source on GitHub, and our implementation of it relies on the object detection examples provided by Stereo labs (also on GitHub). These algorithms are well-established solutions developed by the respective teams at each company, and our team has utilized them as reliable resources in our project [8].*

### 3.2.2 Stereo Vision

Stereo vision is a visual perception mechanism that enables depth perception using two camera. Firstly, image rectification was performed to align the image plane of the left and right camera. Key features of the images were then extract, and matching the correspondence between the left and right image. The disparities map between the left and right feature were then compute. With the disparities know, the depth map can be generate using equation 3

$$Z = fb/d \quad (3)$$

where  $Z$  is the depth,  $f$  is the focal length, and  $d$  is the disparity.

*Note: OpenCV and ZED2 library was used to perform Stereo Vision [9]*

### 3.3. Control System

The computer vision system enables us to extract the  $x$ ,  $y$ , and  $z$  coordinates of the ball. By following the problem formulation section, we can determine the required angles  $\theta_1$  and  $\theta_2$  for the motor. To achieve precise joint angles, a decentralized position controller was implemented, utilizing  $\theta_1$  and  $\theta_2$  as reference inputs. The schematic of the decentralized position controller is depicted in Figure 6.

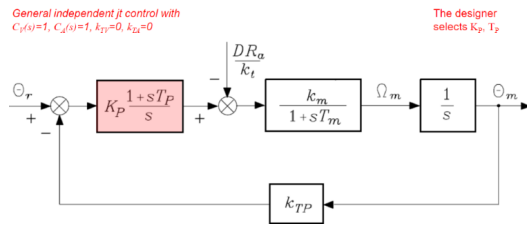


Figure 6. Decentralize Position Control

The system was approached as a black box, where the desired angle served as the input and the current angle as the output. With the completion of all system components, the next step involved thorough testing to assess the system's performance and gather essential data. It is important to note that our approach assumes low torque requirements, considering the lightweight nature of the camera and the fact that the neck does not come into contact with the environment.

## 4. Results and Analysis

To evaluate the performance of the system, we conducted tests in which the camera was positioned at coordinates (0,0)m in the world coordinate frame. Two individuals were positioned 4m apart, with one located at (2,-2)m and the other at (2,-2)m in the same coordinate frame. Initially, the ball was placed next to one of the players, and then it was kicked back and forth along the  $y$ -direction six times before coming to a stop. These tests aimed to fine-tune the PID gains of the neck motors, with the objective of minimizing both the mean and standard deviation of the tracking error. By optimizing these gains, we aimed to enhance the robustness and responsiveness of the system, enabling it to effectively adapt to changes in the ball's position. The optimization process aligned with our specific criteria for controller design and tuning, ensuring the system's performance was optimized for accurate and reliable tracking.

The following plots showcase the results obtained using the best control gains of  $K_P=300$  and  $K_D=1$  for motor 1, and  $K_P=200$  and  $K_D=0$  for motor 2. Thus, Figure 7 presents the Matlab simulation (Peter Corke's Robotics Toolbox) displaying the trajectory of the ball based on the recorded data, as well as the visual model of the robot and camera.

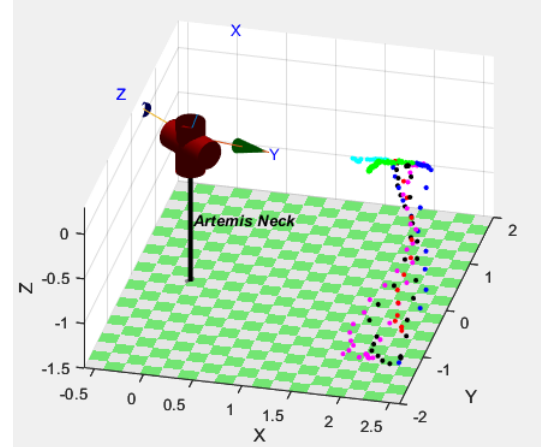


Figure 7. Matlab Trajectory Simulation from Recorded Data

This section refers to Figure 8, which depicts the plot of the  $xyz$  position of the ball. The plot provides meaningful insights into the ball's movement during the experiment. As anticipated, the  $y$ -direction exhibits the most significant displacement since the ball was kicked along this axis. The  $x$ -axis also shows some displacement, indicating that the ball did not follow a perfectly straight trajectory. Moreover, as the ball did not experience any bouncing motion, the  $z$ -direction remains relatively constant. This observation validates the effectiveness of the computer vision system in accurately extracting the relevant depth information from the

tracked ball.

Furthermore, the obtained position data has proven valuable in simulating the desired neck angles. By comparing these simulated angles with the actual angles recorded during the experiment, we can assess the accuracy and reliability of the tracking system. This integration of simulation and experimental data provides a comprehensive understanding of the system's performance and facilitates further analysis and improvement.

It is important to note that the plots exhibit some noise due to camera measurements. However, these effects can be mitigated by applying thresholds and dead zones. These measures ensure that the neck motors do not experience excessive shaking or unnecessary movements because of noise

Position of the Ball - Exp(2) - World Frame XYZ

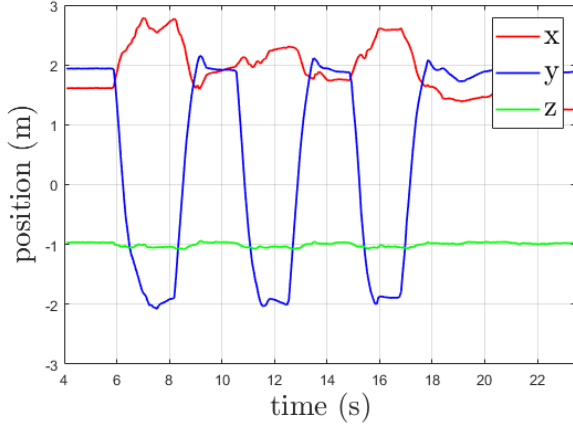


Figure 8. XYZ Position of the Ball in the World Frame

Using the position data, the derivative was taken to obtain the velocity values, which were then plotted in Figure 9. The graph reveals that the y-direction exhibits the highest velocity, reaching a peak of 6.81m/s. This observation aligns with the fact that the ball was kicked along the y-direction during the experiment. Furthermore, the velocity graph shows six distinct peaks, which correspond to the ball being kicked six times.

The real neck motor angles were plotted in Figure 10 and compared with the Matlab simulation model. Motor 1 corresponds to the neck movement left and right. The graph indicates that it is reasonable for motor 1 to exhibit significant displacement in order to track the kicked ball. Conversely, since the ball does not move vertically or farther away, there is no need for the neck to tilt up and down. As a result, motor 2 remains relatively constant. The recorded data and the simulation demonstrate similar results, confirming the accuracy and reliability of the simulation model in replicating the real-world behavior of the neck motors.

The tracking error analysis is presented in Figure 11,

Velocity of the Ball - Exp(2) - World Frame XYZ

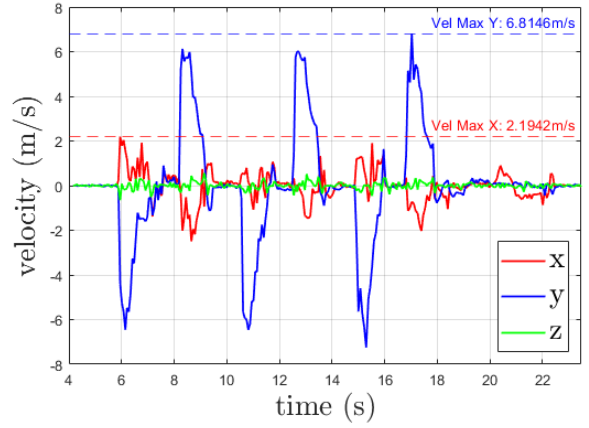


Figure 9. XYZ Velocity of the Ball in the World Frame

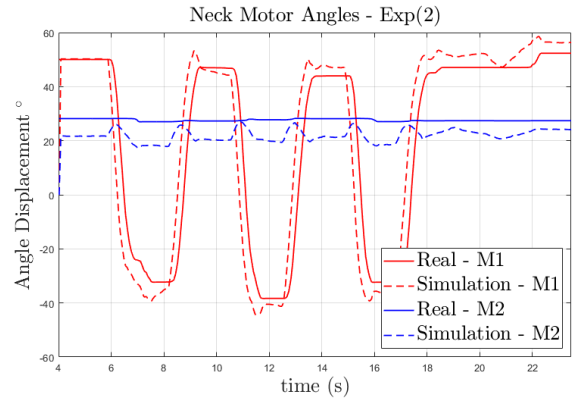


Figure 10. Neck Motor Angle

where the camera's field of view (FOV) of  $110^\circ$  is considered. This FOV sets the maximum allowable deviation from the center of the image, with half of the FOV range, i.e.,  $55^\circ$ , representing the threshold beyond which the ball is no longer visible. In the plot, this threshold is depicted by the green horizontal line. As our objective is to consistently keep the ball in the center of the image, the motor angle displacement plays a crucial role in controller design.

To assess the controller's performance, the error for both motors was calculated using the L2 norm for easier gain tuning. The resulting mean error angle displacement achieved by this controller is  $12.16^\circ$ . It is noteworthy that the tracking error exhibits a slight correlation with the ball speed, with faster-moving balls resulting in slightly higher errors. However, considering the very fast ball speed and the achieved error of only  $12.16^\circ$ , well within the  $55^\circ$  limit, we can confidently conclude that the controller's performance is highly satisfactory.

Lastly, we have included in Table 2 a summary of all the sets of controller gains that we experimented with during



Tracking Error / Gains P1: 300 , D1: 1 , P2: 200 , D2:

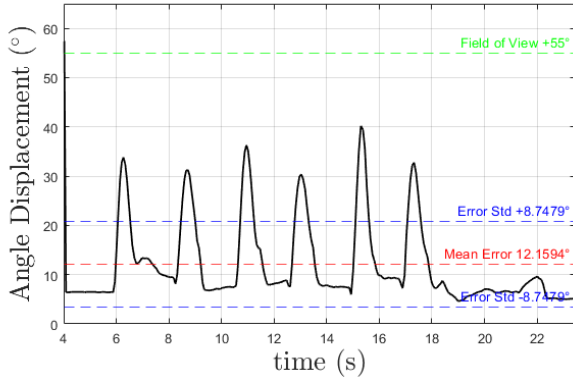


Figure 11. Absolute Tracking Error

the tuning process. In addition, we attached all the plots concerning to all the tests conducted at the lab for reference and comparison. The table presents the corresponding mean error for each set of gains. The best set of gains, which yielded the smallest mean error, has been highlighted in the table. This information provides a comprehensive overview of our exploration and selection of the most effective controller gains for achieving optimal tracking performance.

*Note that in some attempt, the camera was not able to track because the ball was kick too fast (more than 19m/s) to test the controller performance.*

To further validate the system's capabilities, outdoor tests were conducted at the Intra Mural Field of the University of California, Los Angeles, as depicted in Fig. 12. Despite the presence of external factors such as varying daylight conditions, gait instability of Artemis, false positives, and other sources of noise, the system's indoor behaviors remained largely effective during these outdoor tests. This success can be attributed to the modularity of the developed model using ROS, which facilitated seamless transitions between platforms by simply replacing the "command\_bear\_motors" node in Fig. 3 with the corresponding node for controlling Artemis' neck motors. This modularity ensured that the testing conducted on the small upper body setup translated well to the actual robot.

## 5. Discussion and Future Work

In conclusion, this study presents a successful case of visual servoing in a tracking system using computer vision and decentralized position control. The system underwent rigorous testing, with the camera positioned at the origin of the world coordinate frame and multiple ball tracking experiments conducted. Through meticulous fine-tuning of the PID gains for the neck motors, the system achieved improved performance in minimizing the mean and standard deviation of the tracking error. We attribute this success



Figure 12. Deployment of the model in Artemis at the IM field at UCLA

to the utilization of BEAR actuators and the timely publication of desired head poses, both of which contributed to real-time, fast tracking of the ball in different conditions.

In terms of tracking velocities, the system demonstrated exceptional performance by successfully tracking a ball with an estimated average velocity of 6.8 m/s. This performance surpassed the velocities typically achieved by other robots at RoboCup, whose locomotion approaches often rely on rigid brushed motors and large rectangular feet, resulting in maximum velocities of around 3.00 m/s. As shown in Fig. 9, the developed tracking system exceeded our performance expectations for the competition. It is important to note that when running the entire system stack, the tracking module's publishing rate may experience slight slowdowns due to resource allocation to concurrent modules.

Furthermore, the computer vision system proved its accuracy by precisely extracting the position and velocity information of the ball, enabling reliable tracking along the y-direction while maintaining stability in the x and z-directions.

In conclusion, the comparison between real and simulated motor angles provides strong evidence of the effectiveness of the system's control mechanism. The analysis of tracking errors further reinforces the exceptional performance achieved by the developed controller, with the mean error angle displacement well within the camera's field of view limit. This research not only presents a robust tracking system but also highlights its potential applications in various fields, including robotics, surveillance, and sports analysis.

The successful integration of computer vision, control theory, and modular design principles has enabled the development of a precise and reliable tracking system. This interdisciplinary approach demonstrates the power of combining cutting-edge technologies to achieve advanced tracking capabilities.

Table 2. Set of gains selected for tuning

#	$M_1 - K_p$	$M_1 - K_d$	$M_2 - K_p$	$M_2 - K_d$	Mean Error (°)	Standard Deviation (°)	Max Y velocity (m/s)
1	300	0	200	0	12.85	7.73	7.93
<b>2</b>	<b>300</b>	<b>1</b>	<b>200</b>	<b>0</b>	<b>12.16</b>	<b>7.73</b>	<b>7.93</b>
3	300	0	200	1	12.84	9.21	7.8
4	300	2	200	0	14.11	10.10	8.23
5	700	0	600	0	12.65	9.57	5.67
8	1000	0	600	0	15.14	9.54	7.36
10	1100	1	600	0	15.76	8.45	7.02

As a direction for future work, incorporating an estimator based on the dynamic model of ball movement could enhance the tracking system. By anticipating future ball positions, rather than solely reacting to the current position, the tracking process could be expedited. This predictive capability has the potential to further improve the system's responsiveness and overall performance.

## References

- [1] X. Gratal, J. Romero, J. Bohg, and D. Kragic, "Visual servoing on unknown objects," *Mechatronics*, vol. 22, no. 4, pp. 423–435, 2012. Visual Servoing SI. **1**
- [2] C. Sammut, "'robocup federation official website.'" robocup federation official website, 26 feb. 2023, <https://www.robocup.org/>," 2020. **1**
- [3] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. **1**
- [4] P. Durdevic and D. Ortiz-Arroyo, "A deep neural network sensor for visual servoing in 3d spaces," *Sensors*, vol. 20, no. 5, 2020. **1**
- [5] J. Wu, Z. Jin, A. Liu, L. Yu, and F. Yang, "A survey of learning-based control of robotic visual servoing systems," *Journal of the Franklin Institute*, vol. 359, no. 1, pp. 556–577, 2022. **2**
- [6] M. G. Krishnan and A. S, "Object detection methods for image-based visual servoing of 6-dof industrial robot," in *2022 IEEE Region 10 Symposium (TEN-SYMP)*, pp. 1–6, 2022. **2**
- [7] Westwood Robotics, "Bear sdk manual 0.3.1." [https://www.westwoodrobotics.io/wp-content/uploads/2023/04/BEAR\\_SDK\\_Manual\\_0.3.1\\_EN.pdf](https://www.westwoodrobotics.io/wp-content/uploads/2023/04/BEAR_SDK_Manual_0.3.1_EN.pdf), 2023. **2**
- [8] Ultralytics, "YOLOv5." <https://github.com/ultralytics/yolov5>, Year. **4**
- [9] StereoLabs, "ZED SDK." <https://github.com/stereolabs/zed-sdk>, Year. **4**