NAO Robot’s Vision Control and Kick Motion Generation

Abdul Rasheed
CRAIB, Department of Computer Science
Sukkur IBA University
Sukkur, Pakistan
abdul.rasheed@iba-suk.edu.pk

Bilal Ahmed
CRAIB, Department of Computer Science
Sukkur IBA University
Sukkur, Pakistan
bilal_arain@iba-suk.edu.pk

Nimra
CRAIB, Department of Computer Science
Sukkur IBA University
Sukkur, Pakistan
nimra@iba-suk.edu.pk

***Abstract*—This research explores the enhancement of NAO robot's soccer-playing capabilities by integrating vision-based systems. Robot computer vision capabilities such as ball recognition, ball tracking, and motion capabilities like kicking and shooting are also explored. The bottom camera tracks a red ball, and the top camera detects the goal. The robot navigates towards the ball, adapting its position for a precise kick to the left or right. Crucially, safety measures are embedded, ensuring the robot refrains from movement or kicking if the ball is not visible. The process of kick generation and execution is also discussed. This study highlights the feasibility of Nao robot as a soccer player and provides insights into the integration of robotics and programming in sports.**

Keywords- Nao Robot, Computer Vision, Soccer, Robotics.

I. Introduction

The integration of robotics and artificial intelligence (AI) has been transforming various fields, including sports. Robotic players offer unique opportunities to explore new dimensions in the sport, by combining technology with programming to enhance the capabilities of the robots and pushthe boundaries of human-robot collaboration. An area of research that has witnessed significant advancements is theutilization of robots as soccer players. In this paper, we delve into the world of robotic soccer player, specifically focusing on Nao robot and its applications; ball tracking and kicking. The potential of Nao robot as a soccer player is explored, emphasizing the development of static soccer-specificbehaviors such as ball control, ball tracking, and ball kicking. Human-Robot Interaction (HRI) is an interdisciplinary study of interaction between humans and robots. HRI has recently received attention in academic community, labs, technology companies, and media [4]. In case of humanoid robot for education, vision system is one of the main sources for environment interpretation. Humanoid robots have shown to be powerful science, technology, engineering, and maths (STEM) education tools [4].

Corresponding author: Abdul Rasheed

II. Related work

This section explains background knowledge of Nao robots its use cases in SPL (Standard Platform League). Moreover, literature review explains about the available platforms to program Nao robot to play SPL [3]. In conclusion, we discussed reasons to go with Python Programming and to check feasibility level to program Nao robot as a soccer player.

A. The Nao V6 Robot

The Nao robot has got attention since the SPL (Standard Platform League) concept started. Before Nao robot, many other platforms were used such as four-legged robots and mobile robots were used for Soccer. The Nao robot developed by Softbank Robotics [1], is a humanoid robot equipped with advanced sensors, motors, and communication capabilities. Its design and size make it suitable for mimicking human movements and interacting to its environment. The Nao has a humanoid appearance, standing at approximately 58 centimeters (23 inches) tall, with a white plastic body, rounded head, and expressive eyes. It has 25 degrees of freedom [11], enabling it to move its limbs, head, and hands with a wide range of motion, allowing for human-like gestures and interactions. One of the key features of Nao robot is its advanced programming capabilities. It runs on the Naoqi operating system, which is Linux based kernel and provides a framework for developers to create and control various behaviors and applications for the robot. Developers can program Nao robots using several programming languages, including Python and C++, and leverage a rich set of APIs and libraries provided by Softbank Robotics [9]. The Nao robothas been widely used in research, education, and entertainment fields. In educational settings, it has been employed to teach programming and robotics concepts to students. Researchers also use Nao robots to explore human- robot interaction, artificial intelligence, and other related fields. The Nao robot represents a significant advancement in humanoid robotics, providing a platform for developers and researchers to explore the potential of social and interactive robots in various domains. Its versatile programming capabilities and human like appearance make it a popular choice for those interested in studying and developing robotics applications. Figure 1 shows image of Nao robot.

Fig. 1: The Nao Robot.

B. Use Cases of Nao V6 Robot in Standard Platform League(SPL)

The Nao robot offers several applications in SPL. Some of the main applications are:

1. The Nao in the Game: The Nao robots is used as player in the SPL matches. They are equipped with sensors, cameras, and actuators to perceive the environment, interact with the ball, and make decisions based on the game situation. The robots need to navigate the field, dribble the ball, pass to teammates, and score goals autonomously [5].
2. Object Recognition and Tracking: The Nao robot can be programmed to recognize and track objects, such as the ball, and other robots present in the robot field of view with help of its built-in cameras and sensors. This capability allows the robot to perceive the game environment and make decisions based on the positions and movements of the objects.
3. Motion Planning and Control: The Nao robots can perform complex motion planning and control to navigate the soccer field effectively. They need to move quickly, change directions, avoid obstacles, and reach the desired position to play the game. The robot's actuators and motion control algorithms enable it to execute agile movements [5].
4. Strategy and Decision-Making: Teams participating in the SPL need to develop strategies and algorithms for their Nao robots to play effectively. Robots should make intelligent decisions about ball possession, passing, shooting, and defending. This involves implementing algorithms for tactics, positioning, and coordination among teammates [5].
5. Human-Robot Interaction: The Nao robots provides an opportunity for human-robot interaction in the SPL. They can communicate with human teammates and spectators through speech, gestures, and expressions. This interaction can enhance the engagement and experience of the audience and facilitate collaboration between humans and robots on the field [5].

C. Literature Review

The Nao robot can easily be programmed through available software and tools such as; Choregraphe, Naoqi and Webots. These tools can help us to understand the basics of humanoid robot Nao. However, these tools are useful only for beginners so that theyget hands-on with the robot easily. For Researchers, ROS (Robot Operating System) is considered suitable platform. And teams participate in SPL they designed their own controllers for the soccer game. The choice of the platforms is according to the research you do with the robot. Further, the subsequent sections briefly describe the suitable platforms to program Nao robot as a soccer player [3]. Many image processing applications have become more reliable and effective as a result of the usage of computer image processing. Objects can be detected by applying several computer vision techniques [14]. Color is extremely important in image processing. Each image made up of an M×N pixel array is split into M rows and N columns. The red, green, and blue values of each pixel are unique. The RGB colors are identified using thresholds. The intended color of the image is identified using the color recognition process [15].

1. Choregraphe: It is GUI (Graphical User Inter- face) based programming environment developed by Aldebaran Robotics (now known as Softbank Robotics). It provides a user-friendly graphical interface that allows users to create and edit robot behaviors without requiring advanced programming skills. However, Choregraphe is not used for programming Nao robots specifically for the SPL (Standard Platform League) in Robocop. While Choregraphe is a valuable tool for programming and controlling Nao robots in a general context, it is not suitable for SPL as it provides a higher-level, graphical programming environment that does not meet the strict programming requirements set by the SPL rules. Teams in SPL need to implement more fine-grained control over the robot's behaviors, motions, perception, and communication, which is the best achieved through lower-level programming languages and libraries. Therefore, teams participating in SPL typically use the Naoqi SDK to program Nao robots directly using languages such as C++ or Python to meet the specific programming demands of the league [9].
2. Naoqi SDK: In the context of SPL (Standard Platform League) in Robocop, Naoqi plays a crucial role as the underlying software framework for controlling the Nao robots during soccer matches. Naoqi provides the necessary infrastructure and capabilities to program and control Nao

robots' behavior, perception, and motion in the SPL environment [3]. Naoqi can be utilized in SPL as:

I. Perception and Sensing: Naoqi allows teams to access and process sensor data from Nao robots, including the cameras and sensors on the robots. This data can be used for object detection, ball tracking, and localization within the soccer field [7].

II. Motion Control: Naoqi provides the motion control frame- work for Nao robots, allowing teams to control their movements, walk, turn, kick, and perform other dynamic actions on the field. Teams can program motion sequences and behaviors to navigate the robots during game play [7].

III. Behavior Control: Naoqi enables teams to develop behaviors for Nao robots using a behavior-based programming approach. Teams can program the robots to react to specific game situations, follow strategies, communicate with team- mates, and execute coordinated actions [7].

IV. Communication and Team Coordination: Naoqi facilitates communication and coordination between Nao robots within a team. Teams can use Naoqi's communication features to exchange information, share game state, and collaborate in real time during matches [7].

V. Strategy Development: Naoqi allows teams to develop and implement strategies for Nao robots. Teams can program decision-making algorithms, game analysis modules, and high-level strategies to guide the robots' actions on the field [7].

VI. Integration with Robocop Infrastructure: Naoqi interfaces with the Robocop infrastructure, enabling teams to interact with the referee system, receive game commands, and send game related information during SPL matches [7].

3. Webots: In comparison to Choregraphe, Webots provides extra features such as adding obstacles in the environment and tracking the odometer of the robot. The simulator also contains motion libraries to access robot joints and move them through high-level commands. During the motion, the motion library of the robot also controls the physical stability of the robot [13].

4. ROS (Robot Operating System): ROS plays a significant role in the Standard Platform League (SPL) of RoboCup. SPL teams often leverage the capabilities and features provided by ROS to program and control their Nao robots effectively [3]. Some key roles of ROS in SPL:

I. Communication and Message Passing: ROS provides a robust framework for inter-process communication (IPC) and message passing. SPL teams utilize ROS's communication infrastructure to exchange information and commands between various components of their software stack, such as perception,motion control, decision-making, and strategy modules [6].

II. Sensor Integration: ROS facilitates the integration of sensordata from Nao robots into the software architecture. Teamscan leverage ROS's sensor drivers and libraries to acquire data from Nao's cameras, microphones, touch sensors, and inertial sensors. This data can be processed and shared across differentmodules using ROS messages [6].

III. Perception and Processing: ROS supports various libraries and tools for perception and processing tasks. SPL teams can use ROS-compatible packages for vision processing, object detection, ball tracking, localization, and mapping to enable their robots to perceive and interpret the soccer field and gameelements [6].

IV. Motion Control and Actuation: ROS allows teams to control the Nao robots' motions and actuators. By utilizing ROS-compatible motion control libraries or developing custom ROS nodes, teams can program walking patterns, kicking mechanisms, joint control, and other motion-related behaviors for the Nao robots [6].

V. Behavior Coordination and Decision-Making: ROS facilitate the coordination of behaviors and decision-making processes in SPL. Teams can develop ROS nodes and modules to implement game strategies, behavior arbitration, state machines, and team coordination algorithms to guide the Nao robots' actions during matches. ROS provides simulation environments, such as Gazebo, which allow teams to simulate the behavior of Nao robots in a virtual environment. [6]. Table.1 Shows comparative analysis of discussed platform.

Table I: Comparative Analysis

Programming Tools	Advantages	Disadvantages
Choregraphe	-User Friendly Interface -Rapid Behavior Development -Simulation and Testing -Reduced Development Time	-Performance Overhead -No Multi Robot Coordination -Difficult to integrate with custom libraries -Limited Support for Advanced Algorithms
Naoqi	-Robot Specific Features -Rich APIs and Functionality -Behavior Based Programming -Extensible and Customization	-Complex for Beginners -Limited Processing Power -Limited Control and Flexibility -Lack of community support
ROS	-Modular Behavior De- velopment -Inter robot Communi cation -Simulation and Testing -Multi Robot Coordi- nation support -Strategy Development	-Complexity -Hardware Limitations -Maintenance and Up- grades -Performance Limitations -Compatibility Issues

III. METHODOLOGY

The objective of this study is to explore the soccer playing capabilities such as. Ball tracking and kicking for Nao robot using a combination of programming and sensor integration [2][8]. The methodology we have employed to achieve the goal is depicted in Figure 2. The steps involved in this experiment are discussed in subsequent sections.

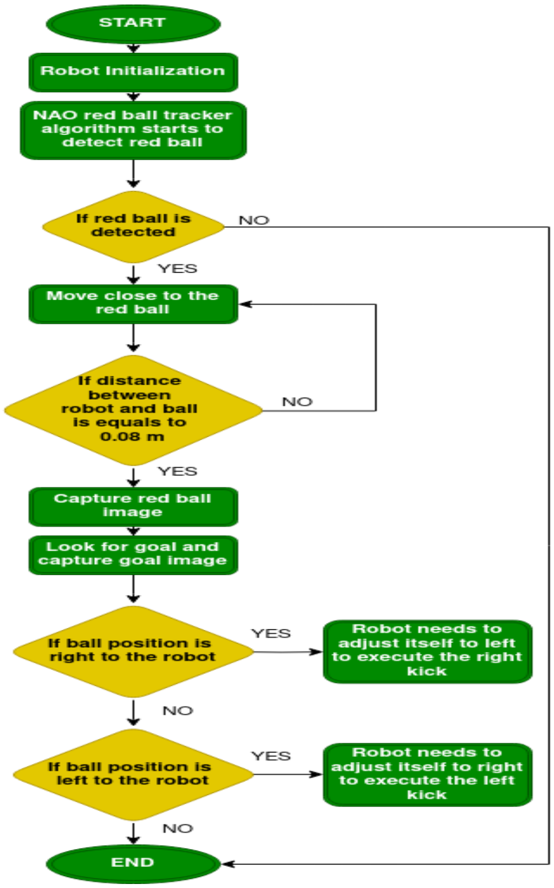


Fig. 2: Methodology flow diagram.

1) Robot Initialization: The initialization process of Nao robot typically involves several steps to configure its hardware and software components. The Nao robot initialization process includes:
I. Hardware setup: Ensure that Nao robot is in a stable position and that all physical components are properly connected. This includes verifying the connections of the robot's batteries, charger, sensors, motors, and any external devices.
II. Naoqi Framework: The Nao robot operates using the Naoqiframework, which is its software platform. It provides a set of APIs and tools for programming the robot's behaviors. Ensurethat the Naoqi framework is installed and running on the robot.
III. Connection to Network: Establish a network connection for Nao robot. This can be done using either a wired or a Wi-Fi connection, depending on the available network infrastructure. It is highly recommended to use wireless network connection when Nao robot is programmed for Soccer. It will be messy and difficult for this application to work with wired network.
IV. Launch PyNaoqi SDK: Launch Naoqi software on a connected computer. This software acts as a bridge between Nao robot and programming environment. In this experiment PyNaoqi-python2.7-2.8.6.23-linux64-20191127-152327version of this software is used [8].
V. Establish Communication and Software Initialization: Establish communication between Nao robot and the Naoqi software running on the computer. This is typically done by connecting to the robot's IP address or host name using the appropriate communication protocol, such as TCP/IP. Once communication is established, initialize the Naoqi software to prepare the robot for operation. This may involve configuring various settings, loading necessary libraries, and setting up the desired operating mode for the robot.
VI. Ready for Operation: Once Nao robot has been initialized, it is ready for operation. At this stage, you can start executing specific behaviors, programming tasks, or applications on the robot using the Naoqi APIs and programming environment.
2) *The Red Ball Tracker:* The red ball tracker is one of the built-in capabilities of the robot's vision system. This red ball tracker algorithm allows the robot to detect and track the position of a red ball in its field of view using its onboard camera. By calling the box of red ball tracker in Choregraphe, you can access the red ball tracker functionality and incorporate it into your custom behaviors and applications. This allows developers and researchers to create various interactivescenarios and utilize the robot's vision capabilities to enhance its functionality.
3) Capture Red Ball Image: The Nao robot is equipped with a vision system that allows it to perceive and interactwith its environment. The vision system consists of cameras mounted on the robot's head, providing visual input for varioustasks. These cameras provide a resolution of up to 2560*1920 at 1 frame per second (fps) or 640*480 at 30 fps. They can be used to identify objects in the visual field such as goals and balls, the top camera provides a forward-facing view and the bottom camera can ease Nao's dribbles andfacing downward [12]. The field of view (FOV) of the Nao robot's cameras can vary depending on the specific model and configuration. As of the Naoqi 2.8 documentation, the technical specifications for the Nao V6 are given in Table II.

Table II: Technical specifications of the Nao V6 Robot

Camera	Model Type	OV5640 System-on-a-chip (SoC)sensor
Imaging Array	Resolution Optical format Active Pixels (HxV)	5Mp 1/4 inch 2592x1 944
CPU	CPU Processor Cache memoryClock Speed	ATOM E3845 2MB 1.91 GHz
RAM		4GB DDR3
Flash Memory		32GB eMMC
View	Field of view Focus type	67.4°DFOV (56.3°HFOV,43.7° VFOV) Auto focus

For this experiment, parameters for the robot are calibrated as; Camera-ID = 18, Camera-Resolution = 320x240 at 30 frames persecond and colorSpaceID = 1 (RGB Colorspace). The bottom camera's field of view is typically wider, covering a larger area. Shown in Figure 3 and Figure 4.

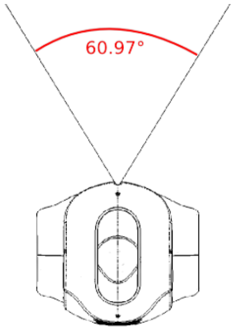


Fig. 3: The Nao robot head with top camera field of view.

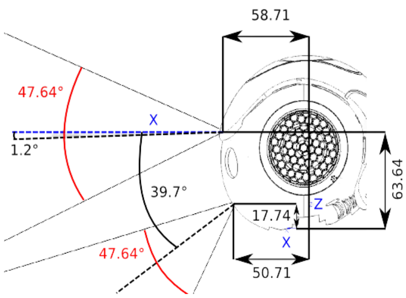


Fig. 4: The Nao robot head with side camera field of view.

IV. Experiments And Results

In Standard Platform League (SPL), all participating teams use identical robots, specifically the Nao robot. As a result, teams primarily focus on software development. These robots function autonomously. In this experiment, we utilized Naoqi, a Python SDK, along with various APIs and modules such as ALVideoDevice, ALMotion, and ALTracker to program the robot. The objective was to enable the robot to detect a red ball, reach to it, and perform a kicking action. Additionally, a soccer field was created specifically for this experiment. Moreover, Figure 5 displays the soccer field view.

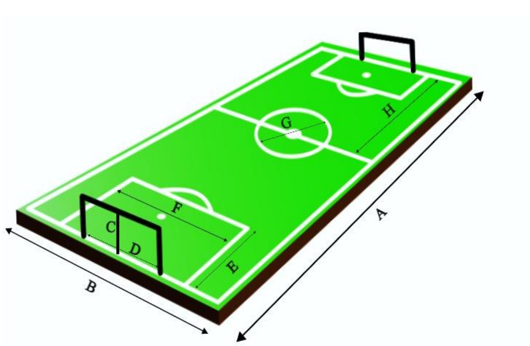


Fig. 5: The Nao robot soccer field view.

Table III: The Nao robot soccer field dimensions

Label	Name	Dimension (inches)
A	Field length	288
B	Field width	144
C	Goal height	36
D	Goal length	56
E	Goal area length	60
F	Goal area width	72
G	Central circle diameter	16
H	Half of the field length (A/2)	144

If red ball is placed between 0.5 to 4 feet within robot frame of vision, the Nao robot start tracking the red ball. Once the robotapproaches the red ball at a distance about 0.08 meters or 0.8cm, it halts its movement. The Nao robot employs its bottom camera to capture an image of the ball. The analysis of this image allowsthe robot to identify the red ball's presence and determineits position. a captured image is then stored for subsequent processing. Figure 6 presents the outcomes obtained from the Nao's bottom camera.

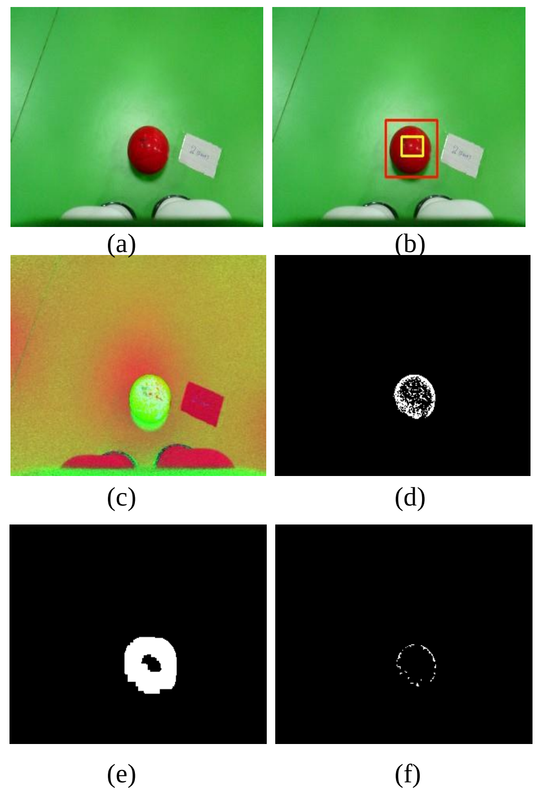


Fig. 6: Utilizing Nao's bottom camera to capture red ballimages (a) RGB ball image (b) ball detected image (c) HSV ball image (d) binary ball image (e) dilate ball image (f) erodeball image

And then Nao robot employs its forehead camera to detect the goal and to capture an RGB red ball image. This image is initially converted into the HSV (Hue SaturationValue) color space, and applied image processing filters such as erosion and dilation. The robot acquires this goal image about distance of 14 feet.

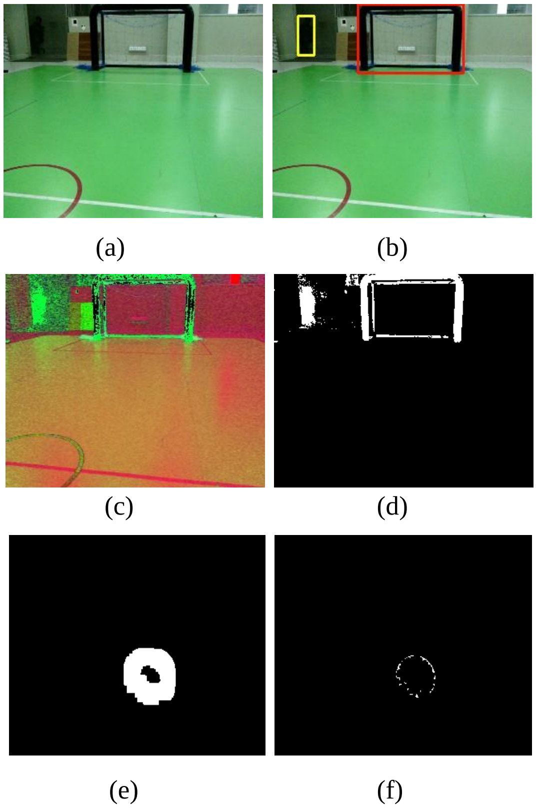


Fig. 7: Utilizing Nao's top camera to capture goal images (a) RGB goal image (b) goal detected image (c) HSV goal image (d) binary goal image (e) dilation goal image (f) erosion goalimage

V. Kick motion generation of the Nao robot using Choregraphe software

The Nao robot kick motion is generated using Choregraphe software by following steps.

I. Launch Choregraphe Software: Run Choregraphe software on your computer, ensuring that you have either version 2.5.10.7 or 2.8.6.23 installed.

II. Select timeline box: Inside a Choregraphe, locate and select the timeline box. This tool allows you to create and manage motion sequences.

III. Choose Kick Direction: Determine whether you want to generate a kick motion for the left or right leg of the robot.

IV. Set Frame Rate: When creating key-frames for the kick motion, be mindful of the frame rate. key-frames generated at 40 frames per second (fps). Remember that a higher frame rate results in a faster motion execution, so handle this setting with care [10]. Generating kick motions can be challenging task if using programming languages. To simplify this process, Choregraphe proves to be the most effective tool. By using the timeline box feature inside the software, you can create precise motion sequences. After creating motions, they can be export to programming languages such as. Python and C++ for further process [10]. The process of generating left kick motion of the robot is shown in Figure 8. After creating left kick motion, for right kick just right click any of the key-frames and click "Flip" option. That will help to create right kick motions straight away.

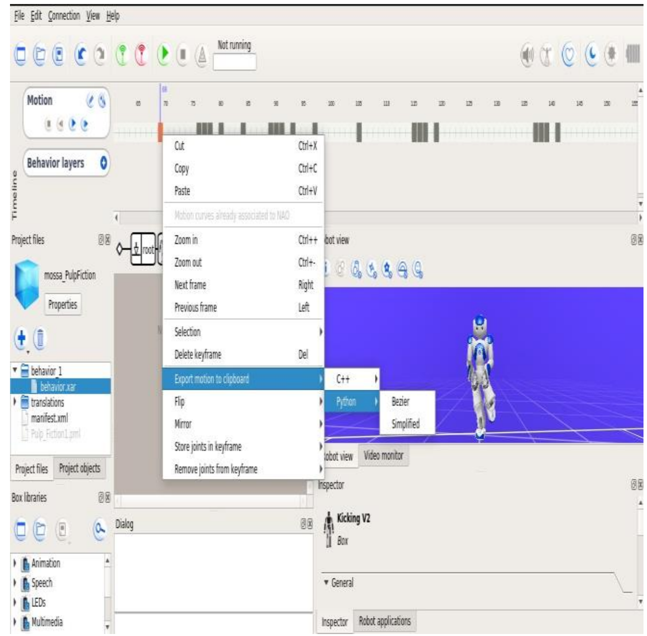


Fig. 8: Left kick motion generation in Choregraphe software.

Motions can be exported in Python by following steps:

I. Export to Python: To export the motion, right-click on any of the frames you have created. A window will pop-up. Choose the" export to clipboard" option. II. Select Python: In option "export motion to clipboard", select Python and then "Simplified" option as shown in Figure 8. III. Copy and Paste: Motion key-frames can be copied to your clipboard. Then paste them into preferred Python integrated development environment (IDE) for next step. Furthermore, the motions exported from Choregraphe, help us to plot the trajectories of the robot joints. Figure 9 and 10 shows these trajectories, representing the right and left kicks respectively.

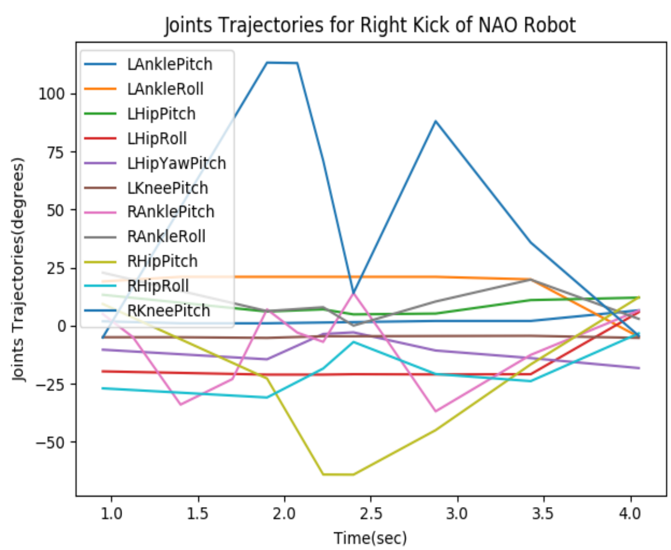


Fig. 9: The Nao robot right kick joints trajectories.

Let's analyze graph shown in Figure 9. This graph has time in (seconds) on x-axis and the angle in(degrees) on y-axis. This graph represents the movement of the eleven joints. Let's take one joint as an example for an understanding. Joint named a "RKneePitch" a cyan colored joint has 0-degree value at time 1 seconds. At time between 1.9 seconds to 2.1 seconds the joint degree value is around 120 and so on. In Python we created a time list and a key list for each joint. The times list represent the time points at which key-frames occur. The keys list store joint values.

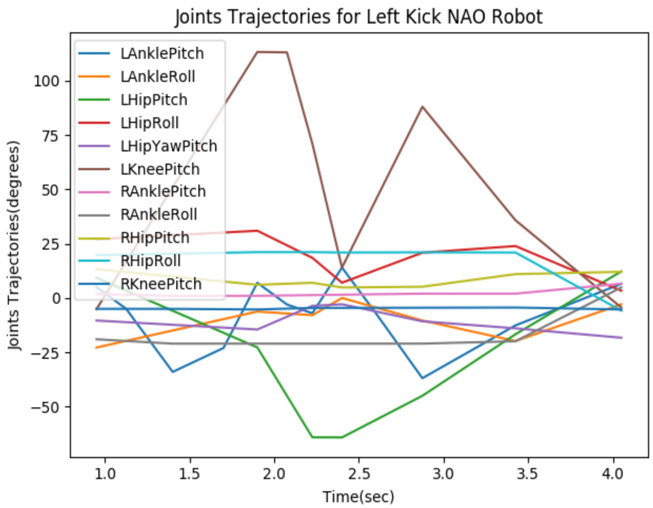
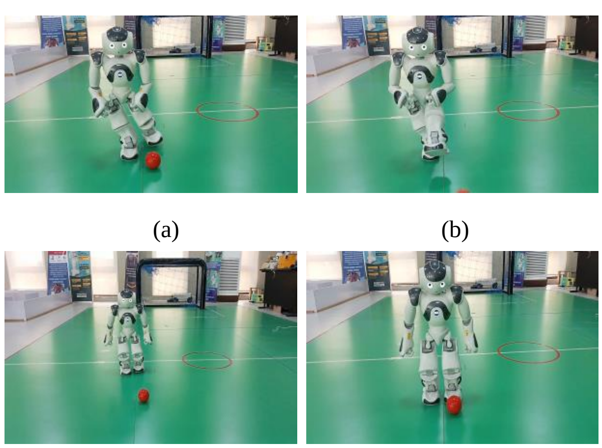


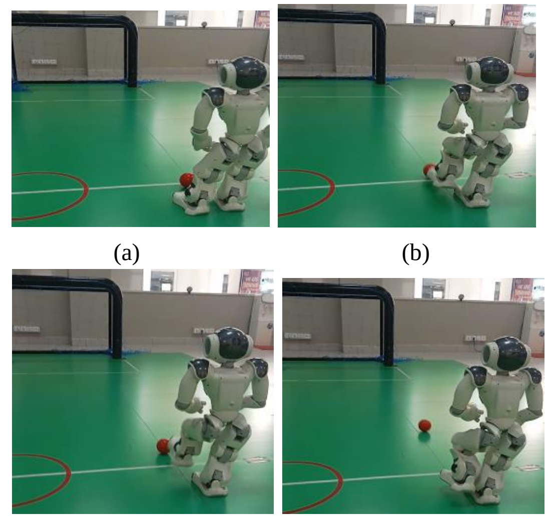
Fig. 10: The Nao robot left kick joints trajectories.

Now Let's analyze graph shown in Figure 10. This graph has time in (seconds) on x-axis and the angle in (degrees) on y-axis. This graph also represents the movement of the eleven joints. Let's take one joint as an example for an understanding. Joint named as" LKneePitch" a brown colored joint has nearly 0-degree value at time 1 second. At time between 1.9 seconds to 2.1 seconds the joint degree value is around 120 and so on. The keys list contains the corresponding key frame data for the movement of the" LKneePitch" joint. Figure 11 shows the key-frames of the actual field placed Nao robot executing it's left kick to place the red ball into the goal.



(a) (b)

(c) (d)



(a) (b)

(c) (d)

Fig. 11: The Nao robot executing 3 different key frames" HipPitch"," KneePitch", and" AnklePitch" with front and back view in the field (a)(e) robot detected the red ball (b)(f) robot move close to the red ball (c)(g) robot adjust itself accordingto the position of the red ball and executing the left kick (d)(h)robot kick the red ball into the goal

VI. Conclusion

Through careful experimentation and analysis, this study has highlighted the Nao robot's capacity to acquire soccer skills, from perceiving the environment to interacting with the ball. The successful integration of computer vision techniques has empowered the robot to interpret visual cues, anticipate movements, and execute precise actions. The developed soccerskills of Nao robots carry promising prospects in fields like education, entertainment. These skills can serve as interactive tools for teaching complex concepts, engaging in collaborative human-robot interactions, and contributing to the developmentof robotic companions. However, there are few drawbacks of Python using in soccer application such as: Python is an interpreted language, which means it can be slower compared to languages like C++ or Java. In applications that require real-time responsiveness, such as robot control in dynamic soccer scenarios, Python's performance may lead to delays or sub- optimal response times. We opted to work with Python as our programming language to facilitate our understanding and learning of hardware programming.

Acknowledgement

We extend our heartfelt gratitude to all those who have contributed to the successful completion of this research paper on" Development of The Nao Robot's Remarkable Soccer Skills". This journey has been marked by collaborative efforts,invaluable guidance, and unwavering support from numerous individuals and institutions. We extend our thanks to Sukkur IBA University, Specially Center of Excellence for Robotics, Artificial Intelligence and

Blockchain (CRAIB) providing us with the necessary resources and facilities that have enabled us to conduct experiments, and analyze results. The research environment and access to cutting-edge technologies have been instrumental in the progress of this work. We express our deepest appreciation to our supervisor and mentor Dr. Ghulam Mujtaba Shaikh, for their constant guidance and insightful feedback throughout writing this paper endeavor. Our gratitude extends to the members of our research team who have contributed their expertise and dedication to various aspects of this project. We also would like to thank our families and friends for their motivation, unwavering encouragement and understanding during the long hours invested in this work.

REFERENCES

[1] Aida Amirova, Nazerke Rakhymbayeva, Elmira Yadollahi, Anara Sandygulova, and Wafa Johal. 10 years of human-nao interaction research: A scoping review. *Frontiers in Robotics and AI*, 8:744526, 2021.

[2] Kaustubh Bhowmick, Rajat Arya Sachin Choubey, and Prachi Purbia. Development of static kick by nao. pages 3167–3172, 2020.

[3] J Boonen, M Houbraken, O Lehmacher, J Robers, and G Theunissen. Research and development of a new robot soccer platform. 2016.

[4] Widodo Budiharto, Anggita Dian Cahyani, Pingkan CB Rumondor, and Derwin Suhartono. Edurobot: intelligent humanoid robot with natural interaction for education and entertainment. *Procedia computer science*,116:564–570, 2017.

[5] E Chatzilaris. *Visual-feature-based self-localization for robotic soccer*. PhD thesis, Diploma thesis, Technical University of Crete, Greece, 2009.

[6] Leonardo Leottau Forero, José Miguel Yánez, and Javier Ruiz-del Solar. Integration of the ros framework in soccer robotics: the nao case. In *RoboCup 2013: Robot World Cup XVII 17*, pages 664–671. Springer, 2014.

[7] Tomás González Sánchez. Artificial vision in the nao humanoid robot. Master's thesis, Universitat Politècnica de Catalunya, 2009.

[8] Roy Chaoming Hsu, Yu-Pi Lin, Chi-Jun Lin, and Li-shun Lai. Humanoidrobots for searching and kicking a ball and dancing. In *2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*,pages 385–387. IEEE, 2020.

[9] Mohd Azfar Miskam, Syamimi Shamsuddin, Hanafiah Yussof, Abdul Rahman Omar, and Muhammad Zaiyad Muda. Programming platform for nao robot in cognitive interaction applications. In *2014 IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pages 141–146. IEEE, 2014.

[10] Emmanuel Pot, Jérôme Monceaux, Rodolphe Gelin, and Bruno Maisonnier. Choregraphe: a graphical tool for humanoid robot programming. In *RO-MAN 2009-The 18th IEEE International Symposium on Robotand Human Interactive Communication*, pages 46–51. IEEE, 2009.

[11] Syamimi Shamsuddin, Luthffi Idzhar Ismail, Hanafiah Yussof, Nur Is marrubie Zahari, Saiful Bahari, Hafizan Hashim, and Ahmed Jaffar. Humanoid robot nao: Review of control and motion exploration. In *2011 IEEE international conference on Control System, Computing and Engineering*, pages 511–516. IEEE, 2011.

[12] Chaoyue Wang. Behaviour design of nao humanoid robot playing tictac toe game. 2015.

[13] Suresh Kumar, M Majeedullah, AB Buriro, et al. Autonomous navigation and real time mapping using ultrasonic sensors in nao humanoid robot. Engineering, Technology & Applied Science Research, 12(5):9102-9107, 2022.

[14] Rayana Antar, Shimaa Alghamdi, Johara Alotaibi, and Manal Alghamdi. Automatic number plate recognition of saudi license car plates. Engineering, Technology & Applied Science Research, 12(2):8266–8272, 2022

[15] Boucetta Kasmi and Abdelouaheb Hassam. Comparative study between fuzzy logic and interval type-2 fuzzy logic controllers for the trajectory planning of a mobile robot. Engineering, Technology & Applied Science Research, 11(2):7011–7017, 2021