IES SMC EdSoc

IEEE TSYP 12

Technical Challenge

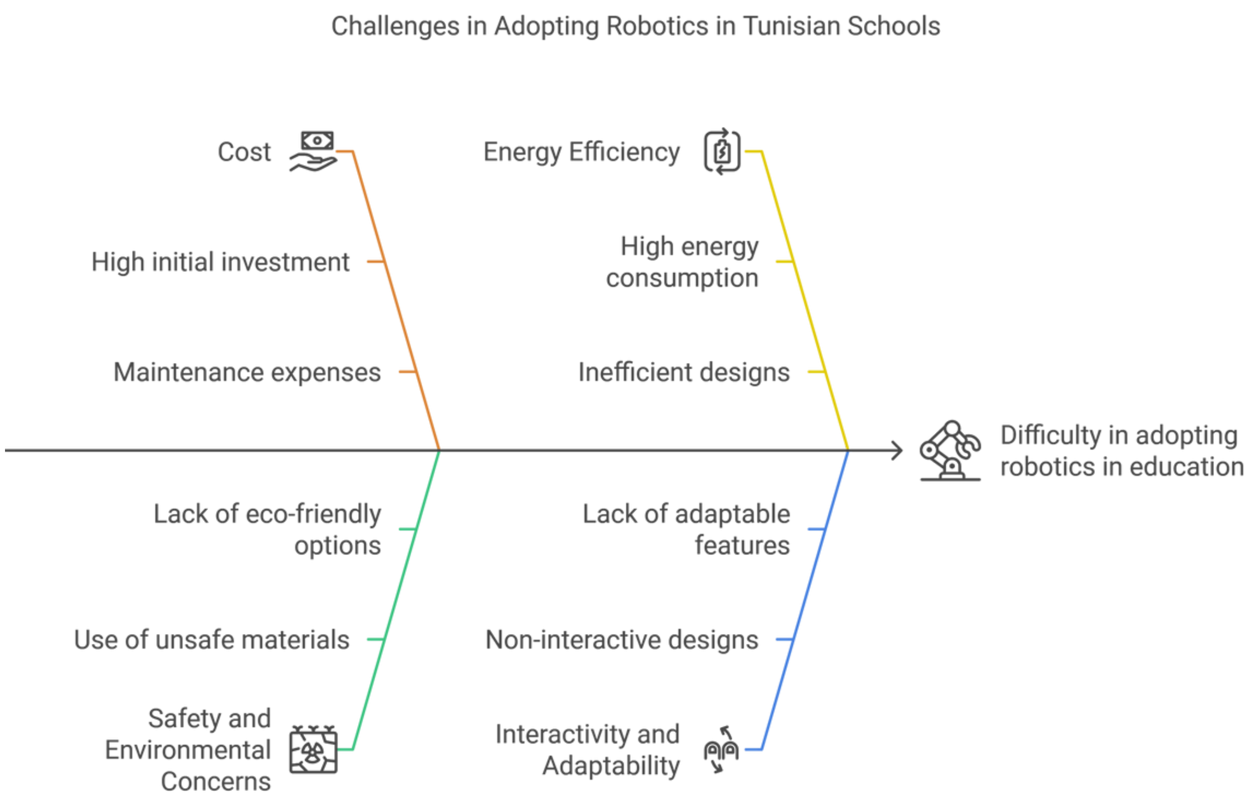
CHESS COACH ROBOT

IEEE ENET'Com SB

TABLE OF CONTENTS

I-Introduction

1.Problem statement

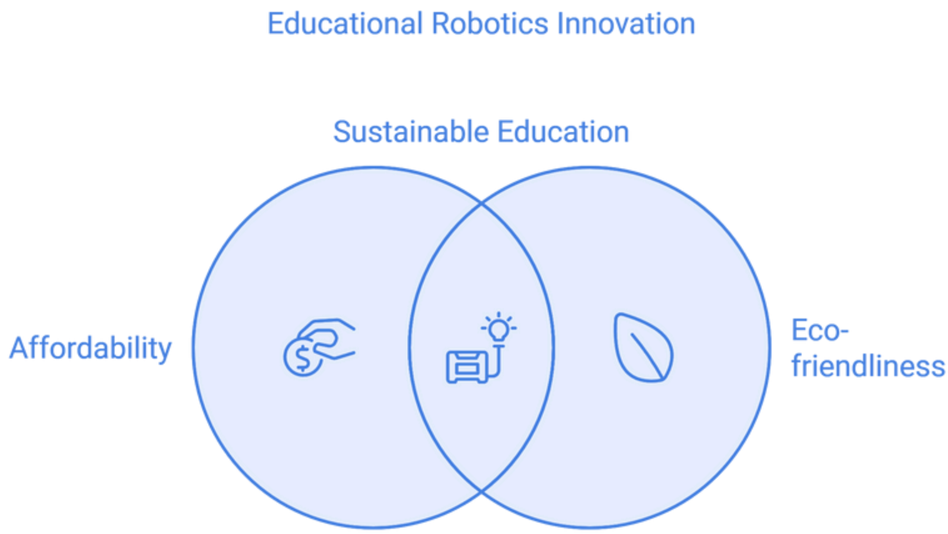


1. **In Tunisia,** the use of robotics in education is limited, despite its potential to enhance learning by combining technology with interactive teaching methods. The lack of adoption is due to several key issues:
2. **High Costs:** Most robotic systems are expensive, making them inaccessible to the majority of educational institutions, particularly in underserved regions.
3. **Limited Sustainability:** Many robotic solutions use materials and energy systems that are not environmentally friendly or cost-effective, raising concerns about long-term sustainability.
4. **Safety Concerns:** Current robotics designs often fail to prioritize safety, particularly for young students. Unsafe materials or mechanisms could pose risks in a classroom setting.

5. **Lack of Interactivity:** Existing robotics applications in education do not engage students effectively, reducing their value as teaching tools. Traditional designs often focus more on technical performance than on interactive and pedagogical elements.
6. **Inflexible Designs:** Robotics systems are often rigid in their design and purpose, limiting their adaptability to diverse educational settings and learning objectives.

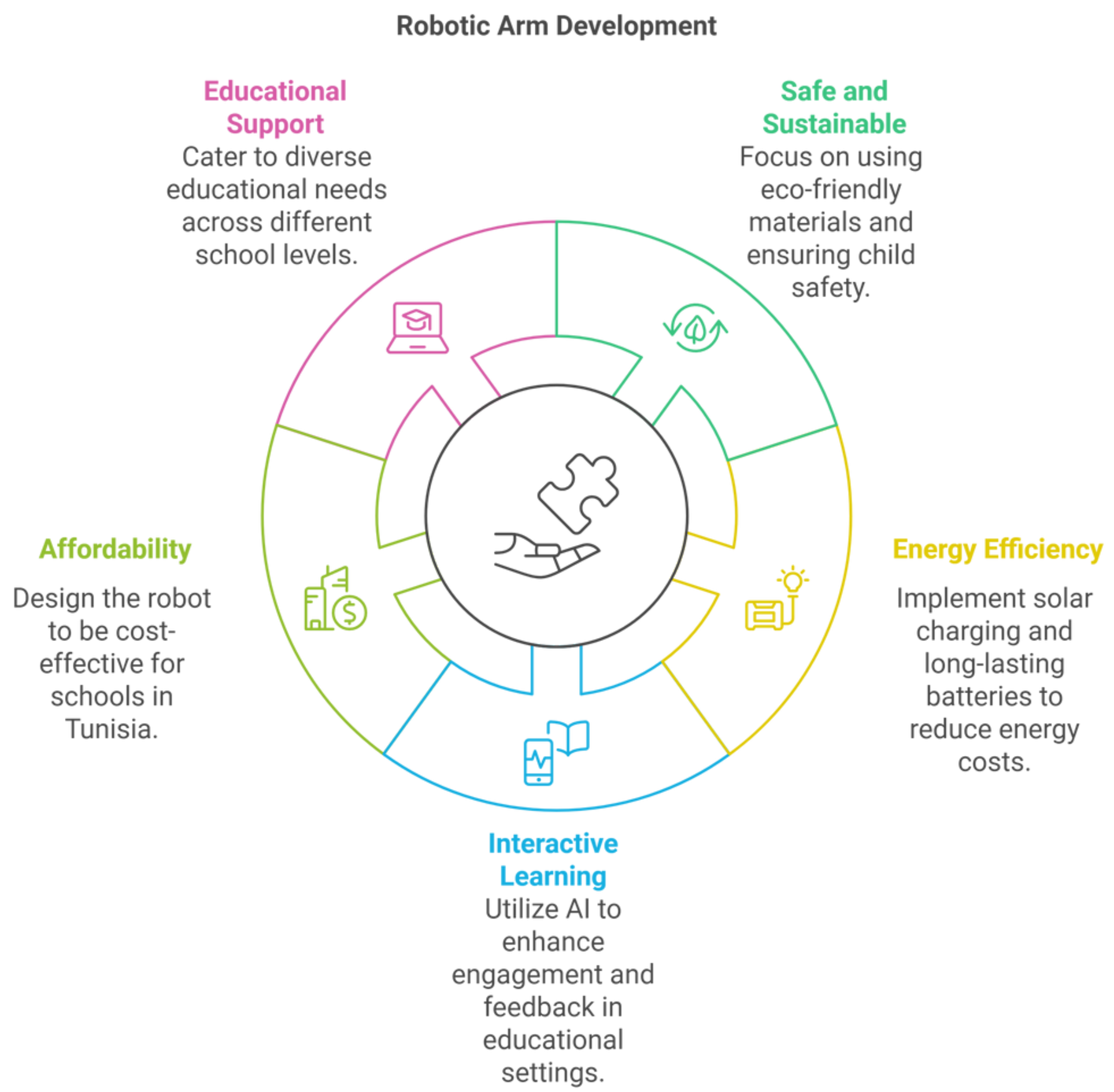
These barriers prevent schools from integrating robotics into their curriculums, leaving students with fewer opportunities to engage with cutting-edge technology and develop skills in STEM fields.

2.Orientation



Our orientation for this problematic aims to develop an affordable, safe, and eco-friendly robotic arm designed specifically for educational purposes. This robotic arm will serve as an interactive chess coach, blending AI technology and robotics with engaging teaching methods. The robot will address the identified challenges through a focus on cost-effectiveness, sustainability, interactivity, and adaptability, ensuring it is suitable for use across schools in Tunisia.

3.Project objectives



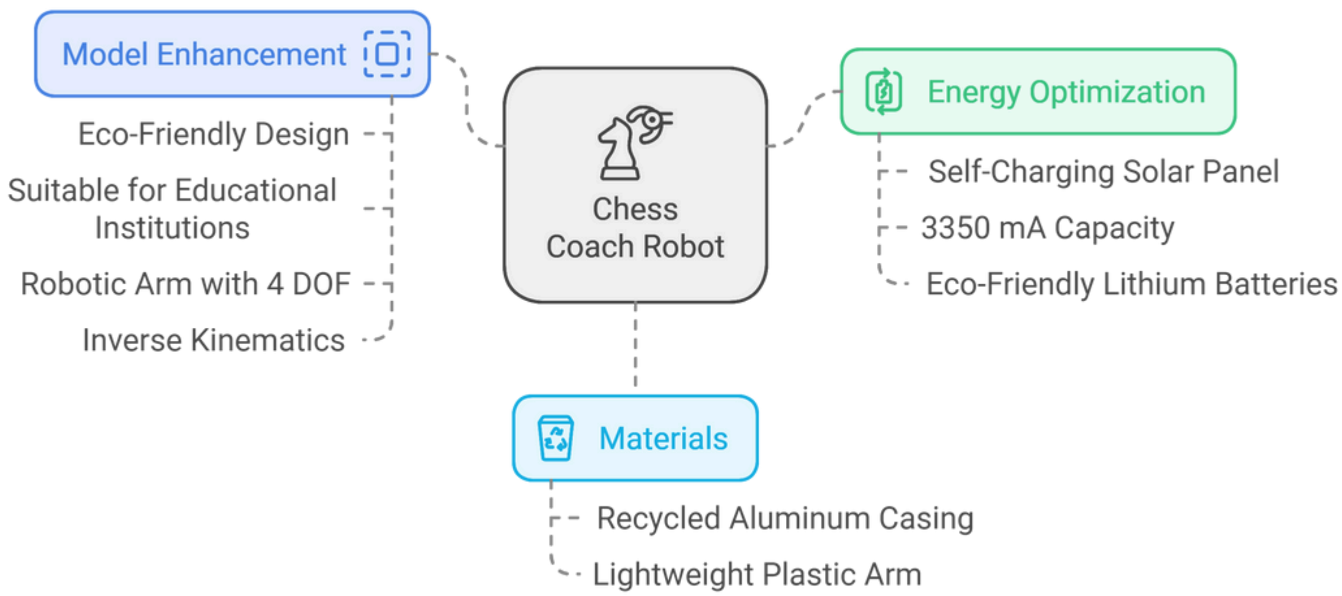
- **Build a Safe and Sustainable Robotic Arm:**
  - Use recycled materials like aluminum and lightweight plastics that are safe for children and better for the environment.
  - Make the robot durable and energy-efficient.
- **Create an Energy-Efficient Design:**
  - Include a solar-powered charging system to reduce energy costs.
  - Use long-lasting lithium batteries to store power.

- **Make Learning Fun and Interactive:**
  - Use AI to move chess pieces, give feedback, and teach strategies during the game.
  - Ensure the robot provides real-time feedback to keep students engaged.
- **Focus on Affordability and Accessibility:**
  - Design the robot to be low-cost so schools in Tunisia can afford it.
  - Make it flexible and easy to use in different educational settings.
- **Support Education Across Tunisia:**
  - Create a robot that fits the needs of schools of all levels, from elementary to high school.
  - Help students learn chess and STEM skills at the same time.

II-Solution overview

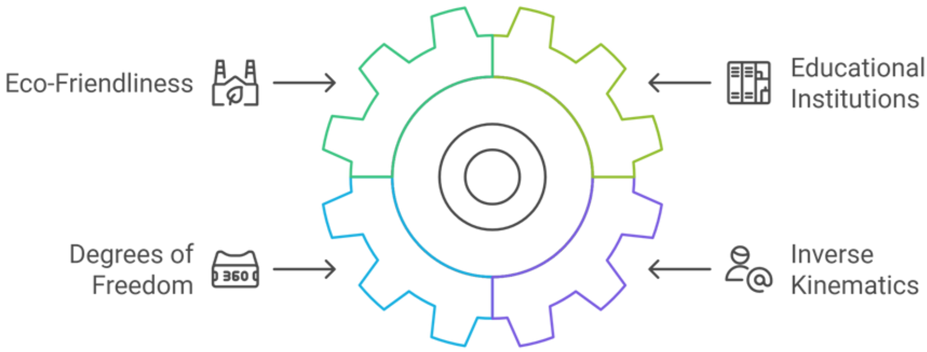
1.Introduction

Robotics and artificial intelligence are transforming industries by enabling innovative solutions to complex problems. This project integrates robotics, AI, and machine learning to address challenges in automation, strategic gameplay, and personalized education.



By combining a ball-sorting robot, a chess-playing and coaching robot, and a system for adaptive learning, this work explores the synergy between human-machine interaction and intelligent systems. The project emphasizes practical applications of inverse kinematics, trajectory generation, AI-driven decision-making, and machine learning-based personalization, contributing to advancements in technology and education.

2. Model enhancement



- Model enhancement was a critical step in our project, as we focused on refining the design and functionality of the systems we developed. For the Chess Coach Robot, we enhanced the construction by carefully selecting superior materials to ensure durability and precision while incorporating safety features to create a student-friendly design. These changes improved both the reliability and usability of the robot in educational settings.
- In terms of energy optimization, we explored and integrated renewable energy solutions, including the innovative "Smart Sunflower" system, to intelligently harvest energy and improve the robot’s sustainability. This allowed us to align the system with modern energy efficiency standards.

- For the AI-driven functionalities, we worked on optimizing decision-making algorithms, ensuring faster and more accurate move calculations. We also enhanced the robot's ability to detect illegal moves and provide detailed feedback, making the gameplay interaction seamless and educational.
- Additionally, we improved the learning models by refining the data pipeline for processing chess game data. By implementing robust classification models and analyzing game patterns, we enhanced the system’s ability to offer personalized learning experiences. These efforts enabled the model to adapt to students' individual needs, making it an effective tool for teaching strategy and tactics.

a.Energy optimization

- Developed a power bank with a self-charging solar panel.
- 3350 mA capacity power bank with 6 lithium batteries.
- Lithium batteries were chosen for their eco-friendliness, safety, and long lifespan.

Solar energy rechargeable power bank

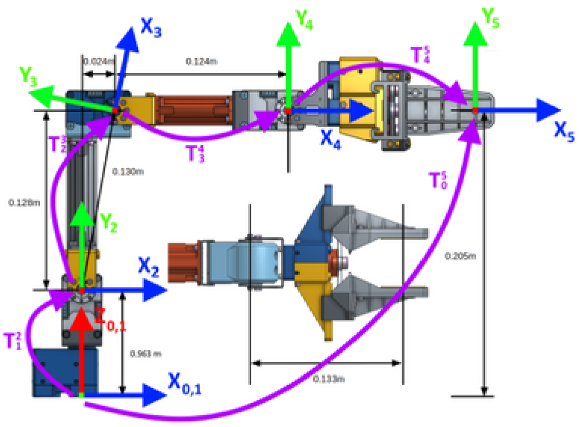
Low energy consumption componets

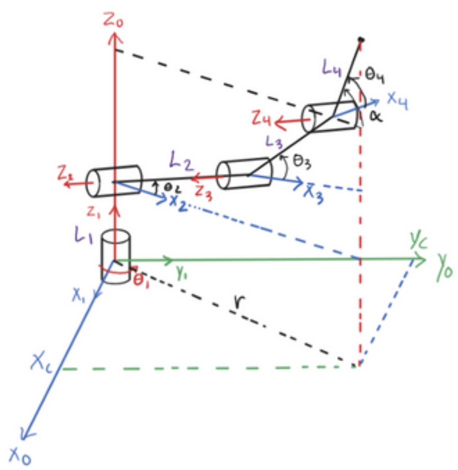
b.Materials

For the body of the robotic arm, we utilized ABS plastic due to its strength, durability, and resistance to high temperatures. These properties ensure that the robot can withstand prolonged use in educational settings without compromising safety. Additionally, ABS's availability in various colors and finishes allowed us to design an aesthetically appealing robot, making it more engaging for students.

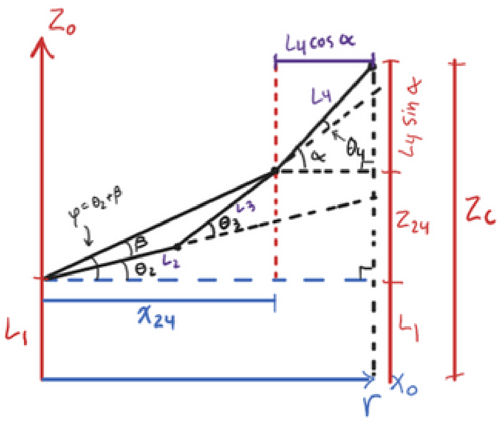
c.Robotic hand mechanics

The robot is tasked with locating the chess pieces and determining the trajectory to grab and sort the different colors. This project required us to explore and implement inverse kinematics, trajectory generation, and a robust computer vision system.

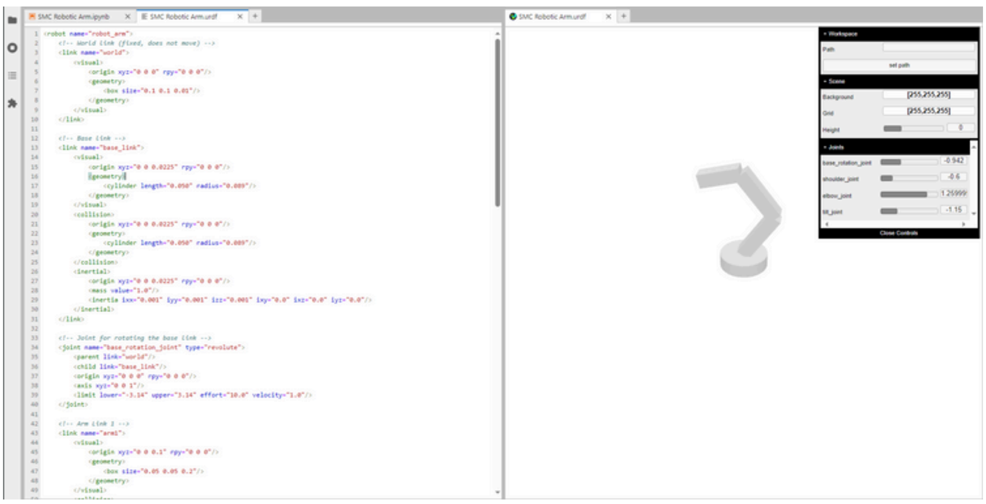




Calculating the inverse kinematics is a lot more complicated because there might be multiple or even infinite solutions for a single position. There might even be no solutions if the position isn't reachable. But ,the general idea is to use trigonometry to calculate possible joint angles ,and then plug those values back into the forward kinematics to see which configurations actually work.
Since the given position has three coordinates (x, y, and z) and there are four unknown joint angles, the angle α is also provided to make the problem solvable where α is the angle between the end-effector and the xy-plane

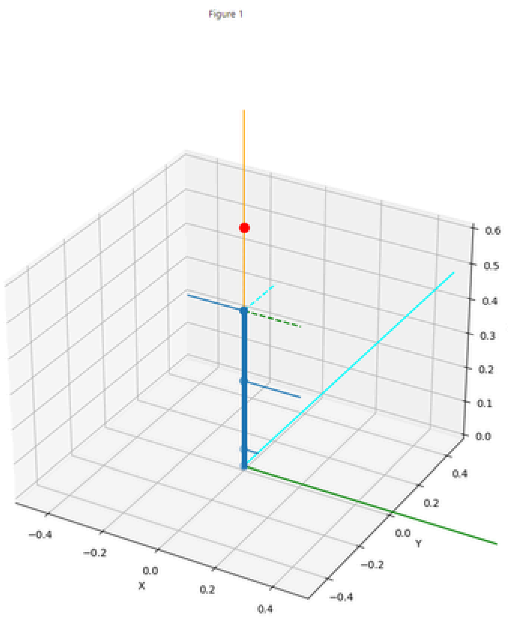


We created a URDF file defining the 4 degrees of freedom for our robotic arm, including the links and joints. This file will be used to simulate the arm's movements along the x, y, and z axes, serving as a foundation for the next step: implementing inverse kinematics.

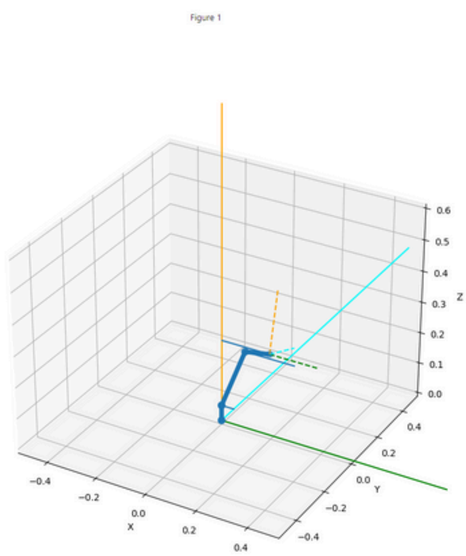


Next, we created an IPython Notebook using the Python programming language in Jupyter Lab. This notebook serves as the main file controlling the movement of the robotic arm. The simulation defined in the URDF file acts as a chain in this notebook. Any change in the variables x,y,z will compute the required joint rotations, which correspond to the servo motors in the functional prototype.

```
target_position = [ 0,0,0.68]

target_orientation = [-1,0,0]
```



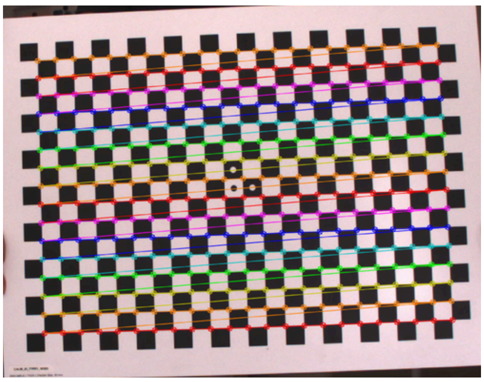

```
target_position = [ 0,3,-2]

target_orientation = [-1,0,0]
```

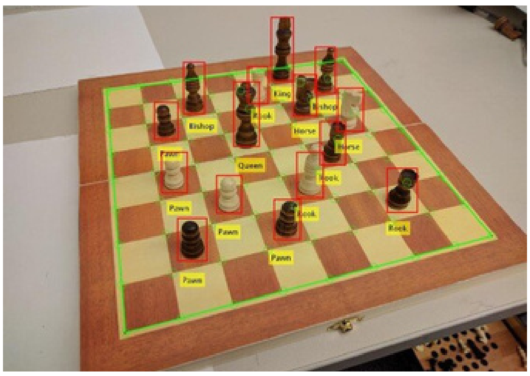


d.Camera integration

- Camera Calibration:
- The camera is calibrated to correct distortions and capture accurate representations of the environment.
- This step ensures the images used for analysis are geometrically accurate.
- Image Processing with the Model:
- The calibrated camera captures images of the environment or objects.

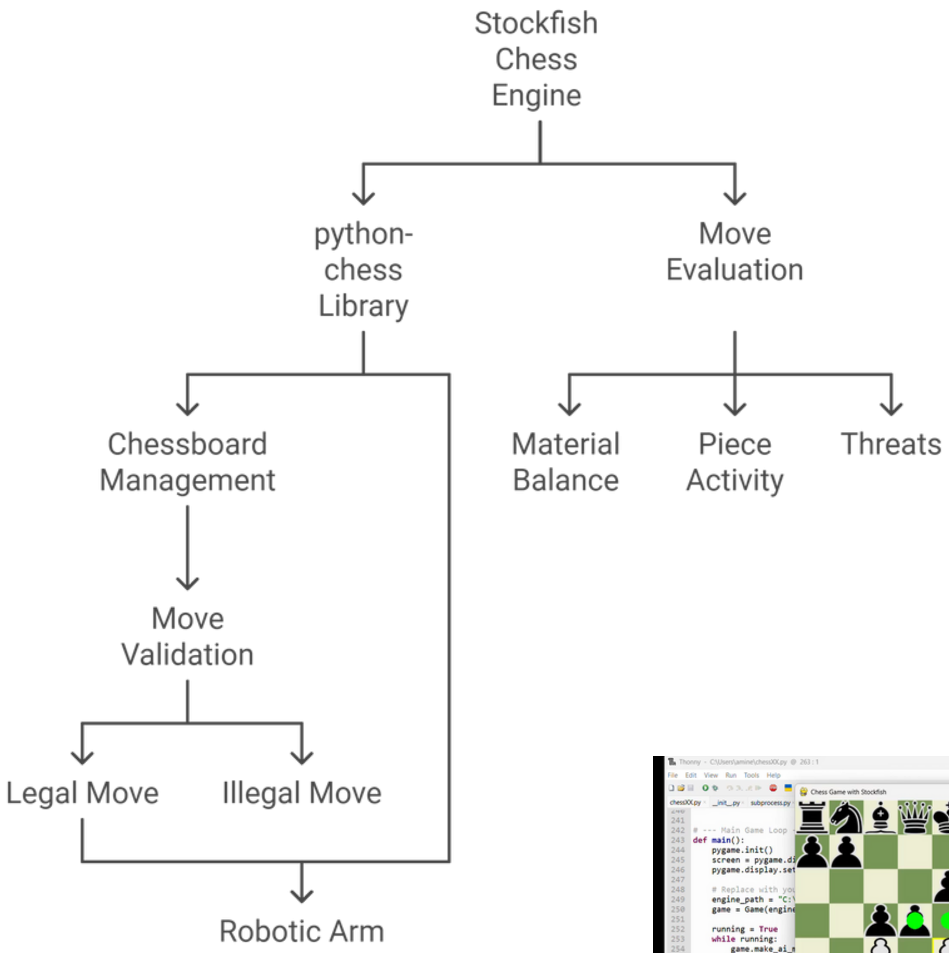
- The model processes these images to extract essential information (e.g., object positions, orientations).
- Data Interpretation:
- The processed image data is converted into actionable insights, such as coordinates or movement vectors.
- Robotic Arm Control:
- The extracted information is sent to the robotic arm's control system (its "brain").
- The control system uses this data to decide how the robotic arm should move or interact with the environment.

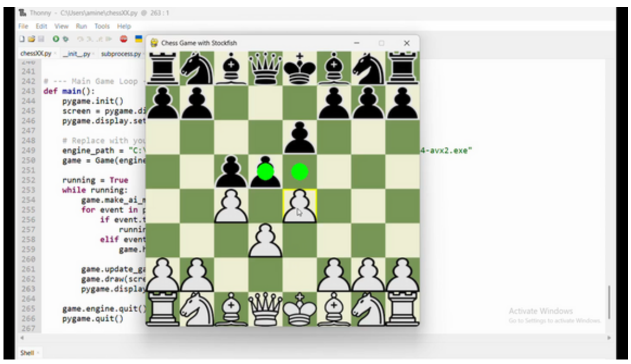




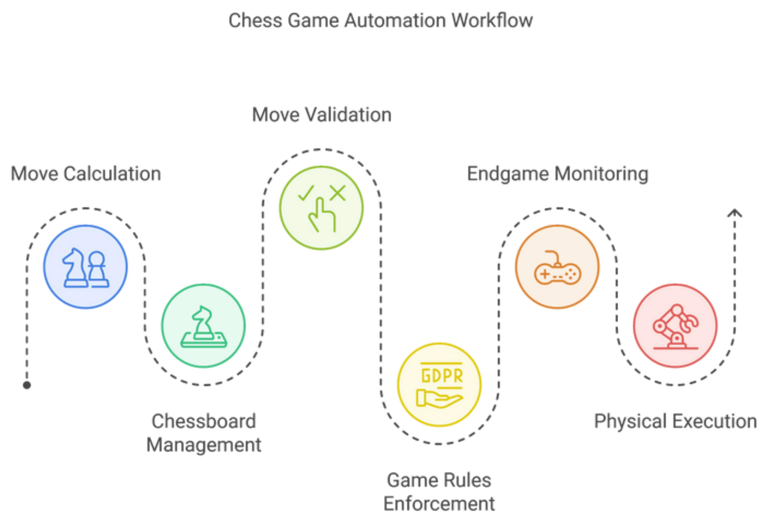
e.Chess API

- System Components:
- Stockfish Chess Engine: Calculates the optimal moves based on the current game state.
- python-chess Library: Manages the chessboard state, legal moves, and interactions with Stockfish.
- Robotic Arm: Physically moves the chess pieces based on calculated moves.





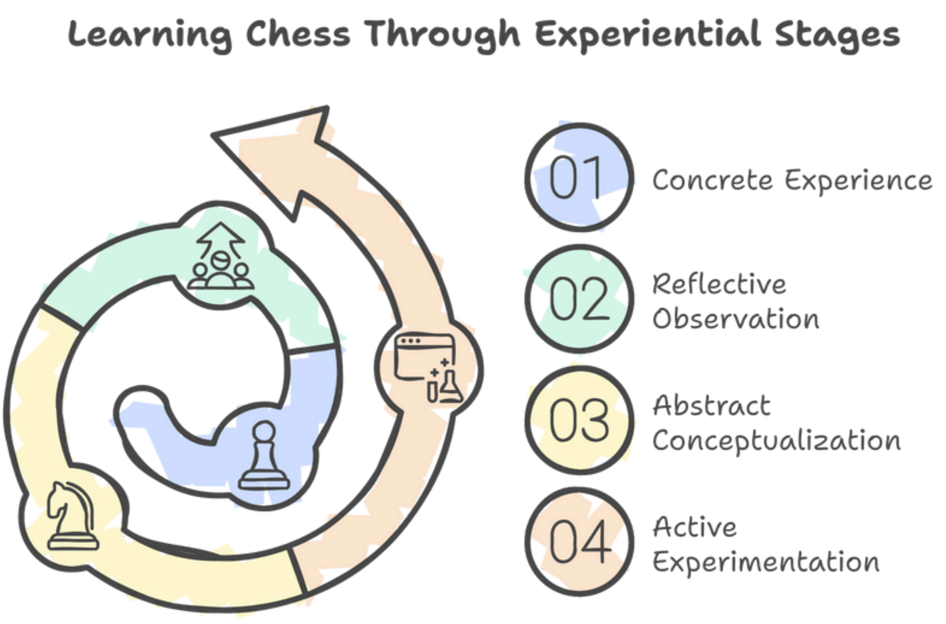
- Workflow:
- Move Calculation:
- Stockfish, accessed via UCI and python-chess, analyzes the board to determine the best move.
- Evaluates factors like material balance, piece activity, and threats.
- Chessboard Management:
- The chess.Board() class tracks the positions, turn order, and special conditions (e.g., checkmate).
- The board state updates after each move.
- Move Validation:
- The system generates and checks legal moves using python-chess.
- Illegal moves (e.g., placing the king in check) are automatically rejected.
- Game Rules Enforcement:
- Ensures all moves follow chess rules.
- Prevents illegal moves like invalid piece movement or moves putting the king in check.
- Endgame Monitoring:
- Detects checkmate, stalemate, and draw using methods like board.is_checkmate() and board.is_stalemate().
- Halts the system when the game ends and declares the outcome.
- Physical Execution:
- The robotic arm translates Stockfish's optimal moves into physical actions to move the pieces on the chessboard.



3. Learning Outcomes

a. Pedagogical Approaches

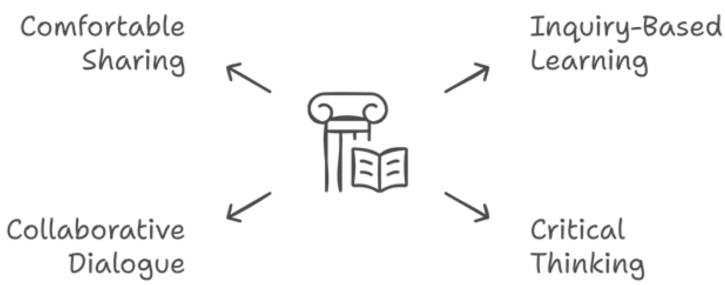
- Concrete Experience:
  - The kids interact with the chessboard, physically moving pieces, and observing gameplay, potentially aided by the robotic arm for demonstrations.
- Reflective Observation:
  - They analyze their moves and outcomes, considering why certain strategies worked or failed, and reflecting on patterns in gameplay.
- Abstract Conceptualization:
  - They connect their observations to broader chess concepts (e.g., openings, tactics, endgames), integrating these ideas into their understanding of the game.
- Active Experimentation:
  - The kids apply their new strategies in gameplay, testing hypotheses and refining their skills based on feedback from their moves and the robotic arm's responses.



- Socratic Method

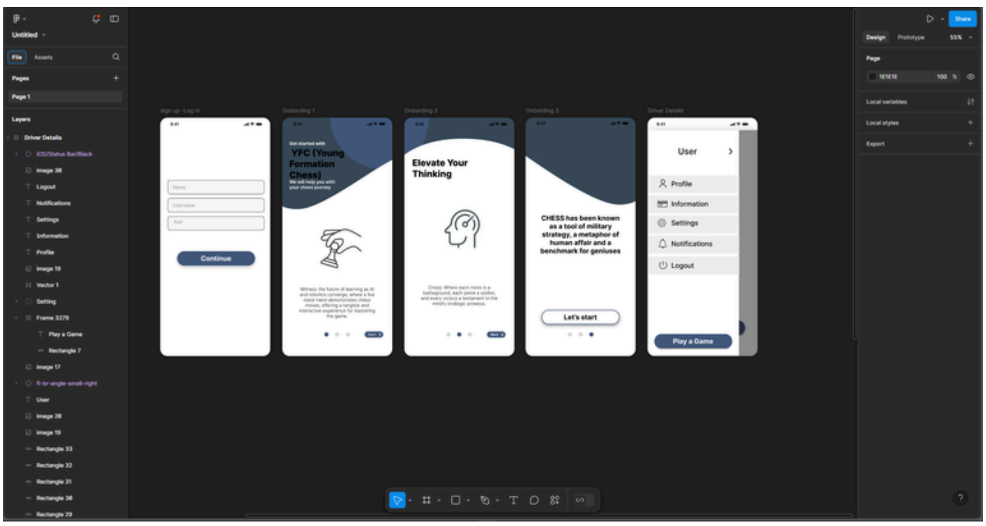
Socratic Method is a pedagogical approach based on inquiry where the trainer and the leaders are both responsible for moving the discussion and dialogue forward through questioning. Much like the philosopher Socrates, the trainer does not pose as an “authority figure” to fill the brain of the leaders with knowledge. He/she/they are much like facilitators and guide through the process. Trainers must have a keen eye to pose questions that would spark conversations and create an atmosphere where students would feel comfortable sharing and engaging into discussions.

Outcomes of Socratic Method



a. Communication

The mobile application is a user-friendly platform designed to facilitate communication and learning for kids playing chess. Developed using the Flutter framework, it ensures compatibility across multiple devices, offering a consistent and smooth user experience. The app features an intuitive and playful interface tailored to engage children and make chess learning enjoyable.



With Firebase as the backend, the application provides real-time interaction capabilities, secure user authentication, and reliable data storage. This allows kids to seamlessly connect, track their progress, and interact with the robotic arm or other features of the chess teaching system.

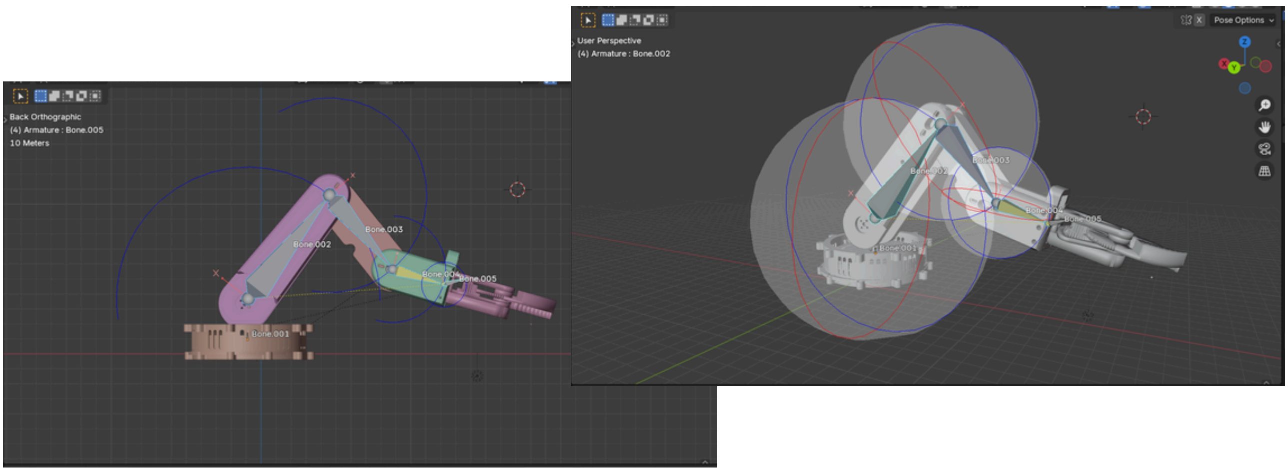
III-Prototype Overview

1.Architecture Design

The robot consists of a base (turning table), upper arm, lower arm, and gripper, all designed for independent movement with four degrees of freedom (DOF).

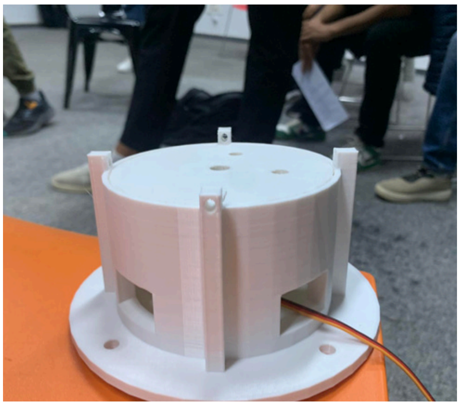
1. **Base (Turning Table)**: Provides stability and allows for rotational movement around a vertical axis, enabling the robot to turn and change direction.
2. **lower Arm**: Attached to the base, it moves along a horizontal plane to extend the robot's reach.
3. **Upper Arm**: Connected to the lower arm, it moves in specific directions to position the gripper accurately.
4. **Gripper**: Located at the end of the upper arm, it enables the robot to grasp, hold, or manipulate objects.

With four degrees of freedom, the robot can perform complex motions, offering flexibility and precision for tasks such as interacting with objects or playing chess.

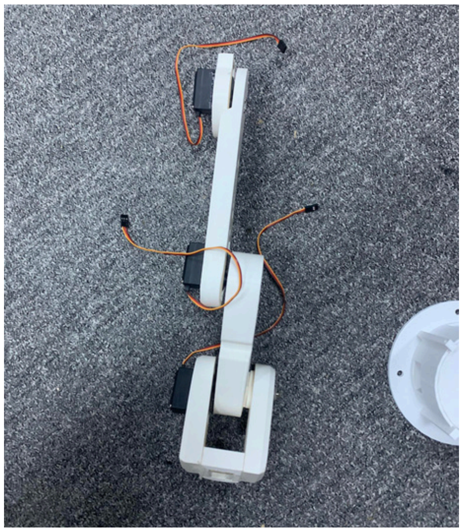


2.Components

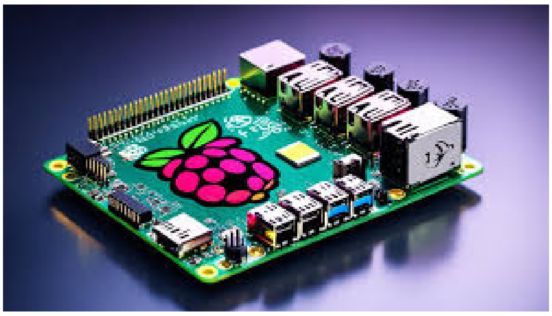
- **Base**



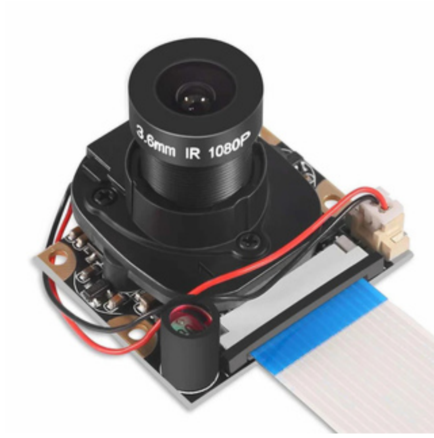
- **arm**



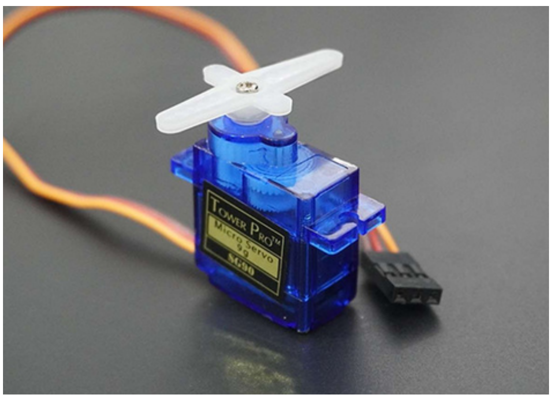
- **carte raspberry pi 4**



- **carte raspberry pi 4**



- **servo motor**



- **cables**

IV-Conclusion

This research explores the development of a cost-effective 4-degree-of-freedom robotic arm powered by renewable energy to teach chess to children. Using inverse kinematics, the design enables precise movements for chess piece manipulation. The concept integrates sustainability and affordability while promoting interactive learning. This study highlights the potential for accessible, eco-friendly robotics in education, laying the groundwork for future innovations.