RB310- Fundamentals of Robotics

Dr. Omar Shalash

Eng. Hossam Elsayed

Mimic Nao

Abdelrhman Saeed

Mostafa Magdy

Introduction

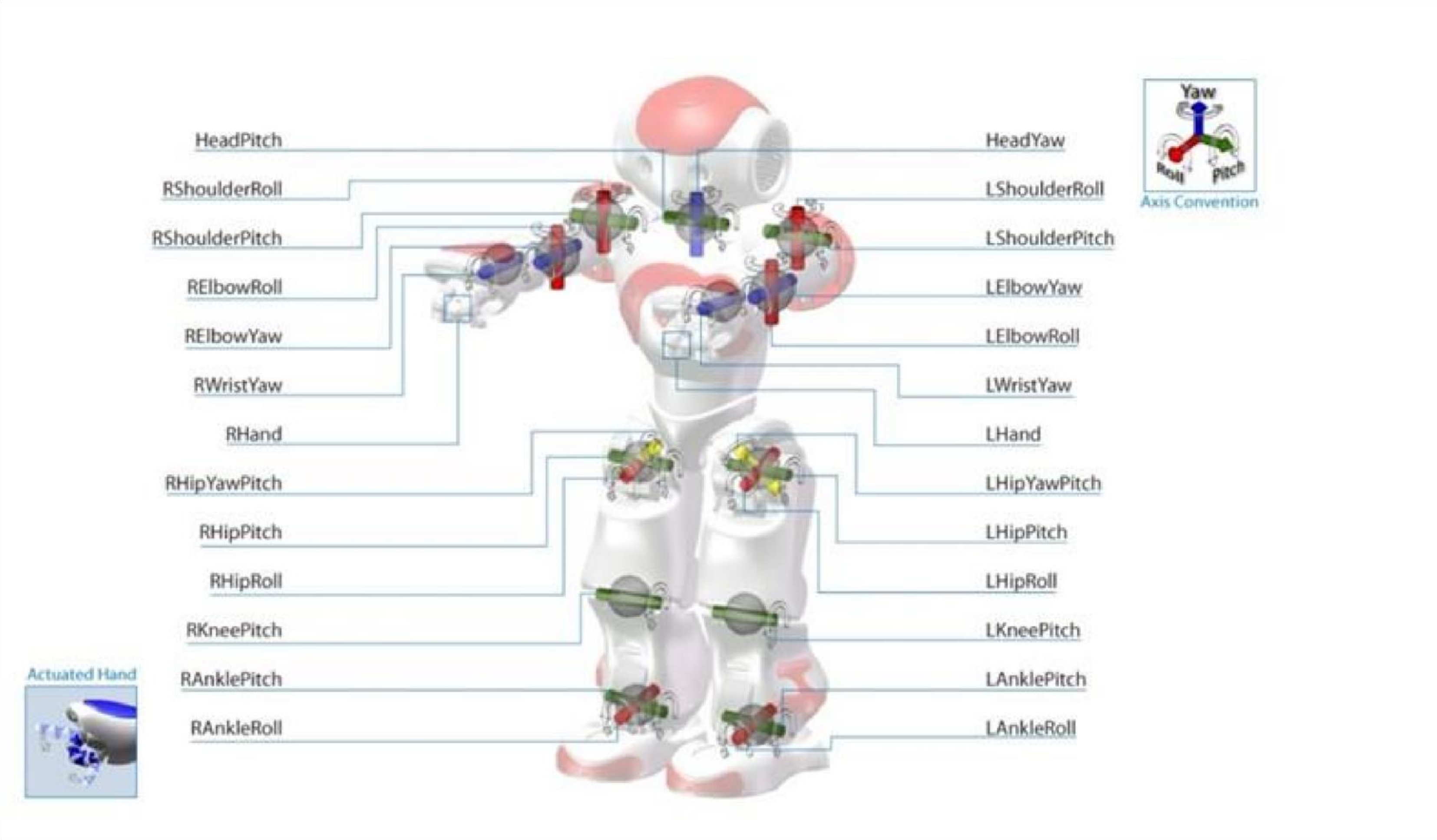
This project integrates robotics with computer vision to create an interactive experience using a NAO humanoid robot. The primary functionality is for the robot to mimic human movements based on pose estimation and to respond to touch interactions.

System Requirements

NAO Robot with NAOqi OS

Computer with Python 2.7 or 3.x

OpenCV (Python library)	MediaPipe (Python library)
Network connectivity between the computer and NAO robot	



Setup and Installation

Ensure that the NAO robot is set up and connected to the same network as the computer.

Install Python dependencies: OpenCV MediaPipe, and NAOqi Python SDK.

Robot Control Script

This script establishes a connection with the NAO robot and controls its actions.

Key Features:

Touch Interaction

Detects and responds to touch events on the robot.

Photo Capturing

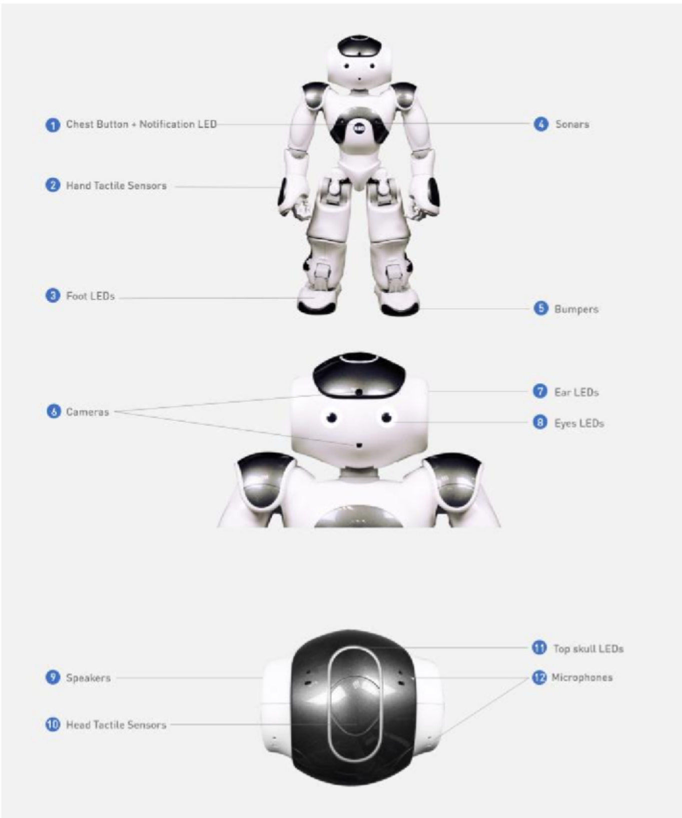
Captures and transfers photos from the robot to the computer.

Code for NAO control

[NAO SCRIPT]

Speech and Movement

Uses the robot's speech and movement capabilities.



Pose Estimation Script

Utilizes OpenCV and MediaPipe for real-time pose estimation from photos captured by the robot.

Key Features:

- Image Processing: Reads and processes images for pose estimation.
- Pose Analysis: Identifies human poses and extracts relevant data.
- Data Output: Writes pose data to a file for the robot to mimic.
- code: Pose_Estimation_Script

Workflow

1. **Robot Interaction**

   Initiate interaction through a touch event on the robot.

2. **Capture and Transfer**

   Robot takes a photo and transfers it to the computer.

3. **Pose Estimation**

   The computer script processes the image and performs pose estimation.

4. **Mimicry**

   The robot reads pose data and mimics the movements.

Usage

Start the robot control script on the computer connected to the NAO robot.

Interact with the robot (e.g., touch its head) to trigger actions.

Limitations and Considerations

Dependence on network connectivity for smooth operation.

Accuracy of pose estimation can vary based on image quality and lighting.

DH model

1 Forward Kinematics

Calculating the position of the end-effector (hand) of the robot's arm in relation to its joint angles.

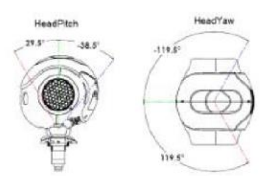
2 Inverse Kinematics

Determining the joint angles required to place the robot's hand at a specific position, such as the desired cup location.

3 Motion Planning

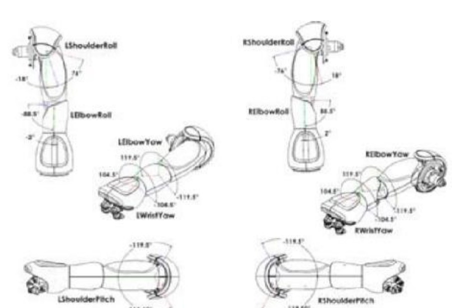
Developing a path planning algorithm to ensure smooth and accurate movements for the robot throughout the game.

4 Rang of joints



Joint Name	Range in Degrees°	Range in Radians
HeadYaw	-119.5° to 119.5°	-2.0857 to 2.0857
HeadPitch	-38.5° to 29.5°	-0.6720 to 0.5149

Figure 3.3: NAO head joints and their operational range



Joint Name	Range in Degrees°	Range in Radians
LShoulderPitch	-119.5° to 119.5°	-2.0857 to 2.0857
LShoulderRoll	-18° to 76°	-0.3142 to 1.3265
LElbowYaw	-119.5° to 119.5°	1.5446 to 0.0349
LElbowRoll	-88.5° to -2°	-0.6720 to 0.5149
RShoulderPitch	-119.5° to 119.5°	-2.0857 to 2.0857
RShoulderRoll	-38.5° to 29.5°	-1.3265 to 0.3142
RElbowYaw	-119.5° to 119.5°	-2.0857 to 2.0857
RElbowRoll	-38.5° to 29.5°	0.0349 to 1.5446
LWristYaw and RWristYaw	disabled	disabled

DH parameters for the left arm chain of the NAO robot :

Frame (Joint)	**a**	$\boldsymbol{\alpha}$	**d**	$\boldsymbol{\theta}$
Base	$A(0, 0, \text{NeckOffsetZ})$			
HeadYaw	0	0	0	θ_1
HeadPitch	0	$-\frac{\pi}{2}$	0	$\theta_2 - \frac{\pi}{2}$
Rotation	$R_x(\frac{\pi}{2})R_y(\frac{\pi}{2})$			
Top Camera	$A(\text{topCameraX}, 0, \text{topCameraZ})$			
Bottom Camera	$A(\text{bottomCameraX}, 0, \text{bottomCameraZ})$			
topCameraX=53.9mm, topCameraZ=67.9mm, bottomCameraX=48.8mm, bottomCameraZ=23.8mm				

Future Enhancements

Expand the range of movements and poses the robot can mimic.

Improve the robustness of the pose estimation algorithm.

Integrate more interactive features, such as voice commands.

VIDEO

Video link