

Design Proposal: Humanoid Robot

This document outlines the design of a software system for a humanoid robot capable of performing complex tasks in dynamic environments. The software supports three core functionalities: object manipulation, human interaction prediction, and task learning through observation. A Command Line Interface (CLI) is integrated to allow users to interact with the robot, issue commands, and monitor its activities. The system also utilizes Python data structures such as lists, stacks, and queues to manage and process environmental data. This design ensures seamless human-robot collaboration in applications such as healthcare, manufacturing, and customer service (Mukherjee et al., 2022).

Models

The system is designed using object-oriented principles, with the following key classes:

- **Perception Class:** This class manages data from the robot's sensory inputs, such as cameras, infrared sensors, and ultrasonic sensors. It includes methods for identifying objects, tracking humans, and mapping environmental features (Scheutz et al., 2007).
- **Planning Class:** This class handles the robot's decision-making processes. It uses sensory data and the robot's current state to determine actions, such as navigating around obstacles or executing tasks (Kragic et al., 2018).
- **Actuation Class:** This class controls the robot's physical movements, such as arm and leg motions. It executes commands like grasping objects, walking, or adjusting posture (Wee, 2014).

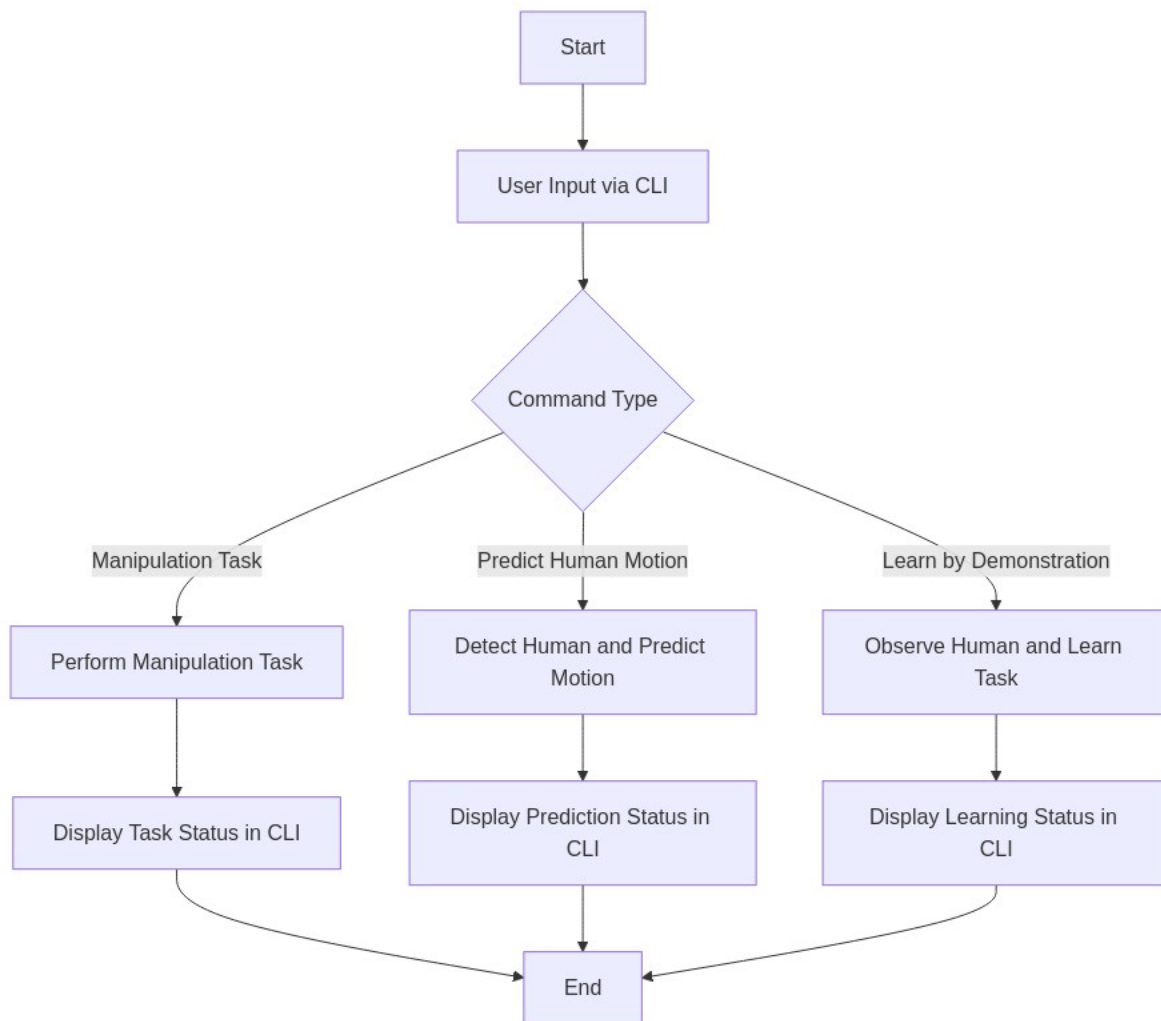
Operations

The software supports three primary operations:

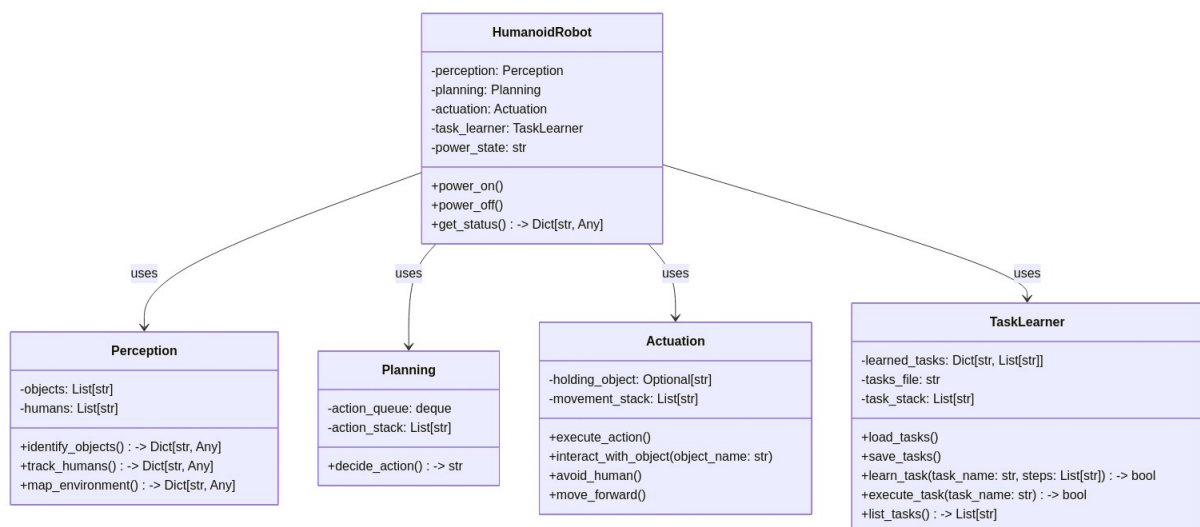
- **Object Manipulation:** The robot performs tasks like picking up and moving objects using its sensors and actuators. For example, it can retrieve a tool from a shelf and hand it to a human worker (Mulko, 2023).
- **Human Interaction Prediction:** The robot detects humans in its workspace and anticipates their actions to ensure safe and efficient collaboration. This is particularly useful in shared workspaces (Bethel and Murphy, 2010).
- **Task Learning Through Observation:** The robot learns new tasks by observing human workers. For example, it can learn to assemble a product by watching a human perform the task (Goertzel et al., 2014).

UML Models

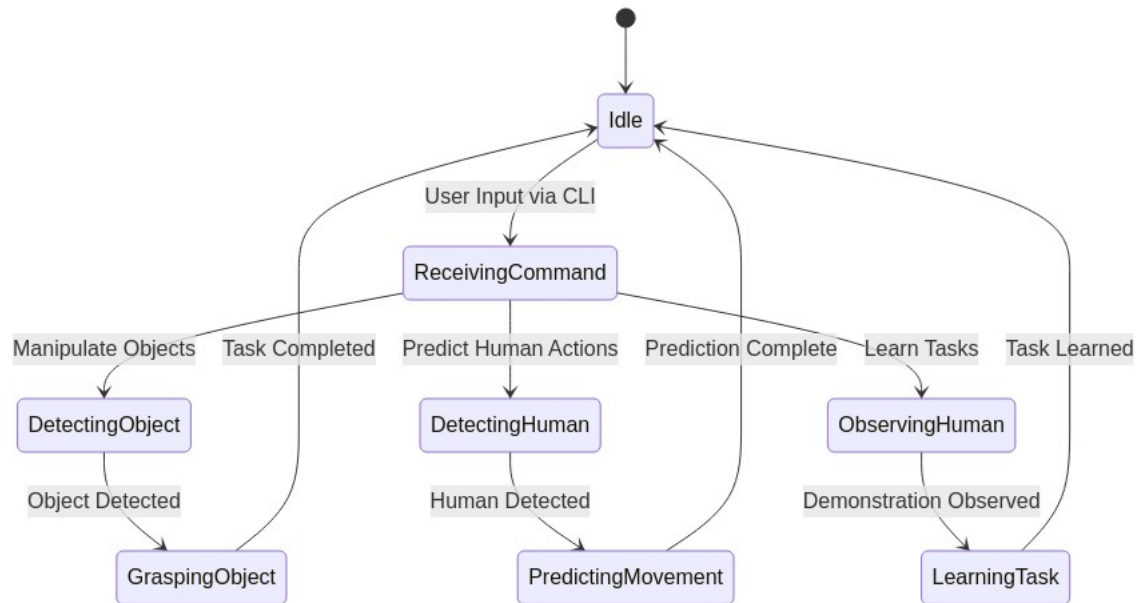
Activity Diagram



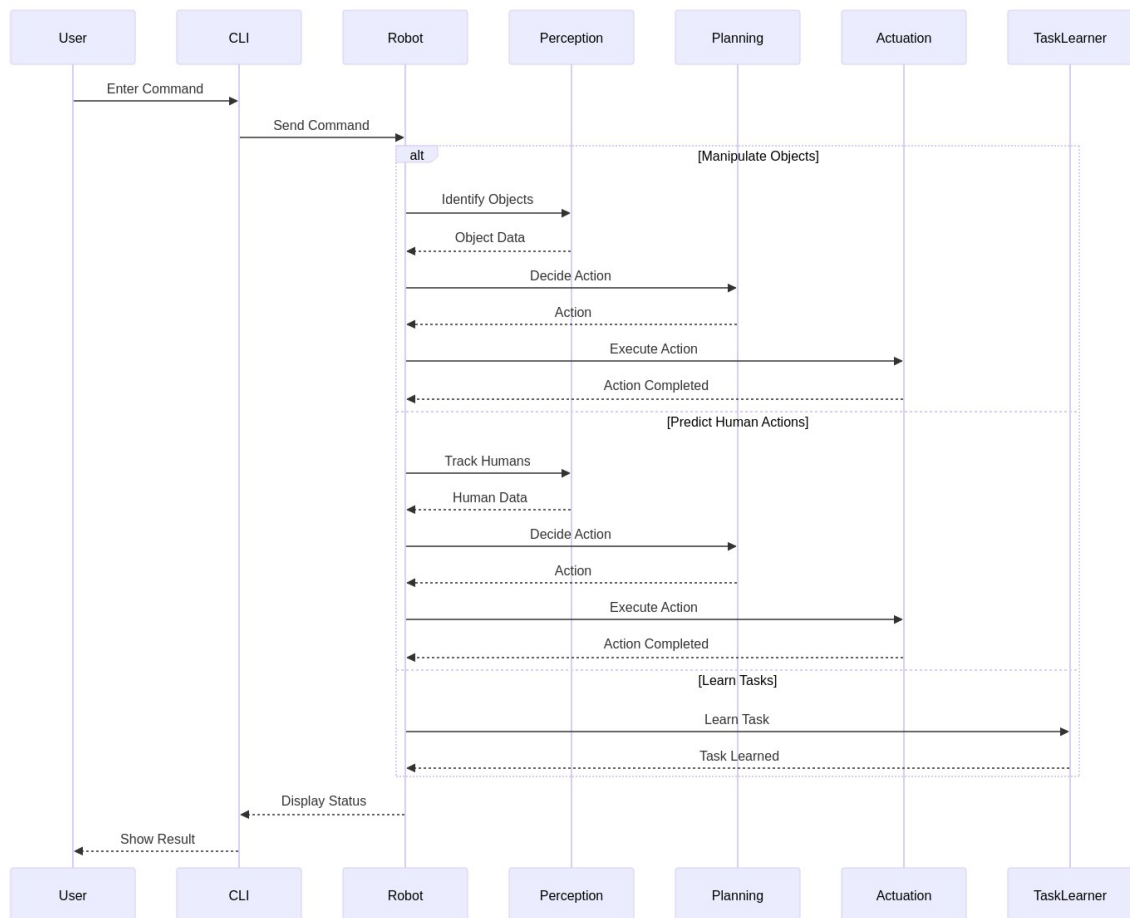
Class Diagram



Sequence Diagram



State Transition Diagram



Data Structures

The system leverages three key Python data structures. These data structures are chosen for their ability to handle dynamic and complex scenarios in human-robot interaction.

- **List:** Lists are used extensively in the Perception class to store and manage collections of objects detected by the robot's sensors. For example, the robot maintains a list of detected objects in its environment, which is updated dynamically as new objects are identified. Lists are ideal for this purpose because they allow for easy addition, removal, and iteration of items, enabling the robot to keep track of its surroundings in real time.
- **Queue:** The Planning class employs a queue to manage tasks that the robot needs to execute. Tasks are added to the queue in the order they are received, ensuring that they are processed sequentially. This is critical for maintaining workflow efficiency and avoiding task conflicts, especially in dynamic environments where multiple tasks may need to be prioritized and executed systematically.
- **Stack:** Stacks are used to manage nested tasks or actions that require backtracking. For instance, if the robot encounters an error during task execution, it can use a stack to revert to previous states and retry the operation. This is particularly useful for handling complex tasks that involve multiple steps, as the stack allows the robot to "undo" actions and recover from errors effectively.
- **Dictionary:** Used to store key-value pairs, such as the robot's joint positions in the Actuation class. Dictionaries provide fast access to specific data points, making them suitable for managing the robot's physical state and configuration.

References

- Ackerman, E. (2023). Humanoid Robots Are Getting to Work. IEEE Spectrum. Available at: <https://spectrum.ieee.org/humanoid-robots>.
- Bethel, C.L. and Murphy, R.R. (2010). Review of Human Studies Methods in HRI and Recommendations. International Journal of Social Robotics, 2(4), pp. 347–359. Available at: <https://doi.org/10.1007/s12369-010-0064-9>.
- Goertzel, B., Hanson, D. and Yu, G. (2014). A Software Architecture for Generally Intelligent Humanoid Robotics. Procedia Computer Science, 41, pp. 158–163. Available at: <http://dx.doi.org/10.1016/j.procs.2014.11.099>.
- Kragic, D., Gustafson, J., Karaoguz, H., Jensfelt, P. and Krug, R., 2018, July. Interactive, Collaborative Robots: Challenges and Opportunities. In IJCAI (pp. 18-25). Available at: <https://www.ijcai.org/proceedings/2018/3>.
- Mukherjee, D., Gupta, K., Chang, L.H. and Najjaran, H. (2022). A Survey of Robot Learning Strategies for Human-Robot Collaboration in Industrial Settings. Robotics and Computer-Integrated Manufacturing. Available at: <https://doi.org/10.1016/j.rcim.2021.102231>.
- Mulko, M. (2023). 5 of the world's most realistic humanoid robots ever. Interesting Engineering. Available at: <https://interestingengineering.com/lists/5-worlds-realistic-humanoid-robots>.
- Scheutz, M., Schermerhorn, P., Kramer, J. and Anderson, D. (2007). First Steps Toward Natural Human-like HRI. Autonomous Robots. Available at: <https://doi.org/10.1007/s10514-006-9018-3>.
- Wee, T.C. (2014). Design and Control of a Humanoid Robot. Doctoral Thesis, Imperial College London. Available at: <https://doi.org/10.25560/24436>