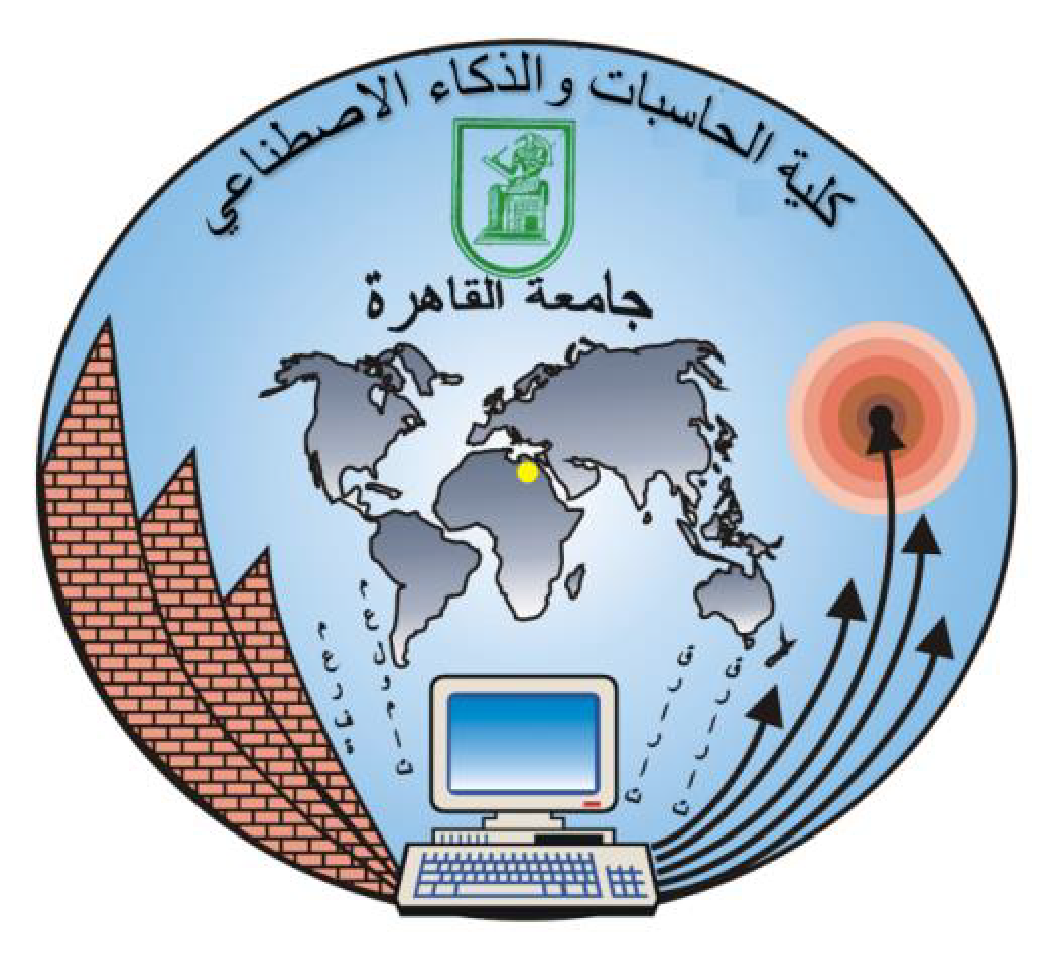
A cartoon of a person playing a musical instrument

AI-generated content may be incorrect.

**Cairo University**

**Faculty of Computing**

**and Artificial Intelligence**

**Course: AI 441- Fall 2025**

**Intelligent Autonomous Robotics Project**

**Final project technical report**

**Supervisor:** **Dr. M. A. Wahby Shalaby**

**TA: Eng. Sherif Magdy**

**25 Dec. 2025**

|  |  |
| --- | --- |
| **Name** | **ID** |
| **Mohamed Islam Khaled** | **20220282** |
| **Ali Mohamed Bahey** | **20220211** |
| **Karim Adel Mohamed** | **20220458** |
| **Seif Eldin Emad Magdy** | **20220448** |
| **Mohammed Ahmed Saeed** | **20220274** |
| **Mostafa Saeed Mowoad** | **20220336** |
| **Mohamed Ahmed Saleh** | **20220276** |
| **Remon Maged** | **20220140** |

**1. Introduction**

Mobile robots represent a core area in intelligent autonomous robotics due to their ability to operate in dynamic, real-world environments with minimal human intervention. In this project, we present the design of a mobile service robot that integrates a differential-drive mobile base with a multi-degree-of-freedom robotic manipulator. The robot is capable of navigating indoor environments such as laboratories or factory floors while performing manipulation tasks including object handling, sorting, and material transportation.  
  
Our interest in this area stems from the growing demand for autonomous service robots in industrial and educational applications. Mobile manipulators combine perception, decision-making, navigation, and manipulation, making them an ideal platform for studying advanced robotics topics such as kinematics, motion planning, control, and sensor fusion. The importance of this type of robot lies in its versatility, as it can adapt to different tasks and environments without requiring fixed infrastructure, unlike traditional stationary robotic arms.

This report builds upon our previous 3-Joint Robot Arm by transforming it into a complete **Mobile Service Robot System**. While the original arm was fixed in one location for tasks like placing chips on circuit boards, this new mobile version can navigate environments like factory floors or laboratories while performing manipulation tasks.

**2. Market Products Survey**

Several mobile robots and mobile manipulators are currently available in the market. Below is an analysis of key platforms that demonstrate the current state of the art:

**2.1 Market Leader**

**1. TurtleBot 3:** A popular educational mobile robot platform widely used in robotics research and teaching.

* **Features:** Utilizes a differential-drive base with onboard sensors (LiDAR and IMU) and supports ROS integration. This validates our project’s decision to use a differential-drive configuration for efficient indoor navigation.
* **Limitation:** It lacks an integrated manipulator, restricting its capabilities to navigation-only tasks, unlike our proposed system which integrates a robotic arm.

**2. KUKA KMR iiwa:** An industrial-grade mobile manipulator combining an autonomous mobile platform with a 7-DOF robotic arm.

* **Features:** Designed for flexible manufacturing environments, offering high precision and advanced safety features. It sets the standard for combining mobility with dexterity, a concept our project aims to replicate.
* **Limitation:** The prohibitive cost makes it unsuitable for small-scale or educational projects, highlighting the market gap for the low-cost solution we are proposing.

**3. Fetch Robot:** A service robot used in research and logistics, equipped with a mobile base, a 7-DOF arm, and a gripper.

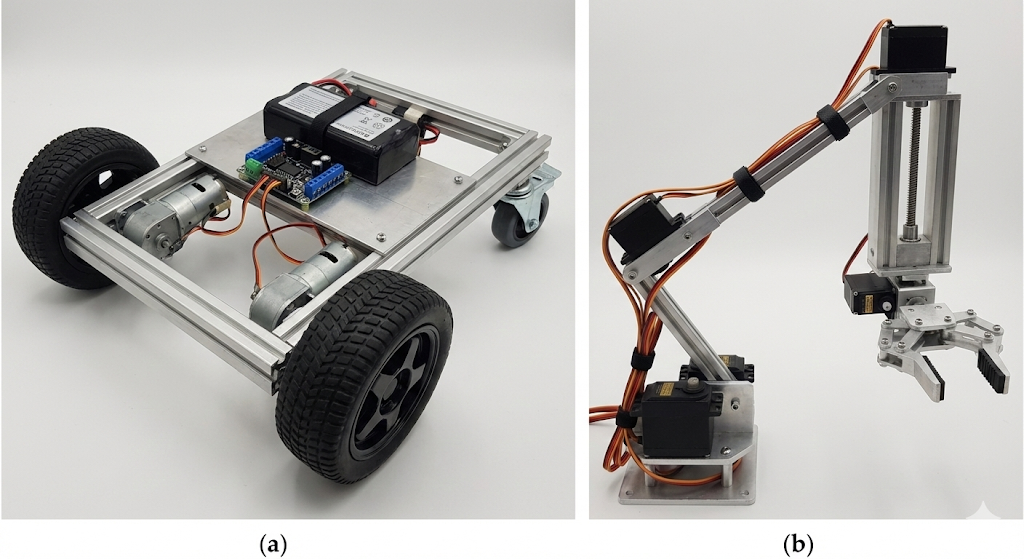
* **Features:** Provides advanced navigation, perception, and manipulation capabilities suitable for complex logistics tasks.
* **Limitation:** Requires complex software and high computational resources. This contrasts with our modular design, which focuses on accessibility and educational implementation without requiring industrial-grade computing power.

| **Feature** | **TurtleBot 3** | **KUKA KMR iiwa** | **Fetch Robot** | **Our System** |
| --- | --- | --- | --- | --- |
| **Primary Capability** | Navigation Only | High-Precision Manipulation | Logistics & Research | Educational Mobile Manipulation |
| **Manipulation** | None (Base Only) | 7-DOF Industrial Arm | 7-DOF Arm | 3-DOF Arm + Gripper |
| **Navigation Tech** | LiDAR / SLAM | Autonomous Industrial | Advanced Perception | Multi-Sensor Fusion (EKF) |
| **Target Audience** | Education / Research | Heavy Industry | Warehousing | Education / Light Industry |
| **Cost Profile** | Moderate | Very High (Industrial) | High | Low (Budget-Friendly) |

**2.2 Comparative Analysis**

**3. Our Mobile Robot**

### 3.1 Robot Chassis, Joints, and Links



The proposed robot consists of two main subsystems: a mobile base (a) and a manipulator arm (b). The chassis acts as the primary structural link, supporting all mechanical and electronic components. The mobile base employs a differential-drive configuration with two actively driven front wheels and two passive rear wheels for stability.  
  
Mounted on the chassis is a 3-joint robotic arm composed of multiple rigid links connected through revolution and prismatic joints. The shoulder and elbow joints provide rotational motion, while the wrist joint allows vertical linear movement. The final link is the gripper, which serves as the end-effector for object manipulation. This configuration enables the robot to perform tasks requiring both mobility and dexterous manipulation.

**Total Count:**

* **Links:** 7 (Chassis + 3 arm segments + 2 Wheels (Left & Right) + Ground)
* **Joints:** 6 (2 wheels + 4 arm joints)

This simple structure allows the robot to move around and manipulate objects in its environment.

### 3.2 Kinematics and Motion Description

The robot's movement is based on two main systems: the mobile base and the robotic arm.

**Mobile Base Kinematics**

* **System Type:** Differential-drive system
* **Drive Wheels:** 2 front wheels (independently controlled)
* **Support Wheels:** 2 rear passive wheels for stability
* **Motion Constraint:** Non-holonomic - cannot move sideways without turning first
* **Key Feature:** Can turn in place (zero turning radius) by spinning wheels in opposite directions

The manipulator arm kinematics can be modeled using forward kinematics based on the Denavit–Hartenberg (DH) convention.

**Manipulator Arm Kinematics**

* **System Type:** 3-degree-of-freedom (3-DOF) arm
* **Movement:** Uses forward kinematics to calculate gripper position from joint angles
* **Method:** Simple trigonometry determines where the gripper will be based on shoulder and elbow angles
* **Purpose:** Allows precise positioning for pick-and-place operations

**Wheel Configuration**

* **Material:** Rubber-coated for good traction
* **Advantages:**
  + Excellent maneuverability in tight spaces
  + Simple control system
  + Stable three-point support (two driven front wheels + rear support)
  + Quiet operation suitable for indoor environment

**Static Stability Features:**

* **Low Center of Gravity:** Batteries and heavy components mounted low
* **Wide Wheelbase:** Maximizes support polygon
* **Balanced Weight Distribution:** Prevents tipping during arm extension

**Dynamic Stability Measures:**

* **Speed Control:** Limits maximum acceleration/deceleration
* **Coordinated Motion Planning:** Arm retracts during high-speed travel
* **IMU Sensor:** Monitors tilt and enables corrective actions

### 3.3 Degrees of Freedom (DOF)

The robot has a total of **6 Degrees of Freedom (DOF)**. This combines the mobility of the mobile base with the articulation of the robotic arm.

**Mobile Base DOF: 2**

* **Forward/Backward motion** : 1 DOF Achieved by both drive motors moving at the same speed.
* **Rotation in place :** 1 DOF Achieved by driving the two front wheels at different speeds or in opposite directions.

**Manipulator Arm DOF: 4**

* **Shoulder Joint:** 1 DOF (Revolute - rotation)
* **Elbow Joint:** 1 DOF (Revolute - bending)
* **Wrist Joint:** 1 DOF (Prismatic - vertical movement)
* **Gripper:** 1 DOF (Revolute - open/close operation) (**operational**)

**Total System DOF = 2 (Base) + 4 (Arm) = 6 DOF**

This high degree of freedom enables the robot to:

* Navigate precisely to any location within its workspace
* Approach objects from multiple orientations
* Perform complex pick-and-place operations
* Adapt to various task requirements with flexibility

### 3.4 Actuators and Specifications

The robot uses a combination of DC motors for movement and servo motors for precise positioning.

**Mobile Base Actuators**

* **2x DC Gear Motors (12V, 300 RPM):** High-torque motors that independently drive the front wheels.
* **L298N Motor Driver:** Controls the speed and direction of both motors through pulse-width modulation (PWM).
* **Specifications:** Selected for sufficient torque for indoor navigation and payload capacity.
* **Approximate Cost:** 180–220 EGP per motor; 70–100 EGP for the driver.

**Manipulator Arm Actuators**

* **3x Servo Motors (MG995):** Provide precise angular control for:
  + Shoulder joint rotation
  + Elbow joint bending
  + Wrist vertical movement (converted to linear motion via linkage)
* **Gripper Actuator (SG90 Servo):** Operates the two-finger mechanical gripper for object manipulation.
* **Why Servos:** Chosen for their ease of control, built-in position feedback, and suitability for lightweight tasks.
* **Approximate Cost:** 150–180 EGP per MG995 servo; 50 EGP for the SG90 gripper servo.

This selection of actuators provides the necessary power for mobility and the precision required for manipulation tasks, all within a practical budget.

### 3.5 Sensors used

These sensors allow the robot to navigate safely, interact intelligently, and perform basic recognition tasks.

The robot uses multiple sensors for intelligent operation:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sensor Type | Function | Example Model | Approx. Price (EGP) | Source | Use Case |
| Ultrasonic Sensor | Detect obstacles ahead | HC-SR04 | 80–100 | Jumia / EG Robotics | Prevents collisions during navigation |
| IR Sensor | Line following or edge detection | TCRT5000 | 50–70 | Jumia / Souq | Follows marked paths on floor |
| Temperature Sensor | Environmental monitoring | DHT11 | 100–120 | EG Robotics | Measures room temperature |
| Camera | Object recognition | OV7725 | 550-600 | EG Robotics/  future robotics | Identifies objects for sorting |
| Encoder | Measures wheel rotation | Optical Encoder Module | 120–150 | Jumia | Calculates distance and speed |
| IMU Sensor | Orientation sensing | MPU-6050 | 150–180 | Jumia / EG Robotics | Maintains stability and balance |
| Color Sensor | Detects object color (RGB) | TCS3200 | 90-120 | EG Robotics / Jumia | Verifies object color for sorting tasks |

### 3.6 Signal processing techniques

For our mobile pick-and-place robot to function intelligently, we need to process various sensor signals. Based on the lecture material, here are the key signal processing techniques we would implement:

**A. Analog-to-Digital Conversion (ADC)**

* **Justification:** All our sensors (ultrasonic, infrared, color sensors) produce analog voltage signals that must be converted to digital values for the microcontroller to process.
* **Implementation:** Using a 10-bit ADC (common on Arduino microcontrollers):
  + Range: 0-5V (R = 5V)
  + Resolution: Q = 5V/(2¹⁰ - 1) = 5V/1023 ≈ 0.0049V per step
  + This means we can detect voltage changes as small as 0.005V, which is sufficient for our application.

**B. Noise Reduction Filtering**

* **Justification:** Sensor readings often contain noise from electrical interference or environmental factors.
* **Implementation:** Using a moving average filter (low-pass filter):
  + filtered value[n] = (x[n] + x[n-1] + x[n-2])/3
  + This is a simple 3-point moving average that smooths ultrasonic distance readings and IR sensor values.
  + This is a linear time-invariant (LTI) system that preserves important signal features while reducing random noise.

**C. Feature Extraction for Color Sensing**

* **Justification:** The color sensor (TCS3200) outputs frequency signals corresponding to red, green, and blue components.
* **Implementation:** We would use Discrete Fourier Transform (DFT) principles to analyze the frequency content:
  + Sample the color sensor output at regular intervals
  + Calculate dominant frequency components to identify color
  + This helps in sorting objects by color before picking them up

**D. Edge Detection for Line Following**

* **Justification:** IR sensors detect transitions between light and dark surfaces for line following.
* **Implementation:** Looking for sudden changes in sensor readings (discrete differentiation):
  + edge detected[n] = |x[n] - x[n-1]| > threshold
  + This helps the robot detect when it's crossing the line boundary

### 3.7 Localization and Trajectory Planning

**A. Localization Approach and Selected Algorithm**

Our robot relies on three complementary localization techniques to estimate its position within indoor environments:

* **Odometry-Based Localization:**  
  Wheel encoders are used to estimate distance traveled using:

This method is simple and cost-effective. However, it accumulates drift over time due to wheel slippage and uneven surfaces.

* **Beacon-Based Localization:**  
  Ultrasonic sensors measure distances from known walls or landmarks, allowing triangulation to refine the robot’s position.  
  This approach improves accuracy but requires a known environment layout.
* **Line-Following Localization:**  
  IR sensors allow the robot to detect and follow predefined floor paths.  
  This is especially effective in structured environments such as factory floors and provides an additional reference to minimize drift.

While each technique has its advantages, odometry alone is prone to cumulative error, and beacon-based or line-following methods can be limited by environmental constraints. Therefore, a sensor fusion algorithm is required to combine these sources and achieve robust localization.

To address this, we evaluated three probabilistic localization algorithms: **Markov Localization, Particle Filter, and Kalman Filter.** After analysis, we selected the **Extended Kalman Filter (EKF)** as the most suitable algorithm for our robot.

**Why We Selected EKF**

The EKF offers several advantages that align with our hardware and operating conditions:

* accurately models differential-drive kinematics
* fuses continuous-valued encoder and ultrasonic measurements
* assumes Gaussian noise consistent with our sensors
* computationally lightweight for microcontroller implementation
* corrects odometry drift using ultrasonic references
* well-suited for structured indoor environments such as labs and factory floors

**Why Markov Localization Was Not Selected**

* relies on discrete grid representations
* requires significant memory and computational resources
* better suited for large or ambiguous environments
* not necessary for our relatively small and known indoor workspace

**Why Particle Filter Was Not Selected**

* requires large numbers of particles for high accuracy
* computationally expensive for embedded systems
* designed for highly uncertain or dynamic environments
* adds complexity without clear benefit for structured paths and beacons

**Conclusion**

By combining odometry, ultrasonic beacons, and line-following with an Extended Kalman Filter, our robot achieves accurate and efficient localization while mitigating the limitations of individual sensing methods. The EKF provides the best balance between performance, hardware constraints, and environmental suitability.

### ****B. Trajectory Planning****

To ensure safe and efficient motion between tasks, our robot uses a trajectory planning approach built around the **Rapidly-Exploring Random Tree (RRT)** algorithm. RRT is used to compute a collision-free path from the robot’s current position to its target location while considering obstacles detected in the environment.

* **Point-to-Point Movement:**  
  RRT generates a feasible path between the pick and place points. Once the path is found, the robot follows it with smooth acceleration and deceleration to maintain stability during motion.
* **Obstacle Avoidance:**  
  Real-time ultrasonic sensor readings are integrated into the RRT planning process. If an obstacle is detected along the current path, the robot triggers a re-planning step so the RRT algorithm can generate an alternate collision-free route.
* **Arm Trajectory Planning:**  
  While the base follows the RRT-generated path, inverse kinematics is used for arm motion. The arm lifts vertically before moving horizontally to avoid collisions and executes smooth joint transitions for accurate grasping and placing.
* **Coordination Between Base and Arm:**  
  The base handles long-range motion via RRT path following, while the arm performs precise manipulation once the robot reaches the target. During travel, the arm remains retracted to maintain balance and prevent interference.

### 3. 8 Control Techniques

**A. Mobile Base Control**

To ensure precise navigation, we implement a PID (Proportional-Integral-Derivative) Controller for wheel speed regulation.

**PID Control Logic:** The controller minimizes the error between the desired speed and actual encoder feedback using the formula:

u(t) = Kp e(t) + Ki ∫ e(τ)dτ + Kd (de(t)/dt)

* + **Proportional**  Corrects immediate speed errors to maintain trajectory.
  + **Integral ():** Eliminates **steady-state error**, ensuring the robot overcomes friction and carries payloads at constant speed.
  + **Derivative ():** Reduces **overshoot** during acceleration and braking, preventing jerky motion.
* **Steering Control**: A differential drive kinematics model adjusts the left and right wheel velocities to execute turns:

*V\_left = V\_base - ΔV V\_right = V\_base + ΔV*

**B. Arm Control**

* **Position Control:** The manipulator relies on independent position control loops for each joint.
* **Coordinated Motion:** To ensure stability, we implement a **Mid-Level** safety check that synchronizes base and arm movements. For example, the arm is forced to retract to the center of gravity before the base initiates high-speed travel, preventing tipping.

**C. Gripper Control**

* **Force-Limited Grasping:** The gripper actuator monitors current draw to estimate grasping force.
* **Logic:** When the current spikes (indicating the object is grasped), the motor halts to prevent crushing delicate objects.

**D. Overall System Control Architecture**

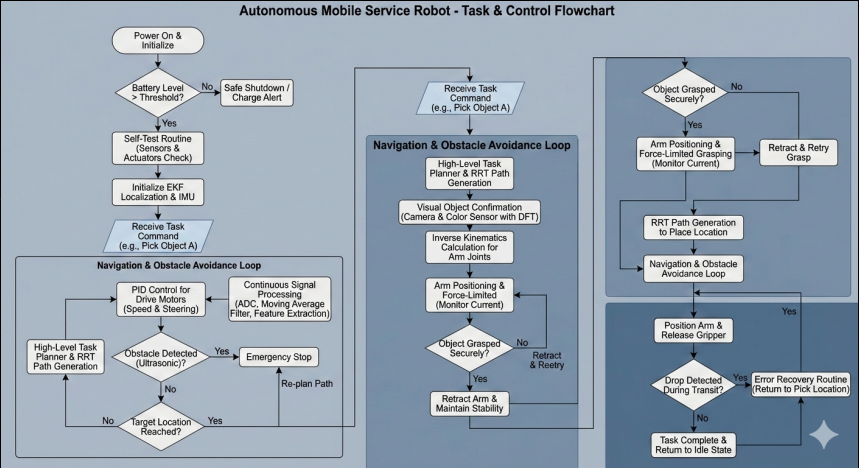
Consistent with the Hierarchical Control framework our system is organized into three vertical layers to decompose complex tasks into manageable subproblems.

* **1. High-Level Control (Task Planner)**
  + *Role:* Situation Awareness & Strategy.
  + *Function:* Receives the mission command (e.g., "Sort Red Objects") and generates a sequence of states: Maps 🡪 Detect 🡪 Pick 🡪 Place.
* **2. Mid-Level Control (Motion Planning)**
  + *Role:* Mode Transitioning & Path Generation.
  + *Function:* Converts high-level goals into geometric paths. It uses **Inverse Kinematics** for the arm and **RRT** algorithms for the base to generate collision-free trajectories.
* **3. Low-Level Control (Execution)**
  + *Role:* Stability & Actuation.
  + *Function:* The **PID controllers** execute the motion commands by adjusting motor voltages 100 times per second (100Hz) to match the target velocities and angles calculated by the Mid-Level layer.

**E. Safety and Error Handling**

* **Emergency Stop:** Triggered immediately if ultrasonic sensors detect an obstacle cm.
* **Battery Monitoring:** Prevents operation if voltage drops below a safe threshold.
* **Error Recovery:** If an object is dropped (gripper load check fails), the system resets the task state to "Pick" to retry

**4. Flowchart of robot’s tasks**



**1. Initialization & Safety Phase**

Before performing any action, the robot runs a self-check to ensure it is safe to operate1.

* **Start-Up:** The system initializes the microcontroller and calibrates sensors (IMU, Encoders).
* **Battery Check:** It verifies the voltage level. If the battery is too low, the robot enters a **Safe Shutdown** state to prevent damage.
* **Home Arm:** The manipulator arm is retracted to a safe, central position to ensure the Center of Gravity (COG) is low and balanced before the mobile base starts moving.

**2. Navigation Phase (The "Travel" Loop)**

Once initialized, the robot waits for a high-level command (e.g., "Pick Object A").

* **Path Planning (RRT):** The system uses the **Rapidly-Exploring Random Tree (RRT)** algorithm to generate a collision-free path from the current location to the target.
* **Localization (EKF):** As the robot moves, it continuously reads data from wheel encoders and the IMU. The **Extended Kalman Filter (EKF)** fuses this data to estimate the robot's exact position (x, y, theta) and correct for wheel slippage.
* **Obstacle Avoidance:** The ultrasonic sensor continuously scans the path.
  + **If an obstacle is detected cm):** The robot triggers an **Emergency Stop** and re-plans the path using RRT.
  + **If clear:** The **PID Controller** adjusts the wheel speeds to keep the robot on the planned trajectory.

**3. Manipulation Phase (The "Pick" Sequence)**

When the robot reaches the target coordinates, the mobile base halts, and the arm takes over.

* **Object Detection:** The camera (or color sensor) identifies the target object and determines its position relative to the gripper.
* **Inverse Kinematics:** The processor calculates the required angles for the shoulder and elbow joints to position the gripper at the object's coordinates.
* **Grasping:** The arm extends, and the gripper closes.
  + **Force Feedback:** The system monitors the current drawn by the gripper motor. A spike in current indicates the object is securely held, triggering the motor to stop closing to prevent crushing the object.
* **Retraction:** The arm lifts the object and retracts to the travel position before the robot moves again.

**4. Error Recovery**

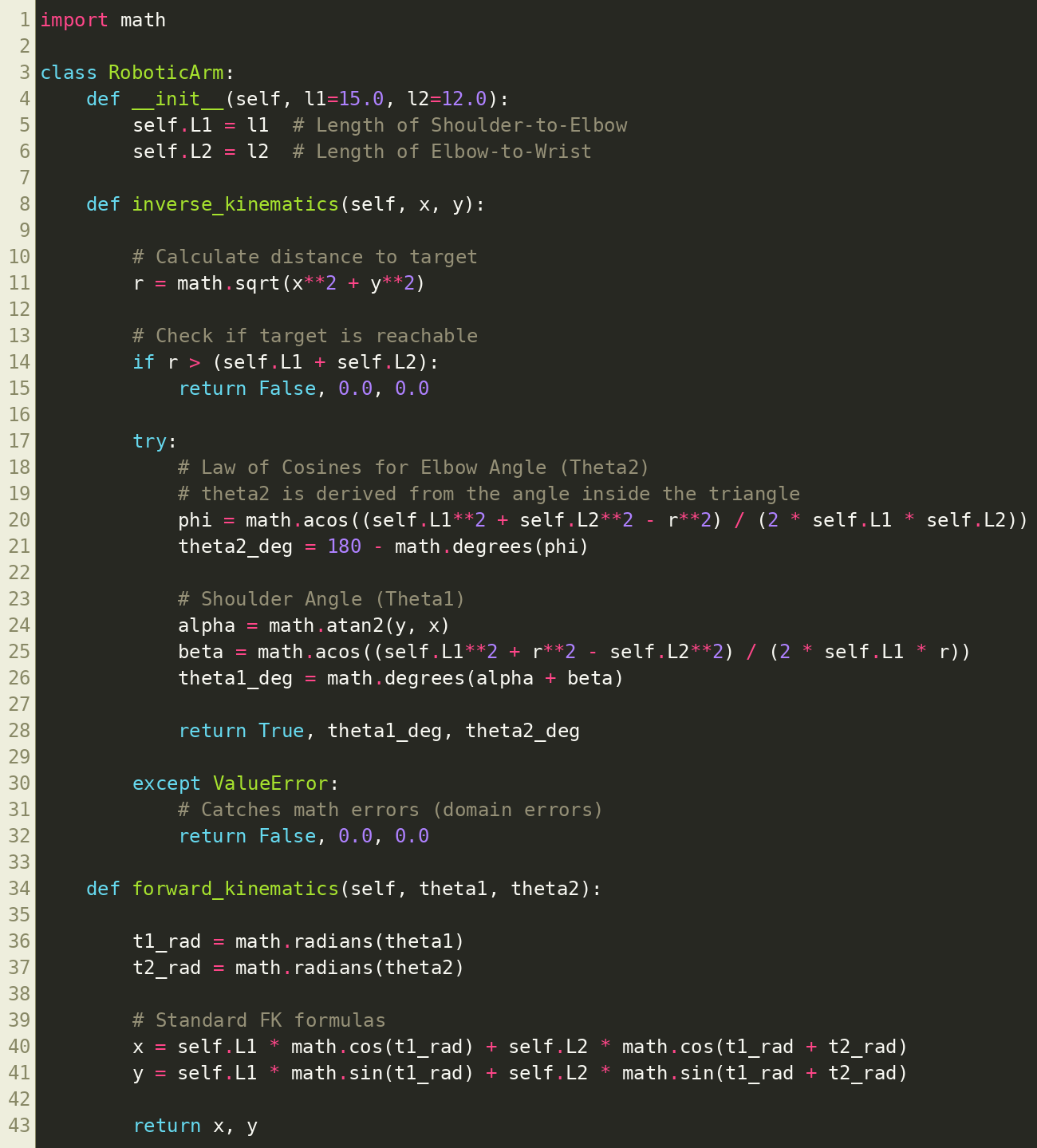
* **Drop Detection:** If the robot detects the object has been dropped during transit, it triggers an **Error Recovery Routine** to return to the pick location and retry the task.

**5. References**

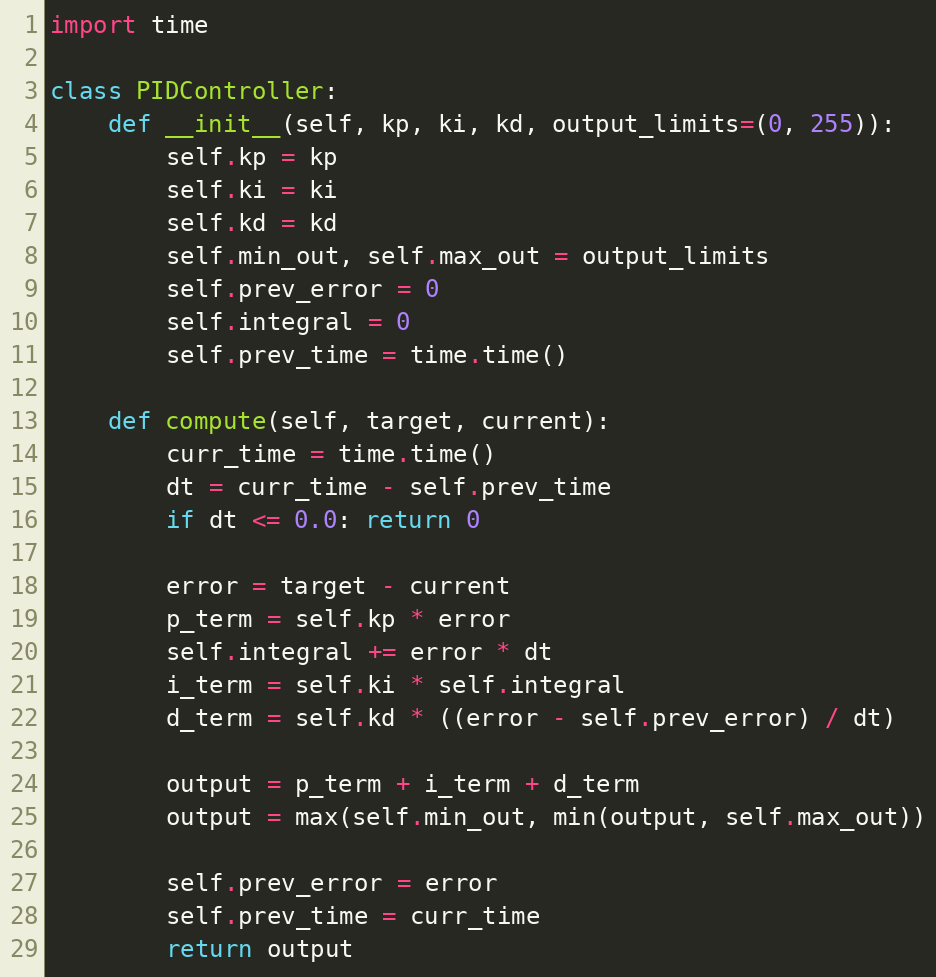
* Dr. M. A. Wahby Shalaby, "Intelligent Autonomous Robotics (AI 441) Course Lectures 1–10: Kinematics, Localization, and Control Systems," Faculty of Computing and Artificial Intelligence, Cairo University, Fall 2025.
* Arduino.cc. (2023). Arduino Reference - Servo Library. <https://www.arduino.cc/reference/en/libraries/servo/>
* HC-SR04 Ultrasonic Sensor Datasheet.
* TCS3200 Color Sensor Module Overview.
* Craig, J. J. (2005). *Introduction to Robotics: Mechanics and Control.* Pearson Education.
* Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2005). *Robot Modeling and Control.* John Wiley & Sons.
* "Mobile Manipulator Systems: Design and Control." International Journal of Robotics Research.
* "Differential Drive Mobile Robot Kinematics." Robotics Knowledge Base.
* Sereinig, M., ... & Vincze, M. (2020). A review of the challenges in mobile manipulation: systems design and RoboCup challenges. Journal of Intelligent & Robotic Systems, 100(3-4), 823-847.
* Kemp, C. C., ... & Jain, D. (2022). The Design of Stretch: A Compact, Lightweight Mobile Manipulator for Human-Centered Autonomy. IEEE Robotics & Automation Magazine, 29(4), 22-33.
* Ullah, I., ... & Kim, J. (2021). Extended Kalman Filter-based localization algorithm by fusing wheel odometer with IMU and external camera for mobile robot. Alexandria Engineering Journal, 60(5), 4899-4910.
* Huang, Z., ... & Li, Y. (2025). Application of multi-sensor fusion localization algorithm based on EKF and RNN in mobile robot. Scientific Reports, 15(1), 1-13.
* Moore, T., & Stouch, D. (2016). A Generalized Extended Kalman Filter Implementation for the Robot Operating System. In Intelligent Autonomous Systems 13 (pp. 335-348). Springer.

**6. Codes of selected algorithms**

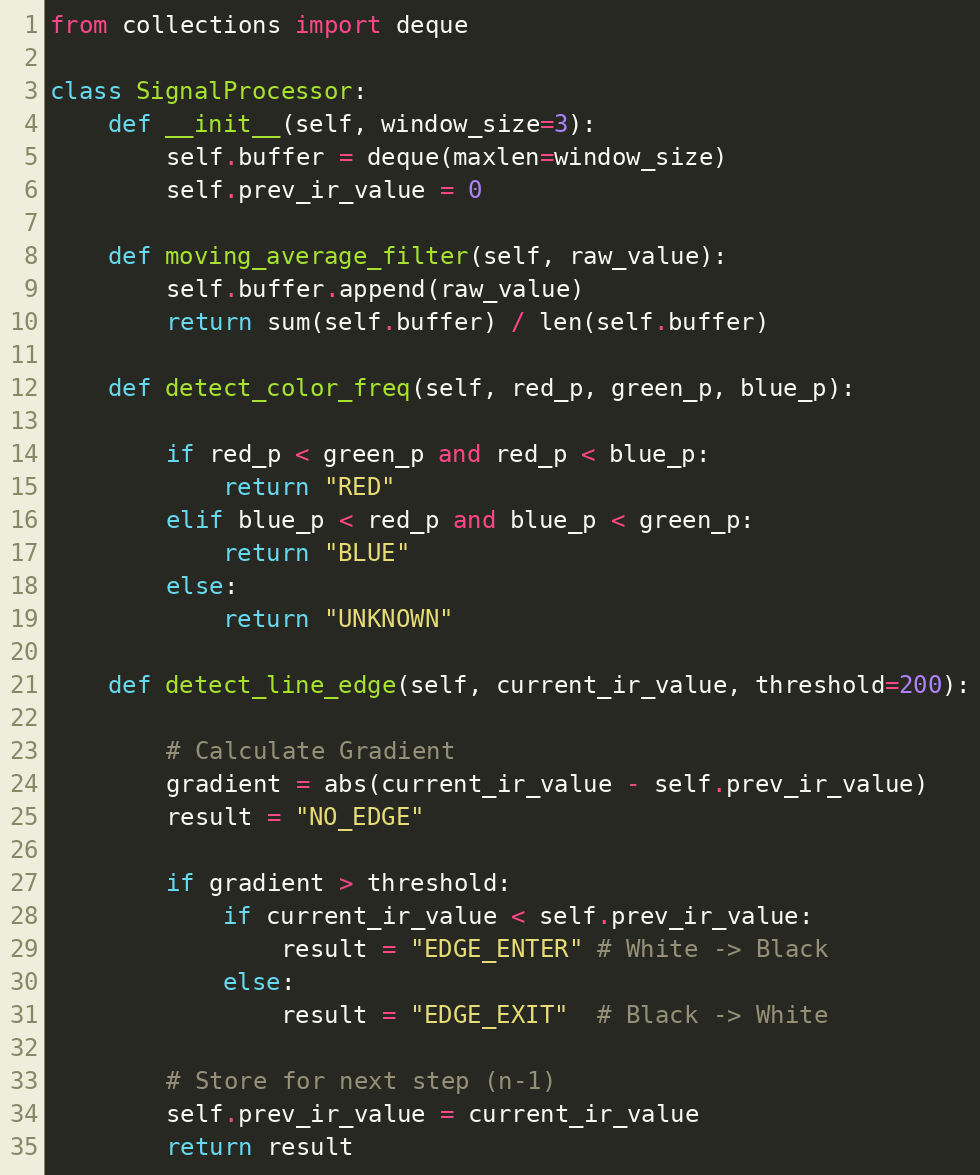
**Kinematics**



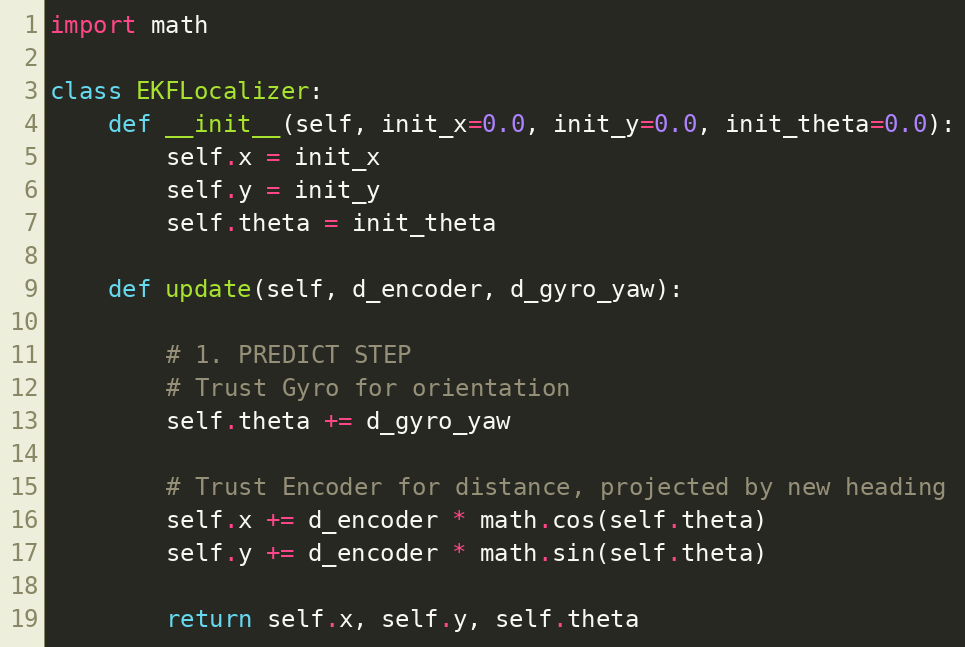
**PID Control (For Mobile Base)**



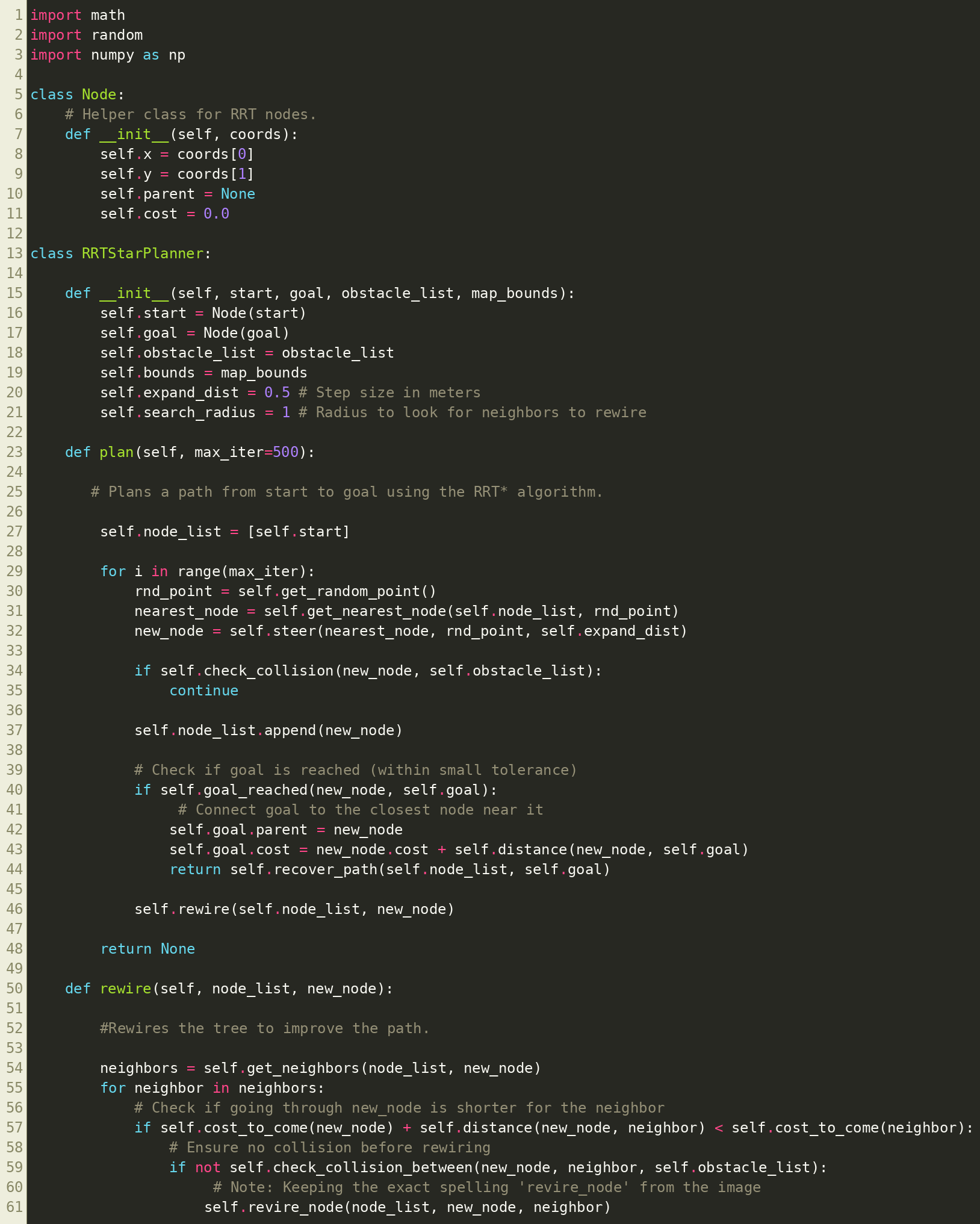
**Signal Processing**



**EKF (Localization)**



**RRT**



**gripper\_control**

