Turtlebot assembly:

1. Introduction:

* Robots with sensory and navigation capabilities
  + Humans and animals interact with external environment using sensory systems – visual, auditory, olfactory, gustatory, tactile, etc. These systems help animals sense and interpret the environment and perform activities like foraging, mating, evading predators, etc. for survival. These sensory systems are also used by robots for sensing and acting in given environment. A robot equipped with visual sensor (camera), olfactory sensor (e.g., chemical sensor), tactile sensor (e.g., touch sensor, airflow detection sensor), etc.
* Importance of olfaction in robotics.
  + These sensory systems can allow robots to perform tasks such as detecting and navigating towards a target odor source in an unknown environment. The technology used in robotics to find odor sources is known as odor source localization (reference 1). OSL has increasingly important applications including fire source finding, air pollution monitoring, detecting chemical gas leaks, marine surveys, etc.
* Challenges of odor source localization: odor plume detection, airflow detection.
  + Development of a robotics system that can
* Moth-inspired algorithm

1. Related works:

* Multi-sensor robot platform development.

1. Methodology:

* Platform development:
  + Turtlebot3 Waffle-Pi:
    - Platform: Turtlebot3 is a popular mobile robot system for research and education. It is highly modular and customizable.
    - ROS: Turtlebot3 uses Robot Operating System (ROS) as its operating system.
    - Raspberry pi
    - OPENCR
    - Given sensors: Turtlebot3 includes Raspberry Pi Camera, 360-degree LiDAR sensor, 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer. These sensors help Turtlebot3 to measure 9-axis inertia.
    - Capabilities: Turtlebot3 has can perform SLAM (simultaneous localization and mapping), Navigation and manipulation tasks with the given sensors and DYNAMIXEL diver. It can be paired with additional sensors for increasing its functionalities.
  + Physical assembly of turtlebot3
  + System setup
    - Remote PC – description and role
      * A remote PC can be used to run custom programs for controlling the Turtlebot3. Turtlebot3 runs on ROS. Both Python and C++ is supported in this platform.
    - SBC, OPENCR – description and role
    - Included sensors: laser, odometry, camera – sensor description and assembly
    - Networking
  + Additional sensor assembly – highlight of the paper:
    - Sensor connection: for odor source localization, additional chemical sensor, airflow speed, wind direction sensors were needed.
      * Chemical sensor –introduction, how it works, connection type, connecting to OPENCR:
        + Introduction: MQ3 sensor is a widely used Metal Oxide Semiconductor (MOS) sensor. It operates on 5V DC and consumes about 800mW. It can detect alcohol concentrations ranging from 25 to 500 ppm.
        + Connecting to OPENCR: its VCC pin connects to 5V output (ping number ? of OpenCR). Its GND connects to pin number ? of OpenCR. Digital/analog pin connection.
      * Airflow and windspeed sensor:
        + Introduction
        + Connecting to OPENCR:
    - Sensor code: Turtlebot3 OPENCR firmware supports additional two touch detecting Bumper sensors, cliff detecting IR sensor, ultrasonic sensor, illumination detecting LDR sensor, etc. However, adding different sensors than the ones listed above require changing the OpenCR firmware code.
      * Chemical sensor edit:
      * Windspeed sensor edit:
      * Wind direction sensor edit:

Uploading changed OPENCR firmware:

* Odor source localization
  + Description of an OSL task: An OSL task have three stages – plume finding, plume tracing and source declaration [source 38]. The first stage is plume finding, which aims to detect plume in the search area. Once the robot detects plume in the search area, the plume tracing stage initiates. In this stage the robot uses the plume detection and airflow state to approach the odor source. Once it reaches the vicinity of the source, it records the source location and terminates.
  + Review of the olfactory-based moth-inspired navigation algorithm: the moth inspired method has two behaviors – moving in cross-wind trajectory to detect plume in the air, and if plume is detected, moving in up-wind trajectory to find odor source. If the robot loses plume in the air, it’ll continue to move in cross-wind trajectory to find plume again.

1. Experiments
   1. Experiment setup: Experiments were conducted in the Automatic Control Lab at the Louisiana Tech University. The lab area was divided into a search area where the robot can navigate and an operation area for the remote PC. The size of the search area is 27X11ft^2. The robot, odor and airflow source were randomly placed in this search area for each trial run. Ethanol vapor was employed as the odor source as it is minimally toxic and commonly implemented in OSL research [40]. A humidifier was used to disperse ethanol vapor consistently as the odor plume. An electric fan was used behind the humidifier to increase odor propagation.
   2. Mapping the experiment area
   3. Sensory input
   4. Experiment run: Fig \*\* depicts the search area and the mobile robot. During an experiment run, the robot sends sensor measurements to the remote PC. The remote PC runs the moth-inspired navigation algorithm to calculate robot’s heading command and transmit it back to the robot. The robot again collects and transmits sensor readings back to the remote PC. This cycle is repeated until the robot gets within 0.2m (the radius is determined based on the search area and robot dimensions) radius of the odor source.
   5. Moth-inspired algorithm
   6. Multiple runs: To evaluate the performance of the moth-inspired algorithm, total 6 tests were conducted. In each run, the odor source was initiated at a separate position.
   7. Description of a specific run
2. Conclusion and future works

* Sensor update
* ML based methods

References