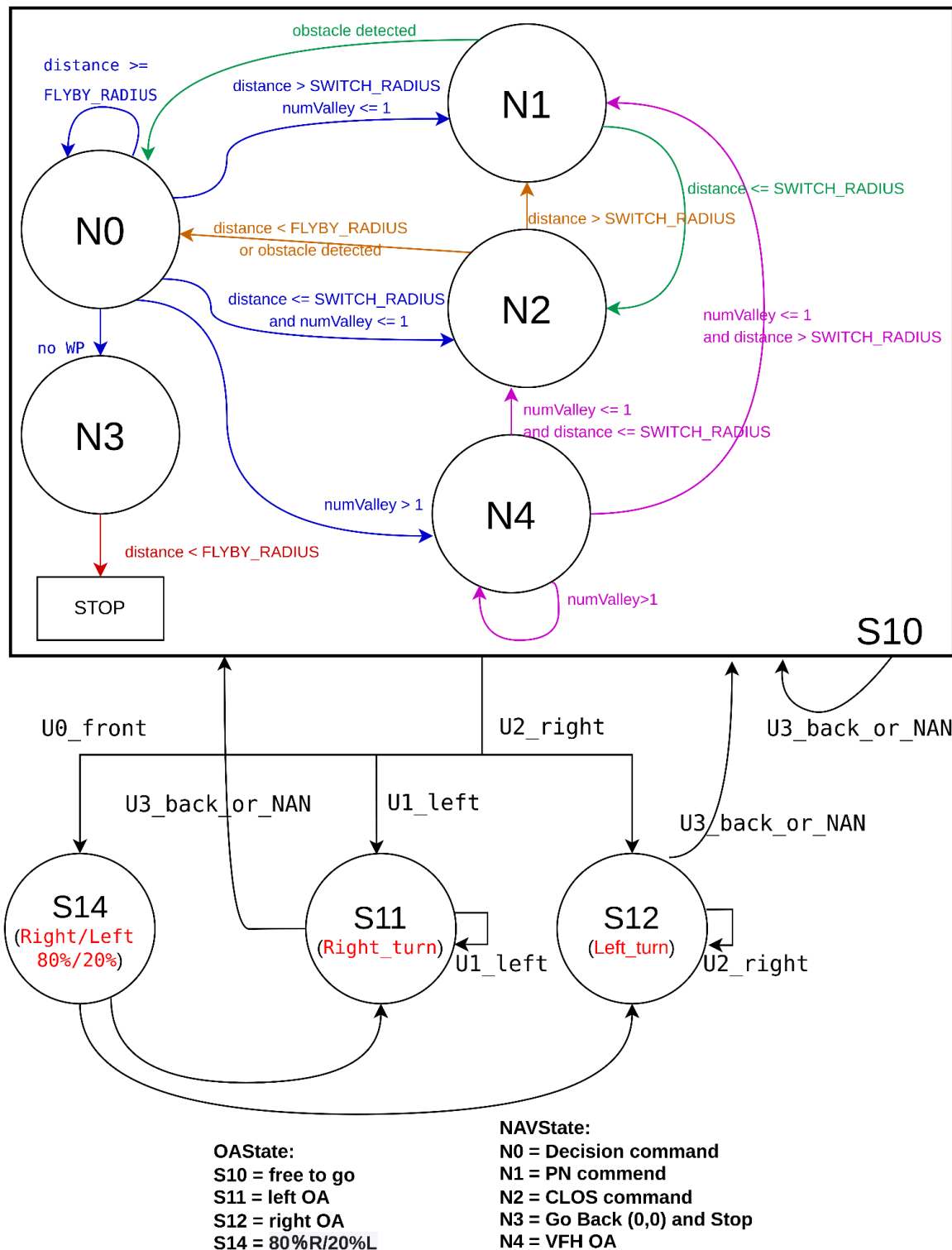


Advanced Programming in Robotic Navigation
 Final-Term report name:Tzu-Ching Chao number:111323083
 state transition diagram :



Subclass Method for State Transition

- N0 (Decision State): Acts as the central hub, evaluating distance and numValley to transition to N1, N2, N3, or N4. It also handles waypoint updates via `handleWaypointCompletion()`.
- N1 (PN command): Uses `PN_control()` to compute the control signal. Transitions to N2 if distance \leq SWITCH_RADIUS, to N4 if numValley > 1 , or to N0 if an obstacle is detected.
- N2 (CLOS command): Uses `CLOS_control()` for precise alignment. Transitions to N1 if distance $>$ SWITCH_RADIUS, to N0 if distance $<$ FLYBY_RADIUS or an obstacle is detected, or to N4 if numValley > 1 .
- N3 (Return to Origin): Uses `CLOS_control()` to navigate to (0,0). Stops when distance $<$ FLYBY_RADIUS.
- N4 : Uses VFH to select a safe direction. Transitions to N1 or N2 when numValley ≤ 1 , based on distance.

Navigation Algorithms

- Proportional Navigation (PN): Computes $\psi_c = \text{PN_GAIN} * d\lambda_mT + \text{HEADING_GAIN} * \text{headingError}$, balancing LOS rate and heading alignment.
- Command to Line-of-Sight (CLOS): Computes $\psi_c = \text{CLOS_GAIN} * \lambda_mT$ for direct target alignment.
- Vector Field Histogram (VFH): Analyzes LIDAR data to compute obstacle density, identify passable regions, and select the optimal direction.

Obstacle Avoidance

- Inheriting the obstacle avoidance function of `OAMyRobot`, it processes the distance sensor data via `STATE_OA()` to cope with obstacles in front, to the left or to the right.
- Using VFH, `getVFH()`, `findValleys()`, and `selectVFHDirection()` in N4 state to identify the passable area and select the obstacle avoidance direction.

Functionality

1. Waypoint Management:

- The `receiveWP()` method retrieves waypoint data from a receiver, dynamically allocating memory to store coordinates and updating the waypoints array.
- The `findNearestWP()` method selects the closest waypoint based on the robot's current position.
- The `handleWaypointCompletion()` method updates the waypoint list when a waypoint is reached (distance $<$ FLYBY_RADIUS), removing it and selecting the next target.

2. Path Navigation:

- Distance navigation (N1 state): Use proportional navigation (PN) control to calculate the control amount based on the rate of change of the angle of view and the orientation error via `PN_control()`.

- Close navigation (N2 state): Use the line of sight command (CLOS) control, through the CLOS_control() according to the line of sight angle adjustment direction.
- When all navigation points have been visited, it enters N3 state and returns to the home position (0,0).