**Main Programme Waypoint Designation Parameters:**

[Waypoint position, Max Wait Time, Sensor 1, Obstacle Posn, Scan Angle, No of scans, Sensor 2, Obstacle Posn, Scan Angle, , No of scans]

X,Y (mm) (sec) (I-R/U-S/Other) X,Y (mm) +/- deg default 2 (I-R/U-S/Other) X,Y (mm) +/- deg default 2

**Navigation Update Logic**

The robot travels round a pre-programmed course from waypoint to waypoint. These waypoints are chosen so that the planned route avoids the known obstacles. Open-loop navigation is achieved by using two odometers (one on each drive wheel) which control the wheel rotation and therefore the distance moved by each wheel. Unfortunately, navigation errors will build up due to the wheels skidding. Scanning sensors on the robot are therefore used to determine and update the true position of the robot in the arena and thereby correct these navigation errors.

Strategic waypoints are designated as ‘Update’ Waypoints, and for each of these waypoints two specific obstacles are chosen to be detected by an on-board sensor to determine and update the true position and heading of the robot in the arena.

At each ‘Update Waypoint’ a sensor slews to point at the expected position of a designated obstacle and does a mini scan either side of this position to detect the nearest obstacle within the arc of the scan. The range and angle of this detected obstacle is recorded and the sensor (or another sensor) slews to detect the second obstacle.

The detected positions of the two obstacles are compared with the actual known positions of these obstacles to calculate and correct the navigation and heading errors. The robot then plots a new direct route to the next waypoint and continues round the planned route to the next ‘Update Waypoint’

**Update Sequence:**

Robot stops at Update Waypoint.

1. Sensor 1 slews to (expected obstacle 1 position – scan angle)
2. Sensor 1 scans to (expected obstacle 1 position + scan angle) and records min range and associated angle of obstacle
3. Sensor 1 scans to (expected obstacle 1 position - scan angle) and records min range and associated angle of obstacle
4. Sensor 1 repeats scan for number of scans designated
5. System calculates average range and average angle from the two or more scans.
6. Sensor 1or2 slews to (expected obstacle 2 position – scan angle)
7. Sensor 1or2 scans to (expected obstacle 2 position + scan angle) and records min range and associated angle of obstacle
8. Sensor 1or2 scans to (expected obstacle 2 position - scan angle) and records min range and associated angle of obstacle
9. Sensor 1or2 repeats scan for number of scans designated.
10. System calculates average range and average angle from the two or more scans.

DIAGRAM

1. System converts sensor ranges and angles to polar coordinates from centre of chassis
2. System calculates X,Y of Obstacles 1 and 2 in arena Cartesian coordinates assuming odometer derived position and angle of chassis is accurate.

DIAGRAM

1. System calculates sensor derived distance between obstacles 1 and 2 and the angle between obstacles 1 and 2 relative to arena North.
2. System calculate actual distance between obstacles 1 and 2 and the angle between obstacles 1 and 2 relative to arena North.
3. **System calculates angular error which will also be the chassis angle error.**

DIAGRAM

1. System corrects chassis angle and also applies this angular correction to sensor obstacle angles (steps 5 and 10)
2. System re-calculates X,Y of Obstacles 1 and 2 in arena Cartesian coordinates assuming odometer derived position and using corrected angle of chassis.
3. System re-calculates sensor derived distance between obstacles 1 and 2 and the angle between obstacles 1 and 2 relative to arena North.

*Note: The distance between obstacles should remain the same, and the angle between obstacles 1 and 2 relative to arena North should now be the same.*

DIAGRAM

1. System calculates the X and Y coordinate errors between the sensed and actual positions of obstacle 1.
2. System calculates the X and Y coordinate errors between the sensed and actual positions of obstacle 2.
3. System calculates average X error and the average Y error for the positions of obstacles 1 and 2.

*Note: This X and Y error is the error in robot position and is used to update the actual position of the robot.*

DIAGRAM

1. System calculates a new direct route to the next waypoint.

A future requirement will be that if the robot is within 500mm of the next waypoint a direct route is required, otherwise a route to intercept the correct route to the next waypoint is calculated to intercept the required route 500 mm from the current position.

1. A future requirement will be for the robot to learn any consistent errors in X, Y, and Heading and apply appropriate corrections for the remainder of the route.