NUST

SMME

Mobile Robotics

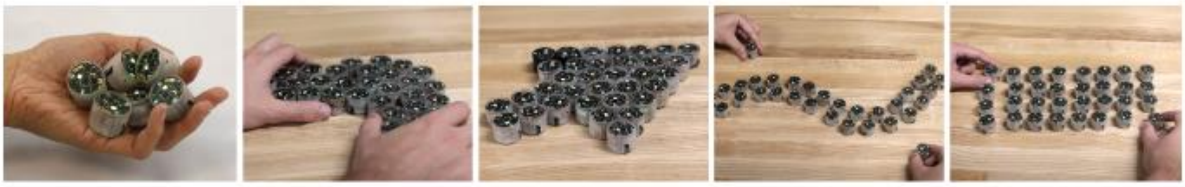
Assignment # 3

Submitted to	**Dr Yasar Ayaz**
Submitted by	**Safi ur Rehman**
Reg No #	**450266**
Program	**MS R&AI**
Semester	**2nd**

Robot Name: Zooids

(Zooid is a single animal that is part of a colonial animal.   Zooids are multicellular; their structure is similar to that of other solitary animals.)

Abstract



Zooids can be held as tokens, manipulated collectively or individually, behave as physical pixels, act as handles and controllers, and can move dynamically under machine control. They are building blocks for a new class of user interface we call swarm user interfaces.

Its an open-source open -hardware for developing table top swarm interfaces.

Swarm User Interface

swarm user interfaces, a new class of human-computer interfaces comprised of many autonomous robots that handle both display and interaction.

Robot Info

Robot is wheeled.

Robot shape is cylindrical, with 2.6 cm in dia.

It uses a radio base-station.

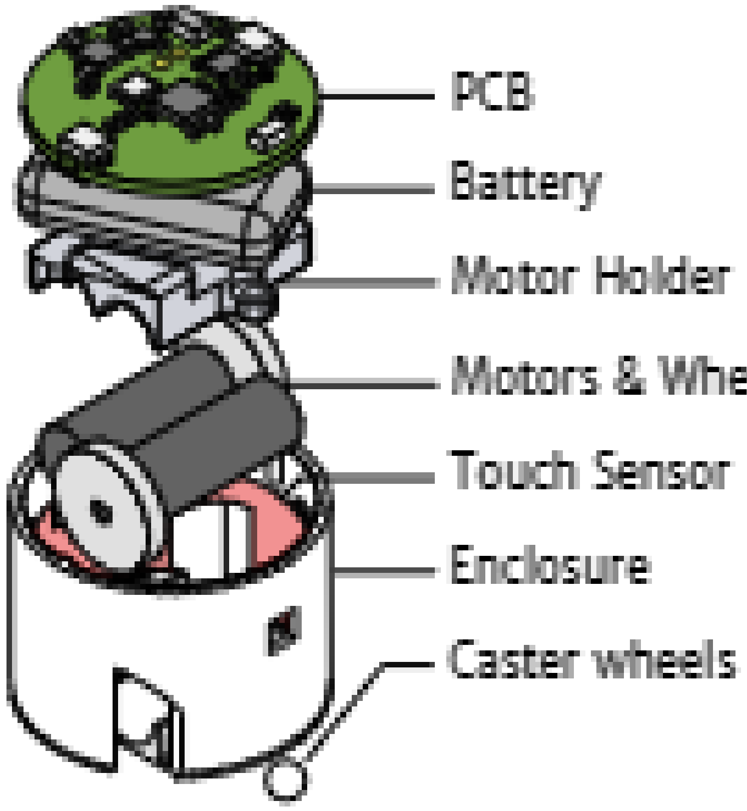
A high speed DLP structured light projector for optical tracking.

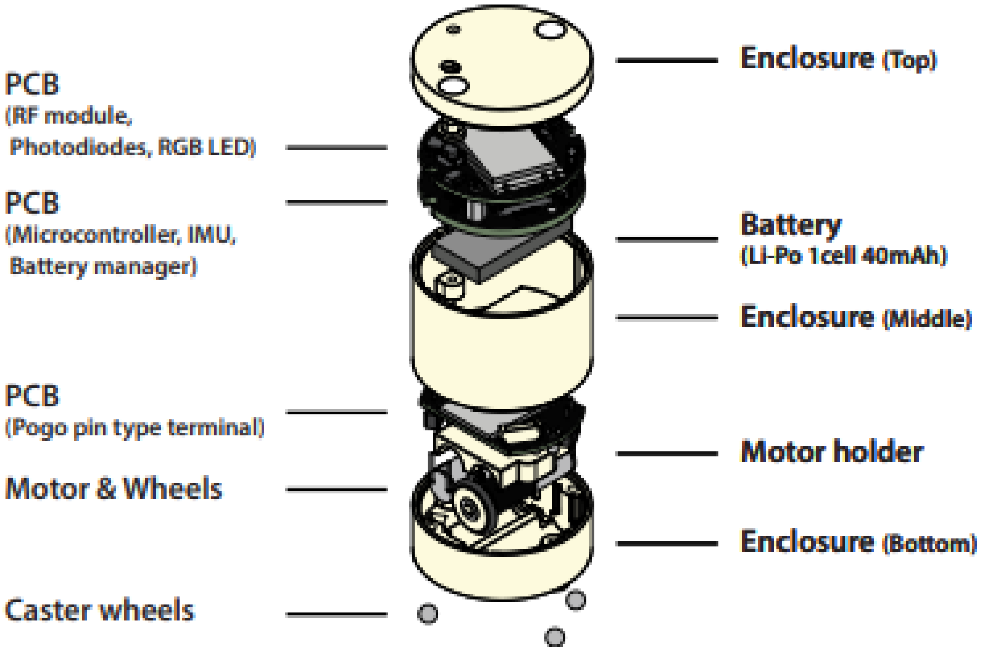
A software framework for application development and control.

It has position and touch sensing capability.

Hardware

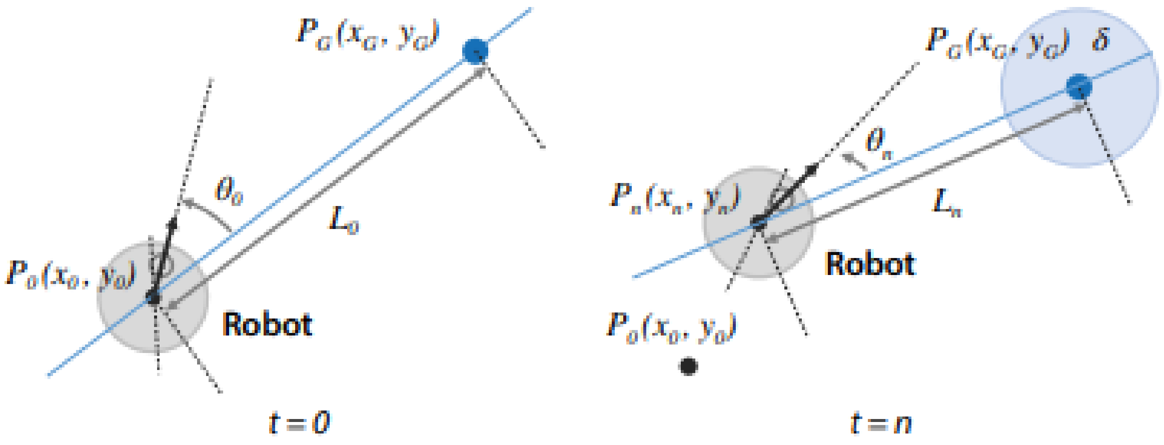
Robot Design





Swarm Robot Control for Following Sub goal Position

The robot receives the subgoal (target) position from the host computer via RF communication and moves to follow it. It is desirable for the robot to follow a smooth path to the subgoal position. In controlling wheeled robots, using a Bezier, spline, or cross-oid curve as the path is common. This curve passes through the current and subgoal positions to avoid sudden changes in angular velocity. However, due to the limited computational resources of the robot's microcontroller, it has been difficult to implement control that sequentially calculates and follows these curves. On the other hand, since the robot can get its absolute position information with the projection-based method, and the subgoal position is updated by the host computer every 100 ms, the distance between the current position and the subgoal position is considered to be close, and we thought that sudden changes in angular velocity would be rare even on a path connecting these two points by a straight line. Thus, we implemented a control model that follows the straight path.



The concept of the control model is shown in Figure 20. Let $P0$ ($x0$, $y0$) be the robot’s initial position at time t = 0 given the subgoal (target) position and PG (xG, yG) the subgoal position. Next, define −−−−→$P0PG$ as the vector connecting points $P0$ and PG , with L (L = $L0$ at t = 0) as the distance between them, and θ (θ = $\theta0$ at t = 0) as the angle between the robot’s direction of motion and the vector −−−−→$P0PG$. (Counter clockwise is defined as positive.) In this case, the following law controls the velocities Vl and Vr of the robot’s. left and right motors.

$$
\begin{aligned}
V_l &= V - \Delta V \\
V_r &= V + \Delta V \\
V &= K_L L \\
\Delta V &= K_\theta \theta + K_{\dot{\theta}} \dot{\theta},
\end{aligned}
$$

where KL, $K\theta$, K ¤θ are constants that represent the control gain. Based on the above control law, −−−−→$P0PG$, the distance Ln, and the angle θn are updated from the robot’s current position Pn (xn, yn) to the subgoal position, and Vl and Vr are calculated and output. However, if this control law is followed, the point PG is theoretically unreachable and will never fully converge, since L = 0 and the velocity will converge to 0 as the robot approaches PG . Therefore, the robot is judged to have converged when it is within a certain distance σ from PG , and the robot stops at that point. In addition, the minimum velocities $Vl{,}min$ and $Vr{,}min$ are set for Vl and Vr, respectively. The maximum speed of motors $Vmax$ is determined by the maximum speed value of the slower of the two motors. This is based on the actual measured speeds of the left and right motors as an upper limit.

This can be written in the formula as follows:

$$V_l = \begin{cases} V_{l,min} & \text{if } V_l < V_{l,min} \\ V_{max} & \text{if } V_l > V_{max} \\ V_l & \text{otherwise.} \end{cases}, V_r = \begin{cases} V_{r,min} & \text{if } V_r < V_{r,min} \\ V_{max} & \text{if } V_r > V_{max} \\ V_r & \text{otherwise.} \end{cases},$$

$$V_{max} = \begin{cases} V_{l,max} & \text{if } V_{r,max} \geq V_{l,max} \\ V_{r,max} & \text{if } V_{r,max} < V_{l,max} \end{cases},$$

where Vl,max and Vr,max are the maximum speed value of left and right motors. If a new subgoal position PG is given before convergence, PG is updated, and the subgoal following continues. Here, the dimension (unit) of the calculated Vl and Vr is [mm/s], but the dimension of the PWM duty of the voltage value, which is a control value that can be input to the motor, is [rpm]. Therefore, the calculated Vl and Vr are converted to the PWM duty value by the following formula:

$$Duty_l = f(V_l)$$
$$Duty_r = g(V_r),$$

here $Dutyl$ and $Dutyr$ are the PWM values of left and right motors, and $f\ (x)$ and $g(x)$ are functions corresponding to the PWM value calculated by the robot’s calibration mode. We confirmed that this control rule can be used to control the robot to successfully reach the goal position even with a microcontroller with limited computational resources