An Approach to Flocking of Robots Using Minimal Local Sensing and Common Orientation

I. Navarro[1] A. Gutiérrez[2] F. Matía[1] F. Monasterio-Huelin[2]

[1]Intelligent Control Group, Universidad Politécnica de Madrid (UPM),
{inaki.navarro,fernando.matia}@upm.es

[2]Departamento de Tecnologías Especiales Aplicadas a la Telecomunicación, UPM
aguti@etsit.upm.es,felix.monasteriohuelin@upm.es

26th September 2008

POLITÉCNICA

Contents

POLITÉCNICA

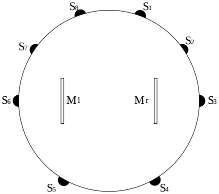
Introduction

- **Flocking** → How to control the movement of a group of robots that move together as group, behaving as a single entity.
- Relative positions between the robots are not fixed.
- External shape is not a requirement.
- Useful in:
  - **Search** tasks, when pattern of the source is complex, (e.g. odor, sound);
  - **Mapping**, measurement redundancy.
- Desired characteristics:
  - Scalable in the number of robots,
  - Local sensing and communications,
  - Decentralized controller,
  - Obstacle avoidance at group level.

Aim

- A distributed and **scalable** algorithm for the control of mobile robots **flocking** that uses very **simple proximity sensors** and information about their own **absolute headings**, in a free of obstacles environment.

Experimental Test Platform (I)



- *E-puck* robots used.
- Wheeled cylindrical robots (7*cm* of diameter)
- 8 infra-red proximity sensors
  - Distributed around the body
  - Used to estimate **angle** and **distance** to nearby robots.
- 3 infra-red ground sensors
- 3-axis accelerometer
- Differential drive system.

POLITÉCNICA

Experimental Test Platform (II)

- *Virtual compass* used to get own heading in a global coordinate system.
  - Robots move in a vertical plane.
  - Magnetic cubic extension added to bottom of the robots, permitting the robots to move attached to a metallic wall.
  - Accelerometers provide global heading.
  - A preliminary calibration is needed to overcome with the accelerometer bias.
- **Communication**, necessary in some parts of the algorithm, implemented through bluetooth and a PC.
- **Webots** simulator used to test the algorithm, using a realistic model of the robots.

POLITÉCNICA

Algorithm (I)

- Each robot reacts to every object detected by its IR sensors, being attracted or repelled depending on the measured distance.
- Robots try to maintain a desired distance between them, just using IR.

$$\vec{V}_{aggregation} = \sum \vec{V}_i \quad (1)$$

$$|\vec{V}_i| = \begin{cases} K_1(desiredDist - dist_i), & \text{if } dist_i \leq desiredDist \\ K_2(dist_i - desiredDist), & \text{if } desiredDist < dist_i \leq maxDist \\ 0, & \text{if } maxDist < distSensor_i \end{cases} \quad (2)$$

$$\arg(\vec{V}_i) = \begin{cases} angle_i, & \text{if } dist_i \leq desiredDist \\ angle_i + \pi, & \text{if } desiredDist < dist_i \leq maxDist \\ 0, & \text{if } maxDist < distSensor_i \end{cases} \quad (3)$$

POLITÉCNICA

Algorithm (II)

- In order to move in the pre-defined desired direction, each robot reacts generating another desired virtual velocity $\vec{V}_{desiredDirection}$:

$$|\vec{V}_{desiredDirection}| = K_3 \quad (4)$$

$$\arg(\vec{V}_{desiredDirection}) = desiredDirection - myHeading \quad (5)$$

- In order to make the robots to move together as a group in the same direction and maintaining the desired distance between them, both virtual velocities are added resulting in the final total virtual velocity:

$$\vec{V}_{total} = \vec{V}_{aggregation} + \vec{V}_{desiredDirection} \quad (6)$$

Low Level Controller

- *Low Level Controller* (LLC) necessary to translate the virtual velocity into wheel speeds in differential drive robots:

$$V_{linear} = K_4|V|cos(\theta) \tag{7}$$

$$V_{angular} = \begin{cases} K_5(\theta + \pi), & \text{if } \theta < -\pi/2 \\ K_5\theta, & \text{if } \pi/2 > \theta > -\pi/2 \\ K_5(\theta - \pi), & \text{if } \theta > \pi/2 \end{cases} \tag{8}$$

$$s_{motor-right} = V_{linear} + B * V_{angular} \tag{9}$$

$$s_{motor-left} = V_{linear} - B * V_{angular} \tag{10}$$

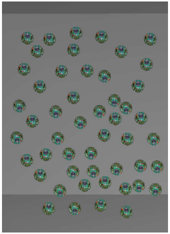
- LLC allows forwards and backwards movement.
- By applying the LLC to $\vec{V}_{total}$ in all the robots, it results in a flocking of the robots towards the pre-defined direction.

POLITÉCNICA

Search for Lost Flock Algorithm

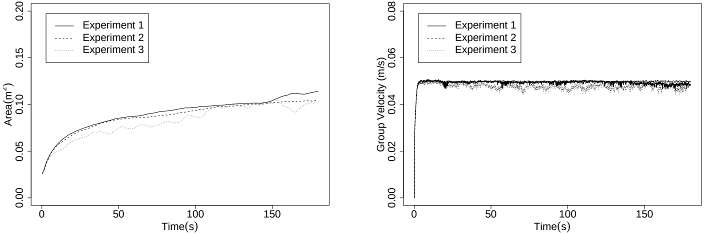
- Eventually a robot may stop detecting any robot and can not follow the flock.
- Simple algorithm to look for the group was designed and implemented:
  1. Lost robot orientates in the direction of the last seen robot.
  2. It moves during few seconds in that direction.
  3. If the flock is still not found, It moves in the direction that the flock is moving (*desiredDirection*)
  4. If after a certain time the flock is not found the robot consider itself as completely lost and stops.

Experimental Results (I)



- Real robots: Flock of 7 robots.
- Simulation: Flocks of 7 & 50 robots.
- 3 types of experiments:
  1. Unbounded arena. Forward movement. (Sim.)
  2. Bounded arena. Back-Forward movement. (R.R. & Sim.)
  3. Unbounded arena. Robots change desired direction of movement progressively with time. (Sim.)
- 3 parameters were measured to analyze the performance:
  - **Group Velocity**;
  - **Area** given by Convex Hull (area of the minimum convex polygon containing all the robots);
  - **Polarization**, mean angle deviation between the group heading and each individual heading.

Experimental Results (II)



7 Simulated Robots

	Group Velocity (m/s)	Area (m^2)	Polarization (rad)
7 Real Rob.	0.04	0.12	0.07
7 Simulated Rob. (3 types experiments)	0.05	0.1	0.05
50 Simulated Rob.(3 types experiments)	0.05	0.81.1	0.05

Values reached in the plateau.

- Polarization reaches its minimum when g. vel. is maximum.

Conclusions

- Algorithm works well according to the carried out experiments.
- A flock results from the local interactions between the robots:
  - moving at the desired velocity
  - in a cohesive way.
- Working with real robots. using simple IR and global heading provided by the *on-board compass*,
- Simulation with 50 robots shows the scalability.
- Group velocity & polarization have reasonable values.
- Absence of leader makes it tolerant to single robot failure.
- Performance of experiments with circular movement shows that the algorithm could be used for movements in which turns are involved.

Future Work

- Use of a real compass instead of virtual one, since magnets and vertical movement were a limiting factor in the velocity and smoothness of the movements.
- If obstacles need to be avoided, an on board relative positioning system to detect nearby robots will be necessary.

Thank you for your Attention!

inaki.navarro@upm.es

POLITÉCNICA