DIS
Project presentation

JAN BRIACHETTI – TEAM 6

07.07.2021
Localization technique: GPS (GNSS)

- **Advantages**
  - Accuracy
  - No error accumulation

- **Drawbacks**
  - Update frequency (1 Hz)
  - External device
  - Reliability (tunnel, inside, etc.)

- **Solution** → Couple with encoders in the Kalman Filter (5-10 times better accuracy)
Obstacle avoidance – Braitenberg FSM

- FSM: threshold (= 1’000) based on distance sensor values
- Example: obstacle world with flocking in real-time
- Future improvement → add bias or noise
Crossing world: Flocking vs Flocking

• Reynolds rules: manual testing until we found good values for rules weights and thresholds
Accelerometer and encoder odometry

- **Accelerometer:**
  - ACC_OFFSET = calibration
  - Orientation: done with wheels encoders

- **Wheels encoders:**

\[
\begin{align*}
\Delta x &= \Delta s \cos(\theta + \frac{\Delta \theta}{2}) \\
\Delta y &= \Delta s \sin(\theta + \frac{\Delta \theta}{2}) \\
\Delta \theta &= \frac{\Delta s_r - \Delta s_l}{b}
\end{align*}
\]
Formation

Crossing Formation

Formation obstacle
Formation: limits and improvements

- **Limits**
  - Calibration: wheels rolling without displacement
  - Control parameters not optimal

- **Ideas for improvement:**
  - Leader informed of the position of followers
  - Random noise to avoid local minima
  - PSO

Metrics formation in obstacle world
My contribution to the DIS project

- Computing the different performance metrics in the supervisor and saving the results into separate files.

- Merging the different algorithms into the same controller and writing the selection process.

- Implementing the Reynolds algorithm based on the intensity of the IR signals of the robots (relative positioning).
Problem with relative positioning

While the robot is encountering an obstacle, the speeds of the wheels will not exactly match the movement of the robot.

This leads afterwards to false positioning of the migration goal.
By using GPS-based odometry, the robots can correct their relative position after encountering an obstacle.
Multi-robot navigation in cluttered and dynamic environments

Maïk Guihard
Localization: Kalman Filter

- Combines GPS and encoder odometry
- Simplified version of the filter
- Prediction step done 1/5 times
- If for some reason gps gives no new read, positioning still works
- Diagonal matrices -> simple equations, hard-coded without matmul

\[ R = \begin{pmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad Q = \begin{pmatrix} 0.025 & 0 & 0 \\ 0 & 0.025 & 0 \\ 0 & 0 & 0.025 \end{pmatrix} \]
Particle swarm optimisation (PSO)

**SUPERVISOR**
- For N epochs:
  - For each particle in swarm:
    - Reset all robot positions
    - Send parameters to robots
    - While not finished:
      - Calculate fitness of swarm
    - Average fitness over time
    - Update particles

**CONTROLLER**
(all robots have the same)
- (Re) initialize gps, enc, kalman
- For J iterations:
  - do odometry + kalman
  - read dist sensors
  - flocking / formation / obstacles
- Send back "finished" to supervisor
Particle swarm optimisation (PSO): parameters

◊ Parameters to optimize:
  ◊ Swarming:
    ◊ Aggregation & Dispersion, weight + threshold
    ◊ Consistency & Migration weight
  ◊ Formation:
    ◊ Forward & rotational control coefficients
◊ PSO parameters (final):
  ◊ Nb particles (30)
  ◊ Nb of index-based neighbors (4)
  ◊ Nb epochs (20)