Event handling with real e-pucks using threshold-based task allocation
December 15th, 2015

Ducret William J. (MT) — Giannakopoulos Athanasios (IN) — Kyritsis Georgios (IN)
Event handling with real e-pucks using threshold-based task allocation

Defining the problem

Threshold-based task allocation in distributed intelligent system

\[ f(x) = \begin{cases} 
  s - \theta > 0, & \text{perform a task} \\
  s - \theta \leq 0, & \text{do not perform a task} 
\end{cases} \]

What we use:

- Homogeneous threshold
- Fixed threshold
- Local stimulus
Event handling with real e-pucks using threshold-based task allocation

Camera-based stimulus and threshold

- 640 X 480 resolution camera
Number of blue pixels captured depends on the distance from the cylinder

$S > \Theta$

Event detected
Event handling with real e-pucks using threshold-based task allocation

E-puck controller pseudo code

```plaintext
init() robot;
while (θ < 2π or event = False) do
    θ = θ + rotate(Δθ);
    takePicture();
    if (blue > red) and (blue > green) then
        event = True;
        state = eventHandling;
    end
end
```

while True do
    if state = eventHandling then
        go to event;
    end
    if state = obstacleAvoidance then
        avoid obstacle;
    end
    if state = randomWalk then
        perform random Walk;
    end
    takePicture();
    if (blue > red) and (blue > green) then
        if close to obstacle then
            state = eventHandling;
        else
            state = obstacleAvoidance;
        else
            state = randomWalk;
        end
    end
    # update speed according to state
    updateSpeed(state);
end
Event handling with real e-pucks using threshold-based task allocation

Part of image kept for analysis

Steer left  Straight ahead  Steer right
Optimization techniques

• Each e-puck starts a rotation around its axis as soon as it handles an event to detect new ones

• Each e-puck changes direction if it detects another e-puck in front of it while traveling to an event location

• Most probably the other e-puck goes for the same event

Abort
Event handling with real e-pucks using threshold-based task allocation

Simulation results (1/2)
Simulation results (2/2)

- [Graph 1]: Distance vs. Threshold value with least squares fit.
- [Graph 2]: Time vs. Threshold value with least squares fit.
Real world implementation

Modifications from webots implementation
  • Red color used for event detection
  • Less pixels due to RAM restriction
  • Modifications in code implementation compared to webots simulation
  • Energy efficiency taken into account
Event handling with real e-pucks using threshold-based task allocation

Problems in real world implementation

What we noticed:

• Environment lighting is of significant importance while using the e-puck camera

• Take into account the computational power

Possible problems:

• Detect obstacle as event collision

• Red color/ambient light might be reflected from the arena walls

• Don’t detect the cylinder due to small angular resolution
Event handling with real e-pucks using threshold-based task allocation
Event handling with real e-pucks using threshold-based task allocation

Demo time