Multi-Agent Search using Voronoi Partition and Voronoi 1D experiment

Fujita Lab, Dept. of Control and System Engineering,
FL07-13-2: July 09,2007
David Asikin

Outline

- Review

- Introduction
- Decreasing density function
- Stability
- Conclusion

- Work progress:
  1. Simulation of Voronoi 2D with density function
  2. Lloyd's Algorithm 1D experiment
- Future Work

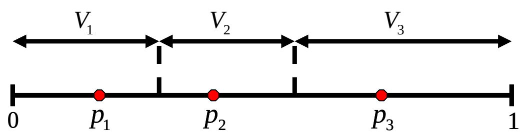
Review

- Review menu:
  1. Voronoi partition in 1D & 2D
  2. Lloyd's Algorithm
  3. Objective function:
     - Sensing performance
     - Density function

Review

- Voronoi partition:
  The set of all points q whose distance from p_i is less than or equal to the distances from all other p_j

$$V_i = \{ q : (\forall j \neq i) \|q - p_i\| \leq \|q - p_j\| \}$$



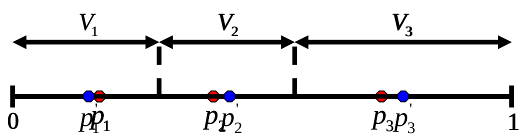
Review

- Lloyd's Algorithm:
  A method for evenly distributing points over an unknown area.

- The steps:
  Step 0: Start with a random area, {*Wi*}, and random points, {*pi*}.
  Step 1: Construct Voronoi partition {*Vi*}, generated by {*pi*}.
  Step 2: Update pi to be the centroid of *Vi*. Return to Step 1.

Review

Step 0: Start with a random area, {*Wi*}, and random points, {*pi*}.
Step 1: Construct Voronoi partition {*Vi*}, generated by {*pi*}.
Step 2: Update pi to be the centroid of *Vi*. Return to Step 1.



Review

- Objective function:

$$H(p,W)=\int_W f(\|q-p\|)\phi(q)dq$$

$f(\|q-p\|)$: Sensing performance (f=big → poor sensing)

$\phi(q)$: Density function

p = agent position *q* = object *W* = partition

- By minimizing H, we get optimum coverage. Why? When H = min, agents move to the area with the highest occurrence possibility.

Review

- We assume: $f(\|q-p\|)=\|q-p\|^2$
  Simplify the objective function using parallel axis theorem.

$$H(p,W)=\int_W f(\|q-p\|)\phi(q)dq$$

↓

$$H(p,W)=H(c_W,W)+M_W\|p-c_W\|^2$$

$$M_W=\int_W \phi(q)dq \quad :mass$$

- To minimize this, $p=C_W$ (=centroid of *W* partition)
- Therefore, use this as an input to make agents go to centroid of the partition.

Introduction

Autonomous *N* agents equipped with sensors deploy themselves in an optimal way over an unknown area.

- Application:
  search & rescue, environmental monitoring, military and defence application, etc.
- Objective: multi-agent search
  Agents deploy themselves optimally in Q while updating (=reducing) uncertainty density function and gather information till the uncertainty is below a certain level.

Decreasing Density Function

- At each iteration, after deploying themselves optimally, the sensors gather information about *Q*, reducing the density function as:

$$\phi_{n+1}(q)=\phi_n(q)\min_i\{\beta(\|x_i-q\|)\}$$

ϕ_n : density function

$\beta(\|x_i-q\|)$: sensing performance

x_i : position of the *i*-th sensor

- $\beta:\mathbb{R}_+\mapsto[0,1]$ is the factor of reduction
- Why $\min_i\{\beta(\|x_i-q\|)\}$?
  Only the agent with the smallest β can reduce the uncertainty by the largest amount.

Decreasing Density Function

- For sensing performance function:

$$\beta(\|x_i-q\|)=1-ke^{-\alpha\|x_i-q\|^2} \qquad k\in(0,1),\ \alpha>0$$

- As x_i approaches *q* → β decreases (=good sensing)
- As x_i go further away from *q* → β increases (=bad sensing)

Conclusion:
When $x_i = q$, β is minimum.

Decreasing Density Function

- For objective function:

$$H_n=\int_Q \Delta\phi_n(q)dq=\int_Q \max\{\phi_n(q)-\phi_{n+1}(q)\}dq$$

$$=\int_Q\left\{\phi_n(q)-\phi_n(q)\min_i\{\beta(\|x_i-q\|)\}\right\}dq$$

$$=\int_Q \phi_n(q)\left\{1-\min_i\{\beta(\|x_i-q\|)\}\right\}dq$$

$$=\sum_i\int_{V_i}\phi_n(q)\{1-\beta(\|x_i-q\|)\}dq$$

$$=\sum_i\int_{V_i}\phi_n(q)\left\{1-\left(1-ke^{-\alpha\|x_i-q\|^2}\right)\right\}dq$$

$$=\sum_i\int_{V_i}\phi_n(q)ke^{-\alpha\|x_i-q\|^2}dq$$

Decreasing Density Function

- Objective function: $H_n = \sum_i \int_{V_i} \phi_n(q) k e^{-\alpha\|x_i - q\|^2} dq$

$$\frac{\partial H_n}{\partial x_i} = \sum_i \int_{V_i} \phi_n(q) k e^{-\alpha\|x_i-q\|^2} (-2\alpha)(x_i - q)\, dq$$

① $\phi_n'(q) = \phi_n(q) k e^{-\alpha\|x_i-q\|^2}$

$$= (-2\alpha)\sum_i \int_{V_i} \phi_n'(q)(x_i - q)\, dq$$

$$= (-2\alpha)\sum_i \int_{V_i} \left(\phi_n'(q) x_i - \phi_n'(q) q\right) dq$$

$$= (-2\alpha)\sum_i \left\{\int_{V_i} \phi_n'(q) x_i dq - \int_{V_i} \phi_n'(q) q dq\right\}$$

$$= (-2\alpha)\left\{\tilde{M}_{V_i} x_i - \tilde{M}_{V_i} \tilde{C}_{V_i}\right\}$$

$$= -2\alpha \tilde{M}_{V_i}\left(x_i - \tilde{C}_{V_i}\right)$$

② $M_V = \int_V \phi(q)\, dq$, $C_V = \frac{1}{M_V}\int_V q\phi(q) dq$

Decreasing Density Function

- From $\frac{\partial H_n}{\partial x_i} = -2\alpha \tilde{M}_{V_i}\left(x_i - \tilde{C}_{V_i}\right)$, we can conclude that the necessary condition for optimality is,

$$x_i = \tilde{C}_{V_i}$$

($\tilde{M}_{V_i}$ and $\tilde{C}_{V_i}$ are respectively the mass and the centroid of V_i with respect to ϕ_n')

- Assume the system as $\dot{x}_i = u_i$. Use the result above as an input:

$$u_i = -k_{prop}\left(x_i - \tilde{C}_{V_i}\right) \qquad k_{prop} > 0$$

This moves the agent towards $\tilde{C}_{V_i}$.

DDF Summary

- Objective:
  Agents deploy themselves optimally in Q while reducing uncertainty density function and gather information till the uncertainty is below a certain level.
- Decreasing density function:

$$\phi_{n+1}(q) = \phi_n(q) \min_i \left\{\beta\left(\|x_i - q\|\right)\right\} \qquad \beta\left(\|x_i - q\|\right) = 1 - k e^{-\alpha\|x_i - q\|^2}$$

- Objective function:

$$H_n = \int_Q \Delta\phi_n(q)\, dq = \sum_i \int_{V_i} \phi_n(q) k e^{-\alpha\|x_i-q\|^2} dq$$

$$\frac{\partial H_n}{\partial x_i} = -2\alpha \tilde{M}_{V_i}\left(x_i - \tilde{C}_{V_i}\right)$$

- On system $\dot{x}_i = u_i$, use this as an input: $u_i = -k_{prop}\left(x_i - \tilde{C}_{V_i}\right)$

Stability

- Consider the $V(X) = -H_n$, where $X = (x_1, x_2, \ldots, x_N)$ represents the configurations of N agents.

$$\dot{V}(X) = -\frac{dH_n}{dt} = -\sum_i \frac{\delta H_n}{\delta x_i}\dot{x}_i$$

$$= \sum_i 2\alpha \tilde{M}_{V_i}\left(x_i - \tilde{C}_{V_i}\right)\dot{x}_i$$

$$= \sum_i 2\alpha \tilde{M}_{V_i}\left(x_i - \tilde{C}_{V_i}\right)\left(-k_{prop}\left(x_i - \tilde{C}_{V_i}\right)\right)$$

$$= -2\alpha k_{prop}\sum_i \tilde{M}_{V_i}\left(x_i - \tilde{C}_{V_i}\right)^2$$

- Since $\alpha > 0$, $k_{prop} > 0$, $\dot{V}$ is a negative definite.

Stability

- By LaSalle's invariance principle, the trajectories of the agents governed by control law:

$$u_i = -k_{prop}\left(x_i - \tilde{C}_{V_i}\right)$$

starting from any initial configuration, will asymptotically converge to centroidal Voronoi partition $\tilde{C}_{V_i}$ with respect to the density function:

$$\phi_n'(q) = \phi_n(q) k e^{-\alpha\|x_i - q\|^2}$$

- Note: $\tilde{C}_{V_i} = \frac{1}{\tilde{M}_{V_i}}\int_V q\phi_n'(q) dq$ $\qquad \tilde{M}_{V_i} = \int_V \phi_n'(q)\, dq$

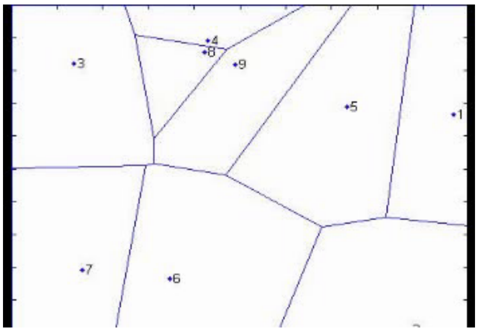
Conclusion

- Objective:
  Agents deploy themselves optimally in Q, gather information in their respective Voronoi partition and hence reduce uncertainty density function.
  (Note: the iterations are continued till the uncertainty in the is below a required level)
- The one-step optimal deployment is the centroidal Voronoi configuration with respect to the reduced density function.
- Proven stable by LaSalle's invariance principle.

Work Progress (1-0-a)

Tokyo Institute of Technology

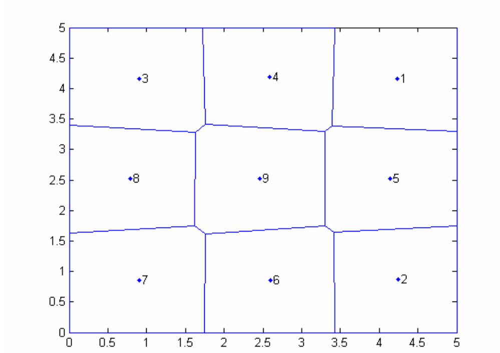
- Simulation of Voronoi 2D with constant density function $\phi = 1$: (n = 9 agents)



Work Progress (1-0-b)

Tokyo Institute of Technology

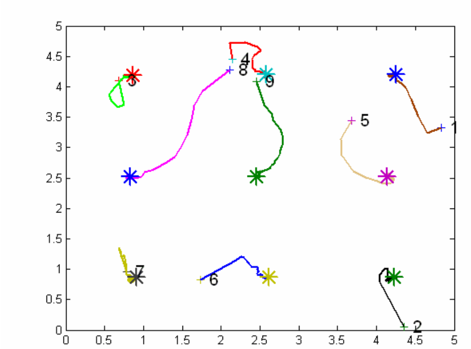
- Last position:



Work Progress (1-0-c)

Tokyo Institute of Technology

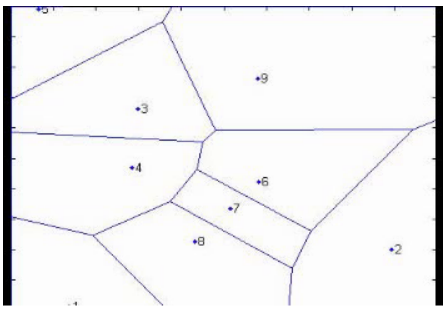
- Trajectory graph:



Work Progress (1-1-a)

Tokyo Institute of Technology

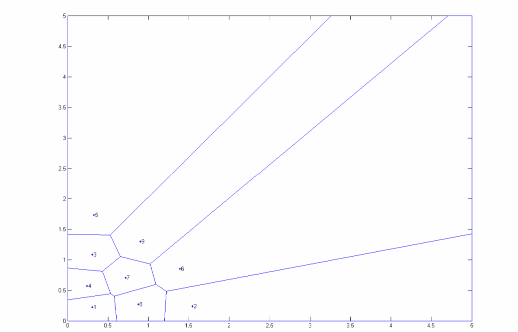
- Simulation of Voronoi 2D with density function $\phi = \exp\left(-x^2 - y^2\right)$: (n = 9 agents)



Work Progress (1-1-b)

Tokyo Institute of Technology

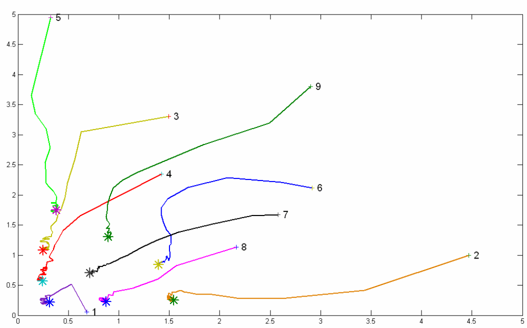
- Last position:



Work Progress (1-1-c)

Tokyo Institute of Technology

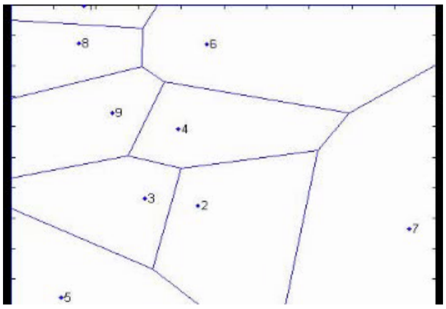
- Trajectory graph:



Work Progress (1-2-a)

Tokyo Institute of Technology

- Simulation of Voronoi 2D with density function $\phi = \exp\left(-x^2 - y^2\right)$: (*n* = 9 agents)

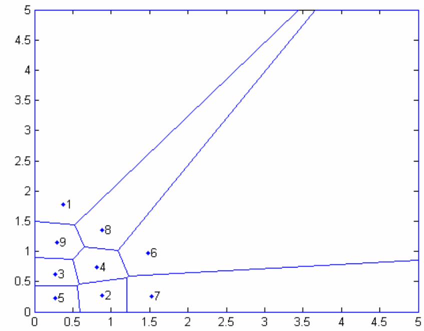


Tokyo Institute of Technology Fujita Laboratory

Work Progress (1-2-b)

Tokyo Institute of Technology

- Last position:

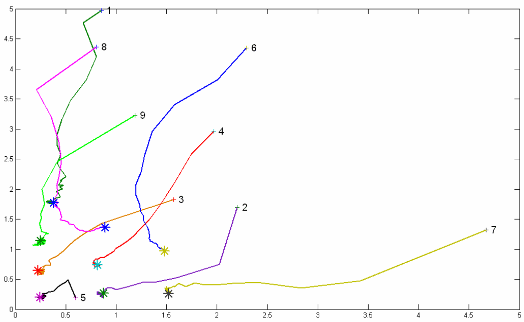


Tokyo Institute of Technology Fujita Laboratory

Work Progress (1-2-c)

Tokyo Institute of Technology

- Trajectory graph:

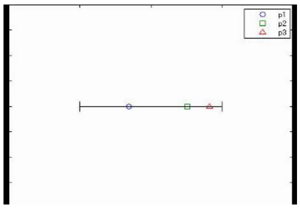


Tokyo Institute of Technology Fujita Laboratory

Work Progress (2)

Tokyo Institute of Technology

- Lloyd's Algorithm 1D simulation with density function $\phi = 1$:



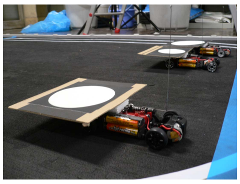
- Objective of the experiment:
  To test the convergence characteristic of Lloyd's Algorithm in 1D using RC cars.

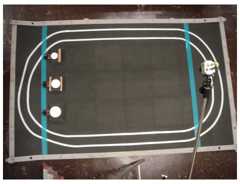
Tokyo Institute of Technology Fujita Laboratory

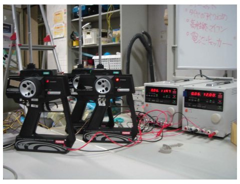
Work Progress (2)

Tokyo Institute of Technology

- Equipment:





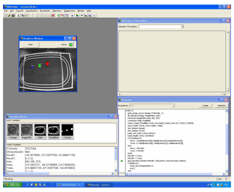


Tokyo Institute of Technology Fujita Laboratory

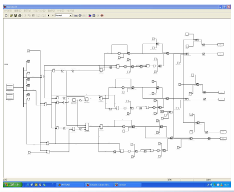
Work Progress (2)

Tokyo Institute of Technology

- Software:



1. Halcon:
   Orders the camera to capture the position of the cars.

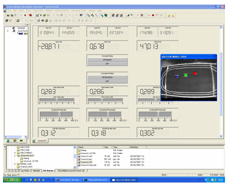


2. Simulink:
   Processes the data captured by Halcon. (Lloyd's Algorithm block is written here)

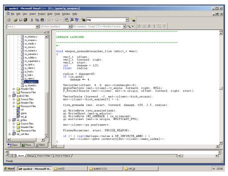
Tokyo Institute of Technology Fujita Laboratory

Work Progress (2)

- Software:



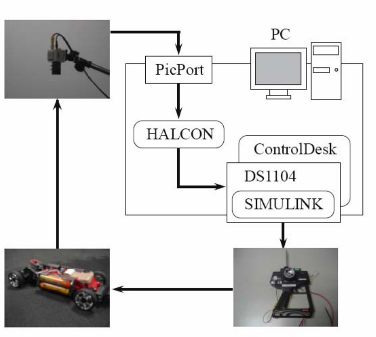
3. Control Desk:
   1. Receives order (output) from Simulink and passes it to RC motors.
   2. Monitors cars position, speed, voltage given, etc.



4. Microsoft Visual C++: Links data between Halcon and Simulink.

Work Progress (2)

- Software diagram:



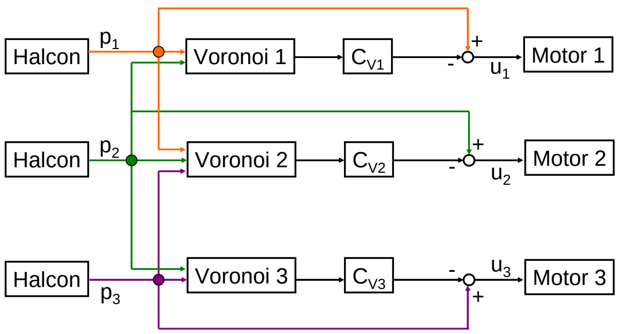
Work Progress (2)

- Experiment order:
1. Make circles out of cardboard (as cars position) for camera to read, and prepare the field.
2. Make Halcon program.
3. Make C++ program to link data from Halcon to Simulink.
4. Make Lloyd's Algorithm block diagram in Simulink.
5. Link Simulink with Control Desk and make monitors in Control Desk.

\+ Adjust voltage, cars speed, direction, camera, etc.
\+ Debug and compile.

Work Progress (2)

- Lloyd's Algorithm 1D block diagram:

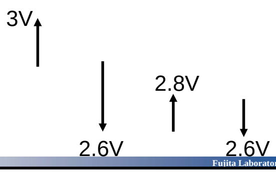


Work Progress (2)

- Trouble:
  - Different time/day, different car and camera characteristic.
  - Adjustment for car speed, direction.
  - Limited power on cars battery reflects on performance.
  - Friction between the field and the tire.
  - Camera's vision is warped (afflicts on video capturing performance).
  - <u>**Backward movement.**</u>
  - Etc.

- Backward movement:

Forward : 3V
Stop : 2.8V
Backward : 2.6 V



Future Work

- Lloyd's Algorithm 1D experiment revision.
- Lloyd's Algorithm 1D experiment with density function.
- Multi-Agent Search 2D simulation.
- Read more coverage control papers, etc.

- If possible, Lloyd's Algorithm 2D experiment with constant density function.

References

- Guruprasad K.R., Debasish Ghose, “Multi-Agent Search using Voronoi Partitions”, ACODS, 2007
- Jorge Cortes, Sonia Martinez, Timur Karatas, Francesco Bullo, “Coverage Control for Mobile Sensing Networks”, IEEE, 2007
- Bruce Francis, “Distributed Control of Autonomous Mobile Robots”, 2006

Any question?