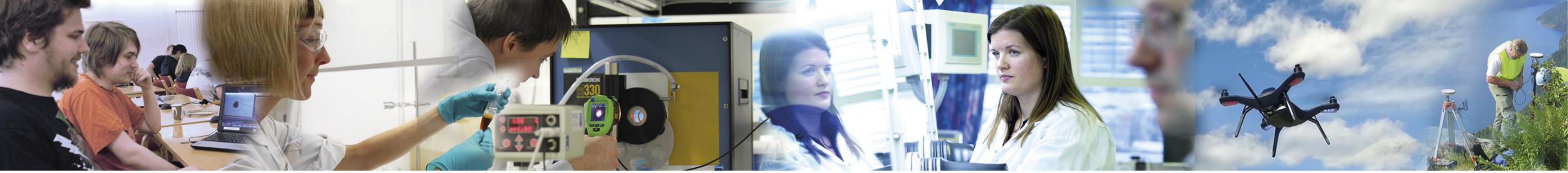


UiO : Department of Technology Systems
University of Oslo

TEK5010/9010 - Multiagent systems 2023
Lecture 5

Swarm robotics 2

Jonas Moen

Highlights lecture 5 – Swarm robotics 2*

- Swarm collective decision-making
  - Terminology and notation
  - The decision-making process
  - Different models (voting, urn, Hegselmann-Krause, etc)

- Swarm case study: Adaptive aggregation, BEECLUST

*Hamann, 2018: chapter 6 and 7

Swarm collective decision-making

Terminology and notation [Hamann, 2018]:

Swarm has to decide over a set of options $O = \{O_1, O_2, \ldots, O_m\}$ with $m > 1$ options. Task is to achieve consensus on one option O_j.

- $q(O_j)$ is quality of option
- A robot i has a defined option o_i at any time
- $\mathcal{N}_i$ defines the neighbourhood of robot i without robot i
- $\mathcal{G}_i$ defines the neighbourhood of robot i including robot i

Swarm collective decision-making

Decision-making process:

Image: Figure 6.5, Hamann, 2018

Swarm collective decision-making

Decision-making process:

1. Exploration phase: robots explore local area in search of information on quality of options.
2. Dissemination phase: robots signal its opinion to neighbours. Typically signal is correlated with quality of opinion, e.g. duration and/or intensity.
3. Opinion switch: robots follow a decision-making rule to switch their opinion, e.g. voter rules.

Swarm collective decision-making

Decision-making process:

- Robots do not have to follow all 3 phases
- Process need not be synchronized among robots
- Signalling needs to be agreed upon

- How to connect micro-rule with global behaviour?

Swarm collective decision-making

The voter model [Clifford & Sudbury, 1973]:

A robot i considers its neighbours' opinions o_j with $j \in \mathcal{N}_i$ and picks a neighbour j at random and switches to its opinion.

- Very simple model
- High accuracy
- Slow convergence

Swarm collective decision-making

The majority rule:

A robot i considers its neighbourhood group $\mathcal{G}_i$ and counts the occurrence w_j of each option in O. The robot them switches its opinion to the most frequent option O_k with $k = \mathrm{argmax}\, w_j$, that is, the majority within its group.

- Fast convergence
- Less accurate than the voter model

Swarm collective decision-making

Urn models:

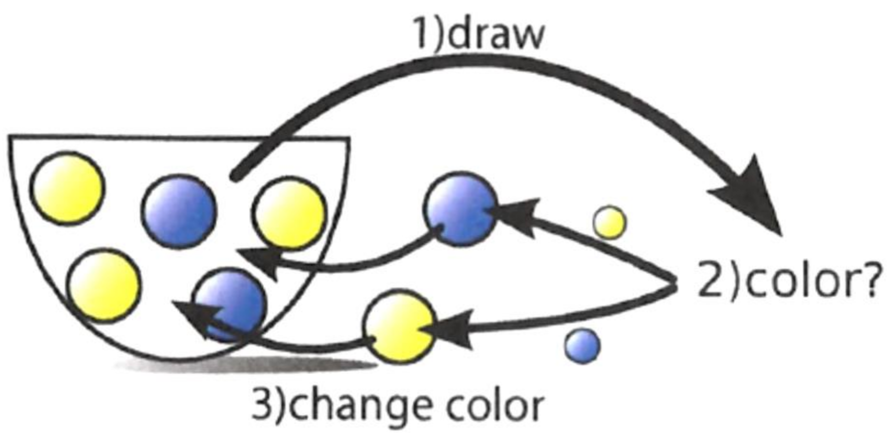


Image: Figure 6.6, Hamann, 2018

Swarm collective decision-making

Urn models:

No spatial information, i.e. a well-mixed density is assumed

- The Ehrenfest model – an introduction to urn models (originally diffusion processes in thermodynamics)
- The Eigen model – self-organization through positive feedback gives perfect consensus
- The swarm urn model – self-organization through positive and negative feedback to avoid perfect consensus

Swarm collective decision-making

Ehrenfest urn model
[Ehrenfest & Ehrenfest , 1907]:

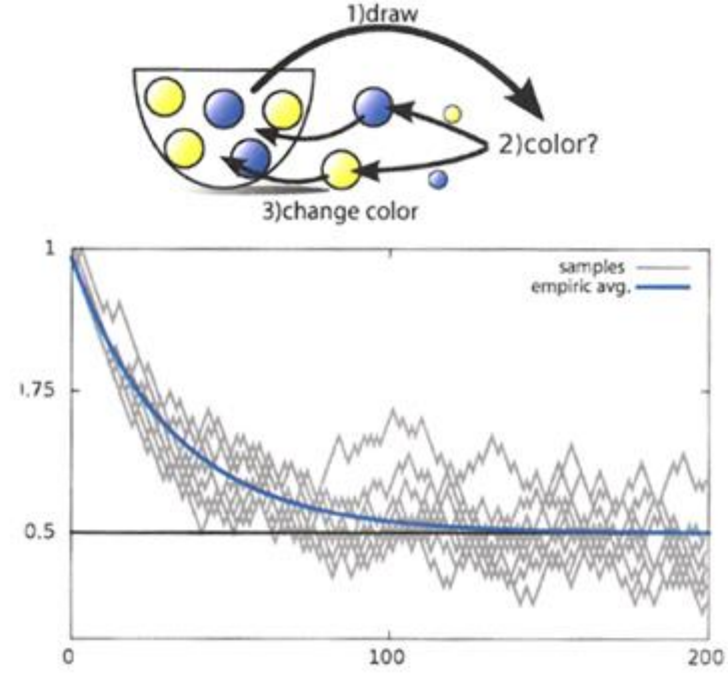


Image: Figure 6.6, Hamann, 2018

Swarm collective decision-making

Ehrenfest urn model [Ehrenfest & Ehrenfest, 1907]:

$$B(t+1) = B(t) + \Delta B\big(B(t)\big)$$

where $B(t)$ is number of balls of colour C at time t

$\Delta B\big(B(t)\big)$ is expected change in balls of colour C

$\Rightarrow$ An exponential convergence is expected

Swarm collective decision-making

Ehrenfest urn model [Ehrenfest & Ehrenfest, 1907]:

Assume 64 balls in urn, 16 Blue and 48 Red:

$$P_{Blue} = \frac{16}{64} = 0.25 \quad \text{and} \quad P_{Red} = \frac{48}{64} = 0.75$$

$$\Longrightarrow \Delta B\left(\frac{16}{64}\right) = (-1)P_{Blue} + (+1)P_{Red} = 0.5$$

Swarm collective decision-making

Ehrenfest urn model [Ehrenfest & Ehrenfest, 1907]:

$$\Delta B\big(B(t)\big) = -2\frac{B}{N} + 1$$

where N is total number of balls

The recurrence $B(t+1) = B(t) + \Delta B\big(B(t)\big)$ can be solved by a generating function assuming $B(t=0)$ is given.

Swarm collective decision-making

Eigen urn model
[Eigen & Winkler, 1993]:

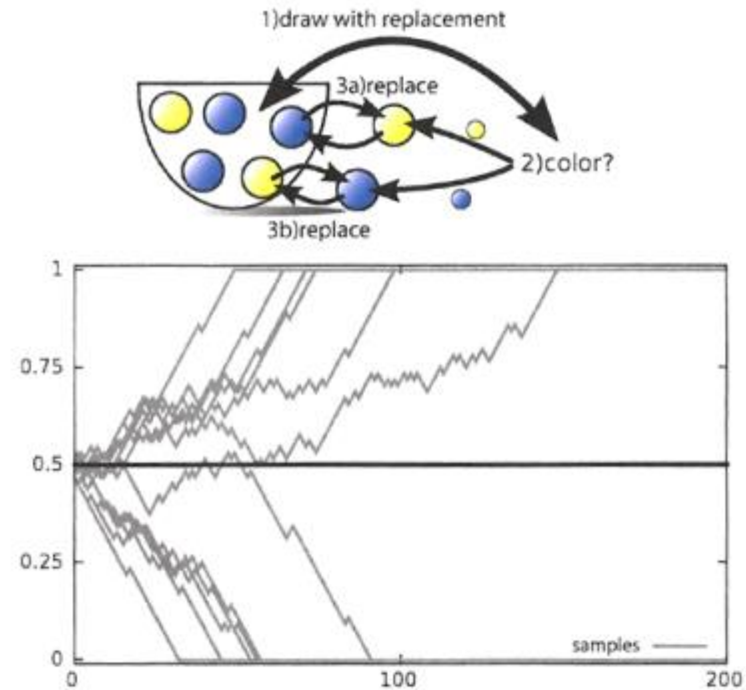


Image: Figure 6.7, Hamann, 2018

Swarm collective decision-making

Eigen urn model [Eigen & Winkler, 1993]:

$$\Delta B\big(B(t)\big) = \begin{cases} 2\frac{B}{N} - 1, & \text{for } B \in [1, N-1] \\ 0, & else \end{cases}$$

The Eigen model is an ‘inverted’ Ehrenfest model.

Special care must be taken for the extreme cases of B=0 and B=N.

Swarm collective decision-making

Swarm urn model
[Hamann, 2013]:

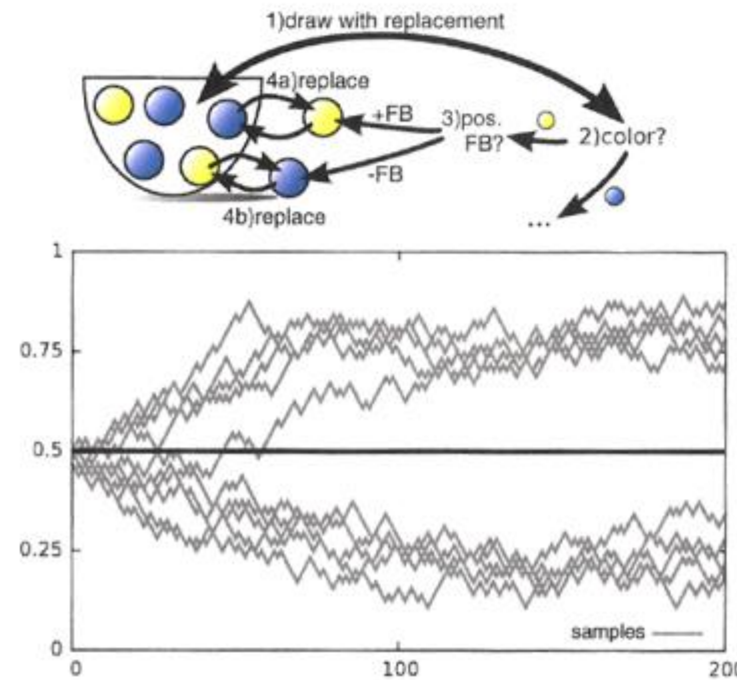


Image: Figure 6.10, Hamann, 2018

Swarm collective decision-making

Swarm urn model [Hamann, 2013]:

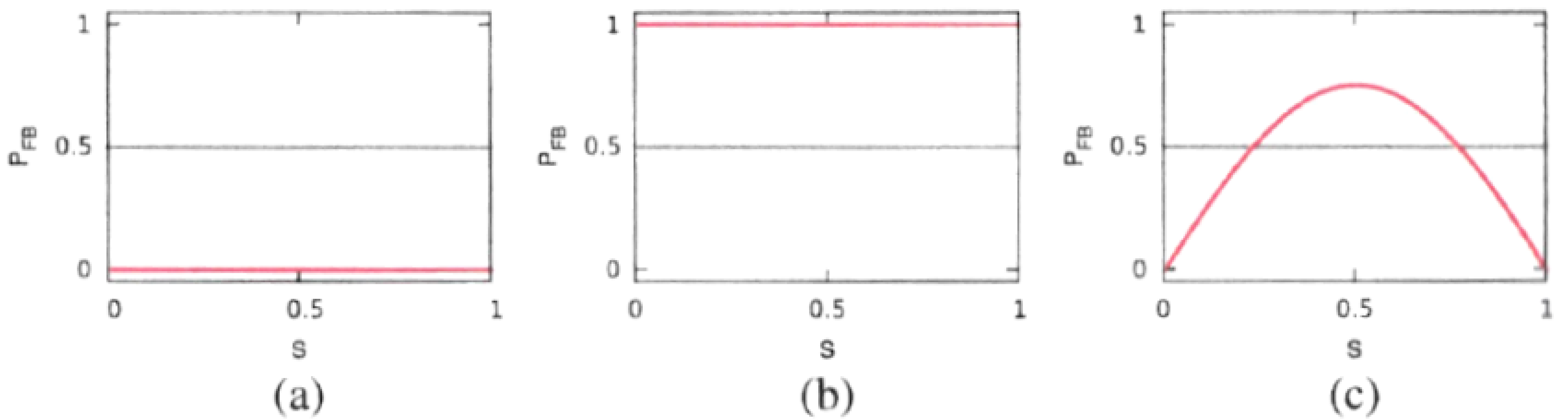


Image: Figure 6.9, Hamann, 2018

Swarm collective decision-making

Swarm urn model [Hamann, 2013]:

$$\Delta s(s) = 4\left(P_{FB}(s) - \frac{1}{2}\right)\left(s - \frac{1}{2}\right)$$

Where Ehrenfest $P_{FB}(s) = 0 \quad \Rightarrow s^* = 0.5$

Eigen $P_{FB}(s) = 1 \quad \Rightarrow s^* = 0 \lor 1$

Swarm $P_{FB}(s) = 0.75 \sin \pi s \quad \Rightarrow s^* = 0.23 \lor 0.77$

Swarm collective decision-making

Hegselmann and Krause [Hegselmann-Krause, 2002]:

Clustering of opinions by having robots move to the centre of gravity of their neighbourhood:

$$x_i = \frac{1}{|\mathcal{G}_i|} \sum_{j \in \mathcal{G}_i} x_j + \varepsilon_i$$

where $\mathcal{G}_i = \{1 \leq j \leq N \colon \|x_i - x_j\| \leq 1\}$ and ε_i is a noise term

Swarm collective decision-making

Hegselmann and Krause [Hegselmann-Krause, 2002]:

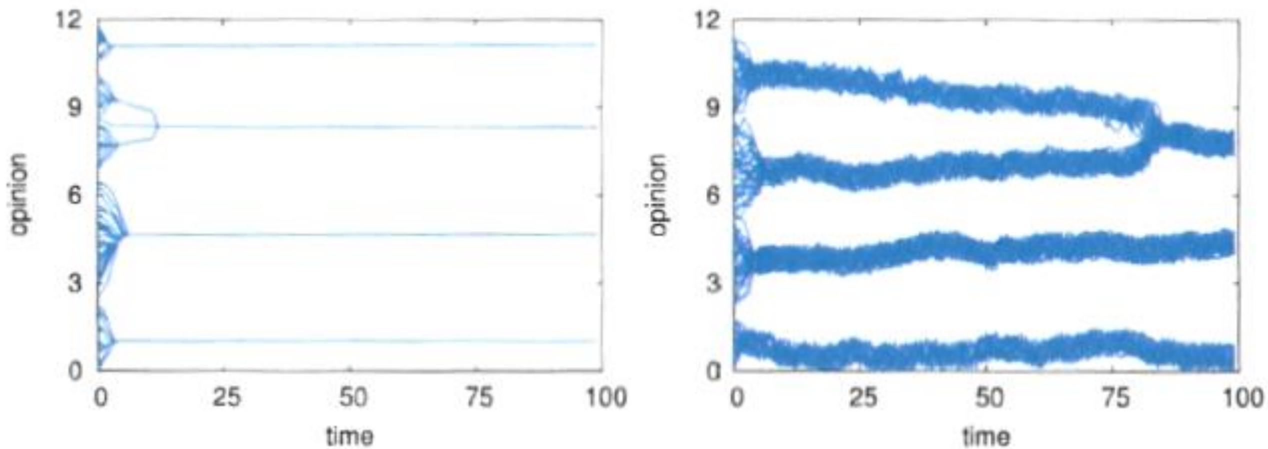


Image: Figure 6.12, Hamann, 2018

Swarm collective decision-making

Various other models:

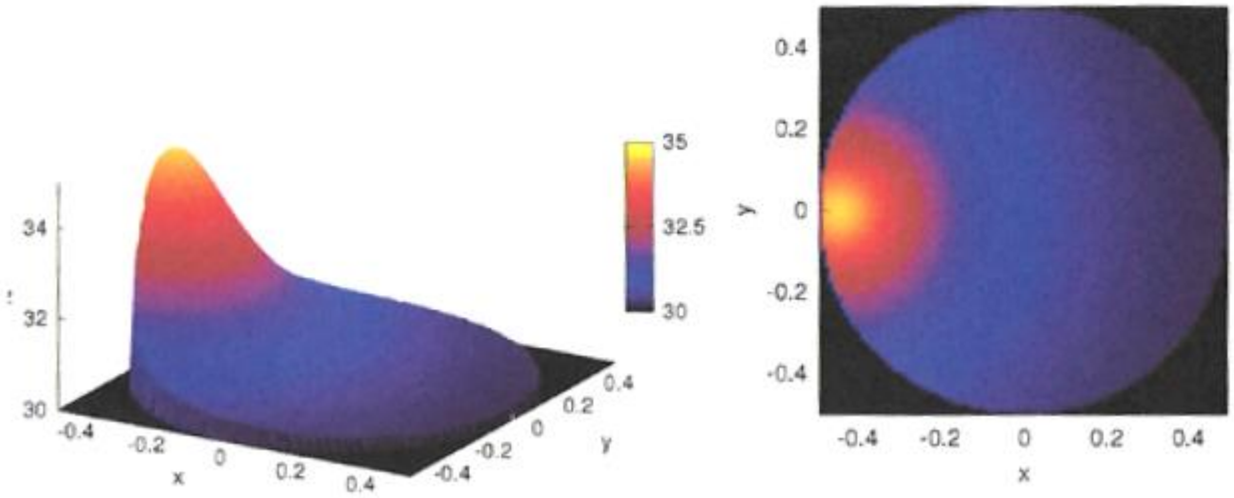
- Kuramoto, inspired by coupled oscillators in physics
- Axelrod, inspired by dissemination of culture in sociology
- Ising, inspired by solid state physics
- Fiber bundle, inspired by texture tensile tests
- Bass diffusion, inspired by how innovative products spread
- Contrarians, inspired by sociology to make a swarm heterogeneous

Use case: Swarm adaptive aggregation

Make a control system that aggregates the robot swarm at a certain spot determined by sensor input but stays flexible to changes in the dynamic environment.

Could be warmest, brightest or most radioactive spot in search area.

Use case: Swarm adaptive aggregation



Possibly multimodal, noisy and/or systematic plateaus

Image: Figure 7.1, Hamann, 2018

Use case: Swarm adaptive aggregation

Alternative modelling approaches:

1. Ad-hoc random search, baseline benchmark
   – Must keep track of position to be effective
2. Gradient ascent and evolutionary optimization
   – Communication improve performance
   – Problems with multimodality and plateaus
3. Positive feedback, inspiration by natural swarm systems
   – The BEECLUST algorithm [Schmickl & Hamann, 2011] inspired by honeybees (bark beetles, ants and cockroaches)

BEECLUST algorithm

Video: Youtube

BEECLUST algorithm

Behavioural model:

- Step 1: move straightforward
- Step 2: obstacle or robots around?
  a) In case of an obstacle: turn away, return to step 1
  b) In case of a robot: stop, measure sensor, wait for some time dependent on sensor reading, u-turn, and return to step 1

Positive feedback since robots are more inclined to stop in high density areas correlated with high sensor readings.

BEECLUST algorithm

Modelling objectives:

1. Capture the ineffective single robot vs the effective robot swarm
2. Explicitly model parameters of the robot control algorithm
3. Spatial modelling
4. Validate model against experiment

BEECLUST algorithm

Microscopic model: The Langevin equation

$$\underbrace{\dot{\boldsymbol{R}}(t) = \alpha \nabla P\big(\boldsymbol{R}(t)\big)}_{\text{Non-stochastic drift term}} + \underbrace{B\boldsymbol{F}(t)}_{\text{Stochastic random term}}$$

where $\boldsymbol{R}(t)$ is position of an agent in 2D space

$\nabla P\big(\boldsymbol{R}(t)\big)$ is the gradient of temperature field P

$\alpha \in [0,1]$ is intensity of drift

$\boldsymbol{F}(t)$ is random perturbation and B is a scalar

BEECLUST algorithm

Microscopic model:

$\alpha = \{0.01, 0.025, 0.05, 0.1\}$

and Brownian $\boldsymbol{F}$

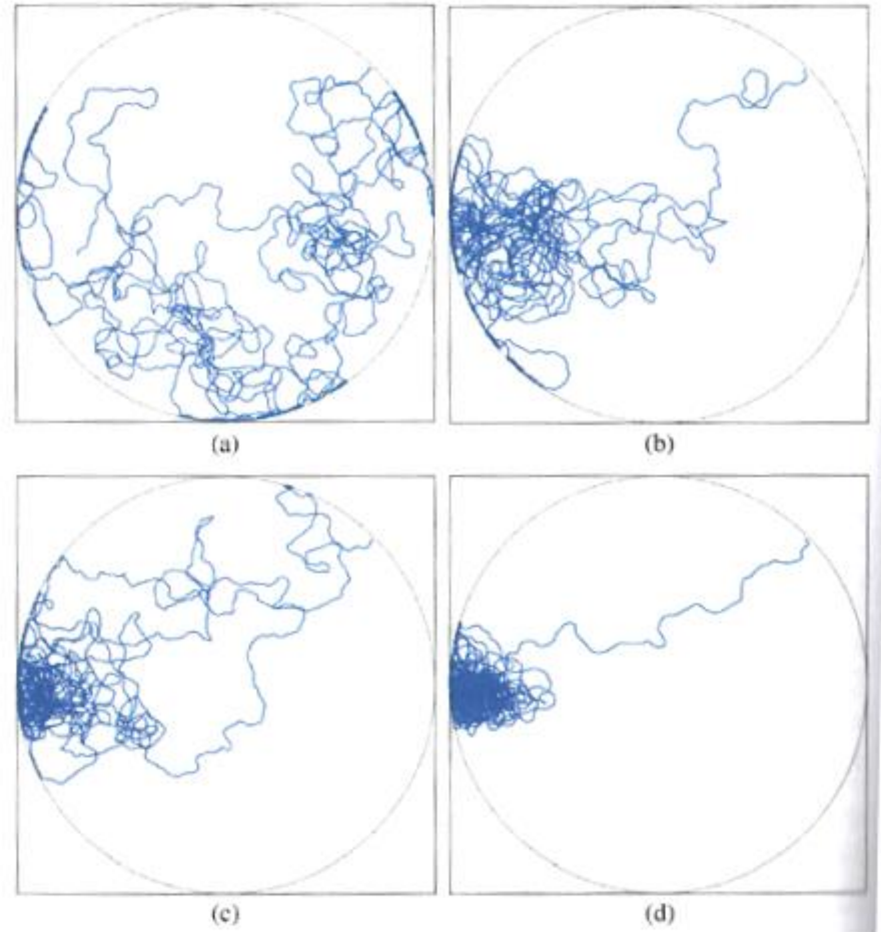


Image: Figure 7.3, Hamann, 2018

BEECLUST algorithm

Microscopic model: The Langevin equation

$$\dot{\boldsymbol{R}}(t) = \alpha \nabla P\big(\boldsymbol{R}(t)\big) + B\boldsymbol{F}(t) = \underbrace{B\boldsymbol{F}(t)}_{\text{Only stochastic random term}}$$

where $\alpha = 0$ i.e. no drift and pure random walk

$\boldsymbol{F}(t)$ is random perturbation and B is a scalar

BEECLUST algorithm

Microscopic model: The waiting time

$$w(\boldsymbol{R}) = \frac{w_{\boldsymbol{max}} P^{\boldsymbol{2}}(\boldsymbol{R})}{P^{\boldsymbol{2}}(\boldsymbol{R}) + c}$$

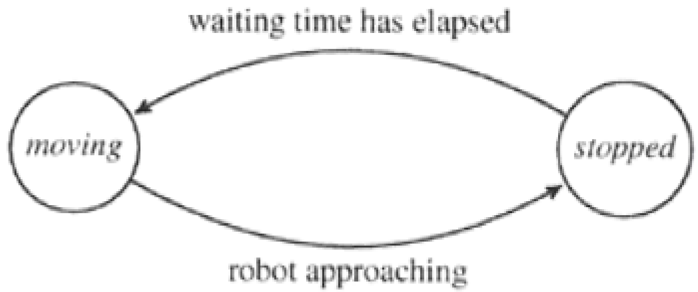
where $P(\boldsymbol{R})$ is temperature at position $\boldsymbol{R}$

$w_{\boldsymbol{max}}$ is maximal waiting time

c is a scaling constant

BEECLUST algorithm

Microscopic model: Finite state machine



Where

moving is: $\dot{\boldsymbol{R}}_m(t) = B\boldsymbol{F}(t)$

stopped is: $\dot{\boldsymbol{R}}_s(t) = 0$

Image: Figure 7.5, Hamann, 2018

BEECLUST algorithm

Microscopic model: Finite state machine

- Agents modelled by simple Langevin equations of random walk.
- Not a completely analytical model since we have to adminster the state transitions, positions, waiting time, check distances to neighbour robots, etc.

BEECLUST algorithm

Macroscopic model: Fokker-Planck equation (from Langevin)

moving: $\dot{\boldsymbol{R}}_m(t) = B\boldsymbol{F}(t) \Longrightarrow \frac{\partial \rho_m(\boldsymbol{r},t)}{\partial t} = B^2\nabla^2\rho_m(\boldsymbol{r},t)$

stopped: $\dot{\boldsymbol{R}}_s(t) = 0 \qquad \Longrightarrow \frac{\partial \rho_s(\boldsymbol{r},t)}{\partial t} = 0$

Where $\rho(\boldsymbol{r},t)$ is density of moving or stopped agents

BEEECLUST algorithm

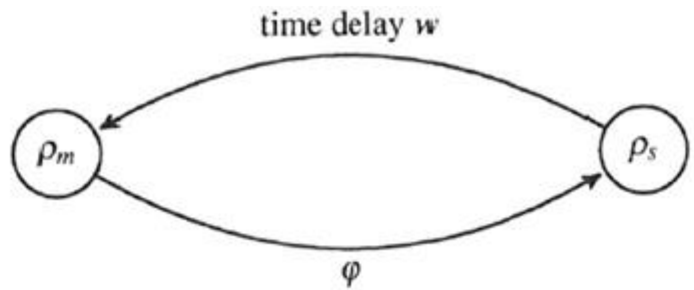
Macroscopic model: Fokker-Planck equation (from Langevin)

We need a way of connecting the state transitions since the Fokker-Planck equation is continous.

We define a stopping rate φ where $\rho_m(\boldsymbol{r}, t)\varphi$ gives us the correct density flow into stopped robots (this is a rather strong assumption).

BEECLUST algorithm

Macroscopic model: Fokker-Planck equation (from Langevin)



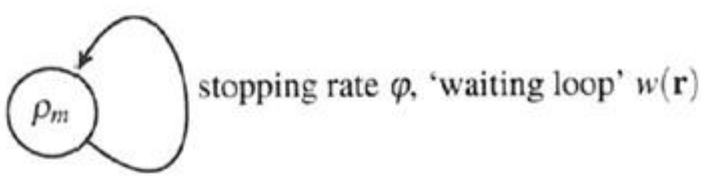


Image: Figure 7.6 and 7.7, Hamann, 2018

BEECLUST algorithm

Macroscopic model: Fokker-Planck equation (from Langevin)

moving:

$$\frac{\partial \rho_m(\boldsymbol{r},t)}{\partial t} = \underbrace{B^2 \nabla^2 \rho_m(\boldsymbol{r},t)}_{\text{Diffusion term}} - \underbrace{\rho_m(\boldsymbol{r},t)\varphi}_{\text{Flow into stopped}} + \underbrace{\rho_m\big(\boldsymbol{r},t-w(\boldsymbol{r})\big)}_{\text{Flow into move}}$$

BEECLUST algorithm

Macroscopic model: Fokker-Planck equation (from Langevin)

Stopped (not necessary to model explicit):

$$\frac{\partial \rho_s(\boldsymbol{r},t)}{\partial t} = \underbrace{\rho_m(\boldsymbol{r},t)\varphi}_{\text{Flow into stopped}} - \underbrace{\rho_m\big(\boldsymbol{r},t-w(\boldsymbol{r})\big)}_{\text{Flow into move}}$$

BEECLUST algorithm

Experimental validation:
[Schmickl et al., 2009;
Kernbach et al., 2009]

15 Jasmine robots measuring ambient light on a 150×100 cm^2 rectangular area.

Image: swarmrobots.org

BEECLUST algorithm

Experimental validation:

[Schmickl et al., 2009; Kernbach et al., 2009]

Two lamps at both ends are operated in

$$\text{mode} = \{\text{off, dimmed or bright}\}$$

and swarm was allowed to converge to steady-state after uniform initial distribution of moving robots.

B was fitted to data.

BEECLUST algorithm

Experimental validation:

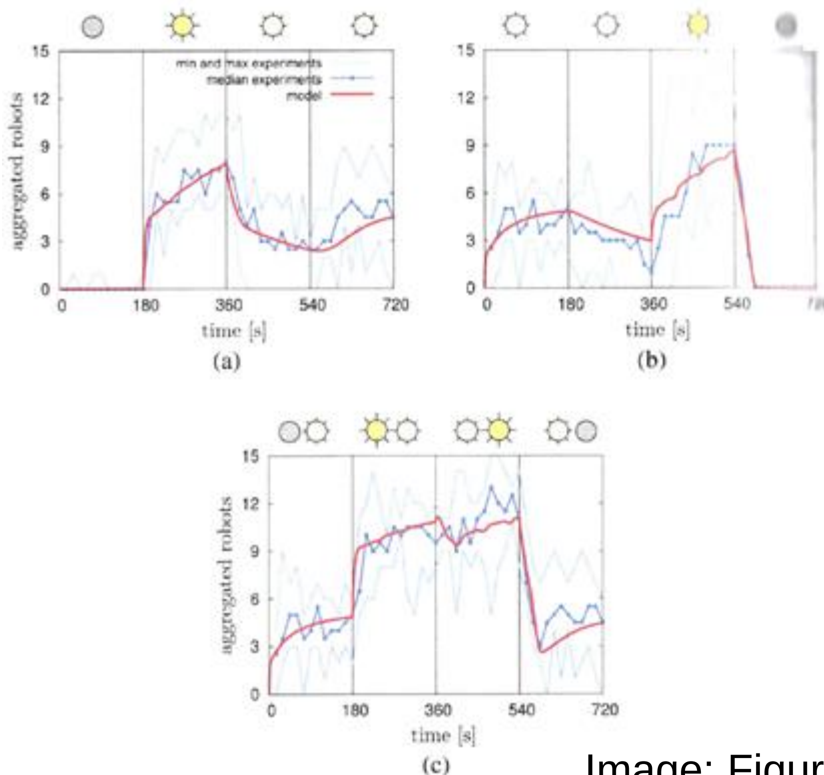


Image: Figure 7.8, Hamann, 2018

BEECLUST algorithm

Video: Youtube

Summary lecture 5 – Swarm robotics 2*

- Swarm collective decision-making
  - Terminology and notation
  - The decision-making process
  - Different models (voting, urn, Hegselmann-Kraus, etc)

- Swarm case study: Adaptive aggregation, BEECLUST

*Hamann, 2018: chapter 6 and 7