# A Self-stabilizing Marching Algorithm for a Group of Oblivious Robots


Y. Asahiro[1]

Joint work with S. Fujita[2], I. Suzuki[3], M. Yamashita[4]

[1]: Kyushu Sangyo Univ., Japan
[2]: Hiroshima Univ., Japan
[3]: Univ. of Wisconsin, Milwaukee, USA
[4]: Kyushu Univ., Japan



PDCAT 2009, Dec.8, 2009

# Outline

# Outline

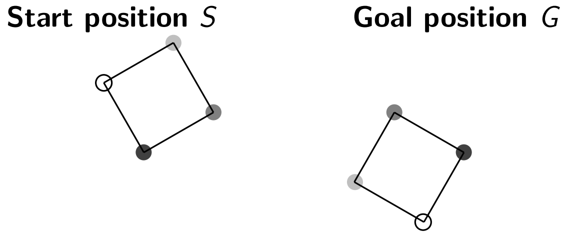
# Outline

1. Problem
   - Marching
   - Problem Setup
   - Related Work: A Time-optimal Motion
2. The Goal of Our Research
3. Robot Model
4. Self-stabilizing Algorithm
   - Self-stability
   - Simple Algorithm $G$ (Greedy)
   - Proposed Algorithm $G+$
   - Proof Idea for Self-stability
   - Simulation Results
5. More robots
6. Summary and Further Research

# Marching by mobile robots

**Marching**: ($\sim$ transportation; many researches...)

- Moving from a start position $S$ to a goal position $G$
- Try to maintain a formation (e.g., line, triangle, etc)



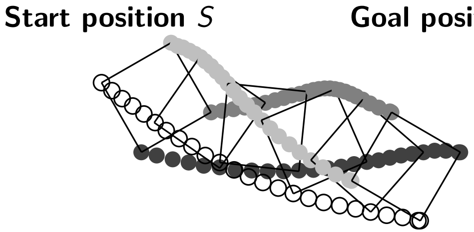

**Focus on two robots case (in this talk):**

**Keep formation = Keep distance between two robots**

# Marching by mobile robots

**Marching**: ($\sim$ transportation; many researches...)

- Moving from a start position $S$ to a goal position $G$
- Try to maintain a formation (e.g., line, triangle, etc)

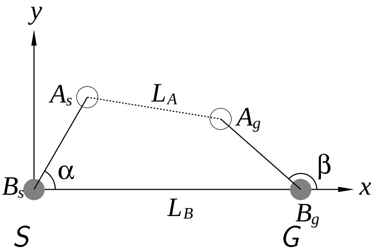



**Focus on two robots case (in this talk):**

**Keep formation = Keep distance between two robots**

# Outline

# Problem Setup(Not so important in this talk)




- **Explain a snapshot by**
  - **Robot $B$'s position $(x, y)$ and**
  - **angle $\theta$ between $x$-axis and segment $\overline{AB}$.**
- **An instance is represented by $L_B$, $\alpha$, and $\beta$**
  **$\alpha$ and $\beta$: angles at $S$ and $G$, resp.**
- **The desirable distance between the robots is set to 1.**
- **x- and y- axes are only for explanation**
  **(Robots do not have any common coordinate system)**

# Outline

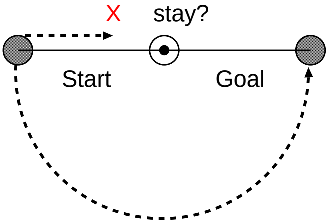
# Related Work: A Time-optimal Motion

**The assumption:**

- **2 robots**
- **Correct formation (distance) is always kept**
- **Both robots always move at the maximum speed** $V$

**by Chen, Suzuki, Yamashita (1997).**

**Example Instance** $I$ **(**$L_B = 2, \alpha = 0^\circ$**, and** $\beta = 180^\circ$**):**



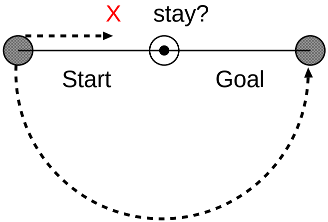
# Related Work: A Time-optimal Motion

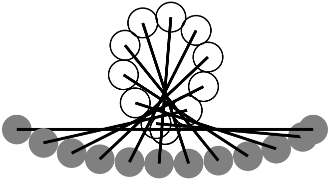
**The assumption:**

- **2 robots**
- **Correct formation (distance) is always kept**
- **Both robots always move at the maximum speed** $V$

**by Chen, Suzuki, Yamashita (1997).**

**Example Instance** $I$ **(**$L_B = 2, \alpha = 0^\circ$**, and** $\beta = 180^\circ$**):**



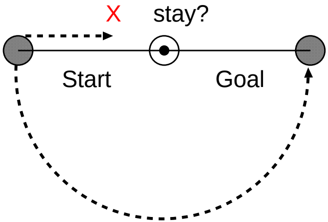
# Related Work: A Time-optimal Motion

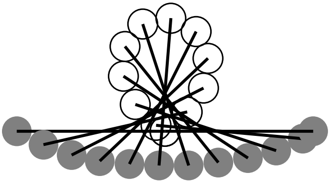
**The assumption:**

- **2 robots**
- **Correct formation (distance) is always kept**
- **Both robots always move at the maximum speed** $V$

**by Chen, Suzuki, Yamashita (1997).**

**Example Instance** $I$ **(**$L_B = 2, \alpha = 0^\circ$**, and** $\beta = 180^\circ$**):**



**With** $V = 0.01$**, finish time of the time-optimal motion is 208 (Motion in left Fig. takes about 314.)**

# A Time-optimal Motion

- **Complicated and basically centralized (whole trajectory is given before motion starts)**
- **Distributed(?): If control error occurs, i.e., the robots sometimes deviate from the given paths, re-calculation is necessary, but it takes a long time:**

**Theorem (CSY97)**

*The time-optimal motion satisfies the following formulas.*

$$\dot{x} = \frac{\dot{\theta}}{2}\sin\theta + cV\cos\theta\sin(\theta+\delta)$$
$$\dot{y} = -\frac{\dot{\theta}}{2}\cos\theta + cV\sin\theta\sin(\theta+\delta)$$
$$\dot{\theta} = 2V\sqrt{1 - c^2\sin^2(\theta+\delta)}$$

*$c$ and $\delta$ in the above meet conditions in the next page.*

## Conditions on $c$ and $\delta$

$$\frac{1}{2}(\cos\alpha - \cos\beta)$$
$$+\frac{\cos\delta}{2c}\left(\sqrt{1-c^2\sin^2(\alpha+\delta)} - \sqrt{1-c^2\sin^2(\beta+\delta)}\right)$$
$$+\frac{\sin\delta}{2c}(F(\beta+\delta,c) - F(\alpha+\delta,c) - E(\beta+\delta,c) + E(\alpha+\delta,c)) = L_B$$

**and**

$$\frac{1}{2}(\sin\alpha - \sin\beta)$$
$$-\frac{\sin\delta}{2c}\left(\sqrt{1-c^2\sin^2(\alpha+\delta)} - \sqrt{1-c^2\sin^2(\beta+\delta)}\right)$$
$$+\frac{\cos\delta}{2c}(F(\beta+\delta,c) - F(\alpha+\delta,c) - E(\beta+\delta,c) + E(\alpha+\delta,c)) = 0,$$

**where**

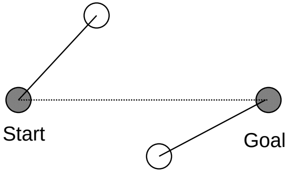
$$F(\phi,k) = \int_0^\phi \frac{d\theta}{\sqrt{1-k^2\sin^2\theta}} \qquad E(\phi,k) = \int_0^\phi \sqrt{1-k^2\sin^2\theta}.$$

# Note on the method

- **Does the assumption Always moving at max speed derive better motions than Allowing reduced speed?**

  **Answer: We do not know! The assumption makes it easier to treat such complicated equations...**

- $c$ **and** $\delta$ **are obtained numerically for simulation. We do not know any easy way to calculate them...**
- **The method can not handle the case** $\alpha < 180°$ **and** $180° < \beta$**, i.e,** $S$ **and** $G$ **locate in different sides of** $x$**-axis.**



# Outline

# The Goal of Our Research

**Designing a motion planning algorithm, s.t,**

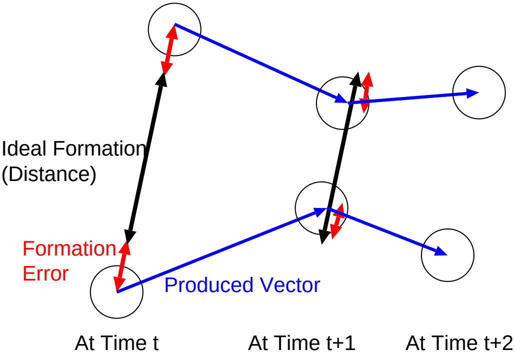
- **Distributed and Simple**
  **Each robot individually determines its motion easily (need not so much time).**
- **Oblivious**
  **Each robot determines its motion only based on current state and goal state ignoring past motions**
- **Self-stabilizing**
  **Even if there exists a finite number of control errors, the robots reach the goal position**
- **Reasonably good performance compared to the time optimal one**
  - **Small finish time**
  - **Smooth motion: Small formation error**

# Outline

# Robot Model

1. **Omni-directional**: freely move
2. **Identity** (ID): but no leader, identical algorithm
3. **Oblivious**: ignores past motions.
4. **Full Visibility** with **Local Coordinate**: know correctly current and goal positions of other robots (distinguishable), but only knows their relative positions based on a local coordinate system (not common among robots)
5. Repeats a cycle processed in a **discrete time** step
   1. **Look** other robots and goal positions
   2. **Compute** a vector based on current and goal positions
   3. **Move** according to the produced vector
6. **No communication**
7. **Synchronous**: all the robots move at the same time.
8. (Ignore collision between robots; unrealistic but simple)

# Robot Motion and Formation Error




- **Formation error: Two robots case:** $(\ell - D)/2$
  - $\ell$**: Ideal distance between robots (Correct formation)**
  - $D$**: Current distance between robots**

# Outline

# Outline

# Self-stability (in our research)

1. **From any initial configuration (robot positions),**
2. **the robots finally reach the goal positions,**
3. **even if there exists a finite number of control errors.**

**$\longrightarrow$ If infinite number of errors occur, it seems impossible to arrive at any target position.**

**$\longrightarrow$ A state right after all errors have occurred can be considered as the initial configuration, and after that no errors occur.**

# Outline

# Simple Algorithm $G$ (Greedy)

**Robots:** $R_1, R_2, \ldots, R_n$
**Current positions of the robots $R_i$'s:** $S = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$
**Goal positions of the robots $R_i$'s:** $G = \{\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_n\}$
**The max speed of the robots:** $V$

## Algorithm $G$ for $R_i$

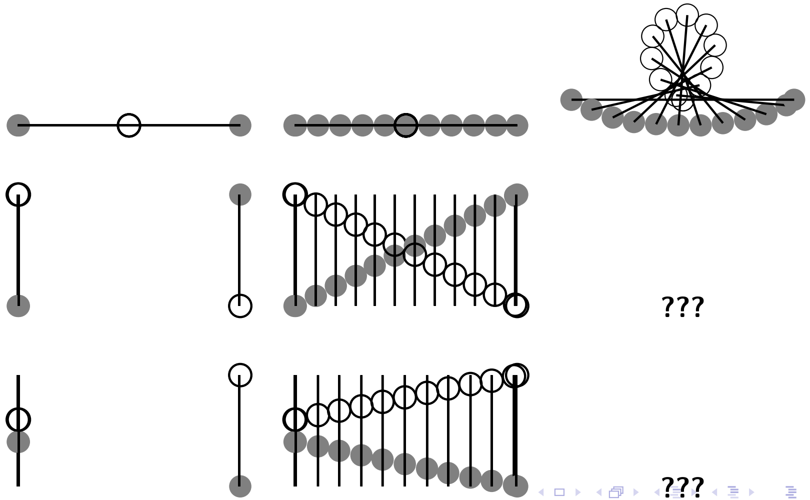
**Produce a vector** $(\mathbf{g}_i - \mathbf{s}_i) \cdot \frac{V}{||\mathbf{g}_i - \mathbf{s}_i||}$**, i.e.,**
**Move straight towards the goal at the max speed.**

**This is a self-stabilizing oblivious algorithm! (No surprise)**

- **Theoretically minimum finish time (lower bound)**
- **Bad formation during motion**

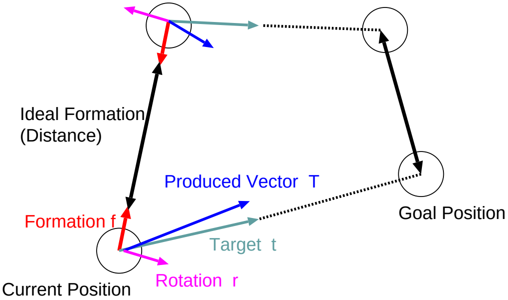
# $G$'s motion

| Instance | $G$ | Time-optimal |
|---|---|---|



# Outline

# Algorithm $G+$ (Fig)




**Produce a vector $T_i$ by summing up three vectors**

- **Target vector: $\mathbf{t}_i$**
- **Rotation vector: $\mathbf{r}_i$**
- **Formation vector: $\mathbf{f}_i$**

**with scaling.**

# Algorithm $G+$ for robot $R_i$

Step 0: **Let $L_i = ||\mathbf{g}_i - \mathbf{s}_i||$ and $L_{max} = \max_{1 \le i \le n}\{L_i\}$.**

Step 1: **If $L_{max} \le V$, move to $\mathbf{g}_i$ and halt. Otherwise, go to Step 2.**

Step 2: **Set target vector $\mathbf{t}_i = t(\mathbf{g}_i - \mathbf{s}_i)$.**

Step 3: **If $\mathbf{s}_i \neq \mathbf{o}_s$, then set rotation vector $\mathbf{r}_i$ to have magnitude $u||\mathbf{s}_i - \mathbf{o}_s||\tan(\min\{|\gamma|, \pi/4\})$, and direction $\alpha_i - \pi/2$ (or $\alpha_i + \pi/2$) Here, $\alpha_i$ is the direction of $\mathbf{s}_i - \mathbf{o}_s$.**

Step 4: **Set formation vector $\mathbf{f}_i = s(\mathbf{s'}_i - \mathbf{s}_i)$.**

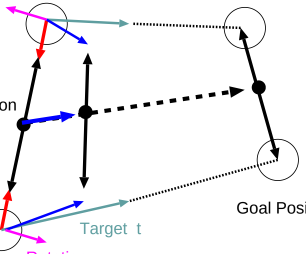
Step 5: **Set $\mathbf{T}_i = \mathbf{t}_i + \mathbf{r}_i + \mathbf{h}_i$.**

Step 6: **Compute $\mathbf{T}_j$, following Steps 2–5 for all robots $R_j$, $j \neq i$. Let $T_{max} = \max_{1 \le i \le n}\{||\mathbf{T}_i||\}$, and $K = \min\{\frac{V}{T_{max}}, \frac{1}{3}\}$.**

Step 7: **Output $K\mathbf{T}_i$ as a produced vector.**

# Outline

# Proof Idea for Self-stability (1/2)

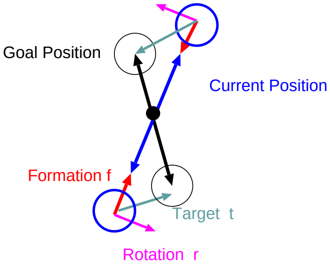



**The center of the formation moves towards the goal:**

- $\sum_i \mathbf{s}_i/n$, $\sum_i \mathbf{g}_i/n$ : **center of current and goal positions**
- $\mathbf{s}_i + \mathbf{T}_i$ **is the position of robot** $R_i$ **at the next time step.**
- $\sum_i \mathbf{f}_i = \sum_i \mathbf{r}_i = \mathbf{0}$
- $\sum_i(\mathbf{s}_i + \mathbf{T}_i)/n = \sum_i(\mathbf{s}_i + \mathbf{t}_i + \mathbf{r}_i + \mathbf{f}_i)/n = \sum_i(\mathbf{s}_i + \mathbf{t}_i)/n$
  $= \sum_i(\mathbf{s}_i + K(\mathbf{g}_i - \mathbf{s}_i))/n = (1-K)\sum_i \mathbf{s}_i/n + K\sum_i \mathbf{g}_i/n$,
  **where** $K$ **is a scaling factor** $< 1$.

# Proof Idea for Self-stability (2/2)

**Finally, center of current form. = center of goal form.**

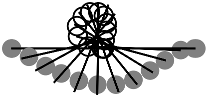
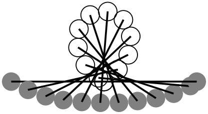
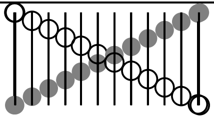
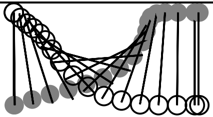
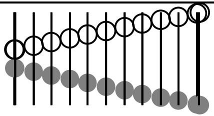
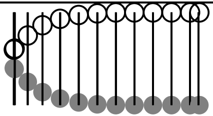



**After that**

- **Rot. vector $r_i$'s adjusts the orientation of the form.**
- **Form. vector $f_i$'s adjusts the distance between the robots.**
- **Target vector $t_i$'s have both effect of $r_i$'s and $f_i$'s.** □

# Outline

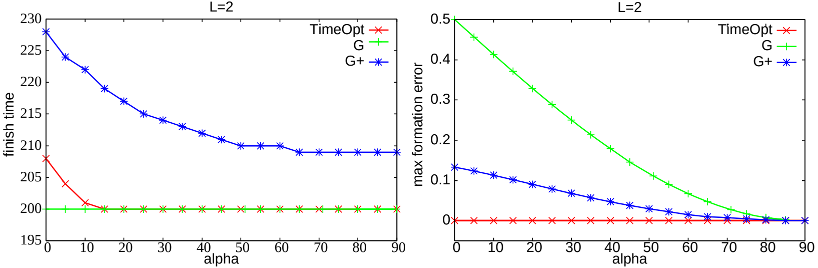
# $G+$'s motion

| $G$ | $G+$ | **Time-optimal** |
|---|---|---|
|  $F = 200,\ E = 0.50$ |  $F = 228,\ E = 0.13$ |  $F = 208,\ E = 0$ |
|  $F = 224,\ E = 0.50$ |  $F = 314,\ E = 0.13$ | **???** |
|  $F = 204,\ \sum E = 81$ |  $F = 221,\ \sum E = 28$ | **???** |

$F$: **finish time,** $E$: **max form. err.** $\sum E$ = **total form. err.**

# Results for a Set of Instances

**Instances:** $L_B = 2$, $0^\circ \leq \alpha \leq 180^\circ$, $\beta = 180^\circ - \alpha$

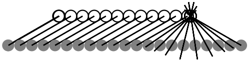
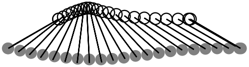
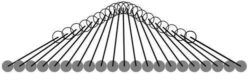
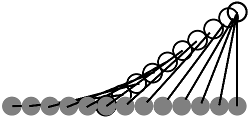
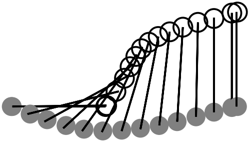
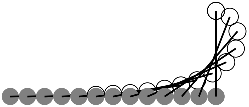



- **Finish time of $G+$ is 5-10% larger than that of the time-optimal one or theoretical lower bound ($G$).**
- **Max formation error is very smaller than that of $G$.**

$\Rightarrow$ **$G+$ has three good properties at the same time:**

- **Fast (Small finish time)**
- **Smooth (Maintain formation)**
- **Self-stabilizing**

# More Figs with Two Robots

| $G$ | $G+$ | **Time-optimal** |
|---|---|---|
|  |  |  |
| $F = 400,\ E = 0.25$ | $F = 412,\ E = 0.07$ | $F = 400,\ E = 0$ |
|  |  |  |
| $F = 240,\ E = 0.10$ | $F = 256,\ E = 0.04$ | $F = 240,\ E = 0$ |

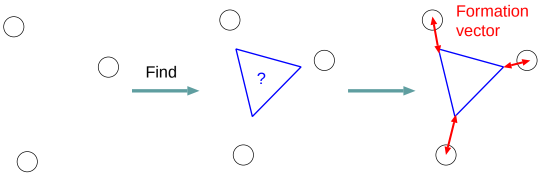
$F$: **finish time,** $E$: **max form. err.**

# Outline

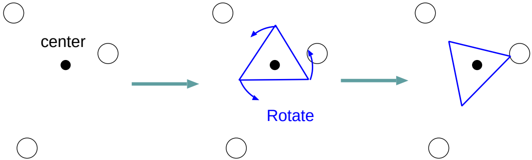
# More robots

**Algorithm $G+$ is self-stabilizing for more than two robots:**

- **In proof,**
  - **no assumption on the number of robots.**
  - **no assumption on the formation at the beginning.**
- **But, what is the ideal current formation?**
  - **In two robots case, the maintained formation is just the distance. $\Rightarrow$ easy**



# Place Ideal Formation




1. **Translate the goal form. $G$ to $G'$ s.t. its center coincide with center of current positions of robots $s_i$'s.**
2. **Rotate $G'$ so as to minimize $\sum ||\mathbf{s}_i - \mathbf{g}'_i||^2$**

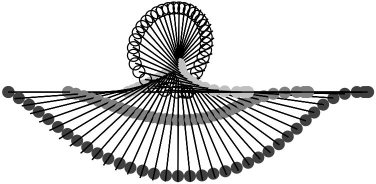
**How to minimize? We can show that**

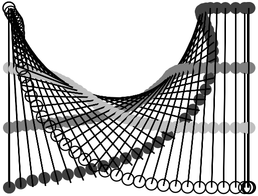
- **the optimal orientation of the formation $=$ argument of $\sum \bar{\mathbf{s}}_i \mathbf{g}'_i$ (in Gaussian plane) and is uniquely determined (each robot individually obtain it).**

  **($\sum \bar{\mathbf{s}}_i \mathbf{g}'_i$ is obtained based on robots' current positions and $G'$)**

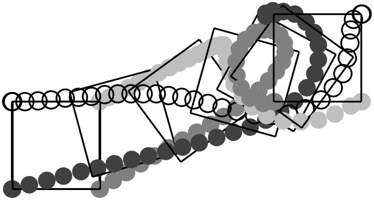
# Example Motions (Marching)

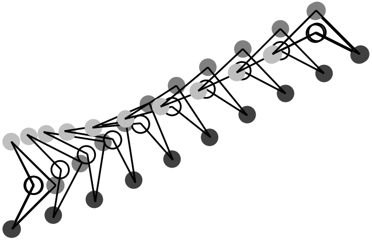
**Line formation with 4 robots:**



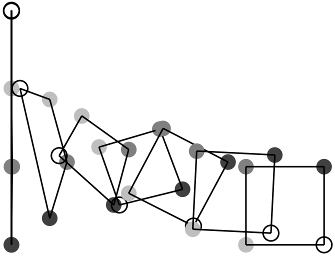
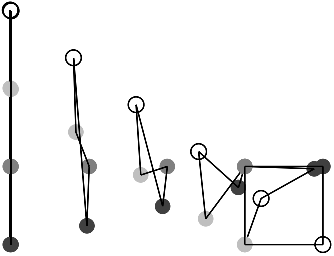
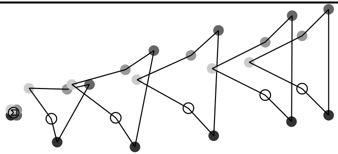
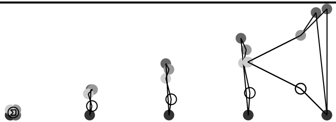


**Square and wedge formations with 4 robots:**





# Example Motions (Morphing?)

| $G+$ | $G$ **(for comparison)** |
|---|---|
|  $F = 658$ |  $F = 584$ |
|  $F = 681$ |  $F = 623$ |

# Outline

# Summary and Further Research (1/3)

**Summary:**

- **Self-stabilizing marching algorithm for a group of oblivious mobile robots.**
  - **Small finish time**
  - **Smooth motion**

**Further Topic:**

- **Self-stability does not help to obtain theoretical guarantee of finish time and max formation error (at the worst case)**
- **Is the method to determine "current ideal formation" by a least square method good enough?**
- **What is a time-optimal motion for more than two robots (and complicated formations)?**

# Further Topic (2/3)

- Comparison with other approaches, e.g.,
  - Leader-follower by Gervasi and Prencipe (2003),
  - Potential function by Shuneider, Wildermuth and Wolf (2000),
  - Practical robots (we do not have...)
- Synchronicity
  The assumption on synchronization in our model is necessary to prove the self-stability; $\sum_i \mathbf{r}_i, \sum_i \mathbf{f}_i, \sum_i \mathbf{t}_i$ can be estimated because of synchronicity.
  - Semi-synchronous model
    Basically synchronized but only a subset of robots is active in each time step.
    Current proof does not work.
  - Asynchronous model
- Visibility: Can distinguish the other robots? Asymmetric formation...

# Further Topic (3/3)

- **Anonymous robots (do not have IDs)**
  **Usually difficult to make a certain formation (Researches by Defago and Samia (2008), etc.)**

  **In our problem,**

  - **Each robot can not know which position in the goal formation is its own goal position.**
  - **but can choose one of the positions as its goal.**

# Further Topic (3/3)

- **Anonymous robots (do not have IDs)**
  **Usually difficult to make a certain formation (Researches by Defago and Samia (2008), etc.)**

  **In our problem,**

  - **Each robot can not know which position in the goal formation is its own goal position.**
  - **but can choose one of the positions as its goal.**

**Thank you very much for your attention!**

# Road to PDCAT (Left to Right)