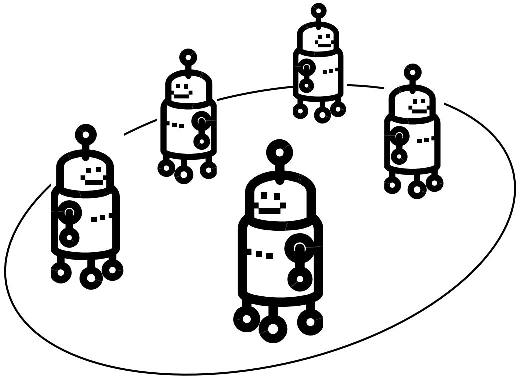
PDCAT 2009
Tutorial 1: Theoretical Aspects of Autonomous Mobile Robots
(2009.12.08 Hiroshima, JAPAN)

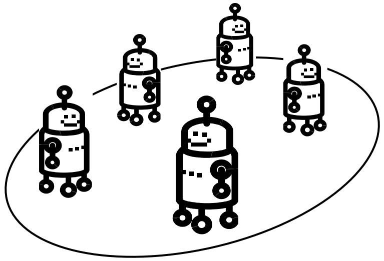
# Rendezvous on Faulty Autonomous Robots

Yoshiaki Katayama
Nagoya Institute of Technology, JAPAN



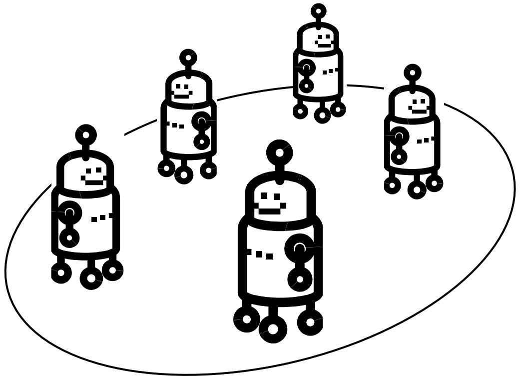
# Outline

- What is Autonomous Mobile Robot Systems ?
  - Everybody has already known … (by the previous session)
- What is Rendezvous ?
  - Gathering & Convergence Problems
- A brief history of Rendezvous Problems and Faulty Robots
  - Considerable Results on the load of tackling the problem.
- Some Algorithms Achieving the Problems
  - The selected algorithms
    - Easy to understand
    - For grasping the images to achieve the problem
- Conclusion



# What is autonomous mobile robot systems ?

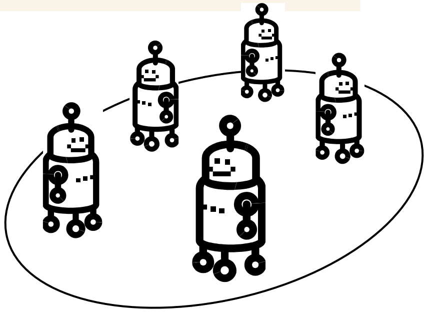
The model considering in this talk



# Preliminary

- The terminology for the robot model is changed.
  - Followings are links that used in previous and this talk.

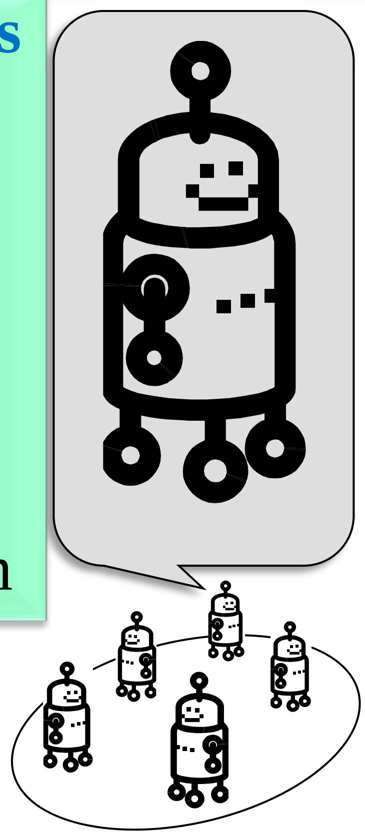
| Previous talk | This talk |
|---|---|
| Full Synchronized | FSYNC (Full SYNChronized) |
| SYm | SSYNC (Semi-SYNChronized) |
| CORDA | ASYNC (ASYNChronized) |
| - | AATOM (ASYNC with ATOMic movement) |



# Autonomous Mobile Robot Systems

The **theoretical model** which consists of two or more robots moving autonomously. We may control them by **deterministic** distributed algorithms equipped on each of them for achieving a goal.

**Common properties**
- anonymous
- oblivious
- no direct comm.
- no volume
- sensor
- an algorithm
- moving mechanism



**Additional Properties**

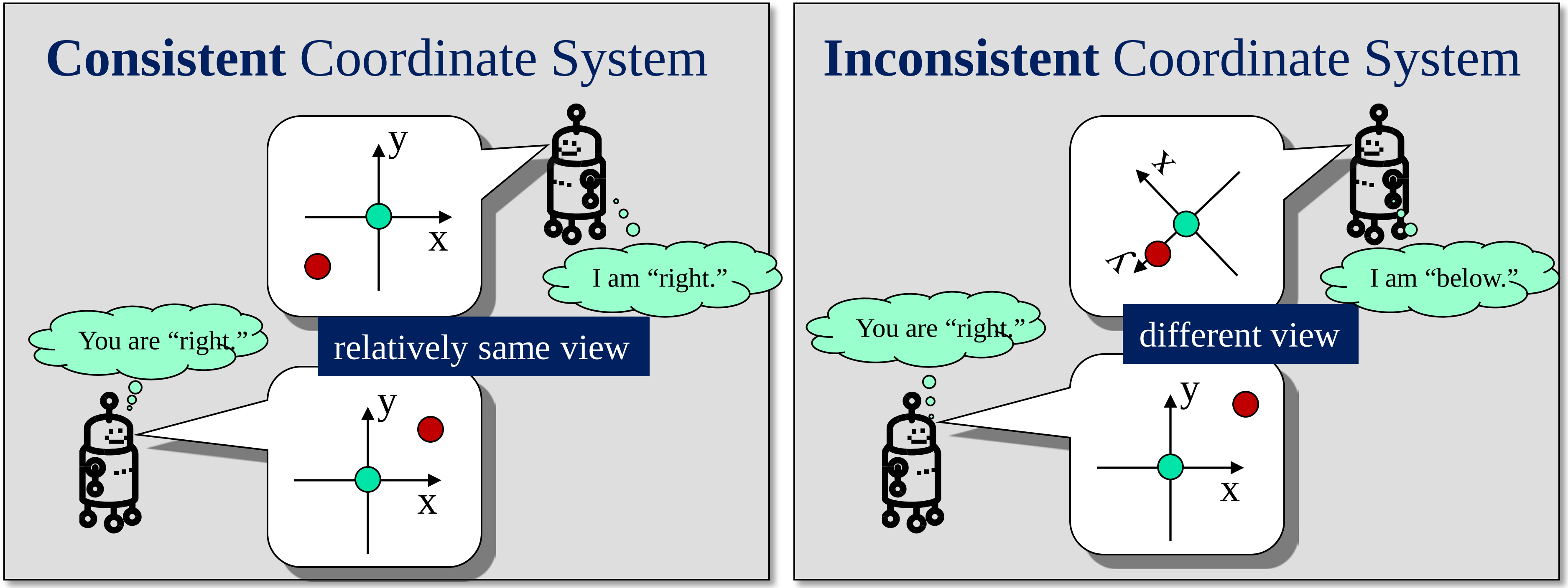
**Execution model**
- full synchronous (FSYNC)
- semi synchronous (SSYNC)
- asynchronous (ASYNC)

**Local coordinate system**
- consistent coordinate system
- inconsistent coordinate system

**Multiplicity detection**
- has
- does not have

# Local Coordinate System

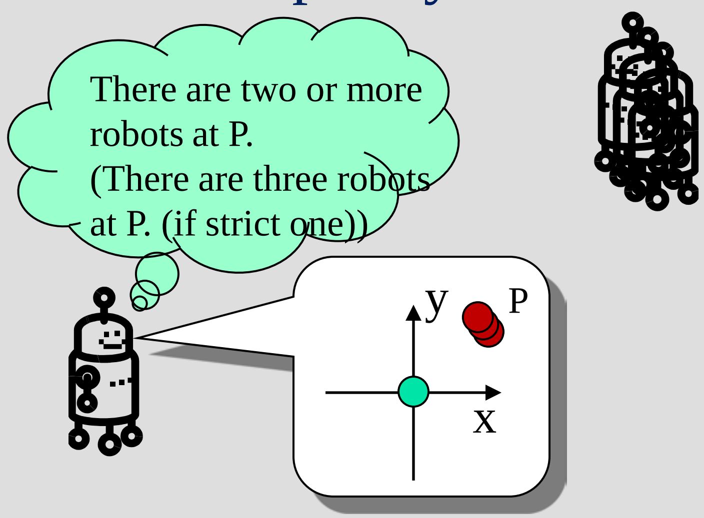
- Each robot has own **local coordinate system**.
- The locations of other robots are mapped on it according to observation by the sensor.



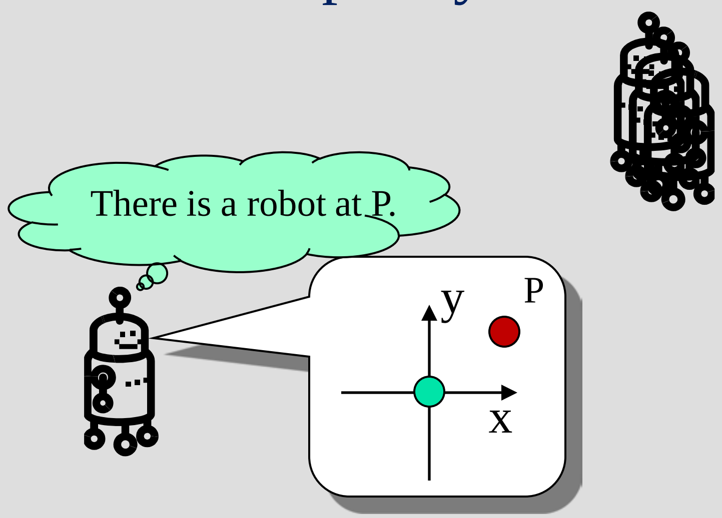
# Multiplicity detection

- **Multiplicity detection** is the ability to detect whether more than one robot is at single point.
  - **Strict Multiplicity detection**
    - If the robot is able to know the number of robots at the point.

**with** Multiplicity Detection

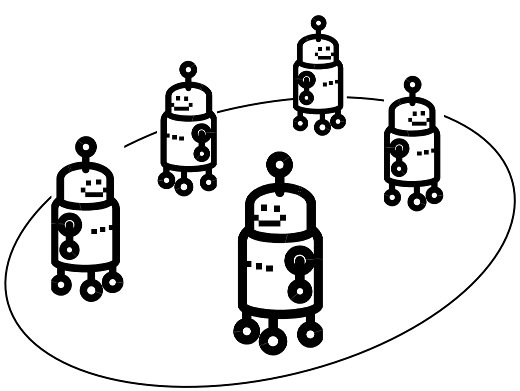



**without** Multiplicity Detection



# What is Rendezvous Problem ?

Definitions of Rendezvous Problem



# Rendezvous Problem

Rendezvous Problem:
Move all the robots toward a non-predefined single point by a deterministic distributed algorithm executing on each robot.

## Why Rendezvous Problem ?

→ This problem is "**(Approximate) Agreement Problem**" !!

one of the most fundamental problem in the distributed system

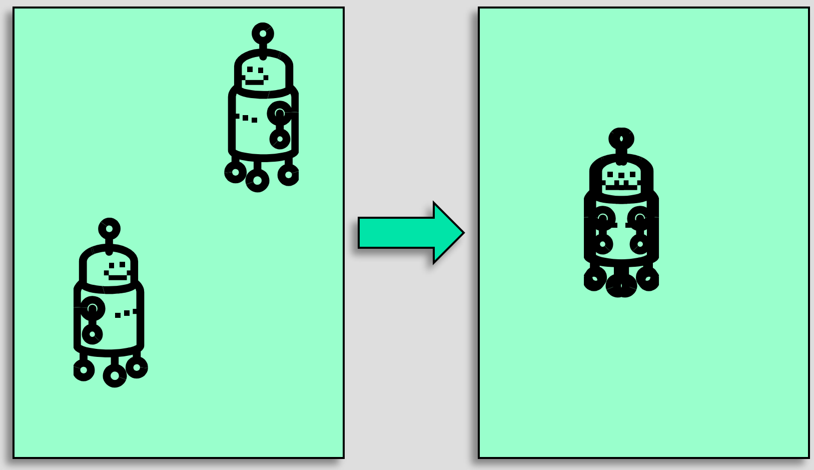
| **Problem** | **We can consider as an agreement problem on** |
|---|---|
| Rendezvous | **the origin** of the global coordinate system |
| Line Formation | **the x or y axis's orientation** of the global coordinate system |
| Circle Formation | **the origin and the unit distance** of the global coordinate system |
| General Formation | **the global coordinate system and its unit distance** |

# The Gathering & Convergence

There are **two types** of the Rendezvous Problem.

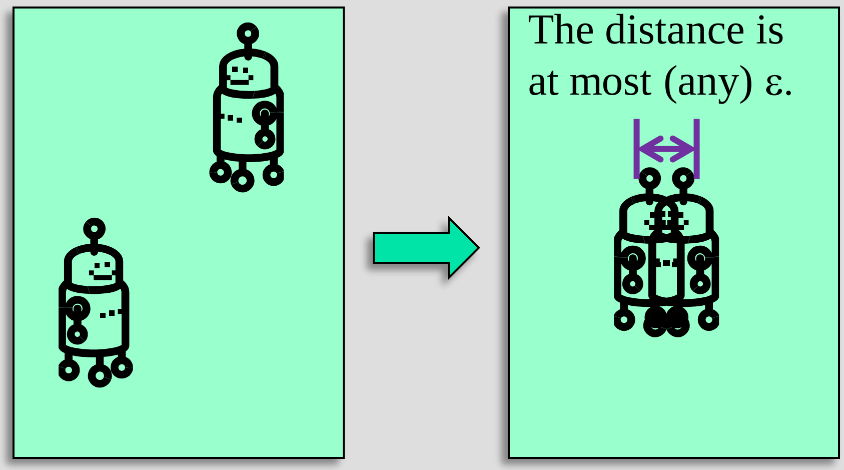
## The gathering problem

All the robots **share** single point within finite time.



## The convergence problem

All the robots **converge** to single point, rather than reach it.



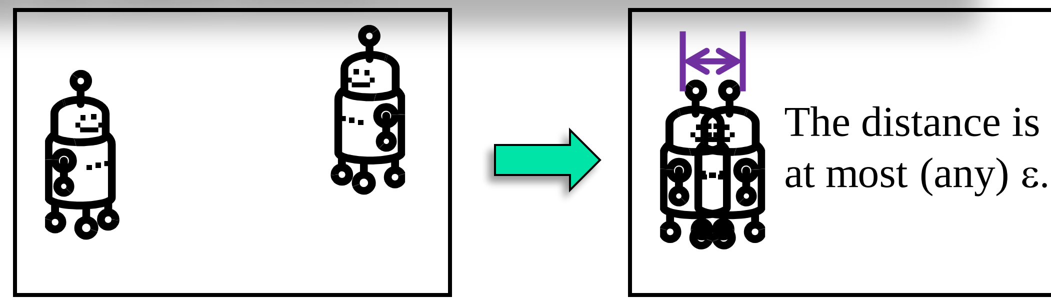

**Are these problems different?**

→ Yes! They are essentially different.

# The Convergence Problem

**Precise definition**:
For every $\varepsilon$>0 there must be a time $t_\varepsilon$ from which all robots are within a distance of at most $\varepsilon$ of each other.

To solve the problem…




**Simple Strategy**
1. Each robot computes **the center of gravity** of all robots,
2. Move toward it.

Simple strategy can achieve the convergence whenever **no faulty robots exist** in the system, even if the weakest robot model is used. (ASYNC, no consistent coordinate system, no multiplicity detection)

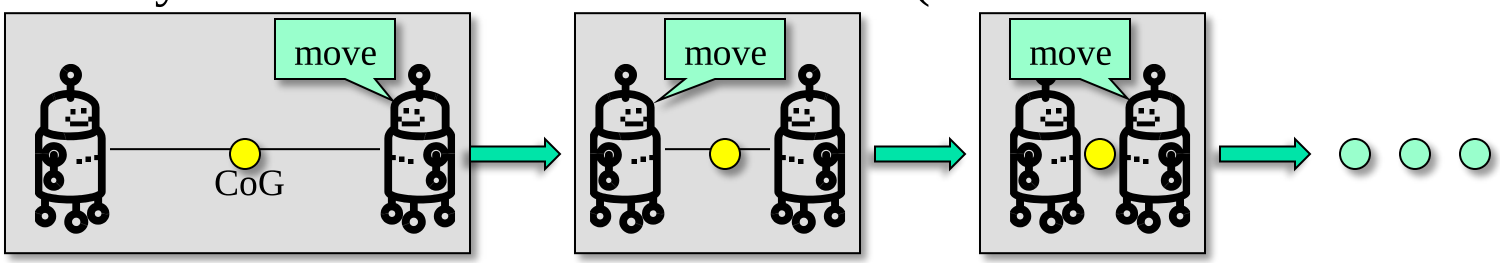
# The Gathering Problem

The gathering problem
All the robots **share** single point accurately within finite time.

**The simple strategy cannot solve the gathering problem!**

Center of gravity (CoG) is variant with respect to robots' movement.
(CoG is invariant when the robot model is FSYNC)

For instance,
If exactly one robot activate at each time… (under SSYNC/ASYNC model)

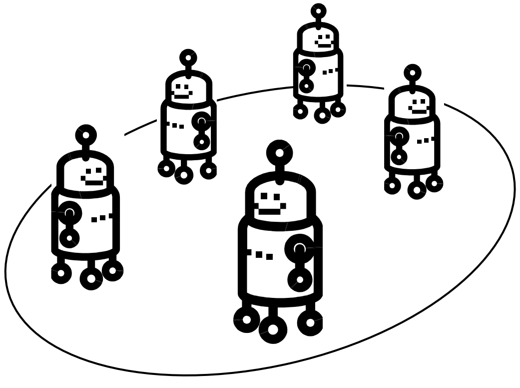



The robots can converge at a point, but never reach a point.

Q. Weber (or Fermat or Torricelli) point is invariant. Can we use it?
A. It is not computable for $n \geq 5$…

# A brief history of
# The Rendezvous Problem

The known results



# The Convergence Problem

The Simple Strategy (if no fault)

1. Each robot computes the center of gravity of all robots,
2. Move toward it.

Using this strategy, the gathering is achievable in the FSYNC model.
But, in the SSYNC or ASYNC model, only the convergence is achievable.

**Theorem [CP05]**
In the ASYNC model, for any $n \geq 2$, in $d$-dimensional Euclidean space, $n$ robots performing an algorithm using the Simple Strategy will converge.

**Theorem [CP05]**
In any execution of the simple strategy (the gravitational algorithm) in the ASYNC model, over every interval of $O(n^2 + nh/\delta)$ time units, the size of the $d$-dimensional convex hull of the robot locations and centers of gravity is halved in each dimension separately.
($\delta$ : smallest distance to move, $h$: the size of convex hull of the robots and its destinations)

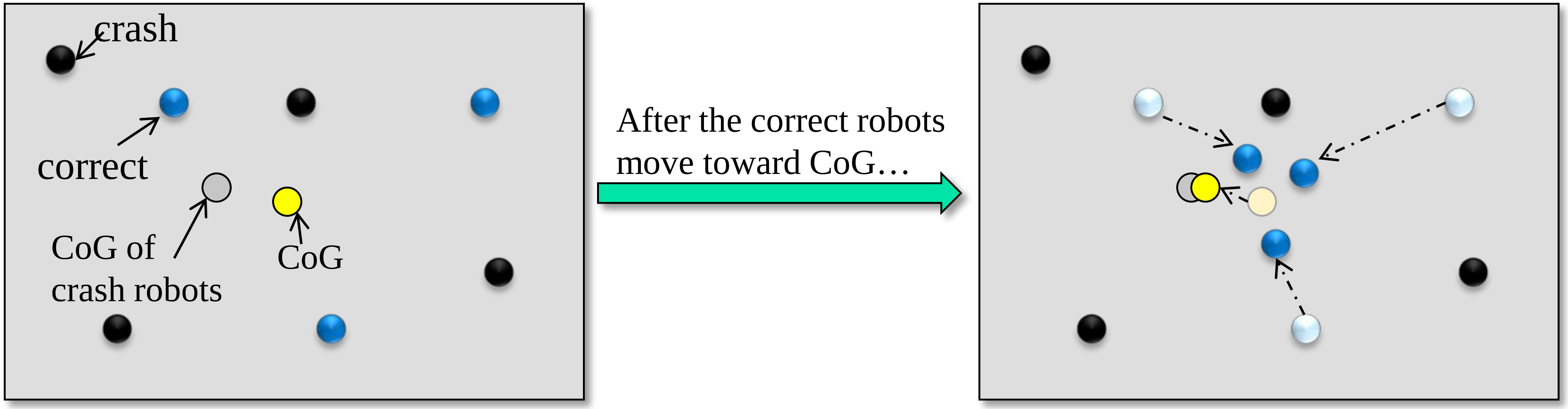
This theorem says "The robots asymptotically approach each other".

[CP05] R.Cohen and D.Peleg: "Convergence properties of the gravitational algorithm in asynchronous robot systems.", *SIAM Journal on Computing*, 34(6), pp.1516-1528, 2005.

# The Convergence with Crash Faults

**Theorem [CP05]**
In the ASYNC model, consider a swarm of $n$ robots that execute an algorithm using the Simple Strategy. If $1 \leq f \leq n-2$ robots crash during the execution, then the remaining $n-f$ robots will converge to the center of gravity of the crashed robots.

Intuitively,
(global) **CoG moves toward CoG of crashed robots asymptotically.**



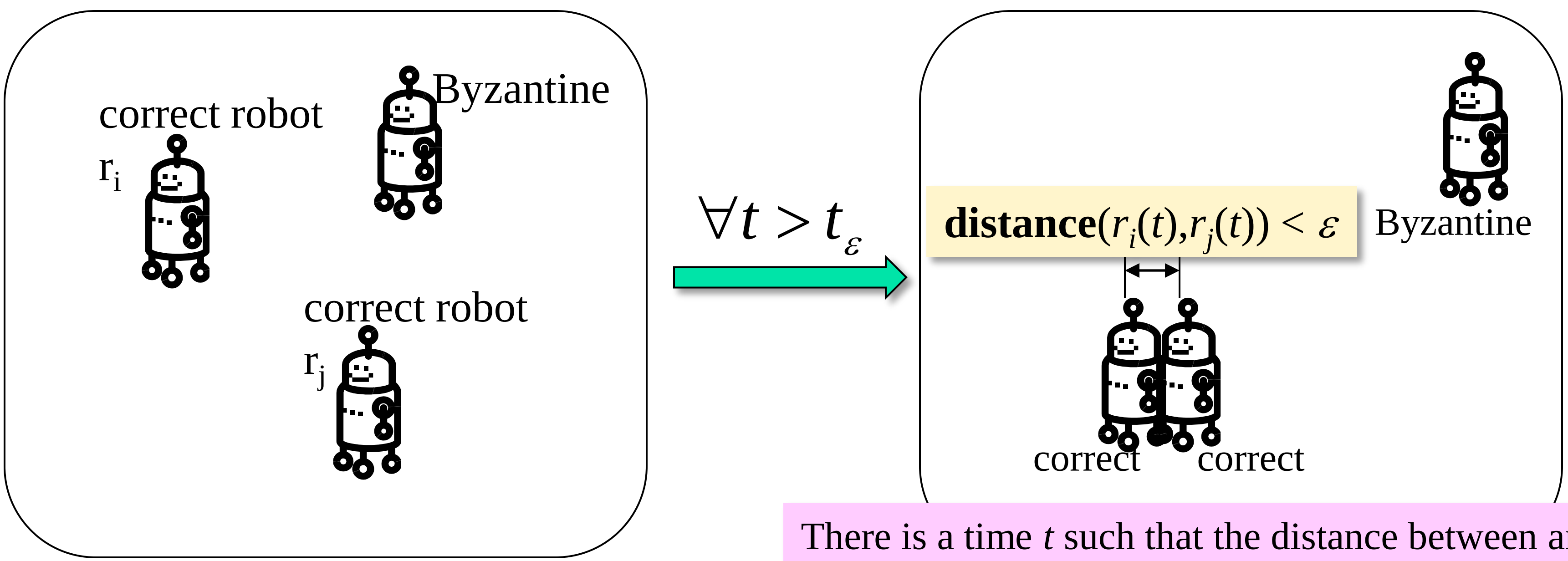

CoG moves toward CoG of crash robots.
→ The Convergence is able to achieved.

# The Convergence with Byzantine Faults

**Precise Definition**: (only for correct robots…)

A robot system satisfies the Byzantine convergence specification if and only if $\forall \varepsilon > 0, \exists t_\varepsilon$ such that $\forall t > t_\varepsilon$, $\forall i, j$, **distance**($r_i(t)$,$r_j(t)$) $< \varepsilon$, where $r_i(t)$ and $r_j(t)$ are the positions of some **correct** robots $r_i$ and $r_j$ at time $t$.

**Only correct robots have to converge.**




There is a time $t$ such that the distance between any two robots become smaller than any value $\varepsilon$.

# The Convergence with Byzantine Faults

The results of the convergence in Byzantine-prone systems.

| Ref. | Robots Model | Bounds |
|---|---|---|
| [AP04] | FSYNC | $n>3f$ (the gathering) |
| [BPT09-1] | FSYNC | **$n>2f$ (OPT)** |
| | AATOM(k-bounded) | **$n>3f$ (OPT)** |
| | ASYNC(k-bounded) | $n>4f$ |
| [BPT09-2] | ASYNC(k-bounded) | **$n>3f$ (OPT)** |
| Bouzid et al. | ASYNC | **$n>5f$ (OPT)** |

**k-bounded** : weak synchronicity such that between any two activations of a particular robot, any other robot can be activated at most k times

HOT !!

This result will be announced at OPODIS2009.

The results of [BPT09-1,2] are in **1 dimensional** space.

**2 or more dimensiona**l space is the **open problem** now.


[AP04] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*,11(14):1070–1078, 2004.

[BPT09-1] Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil. Byzantine-resilient convergence in oblivious robot networks. *ICDCN 2009*, pp. 275–280, January 2009.

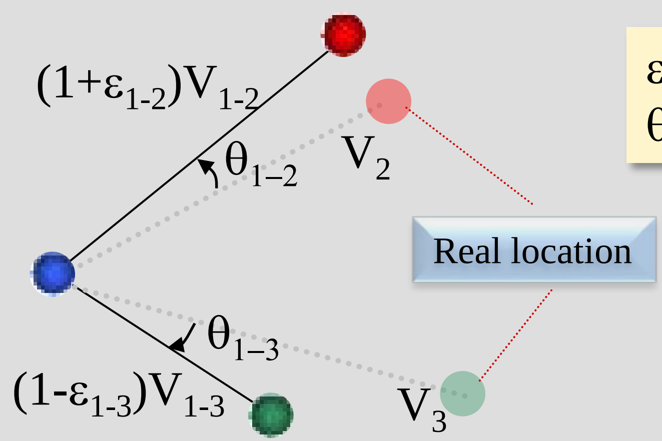
[BPT09-2] Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil. Optimal byzantine resilient convergence in asynchronous robot networks. *CoRR*, *abs/0906.0651*, 2009. ( or also appeared in proceedings of SSS2009.)

# The Convergence with Inaccurate Sensors

## Inaccurate Sensor

If sensing data (locations of observed robots) is error-prone, the gathering becomes difficult even if shared global coordinate system is assumed.

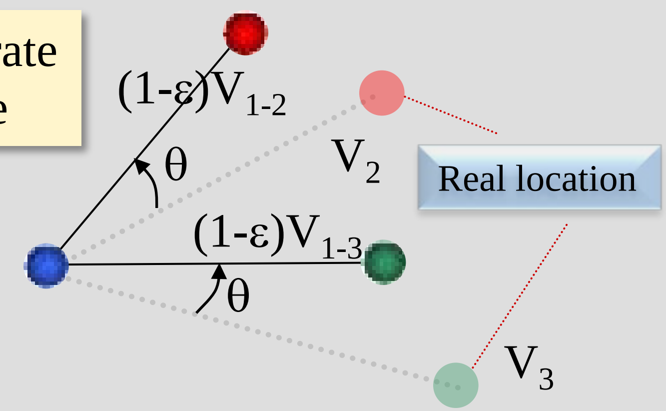
### **non-uniform error**[CP08]




The error rate for each observed robots location is **different.**

ε : distance error rate
θ : angle error rate

### **uniform error**[YIKIW09]



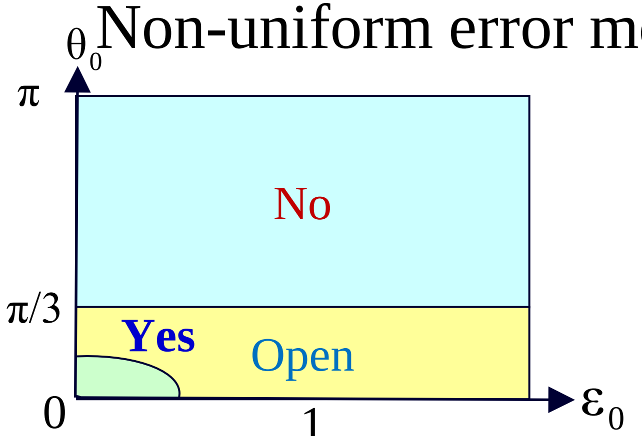

The error rate for each observed robots location is **same**.


[CP08] R. Cohen and D. Peleg. Convergence of autonomous mobile robots with inaccurate sensors and movements. *SIAM Journal on Computing*, 38(1):276–302, 2008. (Actually, this result has been proposed 2004.)
[YIKIW09] K. Yamamoto, T. Izumi, Y. Katayama, N. Inuzuka, and K. Wada. Convergence of mobile robots with uniformly-inaccurate sensors. *SIROCCO2009*, 2009.

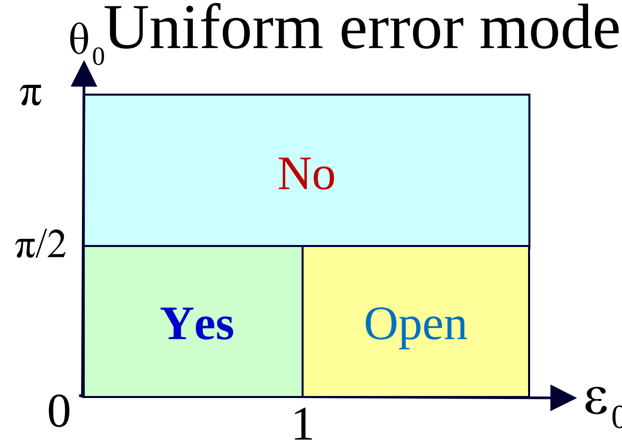
# The Convergence with Inaccurate Sensors

| Observation error model | Distance error | Angle error | Convergence |
|---|---|---|---|
| Non-Uniform | Any | $\theta_0 \geq \pi/3$ | No |
| | $0.2 > \sqrt{2(1-\varepsilon_0)\cdot(1-\cos\theta_0)+\varepsilon_0}$ | | Yes |
| | Otherwise | | Open |
| Uniform | $\varepsilon_0 < 1$ | $\theta_0 < \pi/2$ | Yes |
| | $\varepsilon_0 \geq 1$ | $\theta_0 < \pi/2$ | Open |
| | Any | $\theta_0 \geq \pi/2$ | No |

Non-uniform error model




Uniform error model



# The Gathering Problem

The gathering problem
All the robots **share** single point within finite time.

The fact [SY99, P05]:
The gathering problem is intractable without additional assumptions (multiplicity detection, non-oblivious, coordinate system …), even if the system is fault-free.

The intuitive reason deriving this fact is “the symmetricity.”

Our interests

To solve the gathering problem,
“**What is the minimum assumption ?**”
“**How about the fault tolerance ?**”


[SY99] I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal of Computing*, 28(4):1347–1363, 1999.
[P05] G. Prencipe. On the feasibility of gathering by autonomous mobile robots. *SIROCCO 2005*, volume 3499 of *LNCS*, pages 246–261, 2005.

# The Gathering Problem (Minimum Assumption)

## Memory (non-oblivious)

In [SY99], the authors proposed the gathering algorithm with **memory** in the SSYNC.

They used memory to determine the gathering point from initial configuration.

## Multiplicity Detection Mechanism

In [CFPS03], the authors proposed the gathering algorithm in the ASYNC with **the multiplicity** detection.

Their algorithm ensures that at any time during the execution there is at most one point where two ore more robots sharing; moreover, such a point will eventually be generated. Once this occurs, the robots that are already at that point remain there, while all other robots move towards this unique point.

## Consistent & Stable Common (global) Coordinate Systems

We can give a totally ordered rank to each robot, therefore there is a trivial algorithm such that, for instance, "Move toward a left & upper most robot!!"

[CFPS03] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Solving the robots gathering problem. *ICALP2003*, PP. 1181–1196, 2003.

# The Gathering with Crash & Byzantine

## Crash fault

In [AP04], the algorithm which tolerates **one crash** in the SSYNC was proposed.

## Byzantine fault

In [AP04], the algorithm which tolerate up to ***f*** **Byzantine (*n*>3*f*)** in the FSYNC.

And it was shown that there is **no gathering algorithm tolerating Byzantine faults in the SSYNC**.

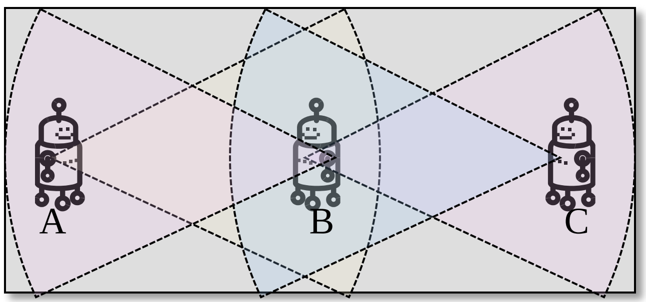
→ Naturally, in the ASYNC too.
(SSYNC can be simulated in ASYNC.)

[AP04] N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*,11(14):1070–1078, 2004.

# The Gathering with Limited Visibility

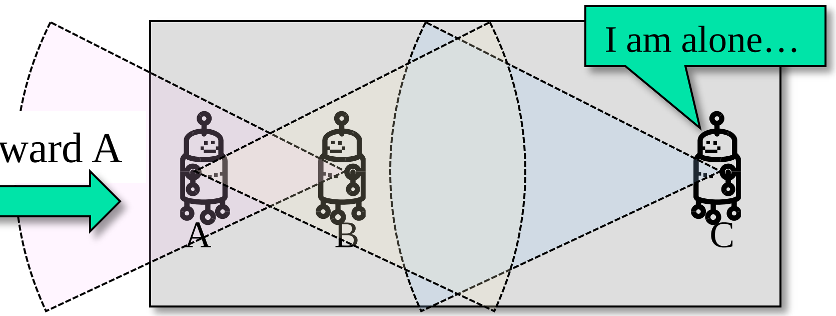
## Limited visibility

If visibility (sensing area) is limited, the gathering become difficult even if shared global coordinate system is assumed.




A and B, B and C can observe each other.
So, all robots are connected.

If B moves toward A




A can observe B, B can observe A, and
C cannot observe anyone. So, C is disconnected.

In [FPSW05], it was showed that **the problem is solvable** as long as **the robots share the knowledge of some direction** (coordinate system) with limited visibility in the ASYNC.


[FPSW05] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous mobile robots with limited visibility. *Theoretical Computer Science*, 337:147–168, 2005.

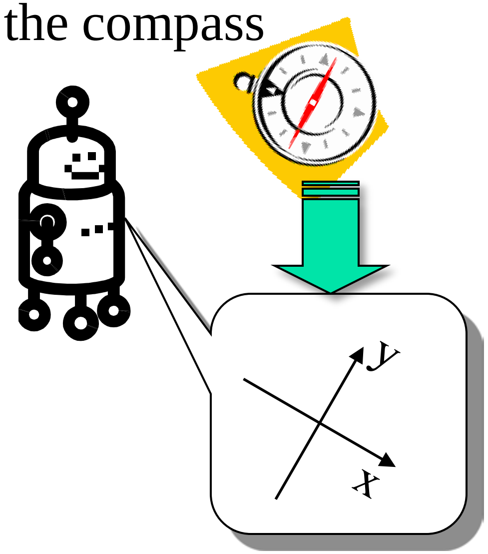
# The Gathering with Inconsistent Compasses

**What is a compass ?**

A classification of inconsistent compasses appears in [KTIIW07].

→ Determining the local coordinate system on each robot.
That is, the compass gives y-axis' positive direction of the local coordinate system.
(If a compass varies, then a local coordinate system varies according to it.)

Two factors defining the compass




stability

- stable compass
  The pointing direction of the compass **does not change** during the execution.
- unstable compass
  The pointing direction of the compass **may change** during the execution.

consistency

- consistent compasses
  All of the compasses on the robots point **same direction**.
- inconsistent compasses
  Each compass on the robots may point **different direction**.

# The Gathering with Inconsistent Compasses

## inconsistent compass (unstable & stable)

If local coordinate systems on each robot are inconsistent or/and unstable, the gathering become difficult even if they almost consistent.

→ There is **no algorithm** which solves the gathering when the compass is **unstable during the robots' movement**. So, we need to assume that **it occurs at only the beginning of Look phase.**

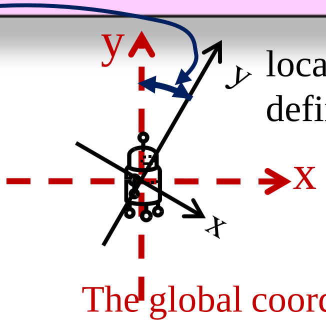
The reason…

The robots cannot move toward the computing point because of the instability of the coordinate system.

**Main Interest**

Where is the bound of degrees of inconsistency (differences of pointed direction between the compasses) that the gathering can be solvable ?

**NOTICE**: The difference (deviation) of the compass means "the difference from the direction of y-axis of the global coordinate system that every robot does not know."



# The Gathering with Inconsistent Compasses

The results of the maximum deviations that allow algorithms to solve the gathering problem for two robots are …

| | ASYNC | SSYNC |
|---|---|---|
| **Stable** | $< \pi$ | $< \pi$ |
| **Unstable** | $< \pi/3$ | $< \pi/2$ |

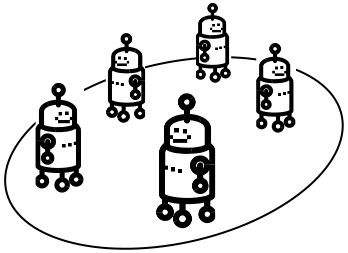
All of these results are **optimal** in the sense of deviations of the compass (except Unstable-SSYNC).

For instance, This result means…

The gathering in 2 robots system is **solvable** when the compass' deviation from the global coordinate system is **less than $\pi/3$** and its pointing direction may change during the execution (**unstable**).

And if the deviation is **more than or equal to $\pi/3$**, the gathering is **not solvable**.

[KTIIW07] Y. Katayama, Y. Tomida, H. Imazu, N. Inuzuka, K. Wada, Dynamic Compass models and Gathering Algorithms for Autonomous mobile Robots, *SIROCCO2007*, pp.274-288, 2007.



# How To Defeat The Faulty ?

**Simple Examples of the Gathering Algorithms**

“The gathering algorithm with inconsistent compasses”

“The convergence algorithm with Byzantine fault”

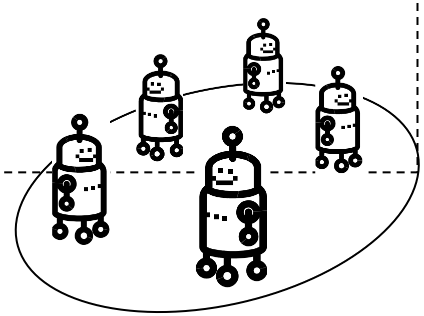
“The convergence algorithm with inaccurate sensors”

# The Gathering Algorithm with Inconsistent Compasses

[KTIIW07]

Y. Katayama, Y. Tomida, H. Imazu, N. Inuzuka, K. Wada, Dynamic Compass models and Gathering Algorithms for Autonomous mobile Robots,

*SIROCCO2007*, pp.274-288, 2007.



# How To Gather with Inconsistent Compasses

The most intuitively understandable algorithm ( I believe… :-)

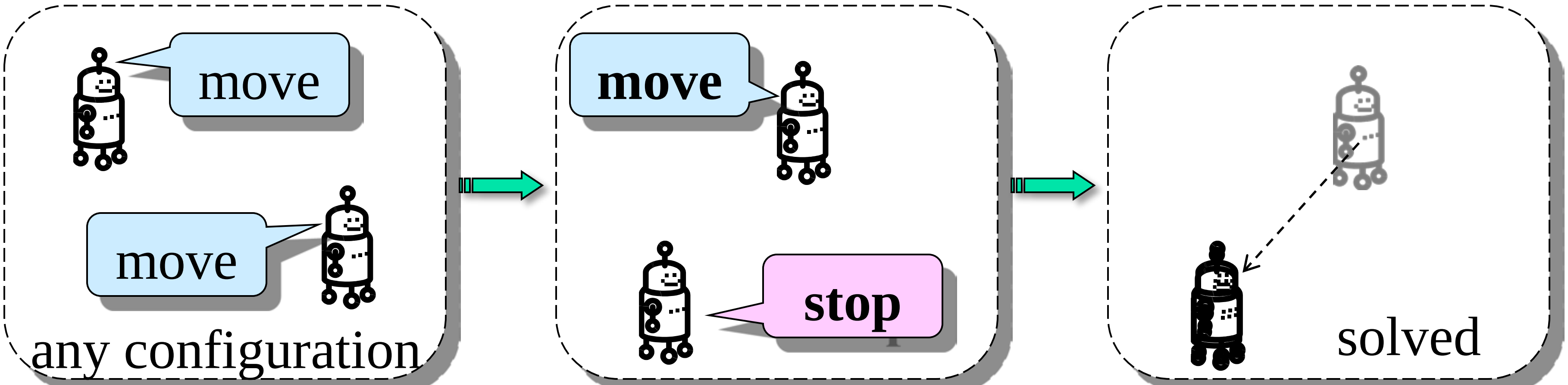
- ✓ The 2 robots gathering,
- ✓ unstable compasses, its deviation is at most **π/4** (not optimal),
- ✓ ASYNC model

| | **ASYNC** | **SSYNC** |
|---|---|---|
| **Stable** | < π | < π |
| **Unstable** | < π/3 | < π/2 |

Basic Strategy: To make the configuration…

**one robot move toward the other and the other does not move.**

- If the configuration is reached such a configuration, the gathering is able to achieve.



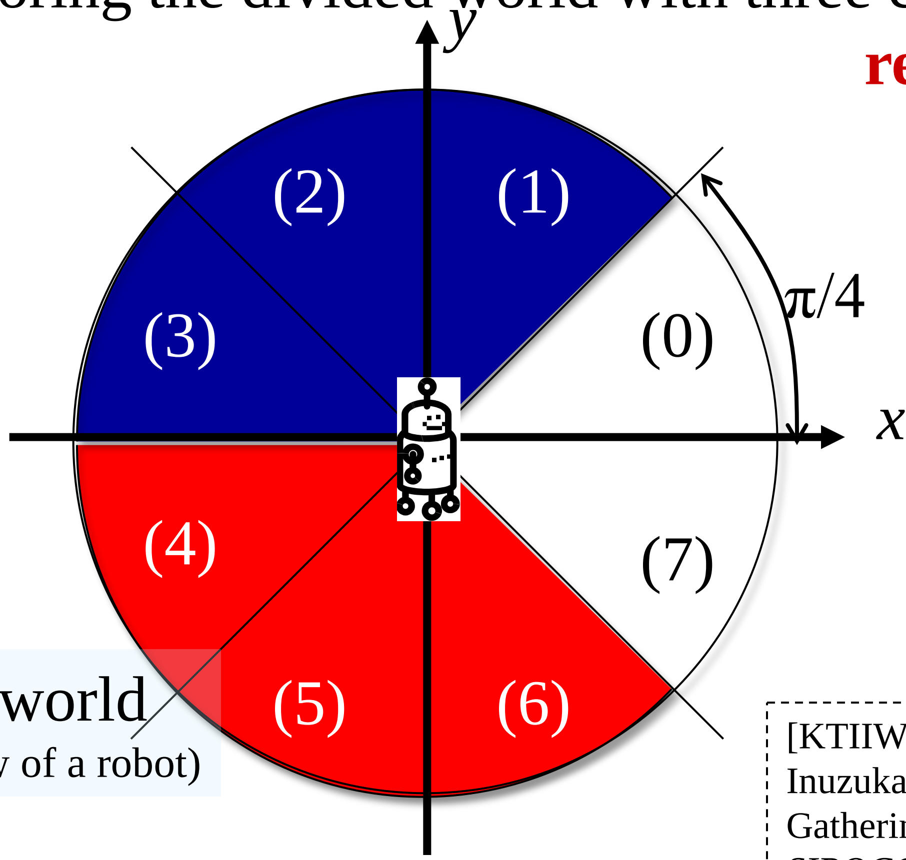
# How To Work The Algorithm?[KTIIW07]

**Point**: How to decide the robots' behavior ?

Dividing the world (a view of a robot) into 8 sectors.

Coloring the divided world with three colors:

**red**, **blue** and **white**.




The world
(a view of a robot)

Each robot **decides its behavior according to the sector** in which the other robot is observed.


[KTIIW07] Y. Katayama, Y. Tomida, H. Imazu, N. Inuzuka, K. Wada, Dynamic Compass models and Gathering Algorithms for Autonomous mobile Robots, *SIROCCO2007*, pp.274-288, 2007.

# The Gathering Algorithm[KTIIW07]

## Algorithm [KTIIW07]

Result of observing the other robot

case: **no robot**

gathering is achieved

case: **in blue sectors (1), (2), (3)**

move toward the other

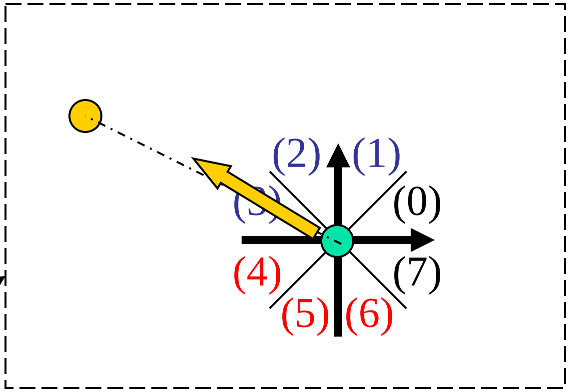
case: **in red sectors (4), (5), (6)**

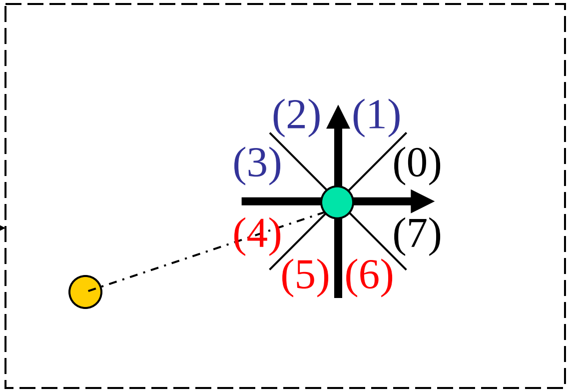
no move

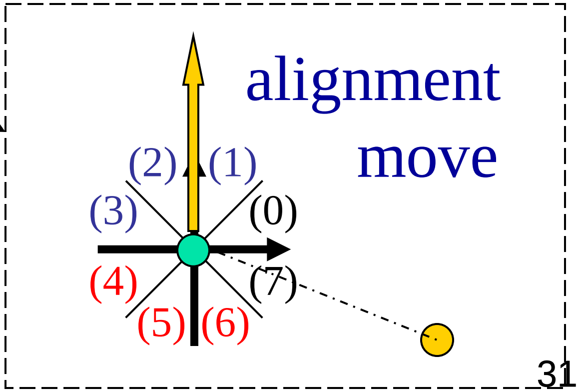
case: **in white sectors (7), (0)**

Alignment move

move toward a right above point where I will be able to observe the other robot in the sector (6)



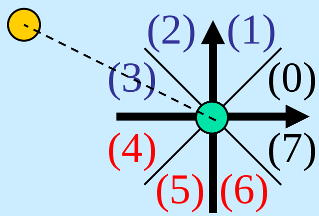



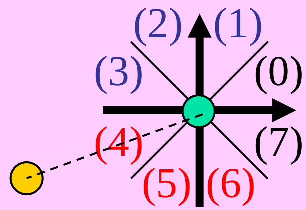
# The Gathering Algorithm[KTIIW07]

## Why the robots can gather ?

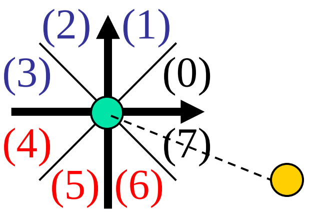
To show correctness, three names of robots are introduced:




**blue robot**
observing the other robot in blue sector.




**red robot**
observing the other robot in red sector.




**white robot**
observing the other robot in white sector.

**Dangerous Configurations**

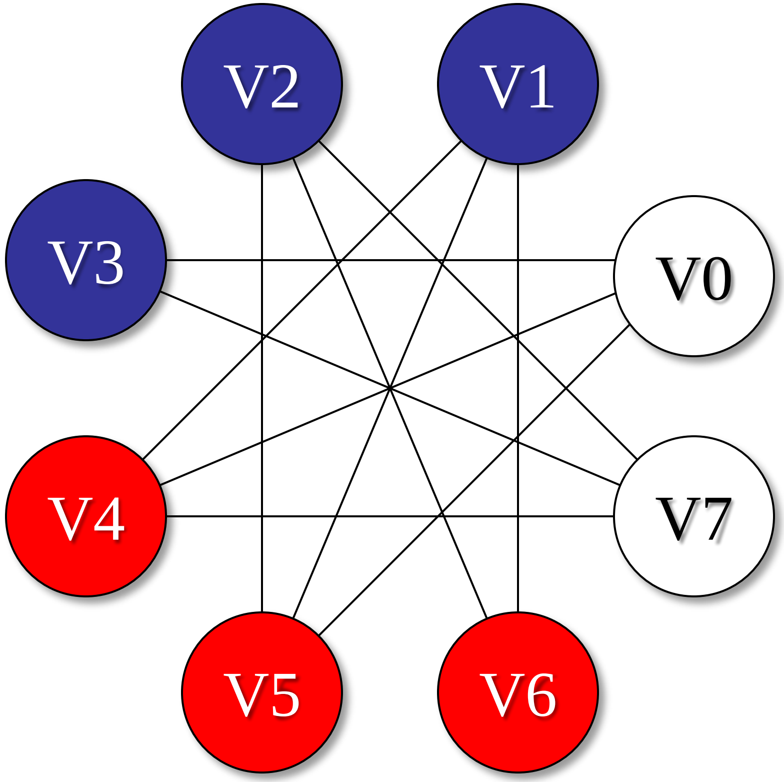
- **red - red** : deadlock!!
- **blue - blue** : swapping!! (loop)

**We must show:**

- **dangerous configurations never occur** **(safety)**
- **blue-red configuration is eventually reached** **(reachability)**

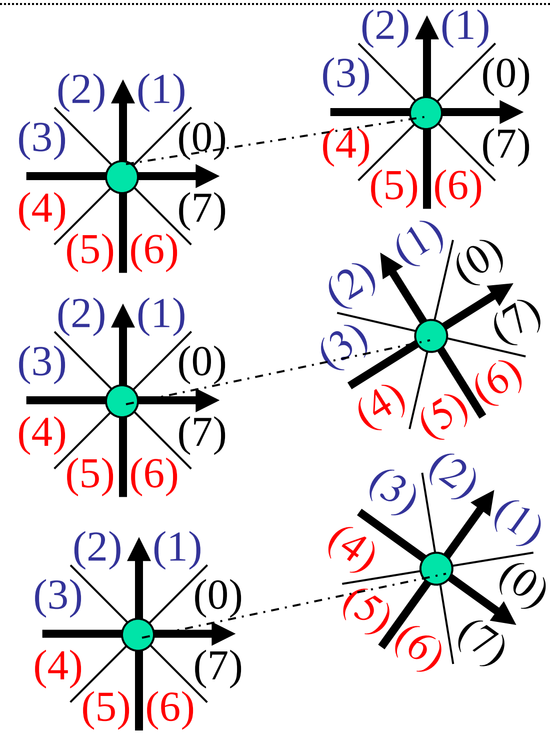
# The Gathering Algorithm[KTIIW07]

The Observation-Relation Graph




All nodes have three edges because of difference of compass.

- **Vi** represents a robot who observes the other in **sector (i)**.
- An edge (**Vi**,**Vj**) represents that a configuration can exist such that robots observe each other in **sector (i)** and **(j)** respectively.



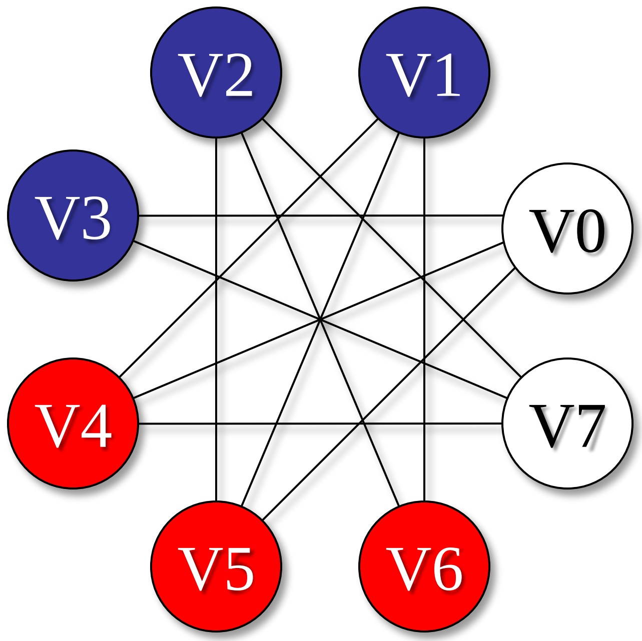

(V0, V4)

(V0, V3)

(V0, V5)

# The Gathering Algorithm[KTIIW07]

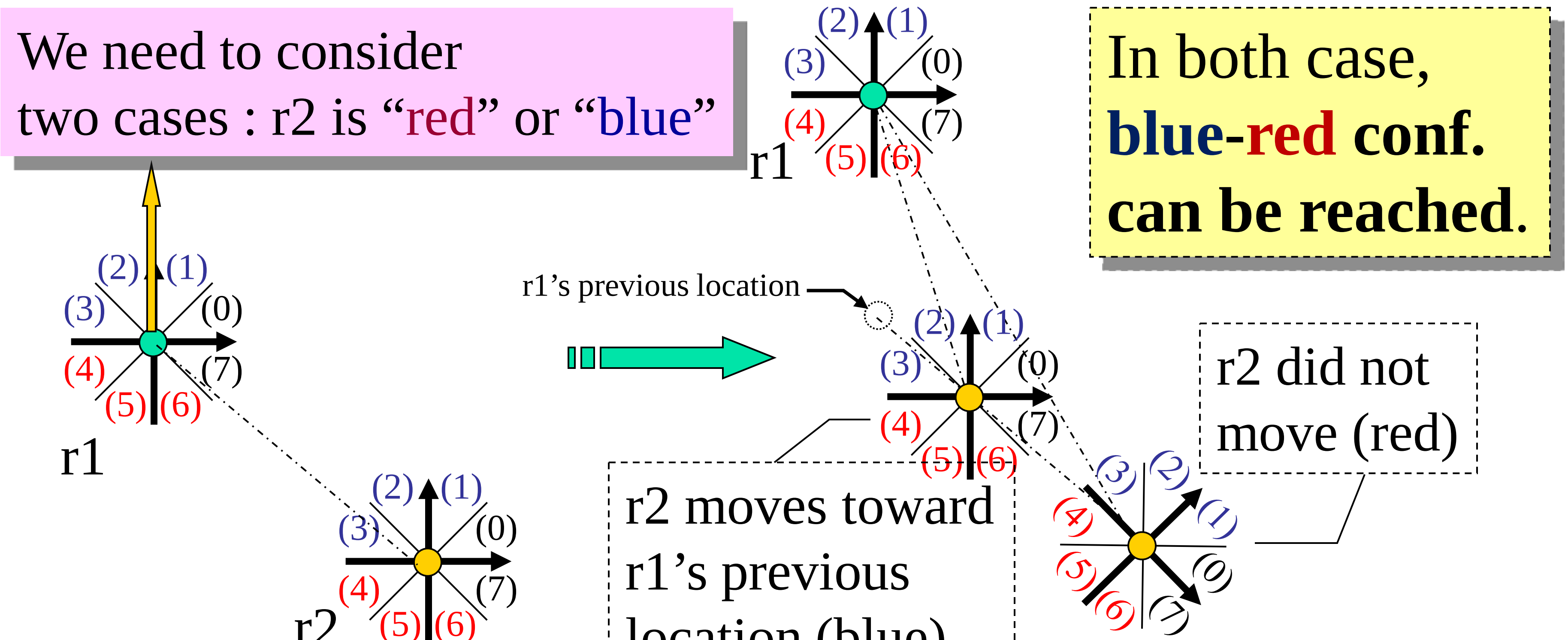
- Dangerous Configuration never occur
  - From the observation-relation graph with the sectoring and coloring, we can know “red-red / blue-blue configurations never occur through executions.”




Only
**blue-red, blue-white, red-white**
configurations can appear.

# The Gathering Algorithm[KTIIW07]

- “Blue-Red configuration” is eventually reached.
  - We need to show
    “From blue-white/red-white configuration, if r1 moves right above, **r1 can always observe r2 in sector (6)**.”



# The Convergence Algorithm with Byzantine Fault

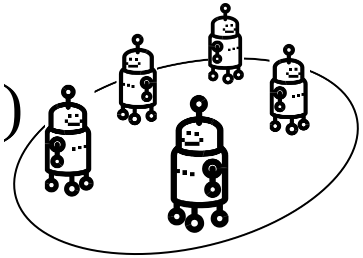
[BPT09-2]

Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil,

Optimal byzantine resilient convergence in asynchronous robot networks,

*CoRR*, *abs/0906.0651*, 2009.

(also appeared in proceedings of SSS2009.)



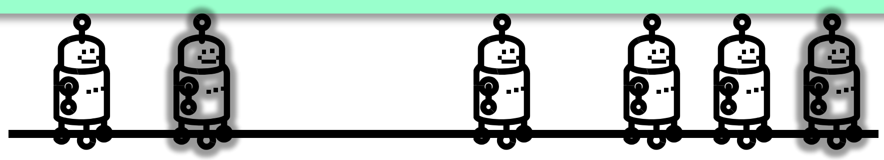
# How To Converge with Byzantine Fault

Byzantine robots and the adversarial scheduler can do any movement that disturbing the gathering.

→ The algorithm have to defy those disturbance.
And gather correct robots at single point

The convergence algorithm in [BPT09-2]

- ✓ The *n* robots convergence,
- ✓ *f* Byzantine robots among *n* robots where $n>3f$,
- ✓ k-bounded ASYNC model
- ✓ 1 dimensional space
- ✓ Strict Multiplicity detection



| Ref. | Robots Model | Bounds |
|---|---|---|
| [AP04] | FSYNC | n>3f |
| [BPT09-1] | FSYNC | **n>2f (OPT)** |
| | AATOM(k-bounded) | **n>3f (OPT)** |
| | ASYNC(k-bounded) | n>4f |
| [BPT09-2] | ASYNC(k-bounded) | **n>3f (OPT)** |
| Bouzid et al. | ASYNC | **n>5f (OPT)** |

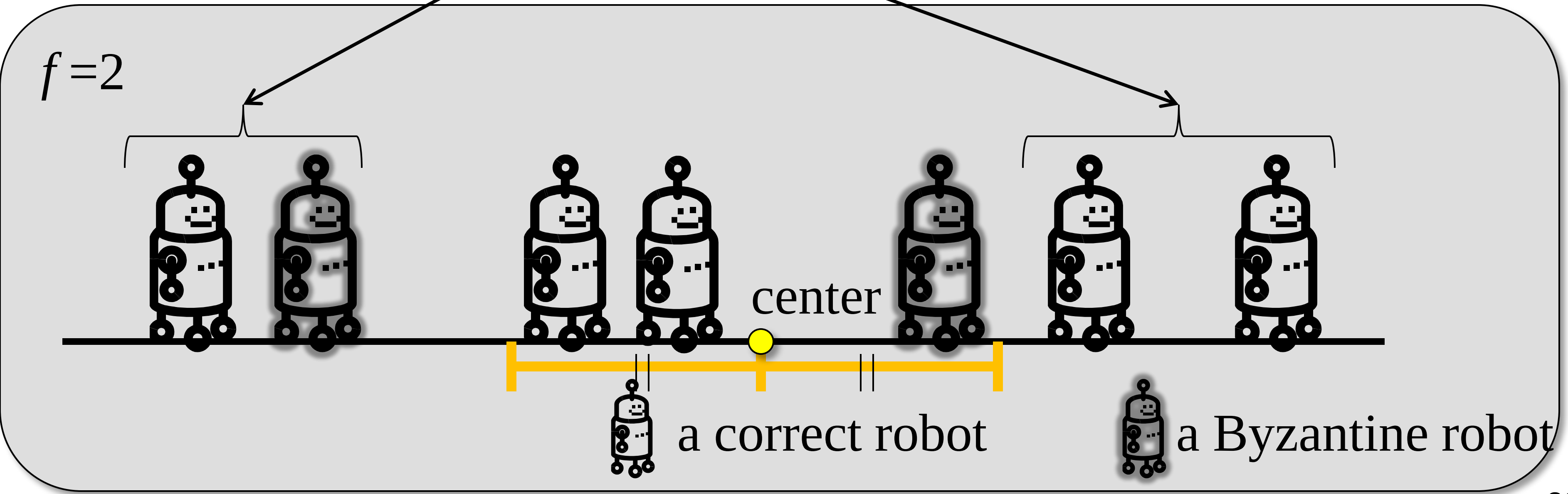
[BPT09-1] Z. Bouzid, M. G. Potop-Butucaru, and S. Tixeuil. Byzantine-resilient convergence in oblivious robot networks. *ICDCN 2009*, pp. 275–280, January 2009.

# The Convergence Algorithm [BPT09-2]

Basic Strategy

Ignoring *f* Byzantine robots' movements that corrupt *n*-*f* correct robots' movements.

The correct robots move toward the center of the set of robots that removing ***f* smallest** and ***f* largest** robots.



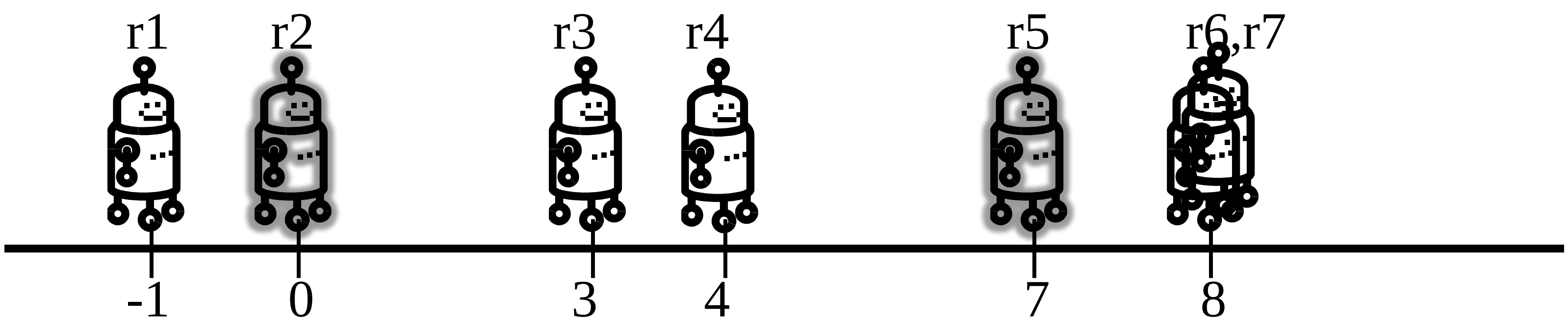
# The Convergence Algorithm [BPT09-2]

The previous strategy cannot achieve when $n > 3f$… ($n>4f$ is OK)

Now, introducing an algorithm in [BPT09-2]

**Variables:**

- $\mathbf{P_i(t)}$ : the position of the robot r*i* at time *t*.
- **P(t)** : the multiset of positions of all robots in the system at time *t*.




$P_{r1}(t) = -1$, $P_{r2}(t)=0$, $P_{r3}(t)=3$, $P_{r4}(t)=4$, $P_{r5}(t)=7$, $P_{r6}(t)=8$, $P_{r7}(t)=8$

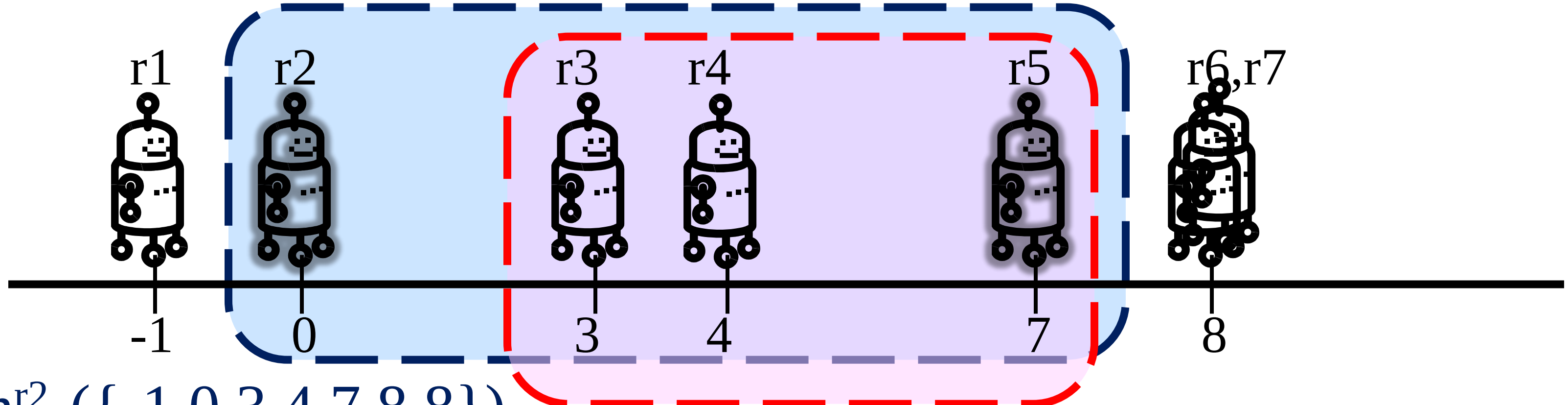
$P(t) = \{P_{r1}(t), P_{r2}(t), \ldots, P_{r7}(t)\} = \{-1, 0, 3, 4, 7, 8, 8\}$

NOTICE:
This example is represented by the global coordinate system.
Actually, in the algorithm, these are represented by the local coordinate system on each robot.

# The Convergence Algorithm [BPT09-2]

**Functions:**

- **$trim^i_f(P(t))$** : removes up to *f* largest positions that are **larger than $P_i(t)$** and up to *f* smallest positions that are **smaller than $P_i(t)$** from the P(t).
- **center** : returns the center of the input range (a set of robots' positions.)




$trim^{r2}_2(\{-1,0,3,4,7,8,8\}) = \{0,3,4,7\}$

$trim^{r3}_2(\{-1,0,3,4,7,8,8\}) = \{3,4,7\}$

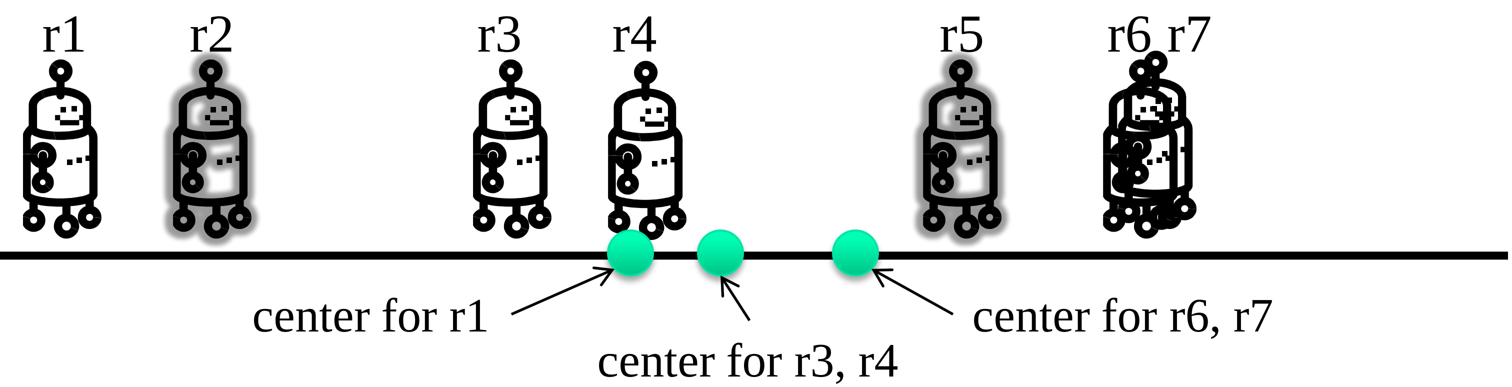
# The Convergence Algorithm [BPT09-2]

**Algorithm [BTP09-2]**

1. Compute **center($\mathrm{trim}^{i}_{f}$(P(t)))**
2. Move toward that point

By trim function, the correct robots can converge at a point in spite of Byzantine robots' disturbance of their adversarial movement.

$n$ =7
$f$ =2




This algorithm can solve the convergence for $n$ robots including $f$ Byzantine robots when $n > 3f$, and satisfies two properties "**Shrinking**" and "**Cautious**".

What is Shrinking? Cautious?

# The Convergence Algorithm [BPT09-2]

**Shrinking Property:**
An algorithm is **shrinking** if and only if

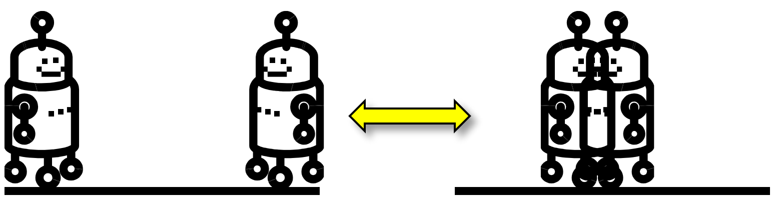
$$\exists \alpha \in (0,1), \forall t, \exists t' > t, \text{ such that}$$
$$\mathrm{diam}(U(t') \cup D(t')) < \alpha \times \mathrm{diam}(U(t) \cup D(t))$$

where <u>$U(t)$</u> and <u>$D(t)$</u> are respectively the multisets of positions and destinations of correct robots, and <u>**diam**</u> means diameter of input range.

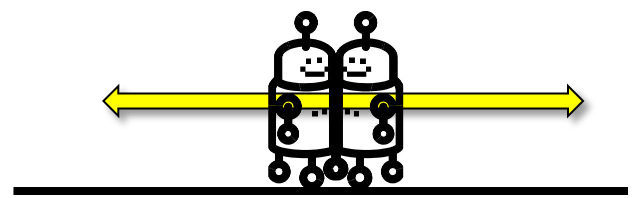
even if the system is fault-free

"Shrinking" is **necessary condition** but **not sufficient** for convergence.

No guarantee of monotonicity and stability.



No monotonicity (pulsation)



No stability (oscillation)

# The Convergence Algorithm [BPT09-2]

Accordingly, to guarantee monotonicity and stability…

**Cautious Property**:

An algorithm is **cautious** if it meets the following conditions:

- **cautiousness**: $\forall t, D_i(t) \in \text{range}(U(t))$ for each correct robot $r_i$.
  $\rightarrow$ all of the correct robots never go out from the present range.
- **non-triviality**: $\forall t,$ if $\text{diam}(U(t)) \neq 0$ then $\exists t' > t$ and a robot $r_i$ such that $D_i(t') \neq U_i(t')$
  $\rightarrow$ at least one correct robot changes its position whenever convergence is not achieved.

where $D_i(t)$ is the last destination calculated by robot $r_i$, and $U_i(t)$ is the multiset of the position of the correct robots as seen by robot $r_i$ at the time.

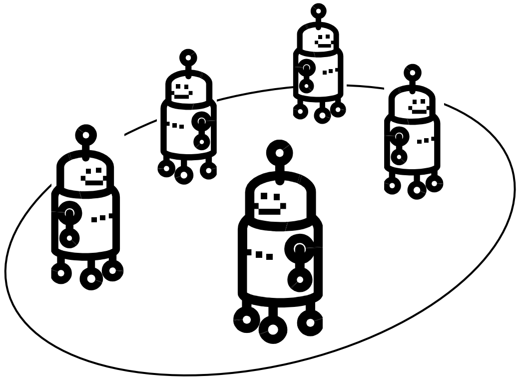
⇨ Totally, this property says that
"the correct robots have to move toward the point that exists inside the range that consists of the correct robots."

# The Convergence Algorithm with Inaccurate Sensors

[YIKIW09]

K. Yamamoto, T. Izumi, Y. Katayama, N. Inuzuka, and K. Wada,

Convergence of mobile robots with uniformly-inaccurate sensors, *SIROCCO2009*, 2009.

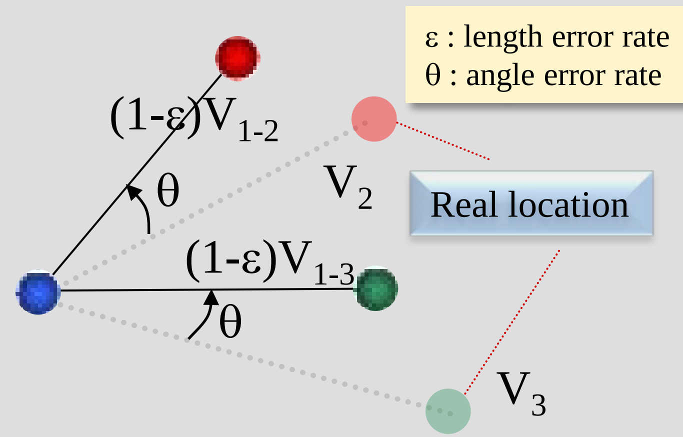


# The Convergence Algorithm with Inaccurate Sensors

The convergence algorithm with uniform inaccurate sensors is…

- ✓ The 2 ASYNC robots convergence,
- ✓ The uniform inaccurate sensor, its maximum error rate is $\theta_0$ ($<\pi/2$) and $\varepsilon_0$ ($0 \leq \varepsilon_0 < 1$).
- ✓ Every robot knows $\theta_0$ and $\varepsilon_0$.

**uniform error**[YIKIW09]




The error rate is **same** among the observed robots

**inconsistent compass**:
Every robot can observe other robots' location precisely.
Consequently, a robot can reach the other robot's point.

The most significant difference

**inaccurate sensor**:
Every robot may observe the phantom of the other robots.
Consequently, the robots may not reach the other robot's point.

The gathering cannot be solved.

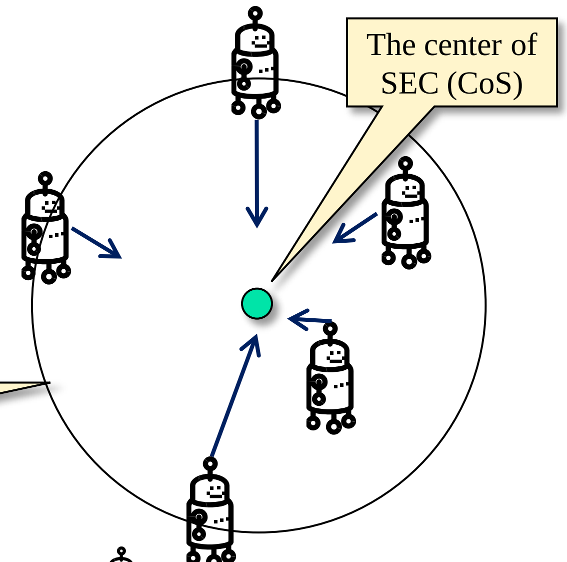
# The Convergence Algorithm [YIKIW09]

Basic Strategy

**Move every robot to make the radius of the smallest enclosing circle (SEC) converge to zero.**

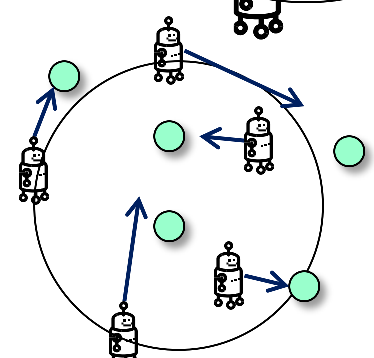
→ Eventually, every robot may converge to the center of SEC.

If every robot is equipped with accurate sensors, that is, **if every robot can compute correct center of SEC, it is easy to solve the convergence.**




↓

Now, the sensors are inaccurate.
**Every robot may compute different SECs** respectively, because each of their observation result includes an observation error that rate may be different.



**The algorithm have to control the robots' movement so that the robot does not go out from SEC.**

# The Convergence Algorithm [YIKIW09]

**The algorithm have to control the robots' movement so that not go out from SEC.**

→ The algorithm determines two parameters "**distance**" & "**angle**" of movement to guarantee the robots are always inside SEC.

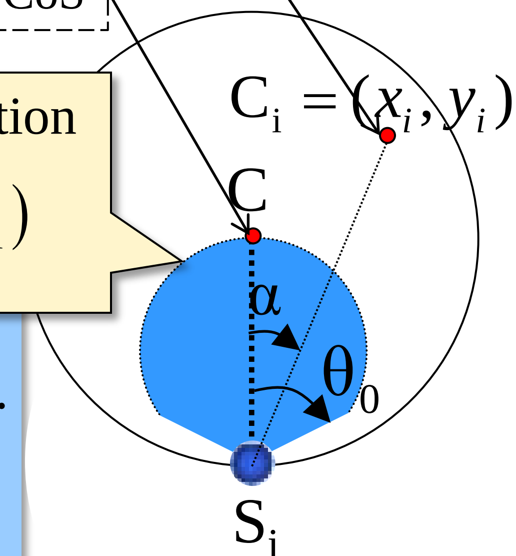
How to determine "distance" and "angle" ?

Ci: computed CoS

C: correct CoS

Area of the destination

$$\frac{1}{1+\varepsilon_0}\cos\alpha \cdot (x_i, y_i)$$




**angle:**

move toward $C_i$

**distance:**

The length of observation is at most $\frac{1}{1+\varepsilon_0}$ times (by length error).
Hence, the algorithm divides the length by $1+\varepsilon_0$ .
And the length should be multiplied by cos α because of the angle error.

But, the robots do not know α (only know $\theta_0$) !!

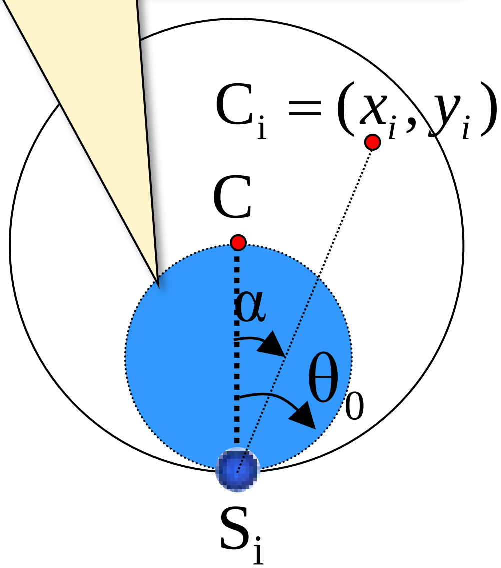
# The Convergence Algorithm [YIKIW09]

But, the robots do not know α (only know $\theta_0$) !!

Since $\alpha \le \theta_0$, the distance of movement is multiplied by $\cos\theta_0$ instead of $\cos\alpha$.

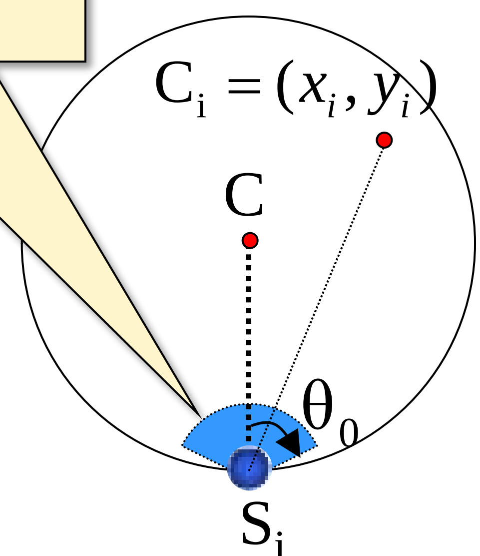
Area of the destination

$$\frac{1}{1+\varepsilon_0}\cos\alpha\cdot(x_i, y_i)$$




**The possible destination point always stays in the SEC.**

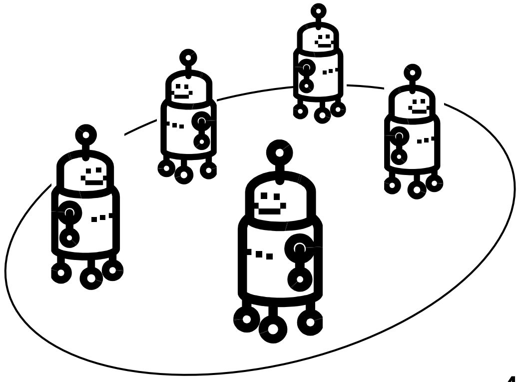
$$\frac{1}{1+\varepsilon_0}\cos\theta_0\cdot(x_i, y_i)$$



# The Convergence Algorithm [YIKIW09]

In [YIKIW09], the authors showed

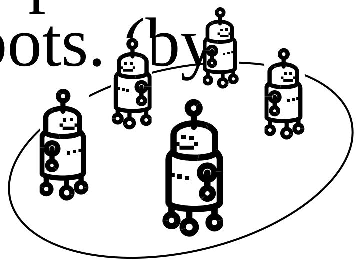
- An algorithm for the convergence problem where the observation error rates are $\theta_0 < \pi/2, 0 \le \varepsilon_0 < 1$.
- No algorithm for the convergence problem exists when the angle error rate is $\theta_0 < \pi/3$.



# Conclusion

# Conclusion

- There is a difference between the gathering and the convergence. (Both problems are Rendezvous Problem)
  - Rendezvous problem is able to consider as a point formation.
  - Symmetricity and Asynchronicity are constitutive difficulties for Rendezvous problem.
- And more, the general pattern formation and the point formation (gathering) is different.
  - In the SSYNC model, all pattern formable with sufficient memory is also formable without memory, except the point formation (i.e., the gathering) for two robots. (by private communication with Prof. Yamashita.)



# Conclusion

- There are open problems in the gathering problem.
  - Many impossibilities has been shown.
  - But, since the problem is very sensitive to the robot models (execution, consistency of the local coordinate system, and so on), to find “some bounds” is still interesting.
- There are many open problems in the convergence problem in error-prone (faulty) systems.
  - Some open problems have been shown in this talk.
- It is also interesting to find equivalencies among
  - fault models
  - robot models (synchronicity, abilities, knowledge, …)

under certain problem settings.