Skalering og feiltoleranse i robotsvermer

Jan Dyre Bjerknes. ITS 12.10.2022



KONGSBERG

KONGSBERG PROPRIETARY: This document contains KONGSBERG information which is proprietary and confidential. Any disclosure, copying, distribution or use is prohibited if not otherwise explicitly agreed with KONGSBERG in writing. Any authorised reproduction in whole or in part, must include this legend. © 2013 KONGSBERG – All rights reserved.

Oversikt

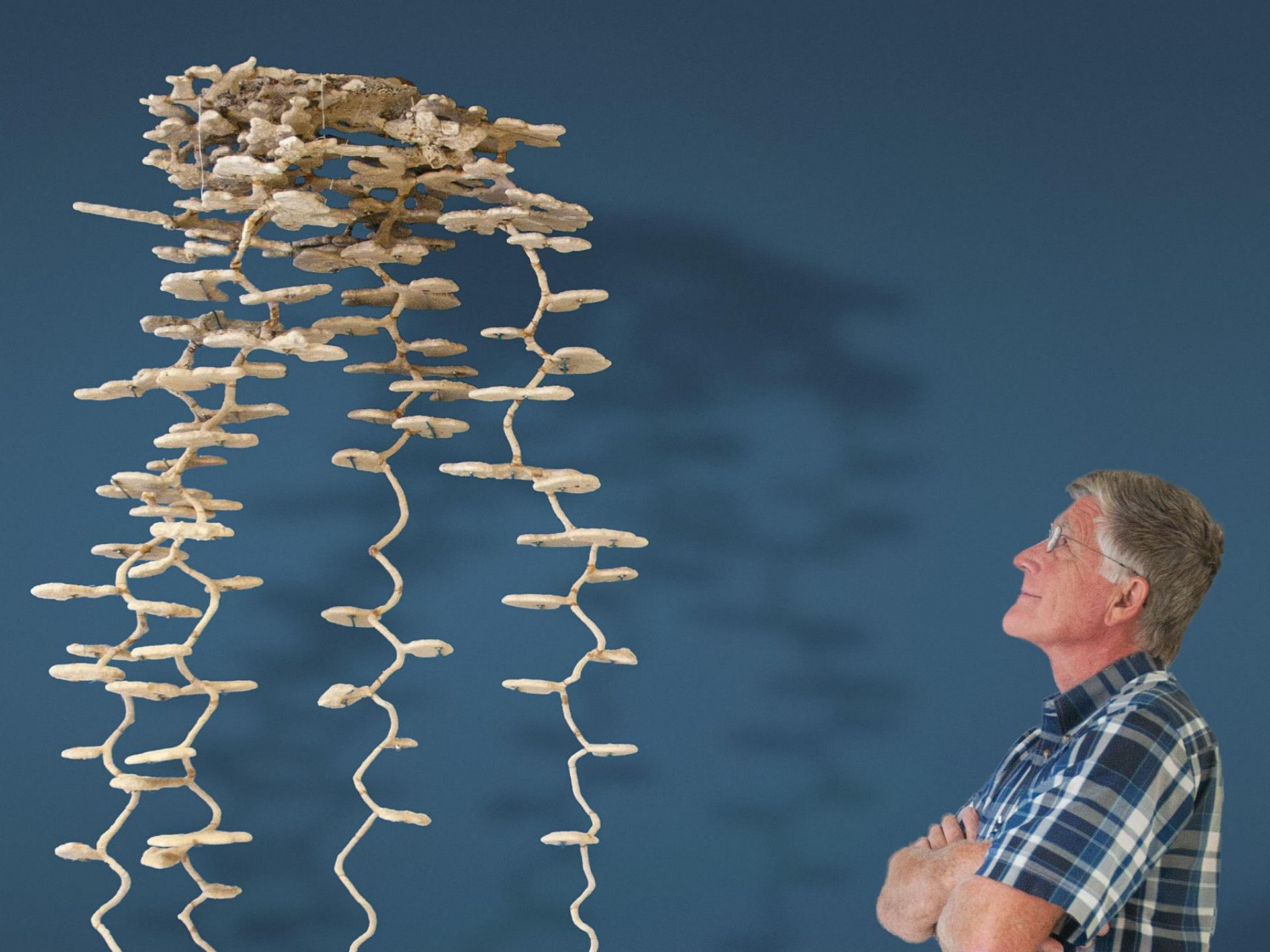
- Litt om inspirasjonen bak robotsvermer.

- Forskjellig typer samarbeid.
  - Parallelt arbeid
  - Gruppearbeid
  - Teamarbeid

- Dicty.
- Vår oppgave
- Matematisk analyse, simulering, hardware

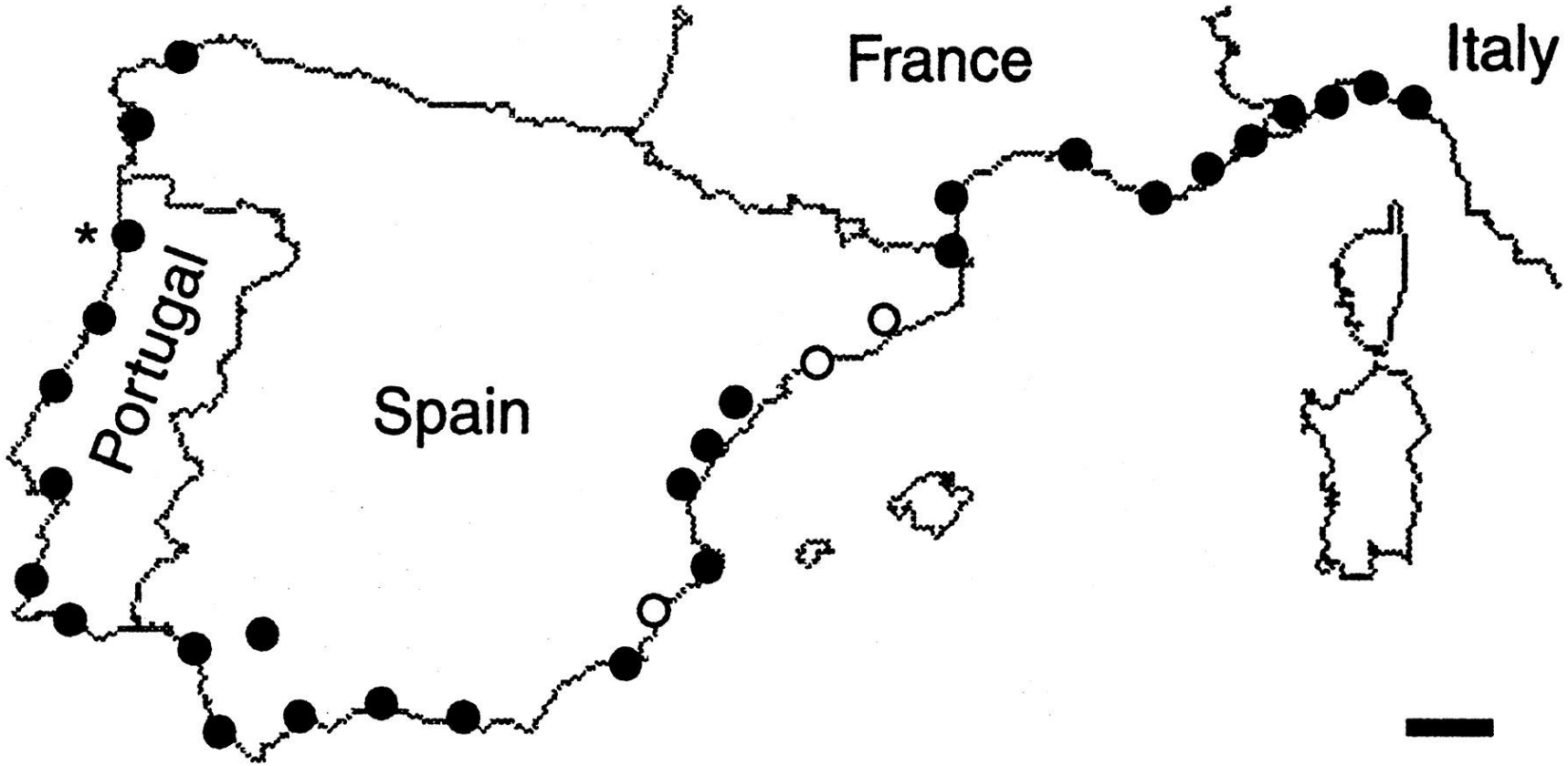
- Skaleringsproblemer
  - Løsninger
- Feiltoleranse.







KONGSBERG









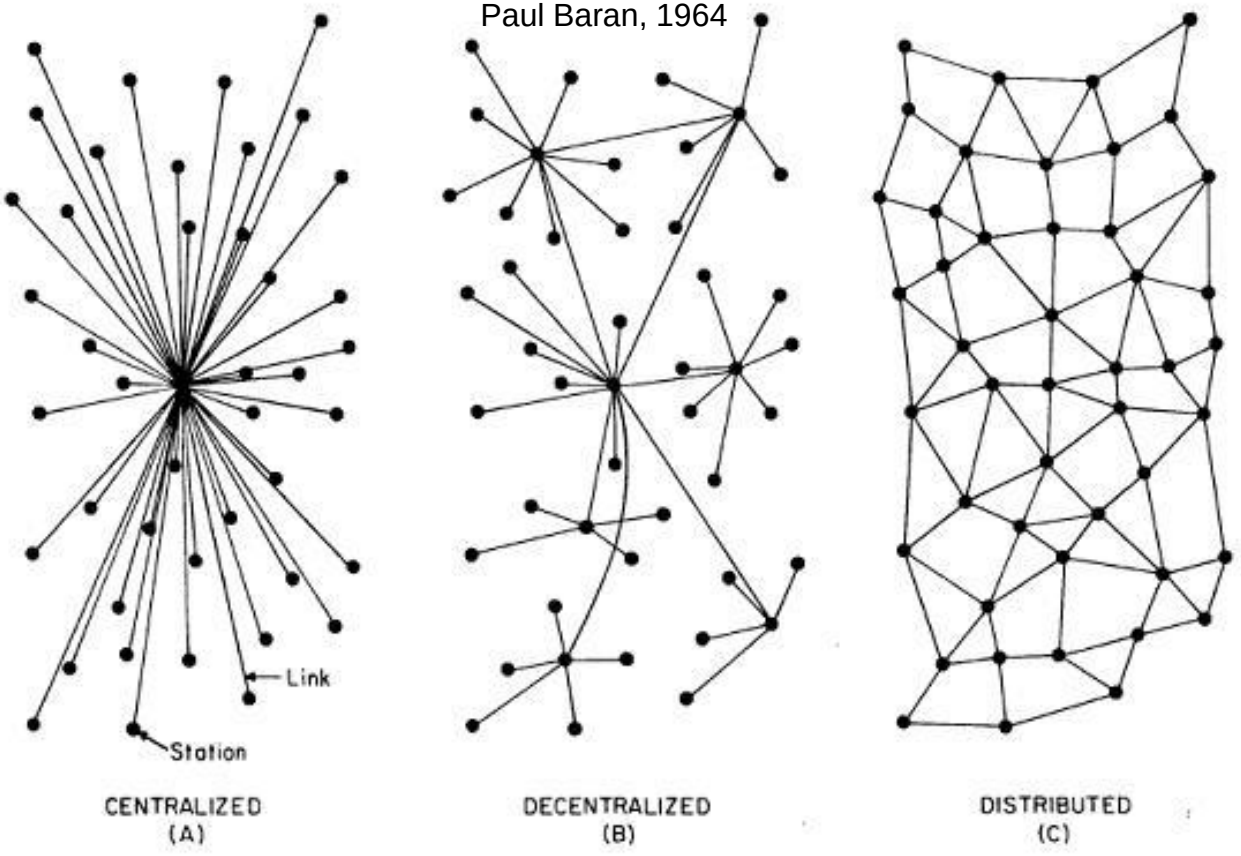


Hva kjennetegner en sverm?

Hva kjennetegner en sverm?

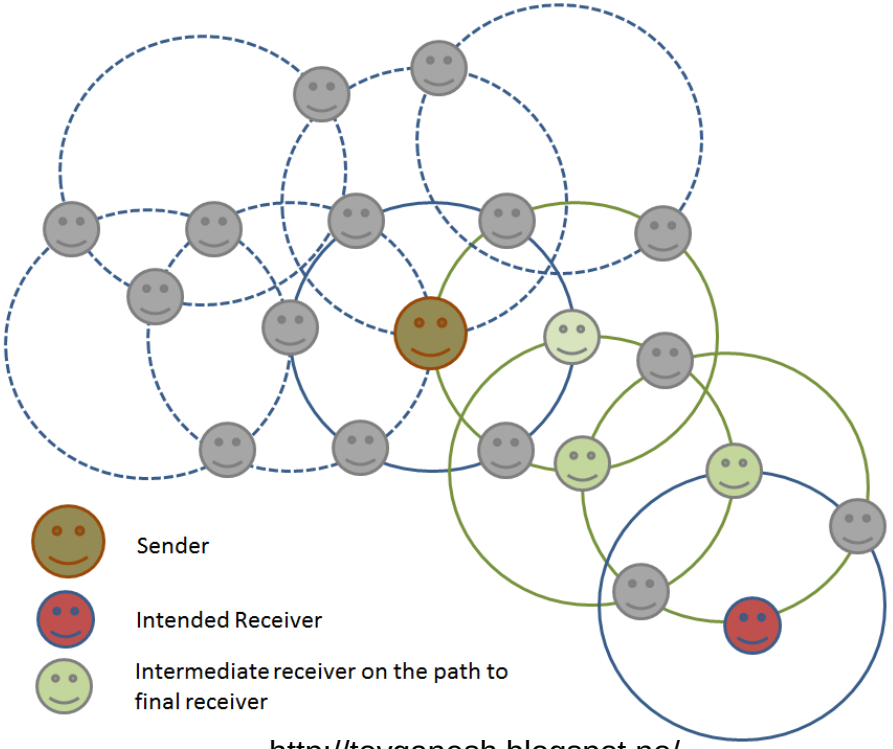
- Distribuert og selvorganisert

Paul Baran, 1964



Hva kjennetegner en sverm?

- Desentralisert og selvorganisert
- Lokal kommunikasjon
  - Med hverandre

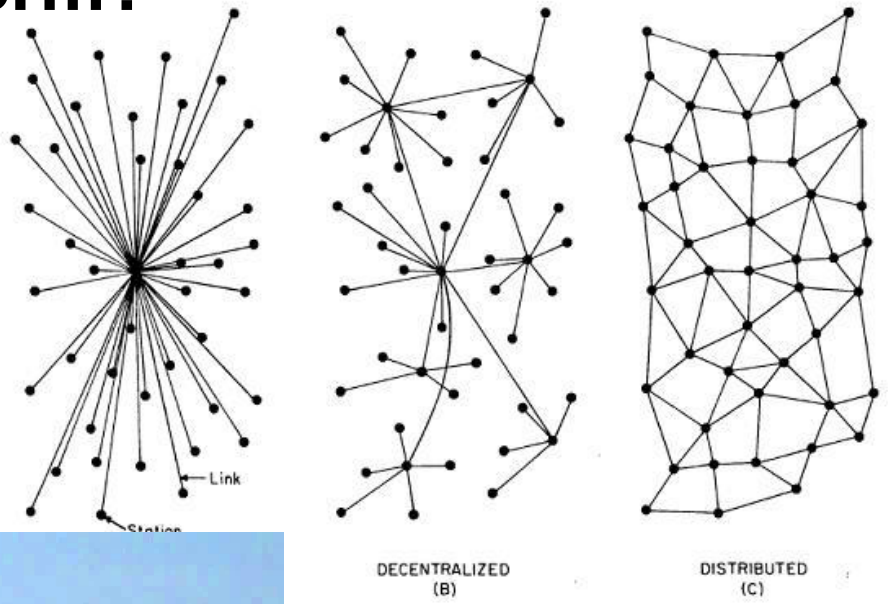


http://tovganesh.blogspot.no/

Erol Sahin (2008): ***Swarm Intelligence: Introduction and Applications***

Hva kjennetegner en sverm?

- Distribuert og selvorganisert

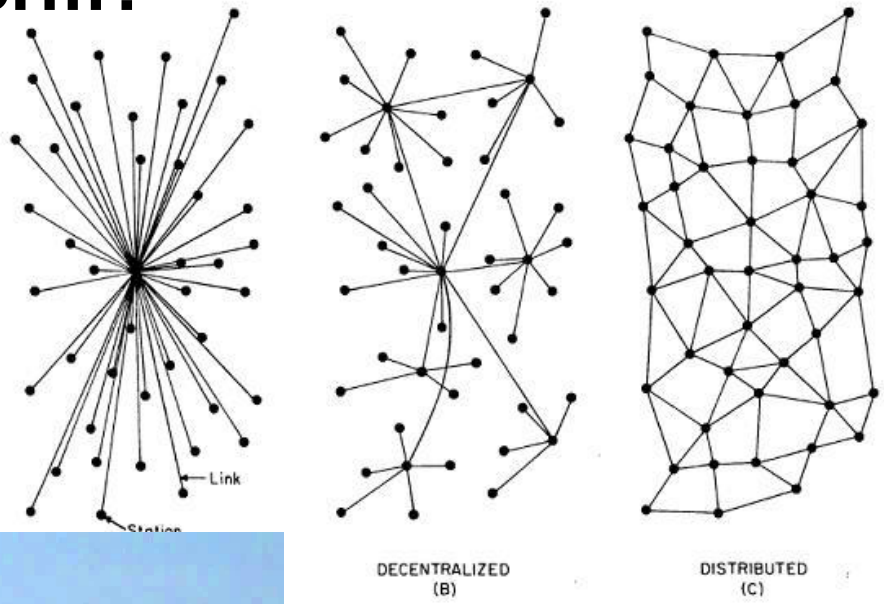


Paul Baran, 1964



Hva kjennetegner en sverm?

- Distribuert og selvorganisert
- Lokal kommunikasjon
  - Med hverandre



Paul Baran, 1964



Hva kjennetegner en sverm?

- Desentralisert og selvorganisert
- Lokal kommunikasjon
  - Med hverandre
  - Via omgivelsene



https://en.wikipedia.org/wiki/Polistes_dominula

Erol Sahin (2008): ***Swarm Intelligence: Introduction and Applications***

Hva kjennetegner en sverm?

- Desentralisert og selvorganisert
- Lokal kommunikasjon
  - Med hverandre
  - Via omgivelsene
- Alle agentene i en sverm har med energi til eget forbruk

Erol Sahin (2008): ***Swarm Intelligence: Introduction and Applications***

Hva kjennetegner en sverm?

- Desentralisert og selvorganisert
- Lokal kommunikasjon
  - Med hverandre
  - Via omgivelsene
- Alle agentene i en sverm har med energi til eget forbruk
- Alle agentene i en sverm har med egen regnekraft

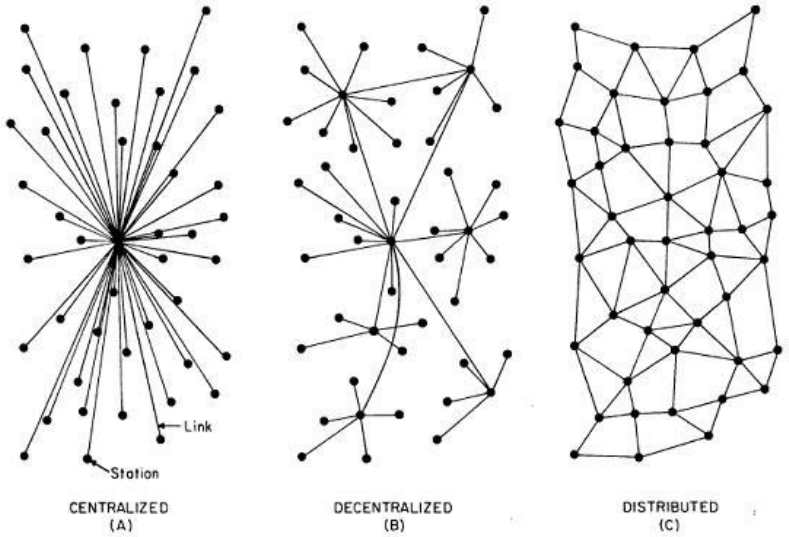
Erol Sahin (2008): ***Swarm Intelligence: Introduction and Applications***

Hva kjennetegner en sverm?

- Desentralisert og selvorganisert
- Lokal kommunikasjon
  - Med hverandre
  - Via omgivelsene
- Alle agentene i en sverm har med energi til eget forbruk
- Alle agentene i en sverm har med egen regnekraft

Tanken er at dette skal gi oss:

- Robusthet



Erol Sahin (2008): ***Swarm Intelligence: Introduction and Applications***

Hva kjennetegner en sverm?

- Desentralisert og selvorganisert
- Lokal kommunikasjon
  - Med hverandre
  - Via omgivelsene
- Alle agentene i en sverm har med energi til eget forbruk
- Alle agentene i en sverm har med egen regnekraft

Tanken er at dette skal gi oss:

- Robusthet
- Fleksibilitet

Erol Sahin (2008): ***Swarm Intelligence: Introduction and Applications***

Hva kjennetegner en sverm?

- Desentralisert og selvorganisert
- Lokal kommunikasjon
  - Med hverandre
  - Via omgivelsene
- Alle agentene i en sverm har med energi til eget forbruk
- Alle agentene i en sverm har med egen regnekraft

Tanken er at dette skal gi oss:

- Robusthet
- Fleksibilitet
- **Skalering**

Erol Sahin (2008): ***Swarm Intelligence: Introduction and Applications***

Skalering

- Sahin (2008):
  - ..a swarm robotic system should be able to operate under a wide range of group sizes. That is, the coordination mechanisms that ensure the operation of the colony should **be relatively undisturbed by changes in the group sizes.**

Skalering

- Sahin (2008):
  - ..a swarm robotic system should be able to operate under a wide range of group sizes. That is, the coordination mechanisms that ensure the operation of the colony should be relatively undisturbed by changes in the group sizes.

- Bondi(2000):
  - (1) the ability of a computer application or product (hardware of software) to function well as it (or its context) is changed in size or volume in order to meet a user need; and (2) the ability not only to function well in the rescaled situation, but to actually **take full advantage of it.**

Forskjellige typer arbeid i svermer

Forskjellige typer svermer

- Parallelt arbeid
- Gruppearbeid
- Teamarbeid

Forskjellige typer svermer

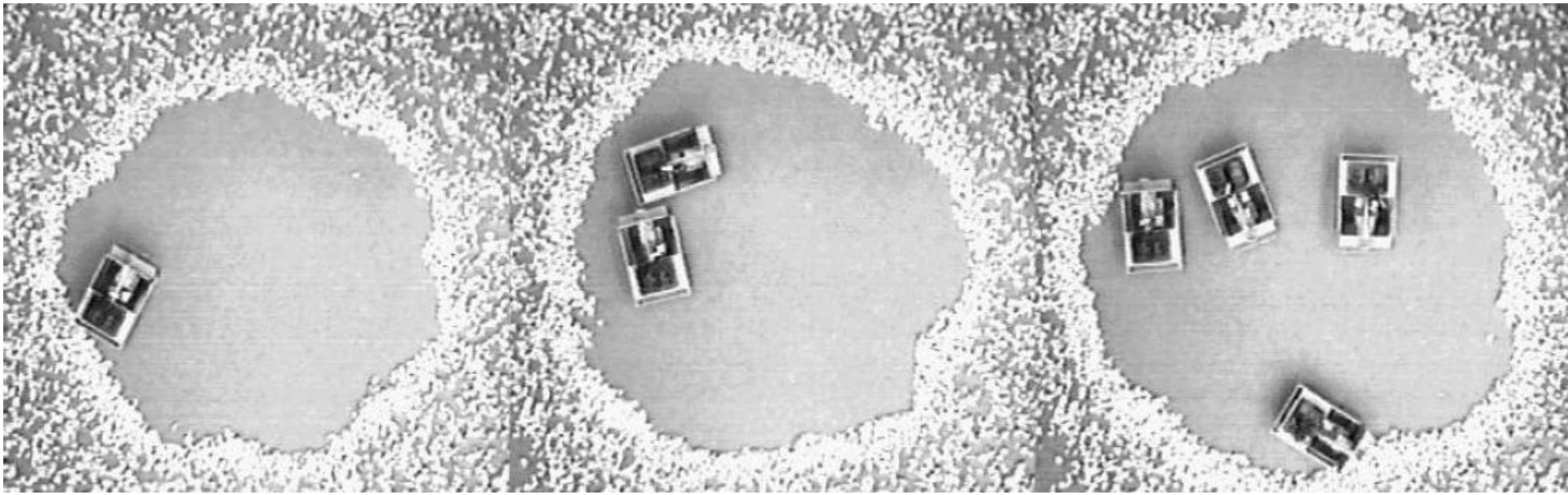
- Parallelt arbeid
  - Arbeid som kan utføres av én robot, men å bruke mange gjør at jobben utføres fortere

Forskjellige typer svermer

- Parallelt arbeid
  - Arbeid som kan utføres av én robot, men å bruke mange gjør at jobben utføres fortere
  - Overvåkning

Forskjellige typer svermer

- Parallelt arbeid
  - Arbeid som kan utføres av én robot, men å bruke mange gjør at jobben utføres fortere
  - Overvåkning
  - Rydding

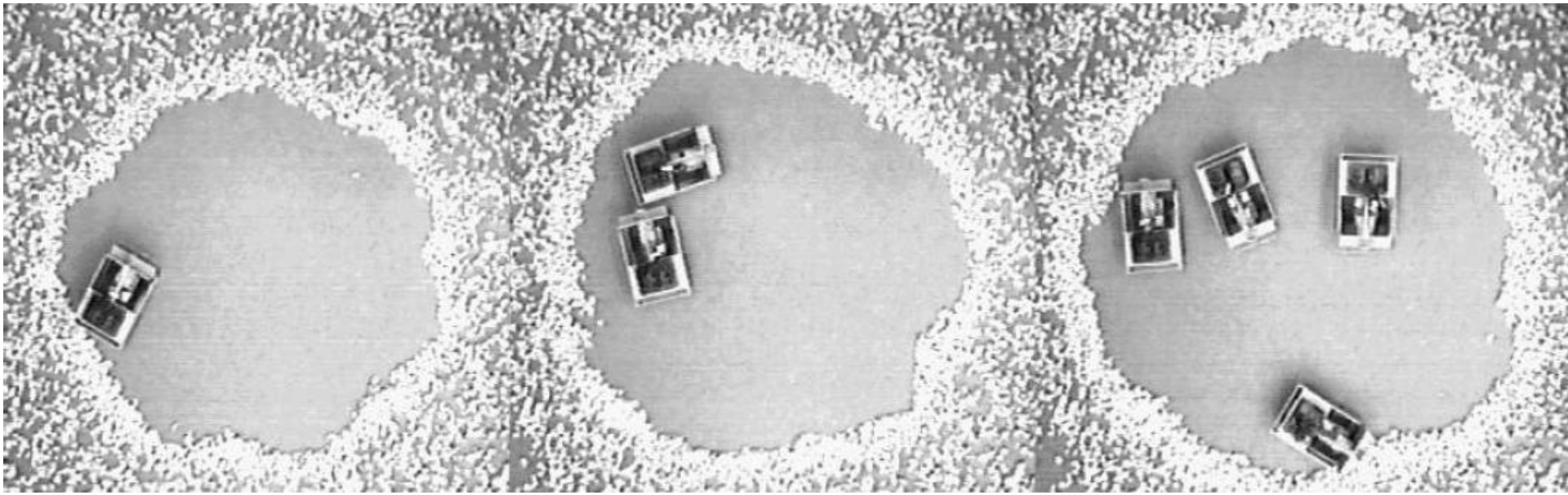


Parker et al (2003): Blind bulldozing: multiple robot nest construction

Forskjellige typer svermer

- Parallelt arbeid
  - Arbeid som kan utføres av én robot, men å bruke mange gjør at jobben utføres fortere
  - Overvåkning
  - Rydding

Helheten er mer enn summen av delene

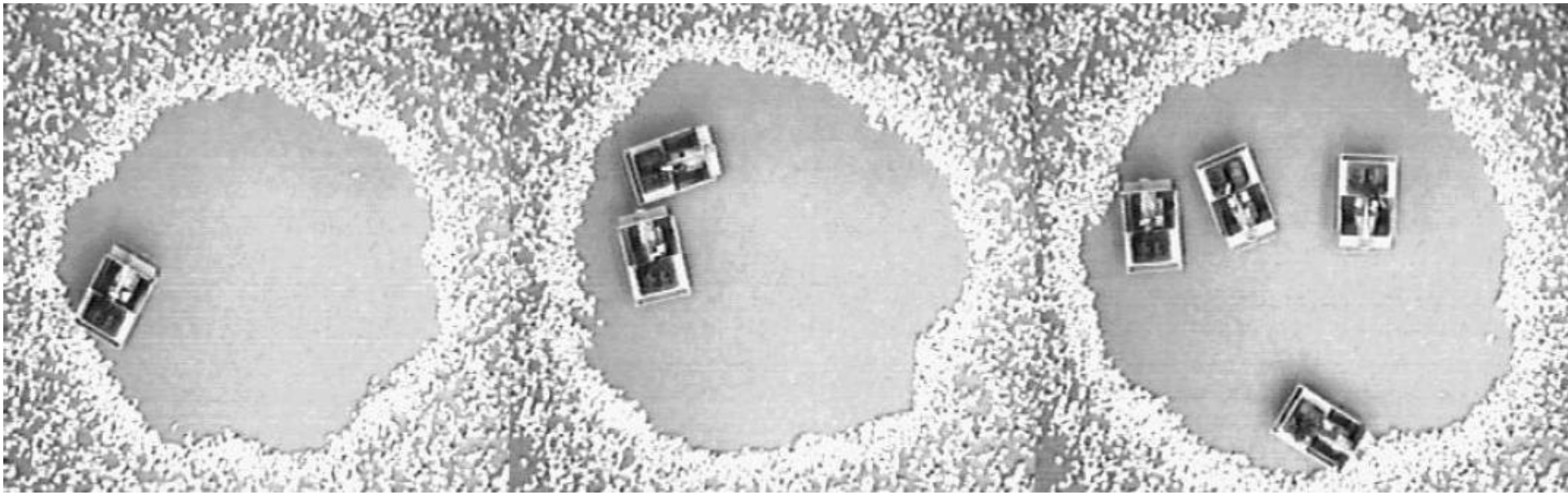


Parker et al (2003): Blind bulldozing: multiple robot nest construction

Forskjellige typer svermer

- Parallelt arbeid
  - Arbeid som kan utføres av én robot, men å bruke mange gjør at jobben utføres fortere
  - Overvåkning
  - Rydding

Helheten er ~~mer~~ mindre enn summen av delene!

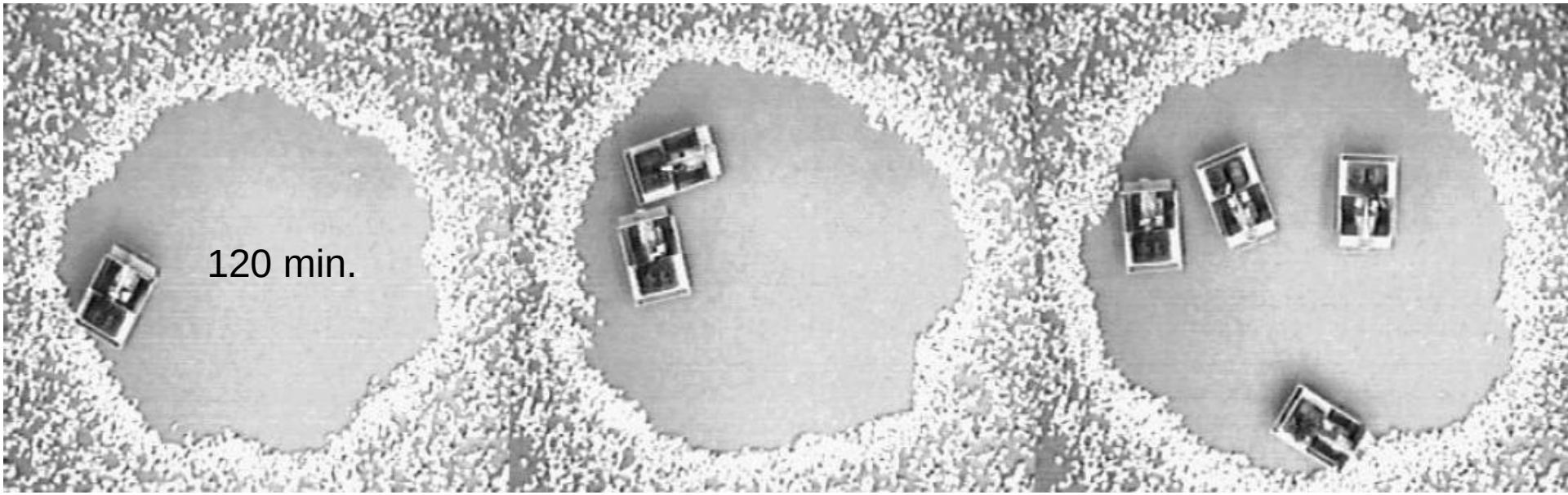


Parker et al (2003): Blind bulldozing: multiple robot nest construction

Forskjellige typer svermer

- Parallelt arbeid
  - Arbeid som kan utføres av én robot, men å bruke mange gjør at jobben utføres fortere
  - Overvåkning
  - Rydding

Helheten er ~~mer~~ mindre enn summen av delene!

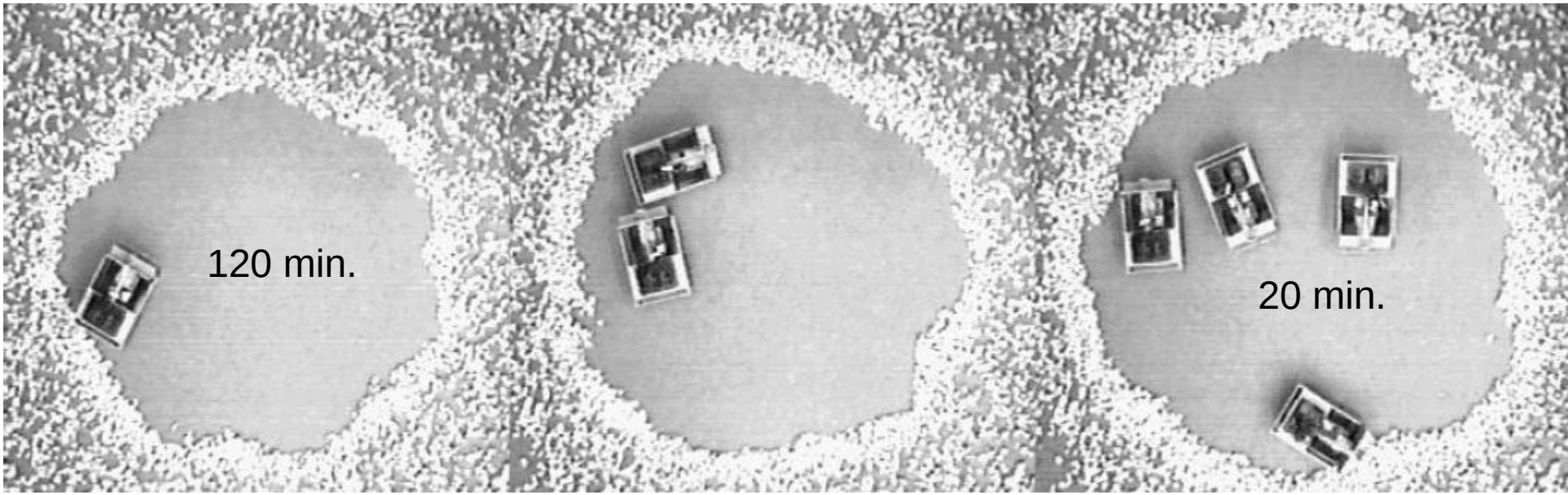


Parker et al (2003): Blind bulldozing: multiple robot nest construction

Forskjellige typer svermer

- Parallelt arbeid
  - Arbeid som kan utføres av én robot, men å bruke mange gjør at jobben utføres fortere
  - Overvåkning
  - Rydding

Helheten er ~~mer~~ mindre enn summen av delene!

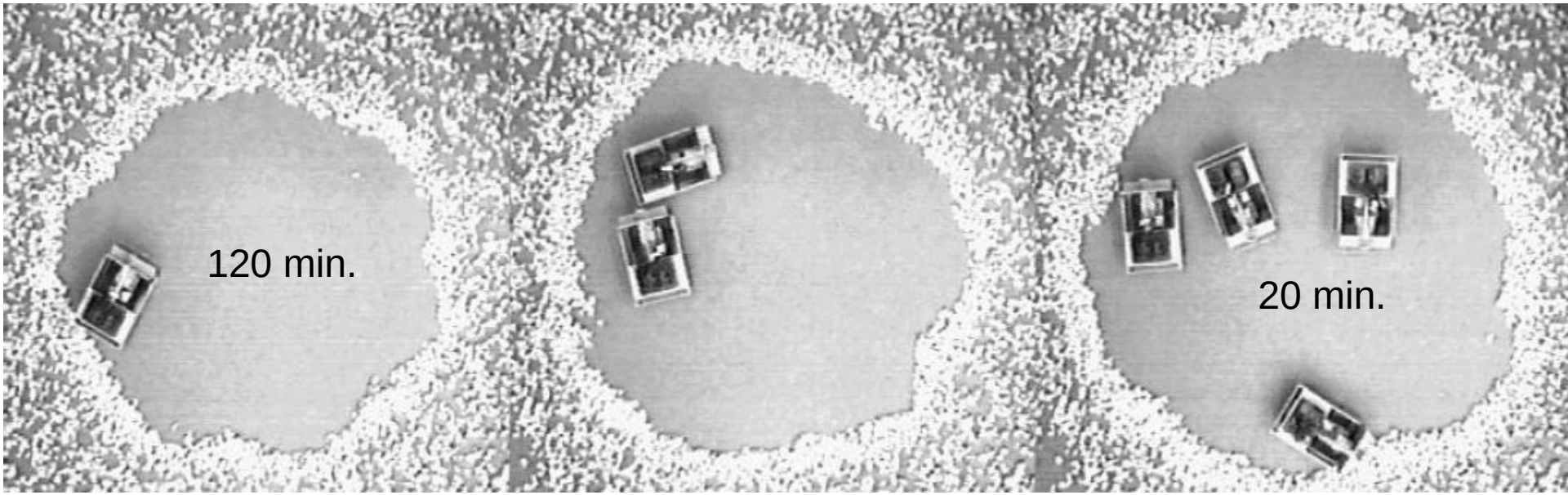


Parker et al (2003): Blind bulldozing: multiple robot nest construction

Forskjellige typer svermer

- Parallelt arbeid
  - Arbeid som kan utføres av én robot, men å bruke mange gjør at jobben utføres fortere
  - Overvåkning
  - Rydding

Helheten er ~~mindre~~ mer enn summen av delene



Parker et al (2003): Blind bulldozing: multiple robot nest construction

Forskjellige typer svermer

- Parallelt arbeid
- Gruppearbeid

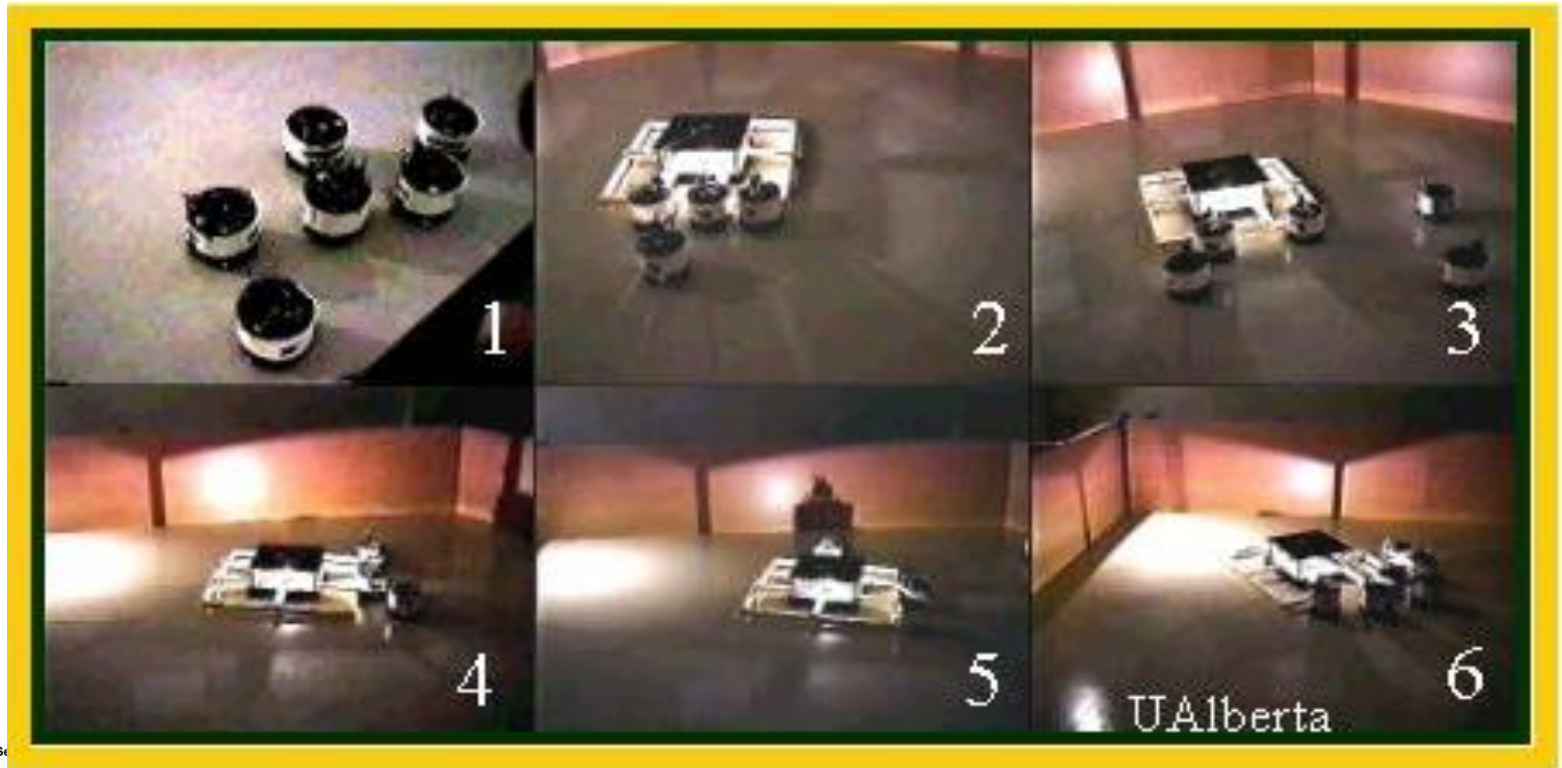
Forskjellige typer svermer

- Parallelt arbeid
- Gruppearbeid
  - Alle robotene bidrar med samme type arbeid
  - Krav om samtidighet

Forskjellige typer svermer

- Parallelt arbeid
- Gruppearbeid
  - Alle robotene bidrar med samme type arbeid
  - Krav om samtidighet

Kube (1997): “Task modelling in collective robotics”



KONGSBERG PROPRIETARY – Se

Forskjellige typer svermer

- Parallelt arbeid
- Gruppearbeid
- Teamarbeid

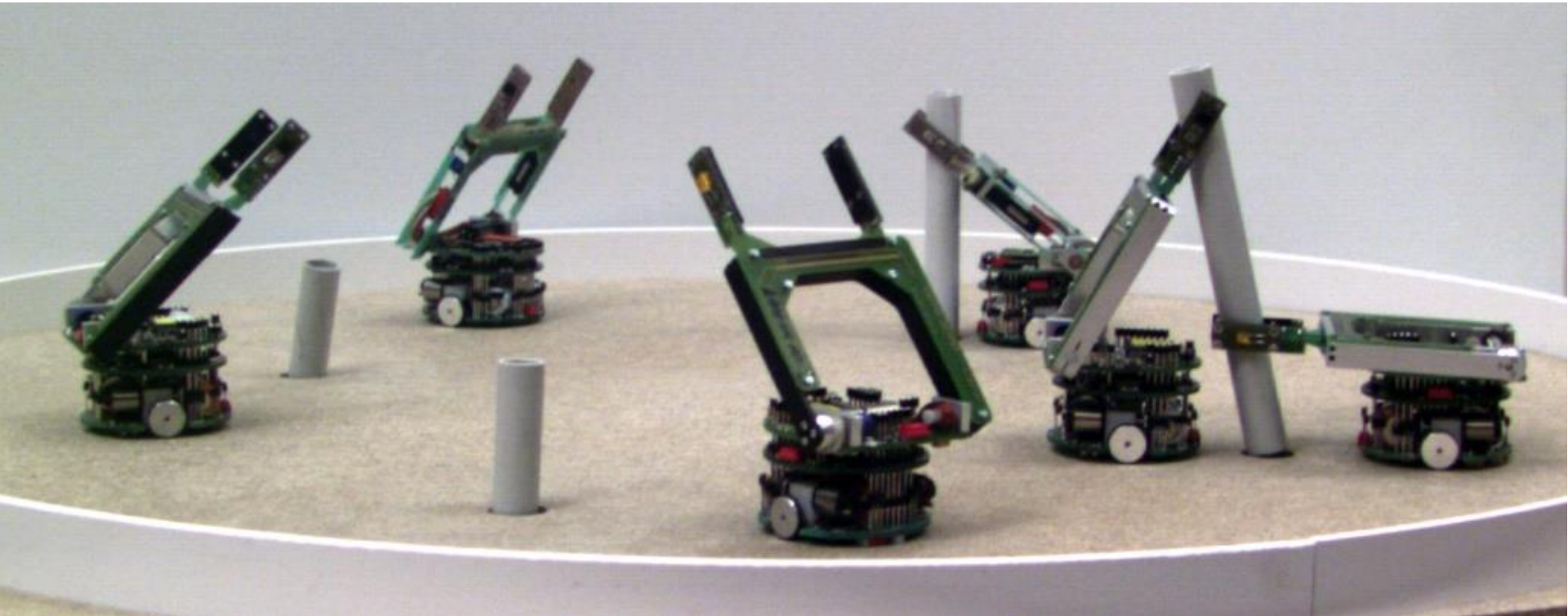
Forskjellige typer svermer

- Parallelt arbeid
- Gruppearbeid
- Teamarbeid
  - Robotene bidrar med forskjellig type arbeid
  - Krav om samtidighet

Forskjellige typer svermer

- Parallelt arbeid
- Gruppearbeid
- Teamarbeid
  - Robotene bidrar med forskjellige type arbeid
  - Krav om samtidighet

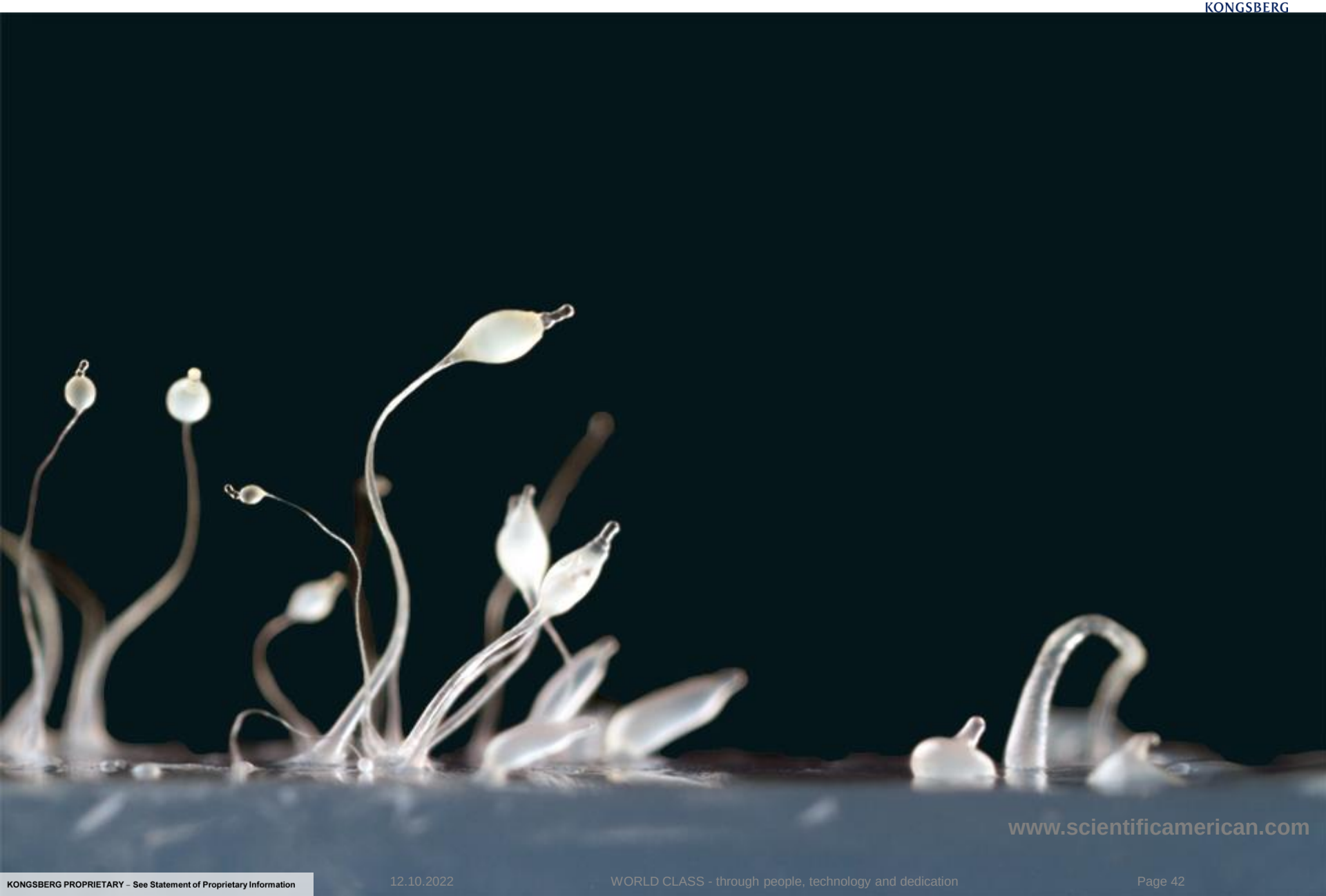
Ijspeert et al (2001): ”Collaboration through the exploitation of local interactions in autonomous collective robotics: The stick pulling experiment”



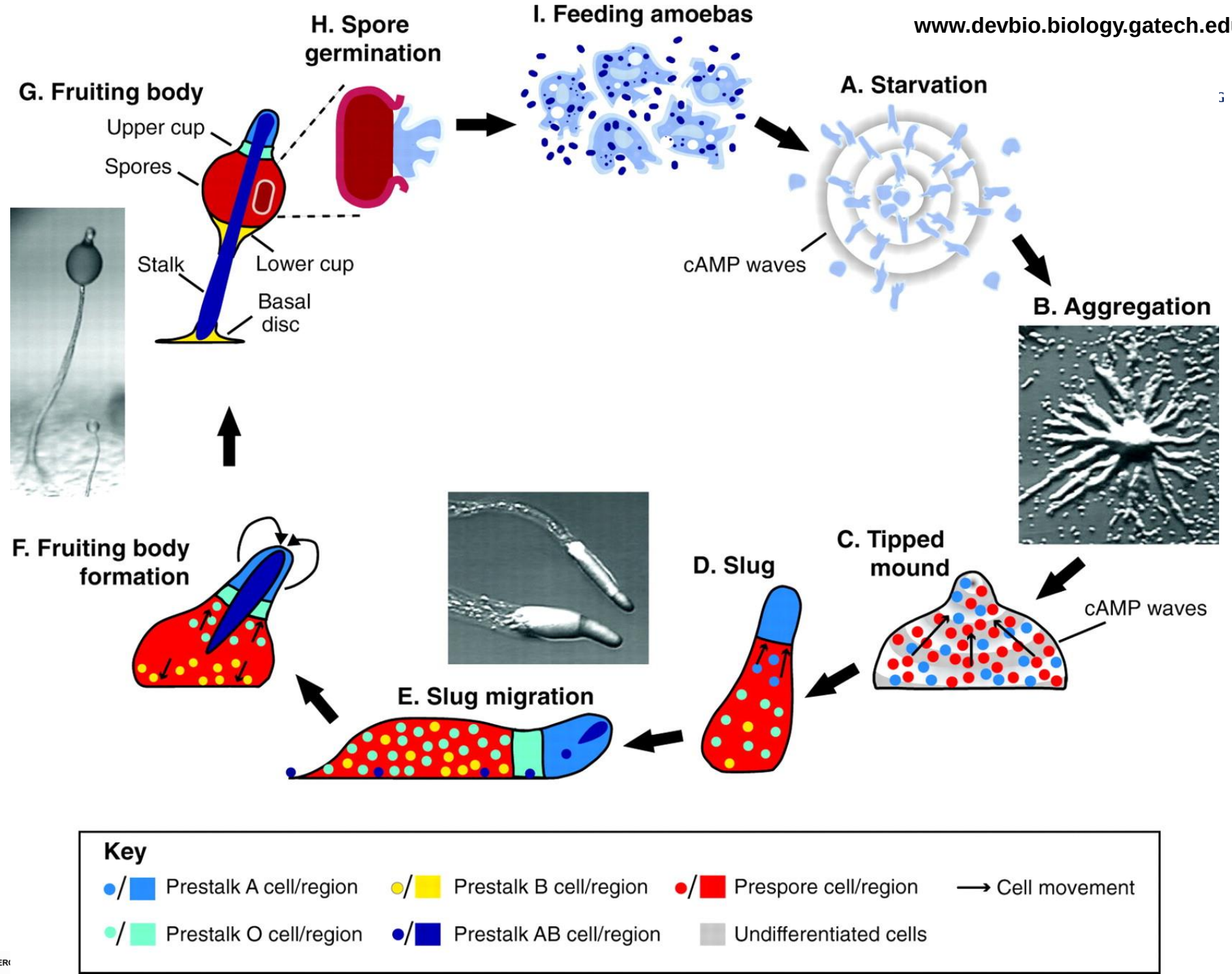
Helheten er annerledes enn summen av delene

Et biologisk eksempel

Dictyostelium Discoideum



www.devbio.biology.gatech.edu



KONGSBER

Dictyostelium Discoideum



Dr. Dirk Dormann, MRC Clinical Sciences Centre, Imperial College London. Nikon photo competetion.

Dictyostelium Discoideum



Credit: Carolina Biologica/Visuals Unlimited/Corbis

Fototaxis med roboter

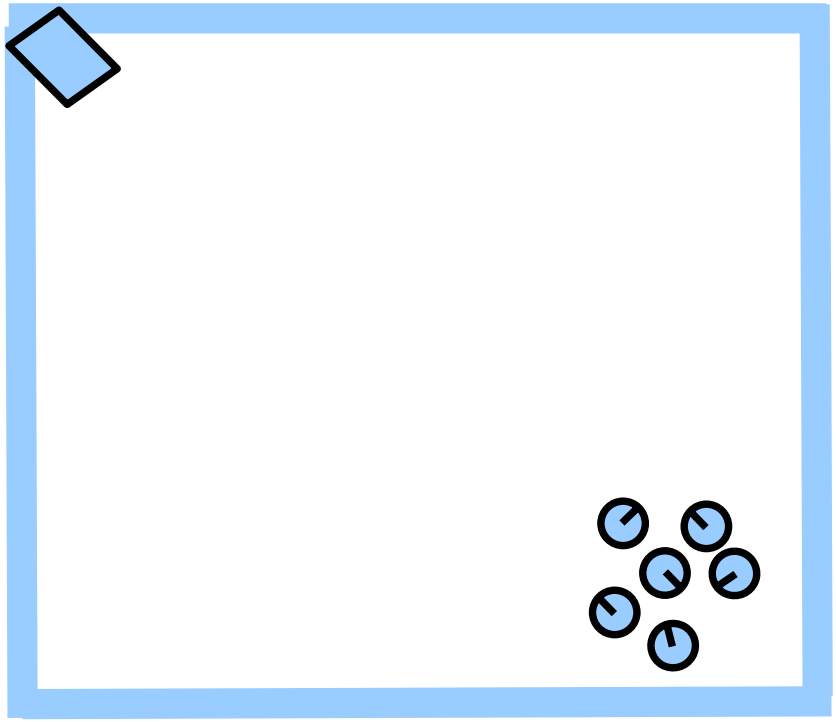
Vi vil at svermen med roboter skal:

- Holde seg samlet i en gruppe
- Bevege seg mot en lyskilde

Fototaxis med roboter

Vi vil at svermen med roboter skal:

- Holde seg samlet i en gruppe
- Bevege seg mot en lyskilde



Fototaxis med roboter

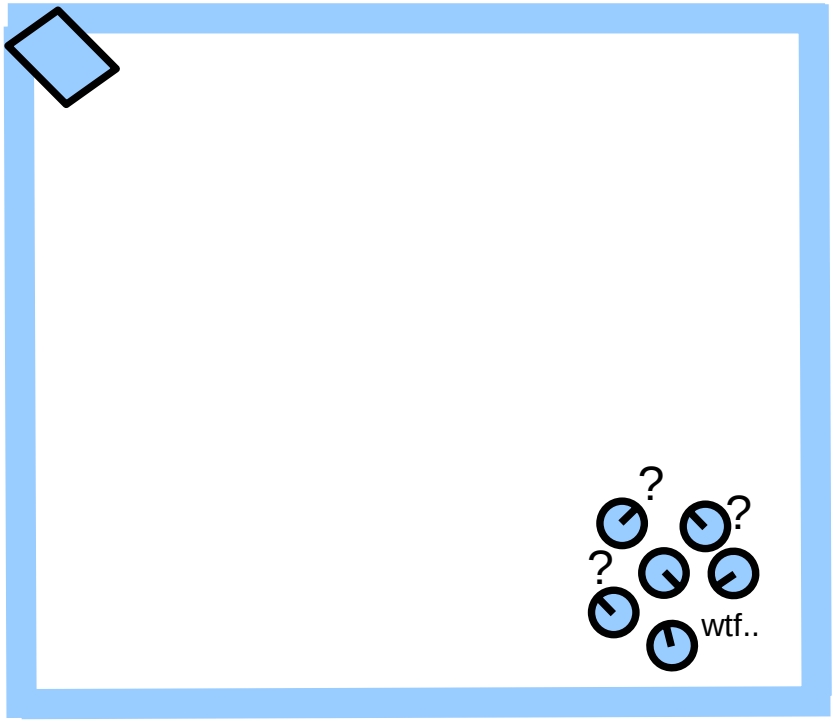
Vi vil at svermen med roboter skal:

- Holde seg samlet i en gruppe
- Bevege seg mot en lyskilde

Men

- Ingen av robotene vet hvor lyskilden er
- Ingen av robotene vet retningen til lyset

- Sensoren for lys er binær og upålitelig.



Fototaxis med roboter

Vi vil at svermen med roboter skal:

- Holde seg samlet i en gruppe
- Bevege seg mot en lyskilde

Men

- Ingen av robotene vet hvor lyskilden er
- Ingen av robotene vet retningen til lyset

- Sensoren for lys er binær og upålitelig.

Hvordan vil svermen vår *skalere?*

- Hvordan vil hastigheten den beveger seg med endre seg ved endring i antall roboter?

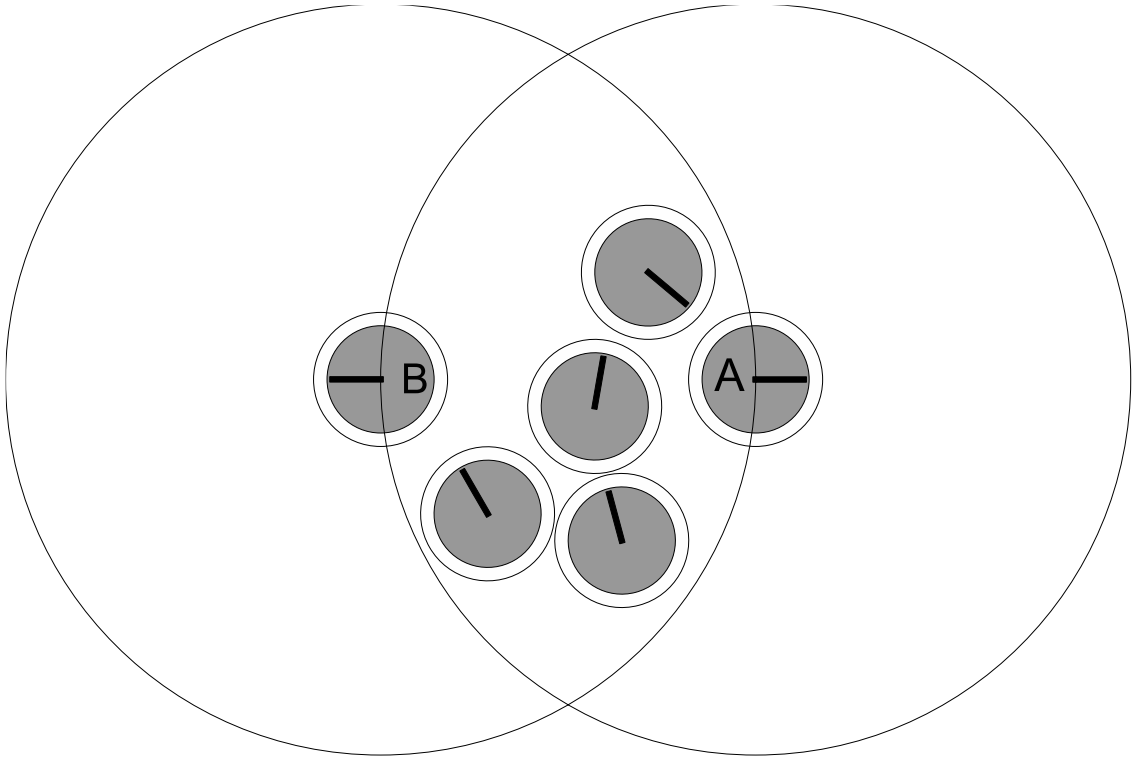
Algoritme #1 (av 2)

Algoritme #1 (av 2)

- Denne algoritmen lar seg ikke implementere i hardware, men har noen egenskaper som gjør at den er velegnet for analyse.

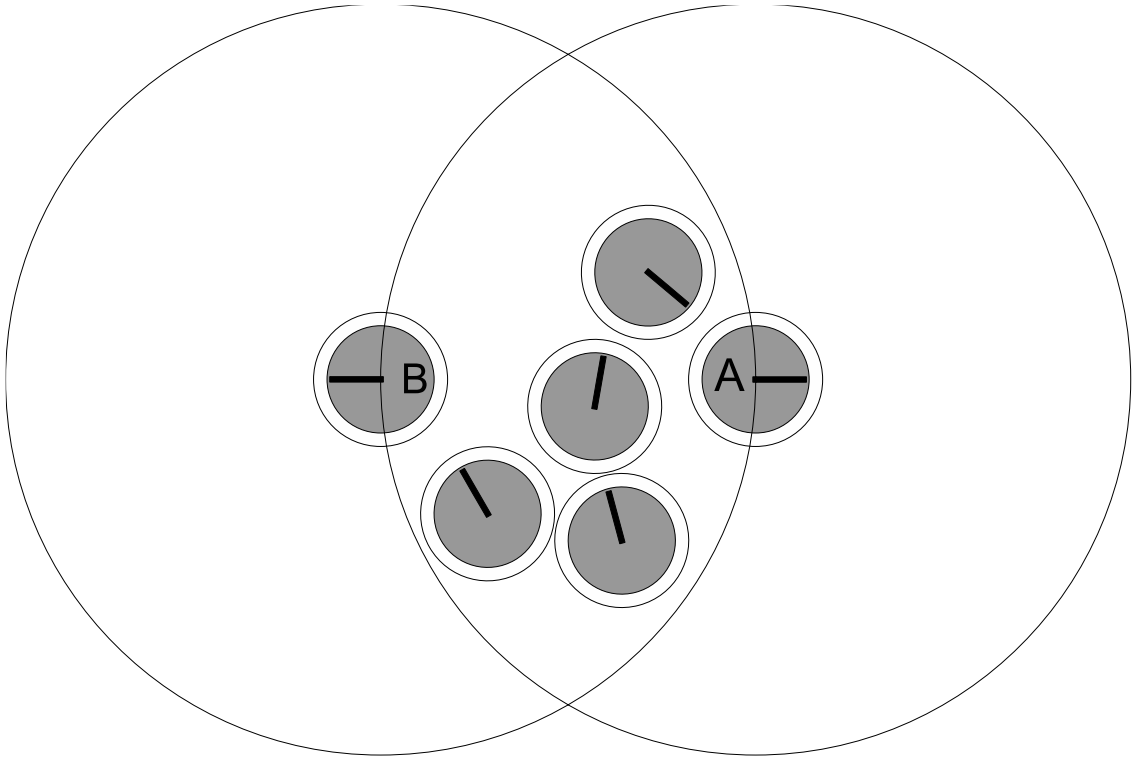
Algoritme #1 (av 2)

- Alle robotene følger de samme reglene:
  - Hvis roboter kommer for langt fra hverandre skal de snu



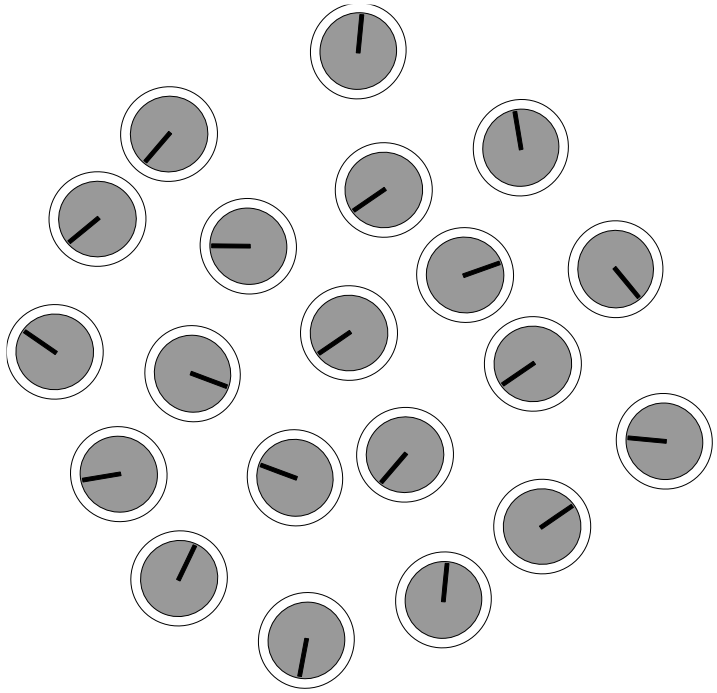
Algoritme #1 (av 2)

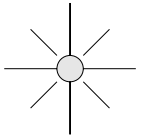
- Alle robotene følger de samme reglene:
  - Hvis roboter kommer for langt fra hverandre skal de snu
  - Hvis roboter kommer for nær hverandre skal de svinge unna



Algoritme #1 (av 2)

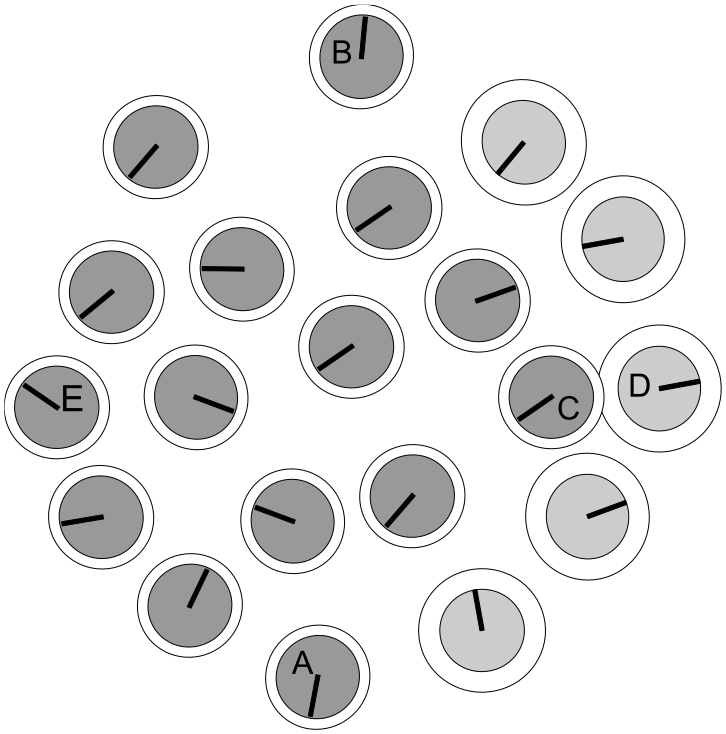
- Alle robotene følger de samme reglene:
  - Hvis roboter kommer for langt fra hverandre skal de snu
  - Hvis roboter kommer for nær hverandre skal de svinge unna
  - Vi må bryte symmetrien så robotene klarer å gå mot lyset

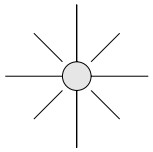




Algoritme #1 (av 2)

- Alle robotene følger de samme reglene:
  - Hvis roboter kommer for langt fra hverandre skal de snu
  - Hvis roboter kommer for nær hverandre skal de svinge unna
  - Vi må bryte symmetrien så robotene klarer å gå mot lyset





Algoritme #1 (av 2)

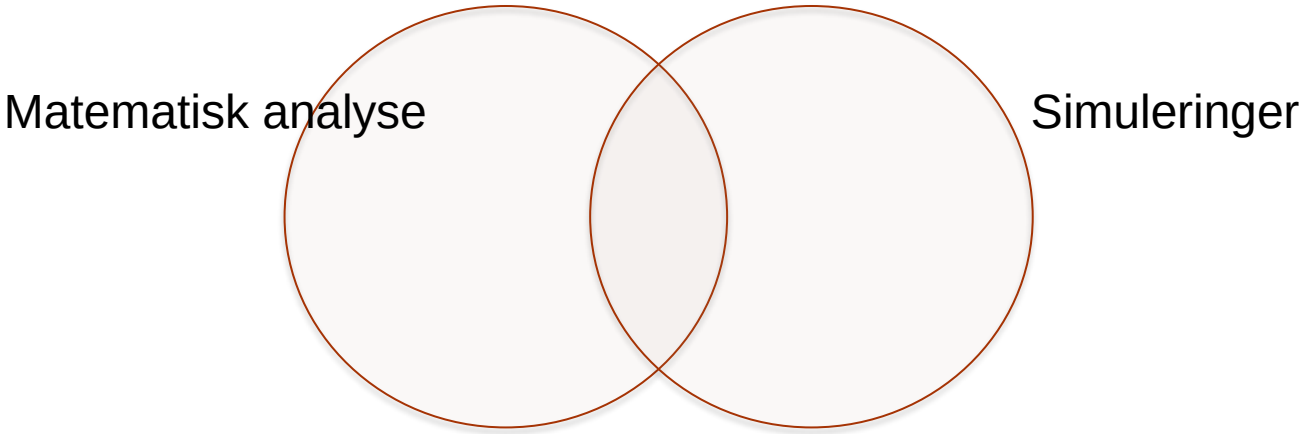
- Vi ser på en simulering i Netlogo.

Algoritme #1 (av 2)

- Hvordan forventer vi at denne algoritmen skalerer? Dvs, hvordan endrer hastigheten seg med antall roboter?

Algoritme #1 (av 2)

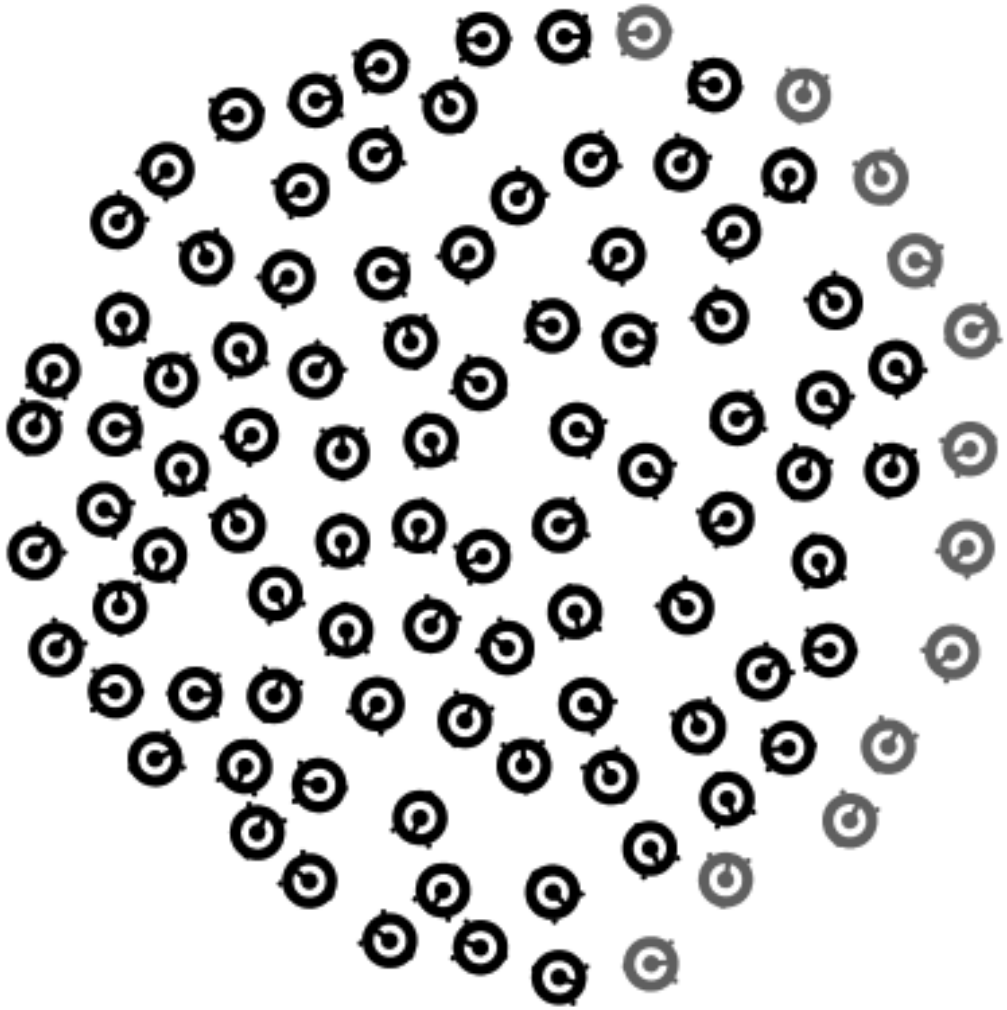
- Hvordan forventer vi at denne algoritmen skalerer? Dvs, hvordan endrer hastigheten seg med antall roboter?



Algoritme #1 (av 2)

- Hvordan forventer vi at denne algoritmen skalerer?

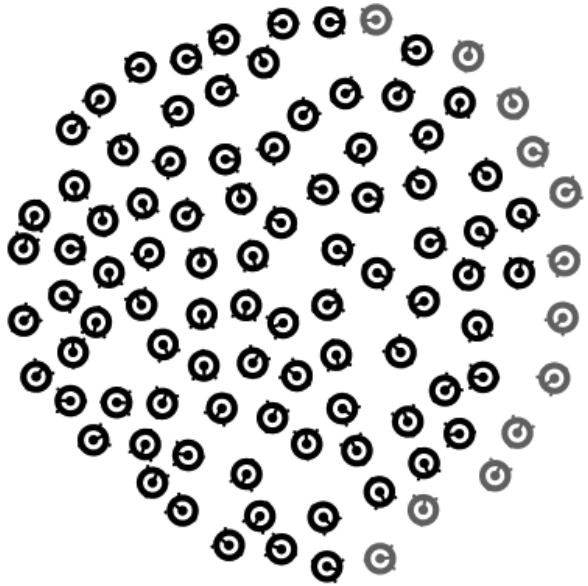




Algoritme #1 (av 2)

- Hvordan forventer vi at denne algoritmen skalerer?
- Vi holder tettheten til svermen konstant

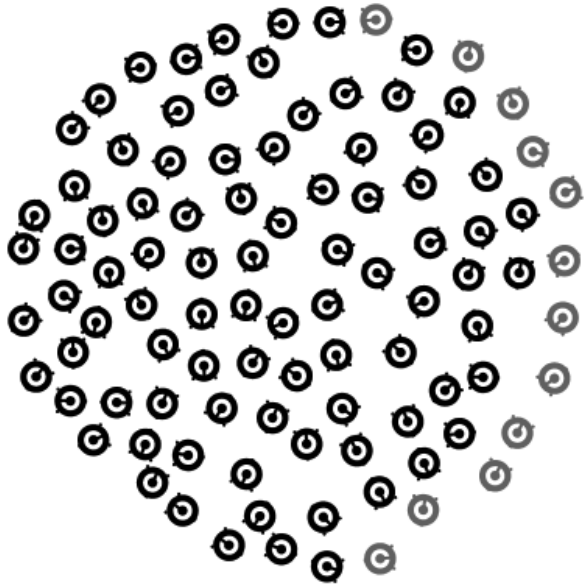




Algoritme #1 (av 2)

- Hvordan forventer vi at denne algoritmen skalerer?
- Vi holder tettheten til svermen konstant
- Vi legger merke til at opplyste roboter befinner seg langs kanten av svermen.

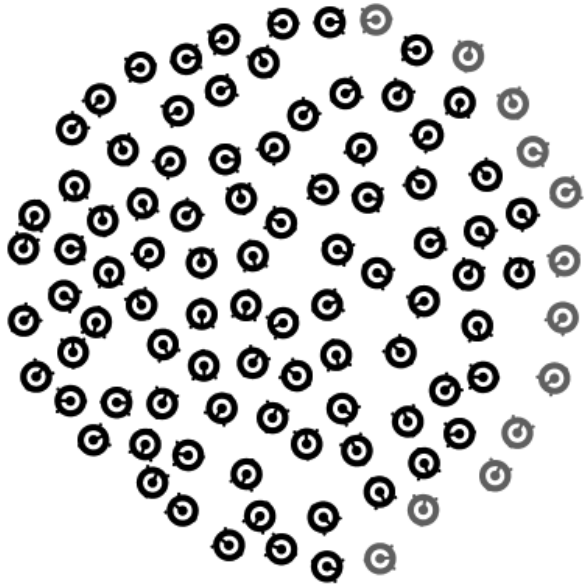




Algoritme #1 (av 2)

- Hvordan forventer vi at denne algoritmen skalerer?
- Vi holder tettheten til svermen konstant
- Vi legger merke til at opplyste roboter befinner seg langs kanten av svermen.
- Arealet som svermen dekker er proporsjonalt med N.

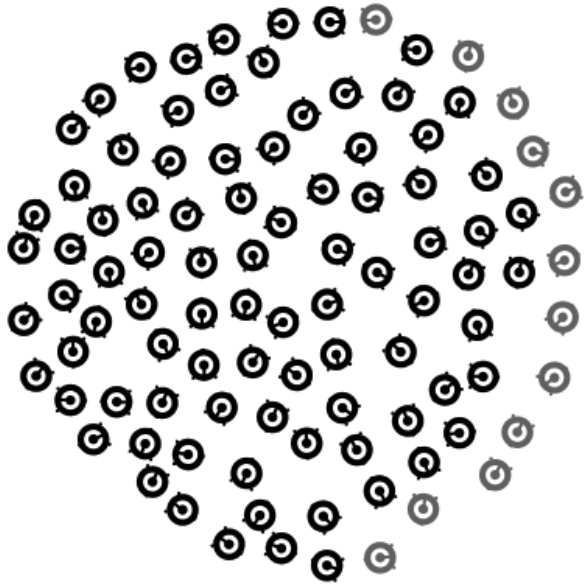




Algoritme #1 (av 2)

- Hvordan forventer vi at denne algoritmen skalerer?
- Vi holder tettheten til svermen konstant
- Vi legger merke til at opplyste roboter befinner seg langs kanten av svermen.
- Arealet som svermen dekker er proporsjonalt med N.
- Opplyste roboter er proporsjonalt med $N^{1/2}$

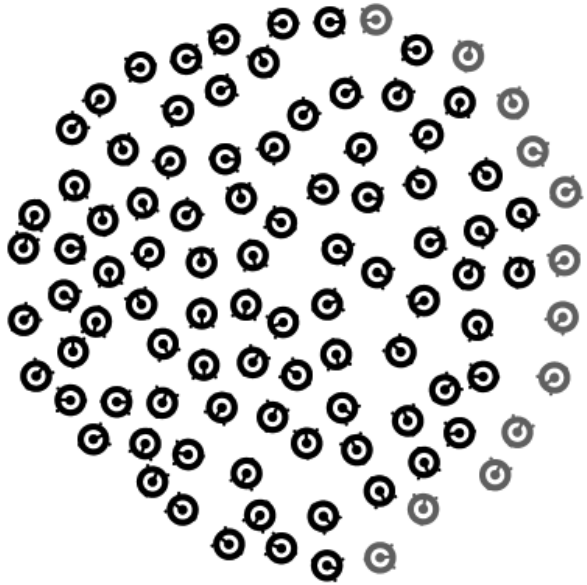




Algoritme #1 (av 2)

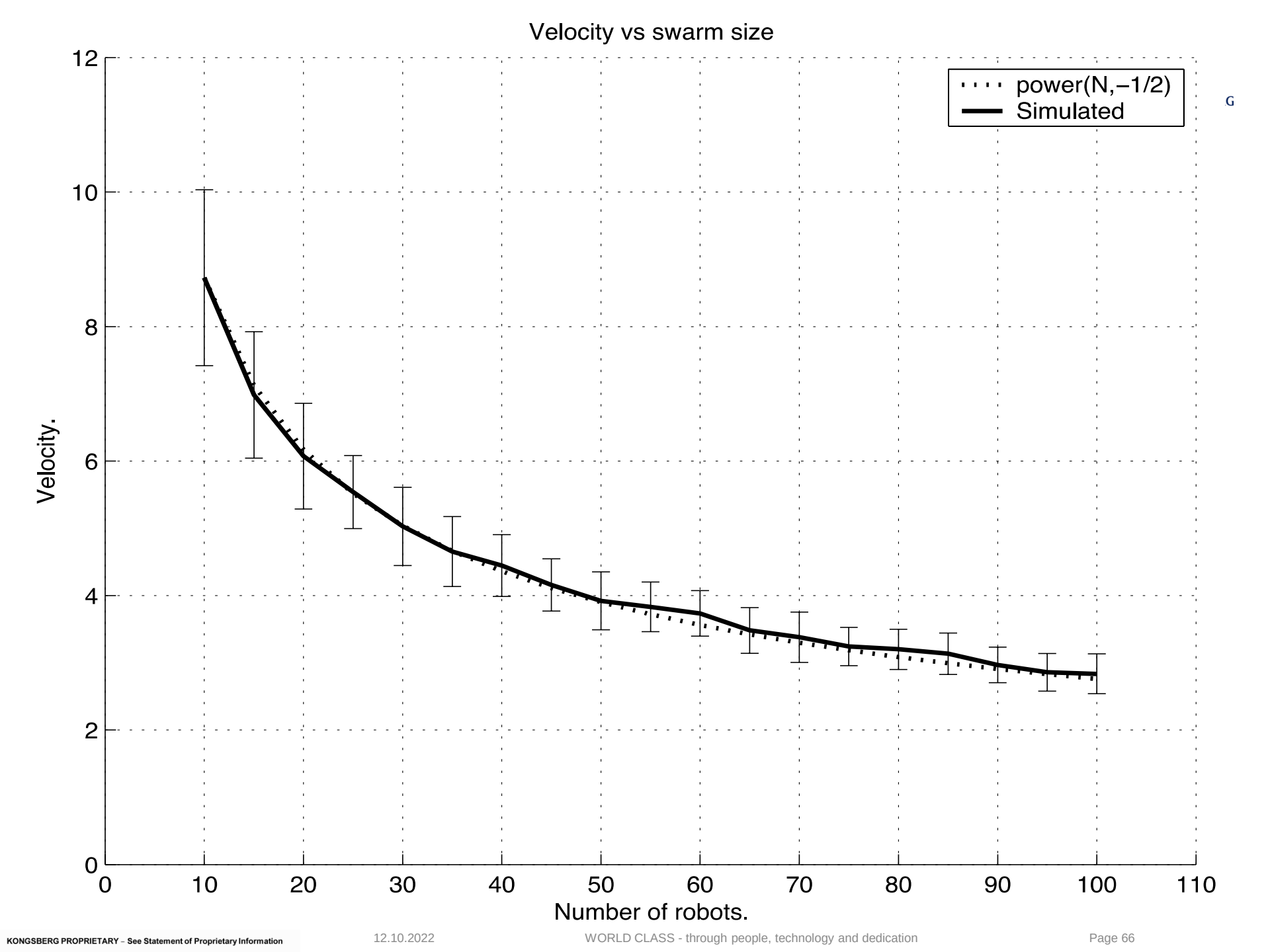
- Hvordan forventer vi at denne algoritmen skalerer?
- Vi holder tettheten til svermen konstant
- Vi legger merke til at opplyste roboter befinner seg langs kanten av svermen.
- Arealet som svermen dekker er proporsjonalt med N.
- Opplyste roboter er proporsjonalt med $N^{1/2}$
- Bidraget fra hver opplyst robot er 1/N





Algoritme #1 (av 2)

$$v_{swarm}(N) = c \times \frac{1}{\sqrt{N}}$$


Velocity vs swarm size
power(N,–1/2)
Simulated
Velocity.
Number of robots.
0
2
4
6
8
10
12
0
10
20
30
40
50
60
70
80
90
100
110

Algoritme #2 (av 2)

Algoritme #2 (av 2)

- Roboter som kommer for nære hverandre svinger unna

Algoritme #2 (av 2)

- Roboter som kommer for nære hverandre svinger unna
- Hvis roboter ikke har svingt unna andre roboter i løpet av en gitt tid, går de mot sentrum av svermen

Algoritme #2 (av 2)

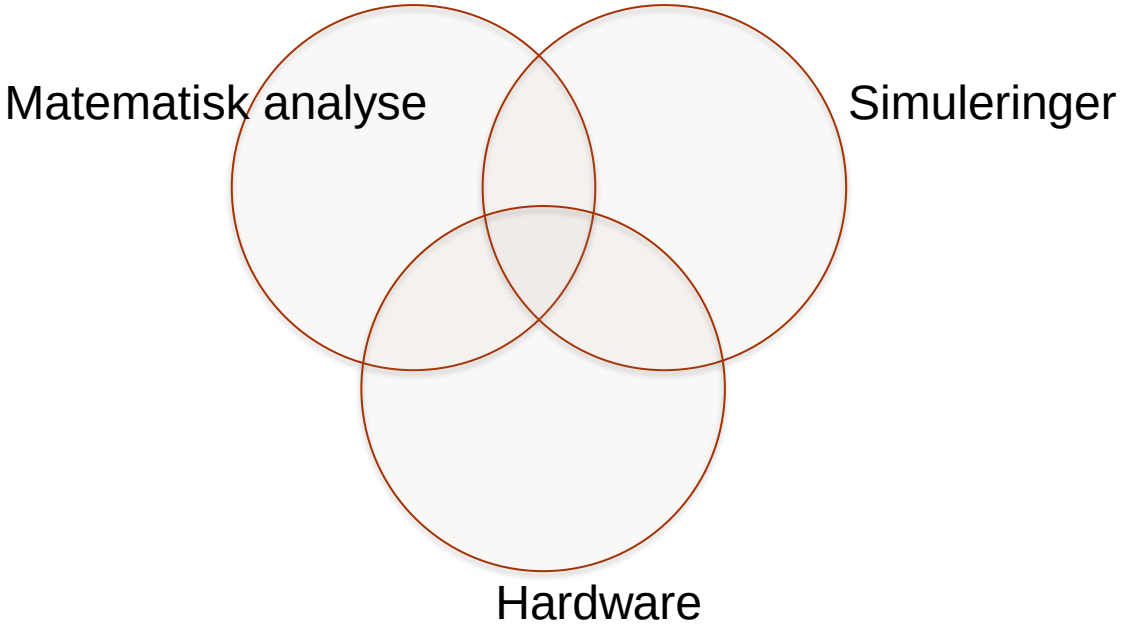
- Roboter som kommer for nære hverandre svinger unna
- Hvis roboter ikke har svingt unna andre roboter i løpet av en gitt tid, går de mot sentrum av svermen
- Symmetrien brytes ved at opplyste roboter har større kollisjonsradius

Algoritme #2 (av 2)

- Roboter som kommer for nære hverandre svinger unna
- Hvis roboter ikke har svingt unna andre roboter i løpet av en gitt tid, går de mot sentrum av svermen
- Symmetrien brytes ved at opplyste roboter har større kollisjonsradius
- Vanskelig å kontrollere morfologi, så skalerer ikke like perfekt som den første algoritmen.

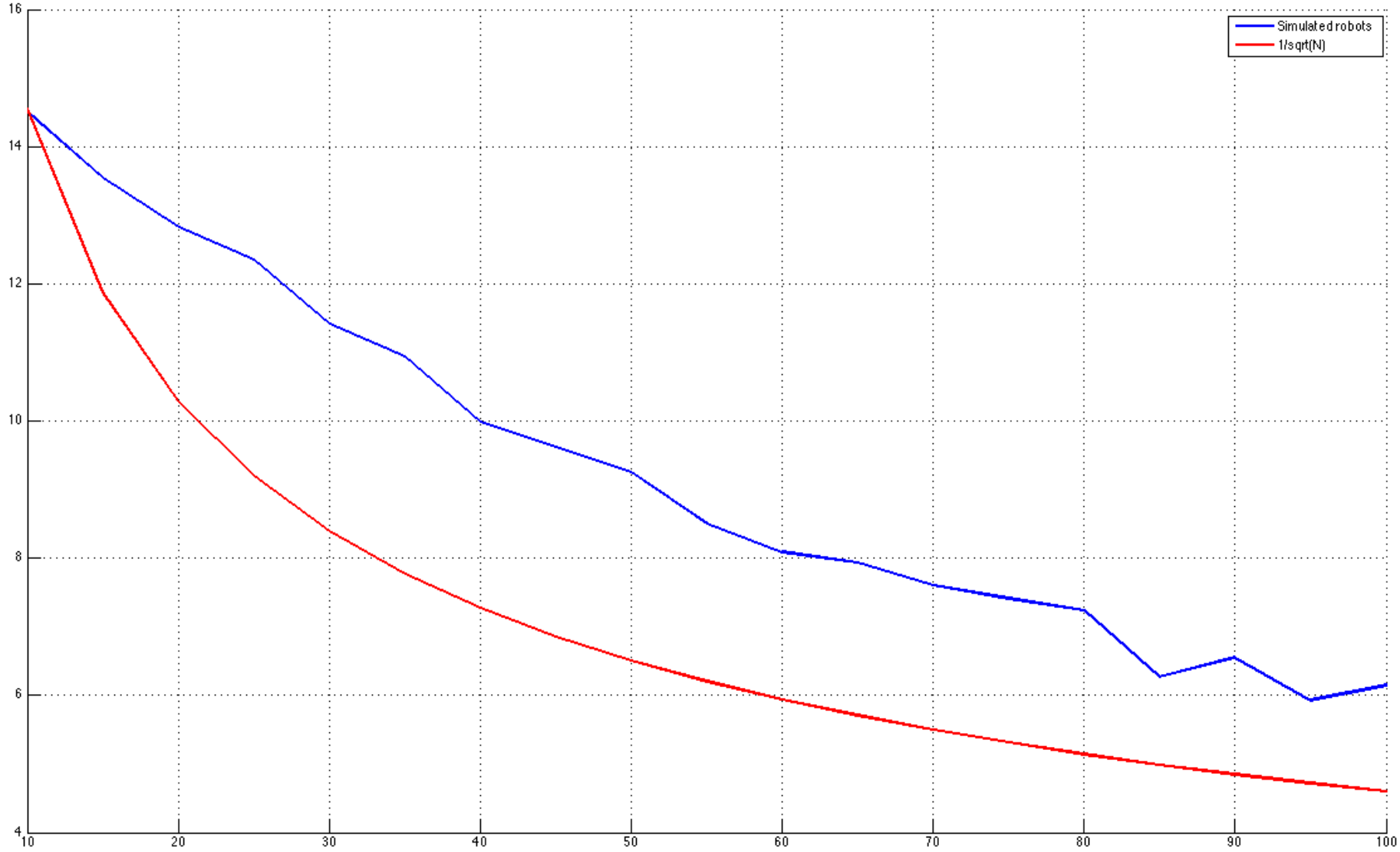
Algoritme #2 (av 2)

- Roboter som kommer for nære hverandre svinger unna
- Hvis roboter ikke har svingt unna andre roboter i løpet av en gitt tid, går de mot sentrum av svermen
- Symmetrien brytes ved at opplyste roboter har større kollisjonsradius
- Vanskelig å kontrollere morfologi, så skalerer ikke like perfekt som den første algoritmen.



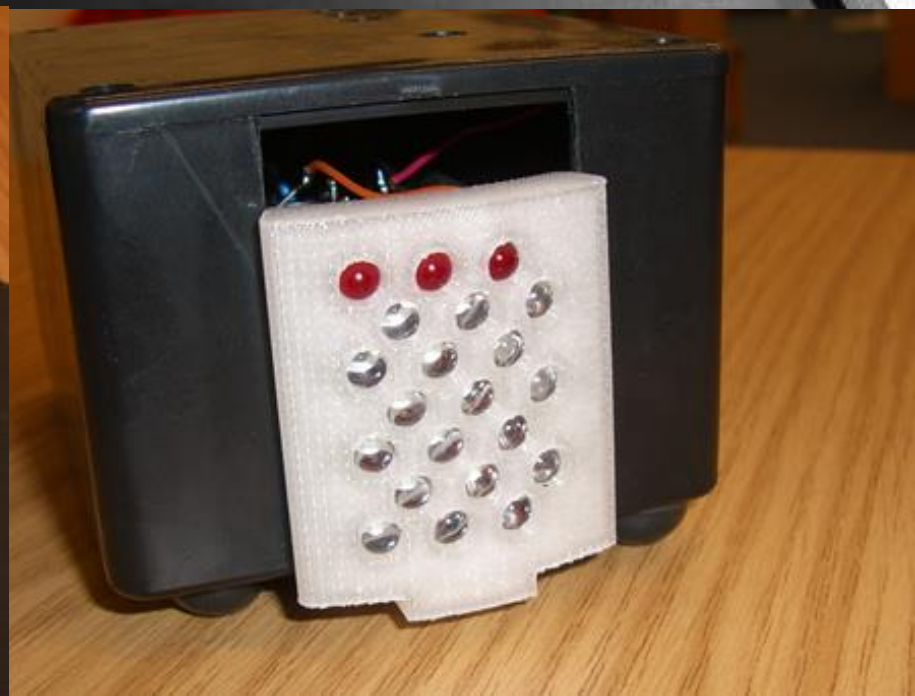
Algoritme #2 (av 2)

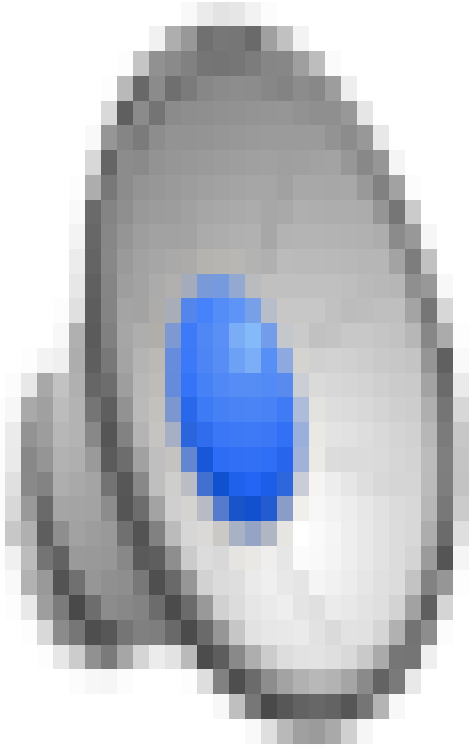
- Vi ser på en simulering i Netlogo.

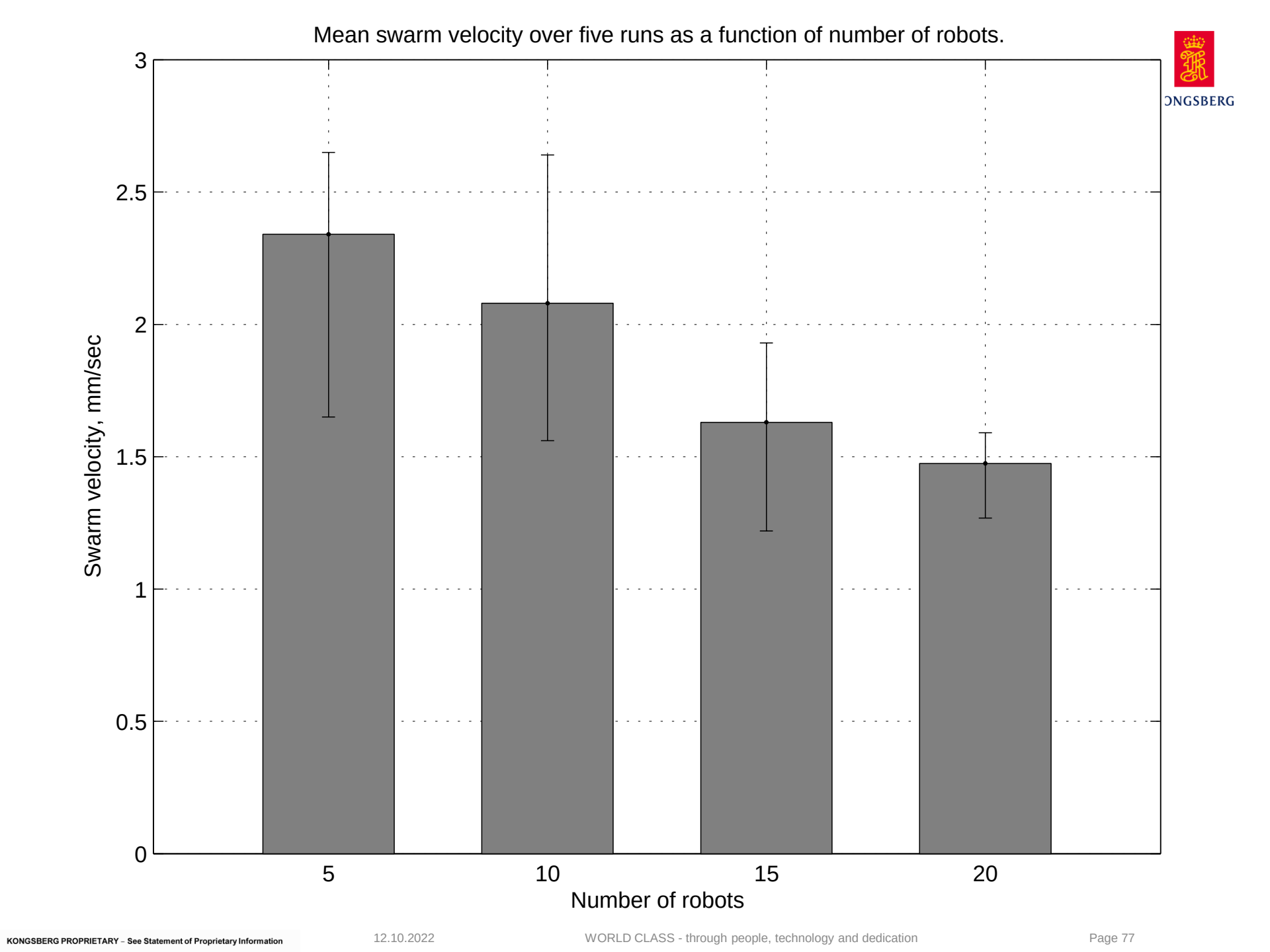

Simulated robots
1/sqrt(N)
16
14
12
10
8
6
4
10
20
30
40
50
60
70
80
90
100




1051



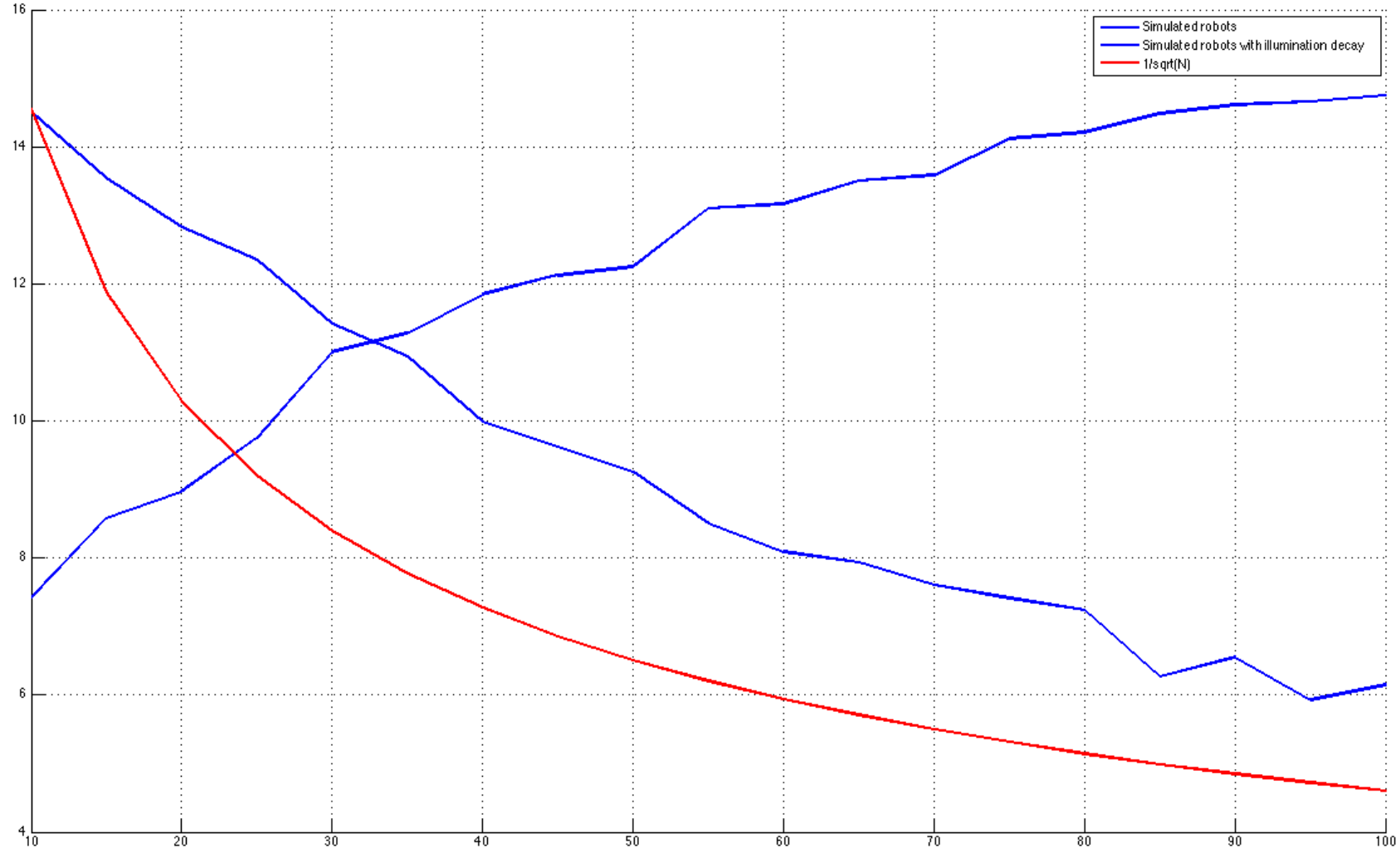



Mean swarm velocity over five runs as a function of number of robots.
Swarm velocity, mm/sec
Number of robots
0
0.5
1
1.5
2
2.5
3
5
10
15
20

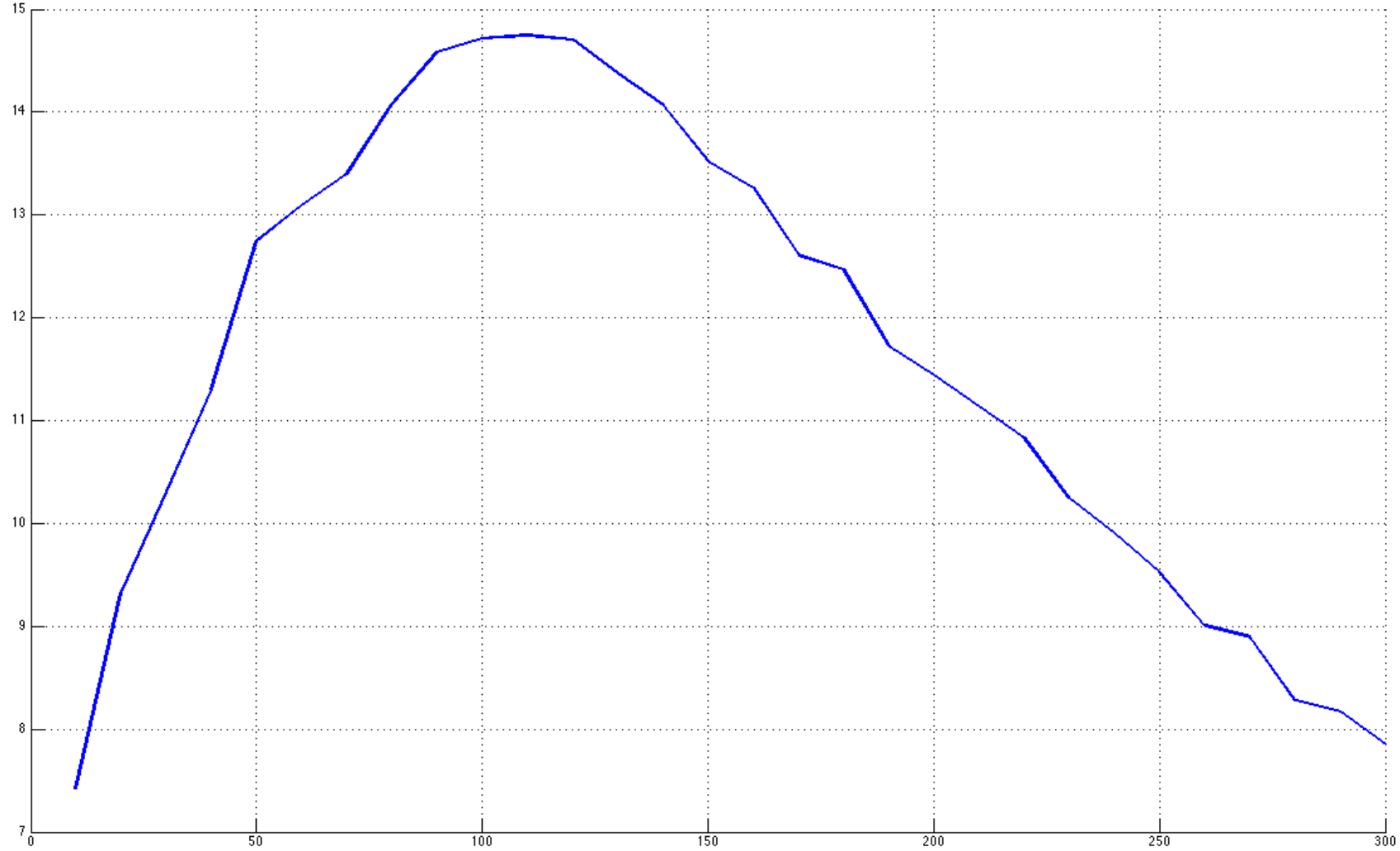
Utfordringen er å få relevant informasjon frem til alle robotene

Utfordringen er å få relevant informasjon frem til alle robotene

KONGSBERG



Utfordringen er å få relevant informasjon frem til alle robotene



Størrelse krever kompleksitet

Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis

Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
  - Robotene har sensorer som de bruker for å
    - Unngå å krasje i andre roboter
    - Finne igjen svermen når de har kjørt for langt bort

Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
  - Robotene har sensorer som de bruker for å
    - Unngå å krasje i andre roboter
    - Finne igjen svermen når de har kjørt for langt bort

  - Hva er konsekvensen dersom flere av robotene opplever en ødelagt sensor?

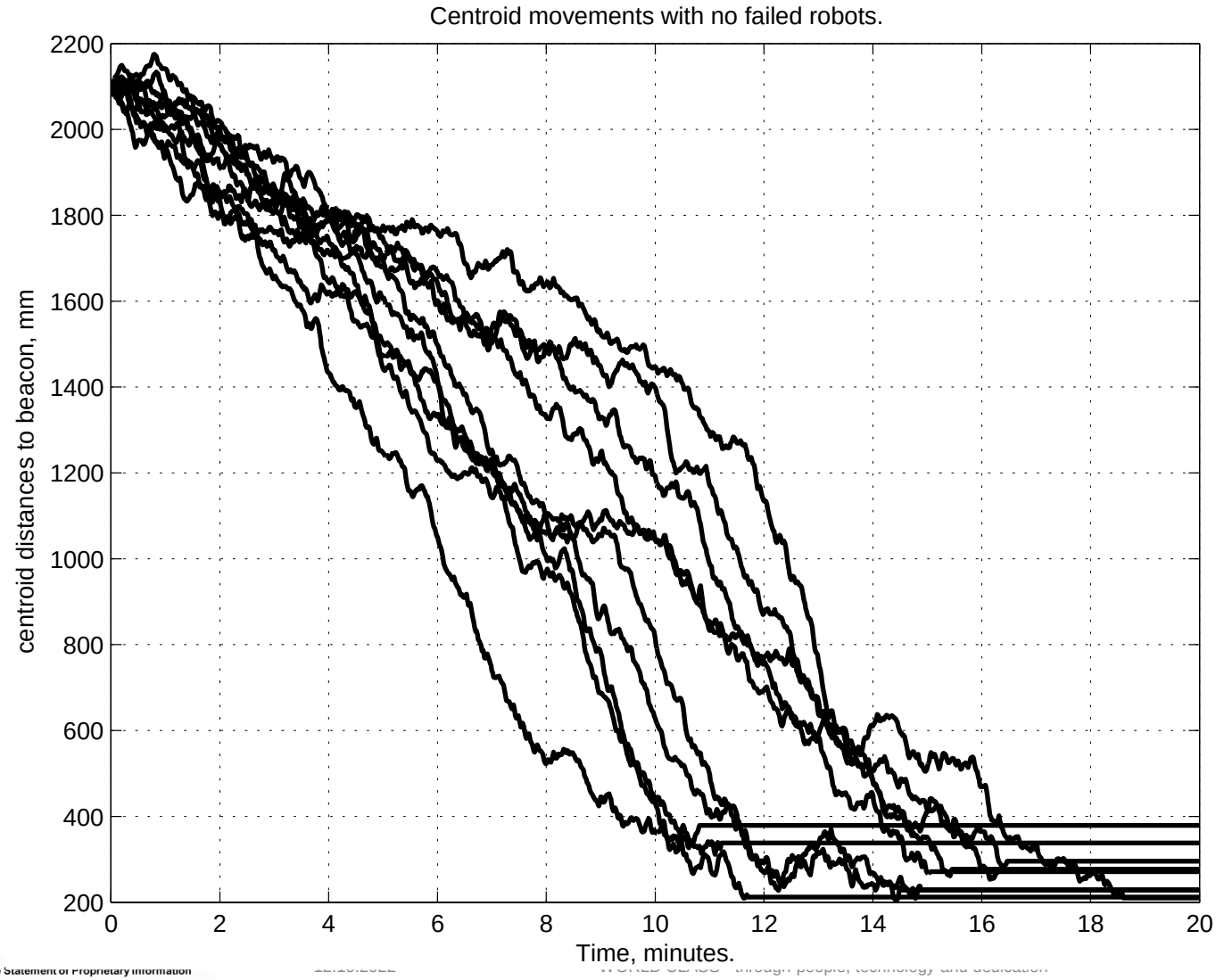
Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
  - Robotene har sensorer som de bruker for å
    - Unngå å krasje i andre roboter
    - Finne igjen svermen når de har kjørt for langt bort
  - Hva er konsekvensen om flere av robotene opplever en ødelagt sensor?
  - Vi kan måle det, men først må vi ha en referanse å måle mot.

Feiltoleranse.

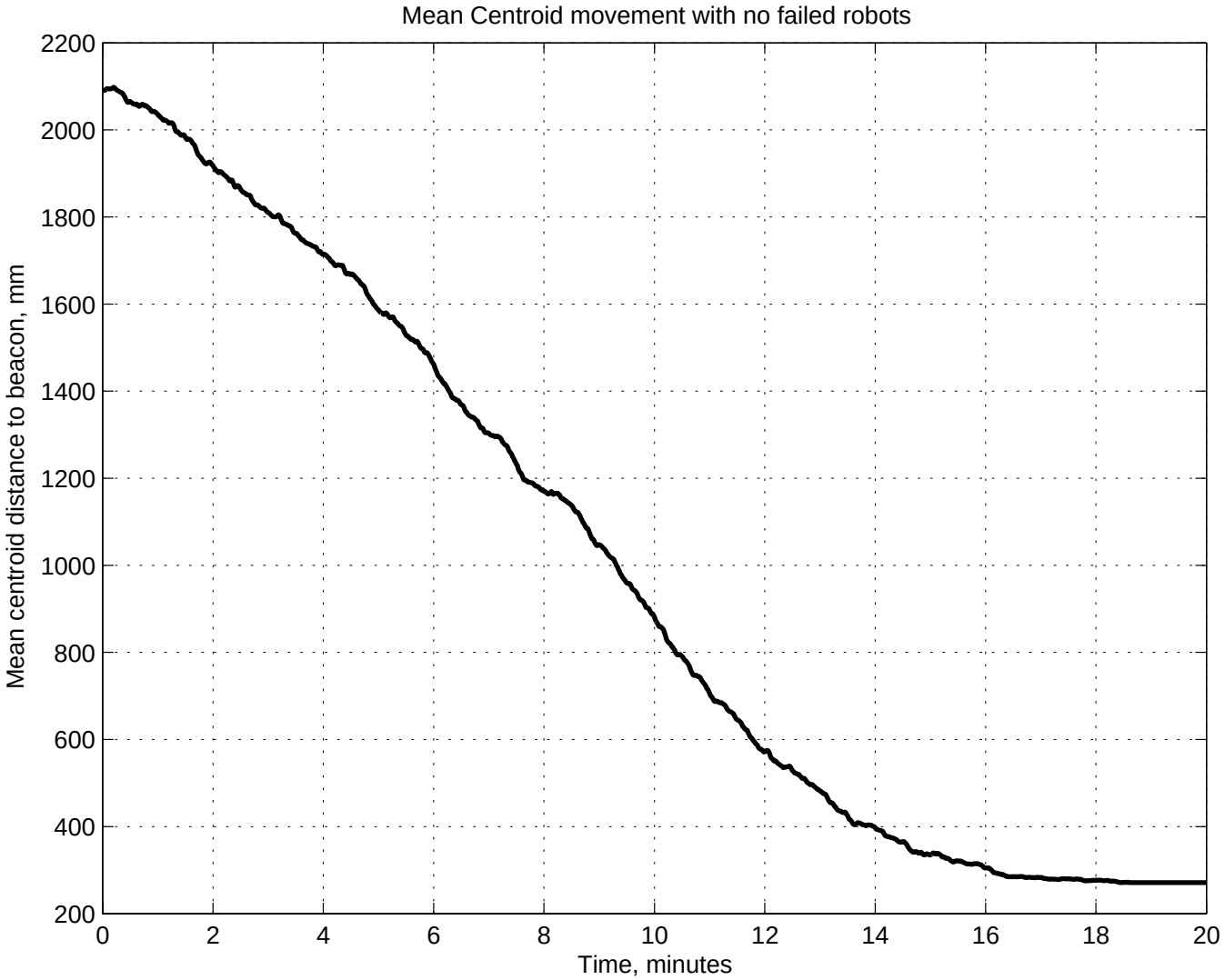
- Hastighet for 10 repetisjoner med fullt fungerende roboter.



KONGSBERG PROPRIETARY – See Statement of Proprietary Information

Feiltoleranse.

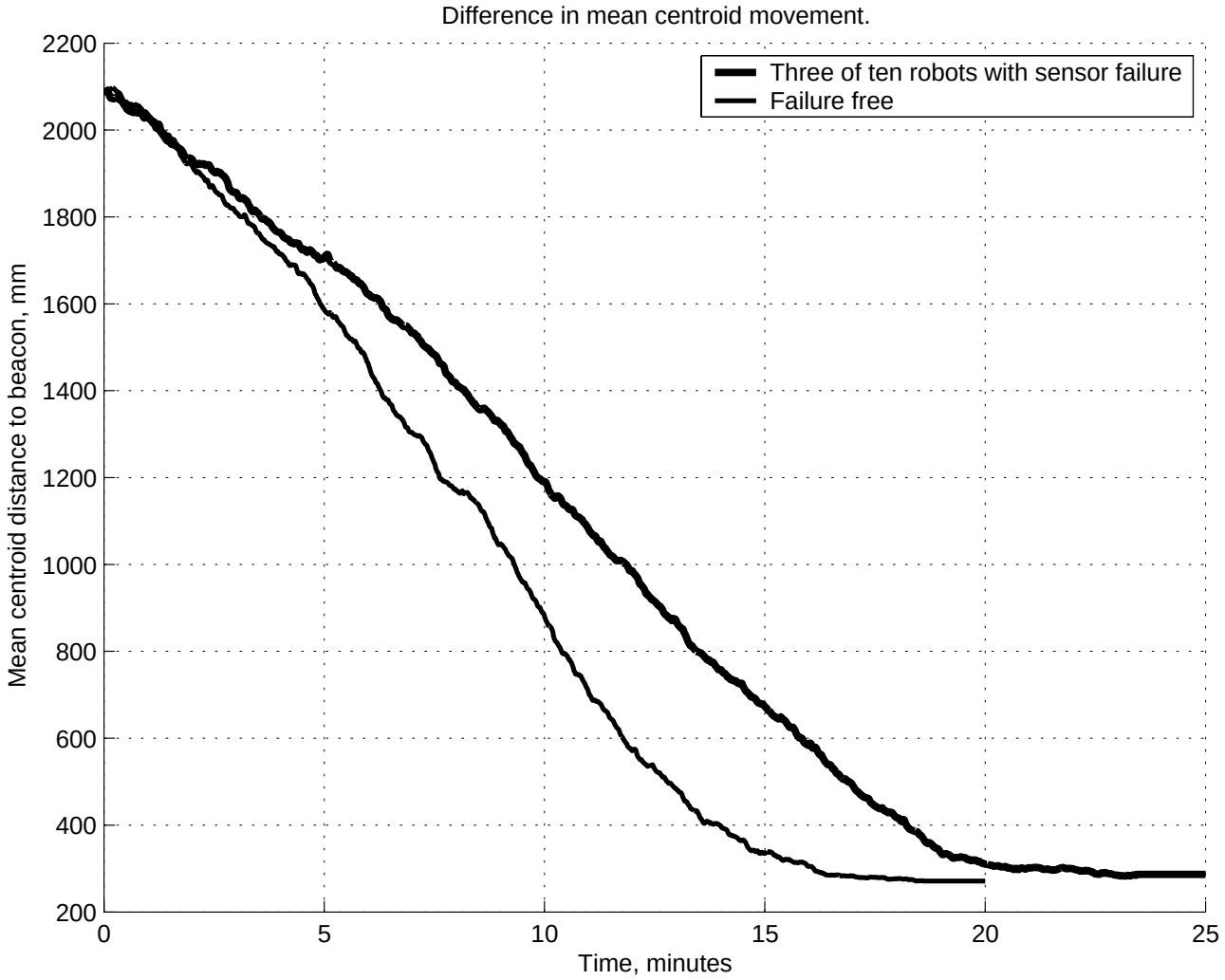
- Hastighet for 10 repetisjoner med fullt fungerende roboter.



KONGSBERG PROPRIETARY – See Statement of Proprietary Information

Feiltoleranse.

- Fungerende roboter vs roboter med sensorfeil



Feiltoleranse.

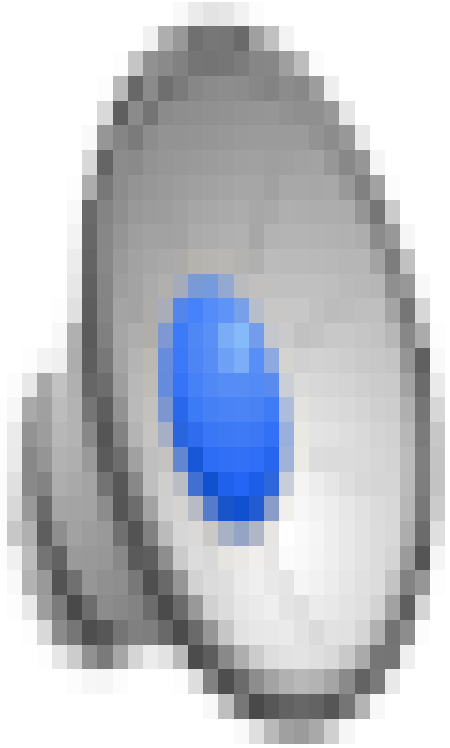
- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
- F2: Døde roboter.

Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
- F2: Døde roboter.
  - Vi antar at et lite antall døde roboter vil midlertidig forsinke svermen, inntil de døde robotene er forlatt.



Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
- F2: Døde roboter.
  - Vi antar at et lite antall døde roboter vil midlertidig forsinke svermen, inntil de døde robotene er forlatt.
  - Vi ser at dersom to av robotene i svermen dør, vil svermen klare å *reparere seg selv*.

Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
- F2: Døde roboter.
  - Vi antar at et lite antall døde roboter vil midlertidig forsinke svermen, inntil de døde robotene er forlatt.
  - Vi ser at dersom to av robotene i svermen dør, vil svermen klare å *reparere seg selv*.
  - Tiden det tar å reparere svermen ved to komplette feil varierer mellom 3 og 11 minutter (ved 10 repetisjoner).

Feiltoleranse.

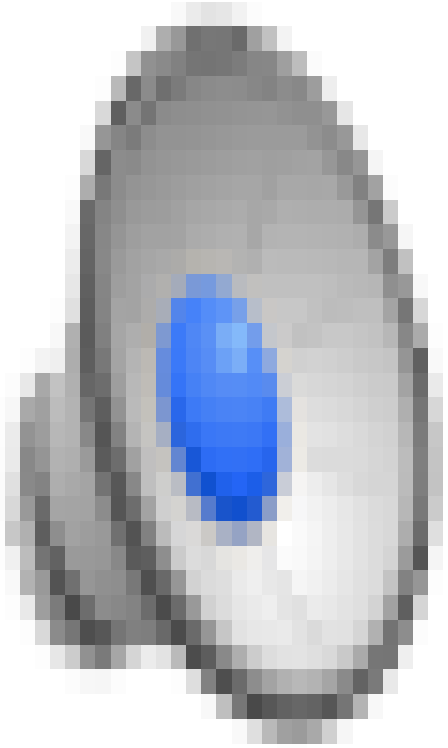
- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
- F2: Døde roboter.
- F3: Roboter med ødelagte motorer, men ellers velfungerende.

Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
- F2: Døde roboter.
- F3: Roboter med ødelagte motorer, men ellers velfungerende.
  - Vi antar at et lite antall delvis ødelagte roboter vil ha katastrofale følger for svermen. Roboter med ødelagte motorer vil potensielt forankre svermen, og forhindre den i å bevege seg videre.



Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
- F2: Døde roboter.
- F3: Roboter med ødelagte motorer, men ellers velfungerende.
  - Vi antar at et lite antall delvis ødelagte roboter vil ha katastrofale følger for svermen. Roboter med ødelagte motorer vil potensielt forankre svermen, og forhindre den i å bevege seg videre.
  - Med to ut av ti roboter delvis ødelagt er svermen for alle praktiske formål ubrukelig.

Feiltoleranse.

- Vi starter med en FMEA –Failure Mode and Effect Analysis
  - Vi tenker på forskjellige feil som kan oppstå, og prøver å forstå effekten det vil ha på hele svermen.

- F1: Sensorfeil.
- F2: Døde roboter.
- F3: Roboter med ødelagte motorer, men ellers velfungerende.
  - Vi antar at et lite antall delvis ødelagte roboter vil ha katastrofale følger for svermen. Roboter med ødelagte motorer vil potensielt forankre svermen, og forhindre den i å bevege seg videre.
  - Med to ut av ti roboter delvis ødelagt er svermen for alle praktiske formål ubrukelig.
  - Av ti eksperimenter klarte bare svermen å komme seg løs i seks av tilfellene. I de fire andre eksperimentene sto svermen fast til batteriene gikk tomme.

Feiltoleranse.

- Så langt har vi sett på tre forskjellige typer feil som kan oppstå, og konsekvensen det har for svermens ytelse.

Feiltoleranse.

- Så langt har vi sett på tre forskjellige typer feil som kan oppstå, og konsekvensen det har for svermens ytelse.
- Basert på eksperimentene vi har sett så langt, la oss gjøre en mer formell analyse av svermens pålitelighet.

Feiltoleranse.

- Så langt har vi sett på tre forskjellige typer feil som kan oppstå, og konsekvensen det har for svermens ytelse.
- Basert på eksperimentene vi har sett så langt, la oss gjøre en mer formell analyse av svermens pålitelighet.
- Vi starter med en vanlig modell for pålitelighetsanalyse, **k-out-of-N:G**

k-out-of-N:G

- Et k-out-of-N:G system er velfungerende (Good) hvis minst k av N deler er velfungerende (Good).

k-out-of-N:G

- Et k-out-of-N:G system er velfungerende (Good) hvis minst k av N deler er velfungerende (Good).

- p = sannsynligheten for at en robot virker

k-out-of-N:G

- Et k-out-of-N:G system er velfungerende (Good) hvis minst k av N deler er velfungerende (Good).

- p = sannsynligheten for at en robot virker
- q = sannsynligheten for at en robot ikke virker (1 – p)

k-out-of-N:G

- Et k-out-of-N:G system er velfungerende (Good) hvis minst k av N deler er velfungerende (Good).

- p = sannsynligheten for at en robot virker
- q = sannsynligheten for at en robot ikke virker (1 – p)

$$P(k, N) = \binom{N}{k} p^k q^{N-k}$$

k-out-of-N:G

- Et k-out-of-N:G system er velfungerende (Good) hvis minst k av N deler er velfungerende (Good).

- p = sannsynligheten for at en robot virker
- q = sannsynligheten for at en robot ikke virker (1 – p)

$$P(k,N) = \sum_{i=k}^{N} \binom{N}{i} p^i q^{N-i}$$

k-out-of-N:G

- Et k-out-of-N:G system er velfungerende (Good) hvis minst k av N deler er velfungerende (Good).

- Utfordringen for oss er å tallfeste *p*, sannsynligheten for at en gitt robot er i live. Heldigvis finnes det en måte å gjøre det på.

k-out-of-N:G

- Et k-out-of-N:G system er velfungerende (Good) hvis minst k av N deler er velfungerende (Good).
- Utfordringen for oss er å tallfeste *p*, sannsynligheten for at en gitt robot er i live.
- MTBF = Mean time Between Failures.

k-out-of-N:G

- Et k-out-of-N:G system er velfungerende (Good) hvis minst k av N deler er velfungerende (Good).

- Utfordringen for oss er å tallfeste *p*, sannsynligheten for at en gitt robot er i live.

- MTBF = Mean time Between Failures.
- 480 minutter (Carlson et al. 2003)

k-out-of-N:G

- Et k-out-of-N:G system er velfungerende (Good) hvis minst k av N deler er velfungerende (Good).

- Utfordringen for oss er å tallfeste *p*, sannsynligheten for at en gitt robot er i live.

- MTBF = Mean time Between Failures.
- 480 minutter (Carlson et al. 2003)

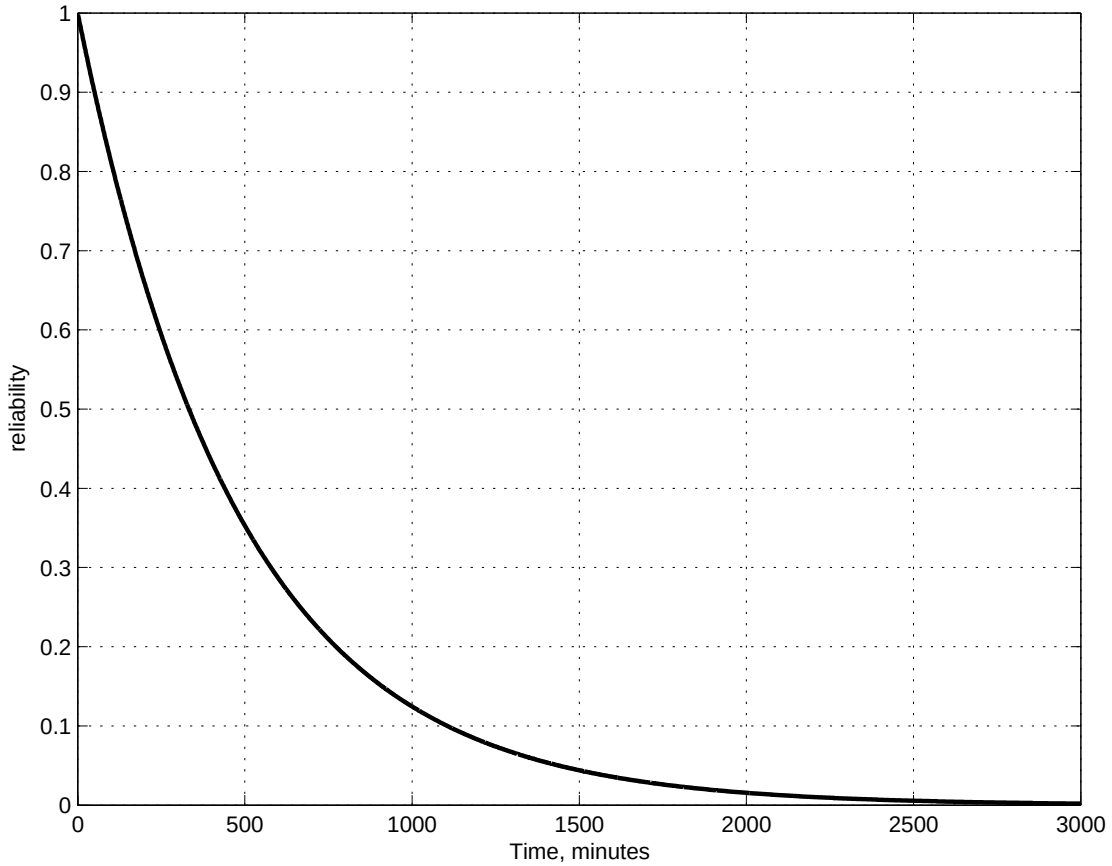
- Og fra MTBF kan vi finne feilraten:
- $\lambda = (1/MTBF)$

Forholdet mellom sannsynlighet og λ

$$p = e^{-t\lambda}$$

Forholdet mellom sannsynlighet og λ

$$p = e^{-t\lambda}$$



k-out-of-N:G

- Vi kan nå erstatte *p* med $e^{-t\lambda}$ og *q* med $1-e^{-t\lambda}$

$$P(k,N) = \sum_{i=k}^{N} \binom{N}{i} p^i q^{N-i}$$

k-out-of-N:G

- Vi kan nå erstatte *p* med $e^{-t\lambda}$ og *q* med $1-e^{-t\lambda}$

$$P(k,N,t) = \sum_{i=k}^{N} \binom{N}{i} (e^{-t\lambda})^i (1-e^{-t\lambda})^{N-i}$$

k-out-of-N:G

- Vi kan nå erstatte *p* med $e^{-t\lambda}$ og *q* med $1-e^{-t\lambda}$
- La oss bruke tallene fra Carlson et al, med en MTBF = 480 minutter.

$$P(k,N,t) = \sum_{i=k}^{N} \binom{N}{i} (e^{-t\lambda})^i (1-e^{-t\lambda})^{N-i}$$

k-out-of-N:G

- Vi kan nå erstatte *p* med $e^{-t\lambda}$ og *q* med $1-e^{-t\lambda}$
- La oss bruke tallene fra Carlson et al, med en MTBF = 480 minutter.
- La oss videre anta at vi starter med en sverm på 10 roboter (N = 10)

$$P(k,N,t) = \sum_{i=k}^{N} \binom{N}{i} (e^{-t\lambda})^{i} (1-e^{-t\lambda})^{N-i}$$

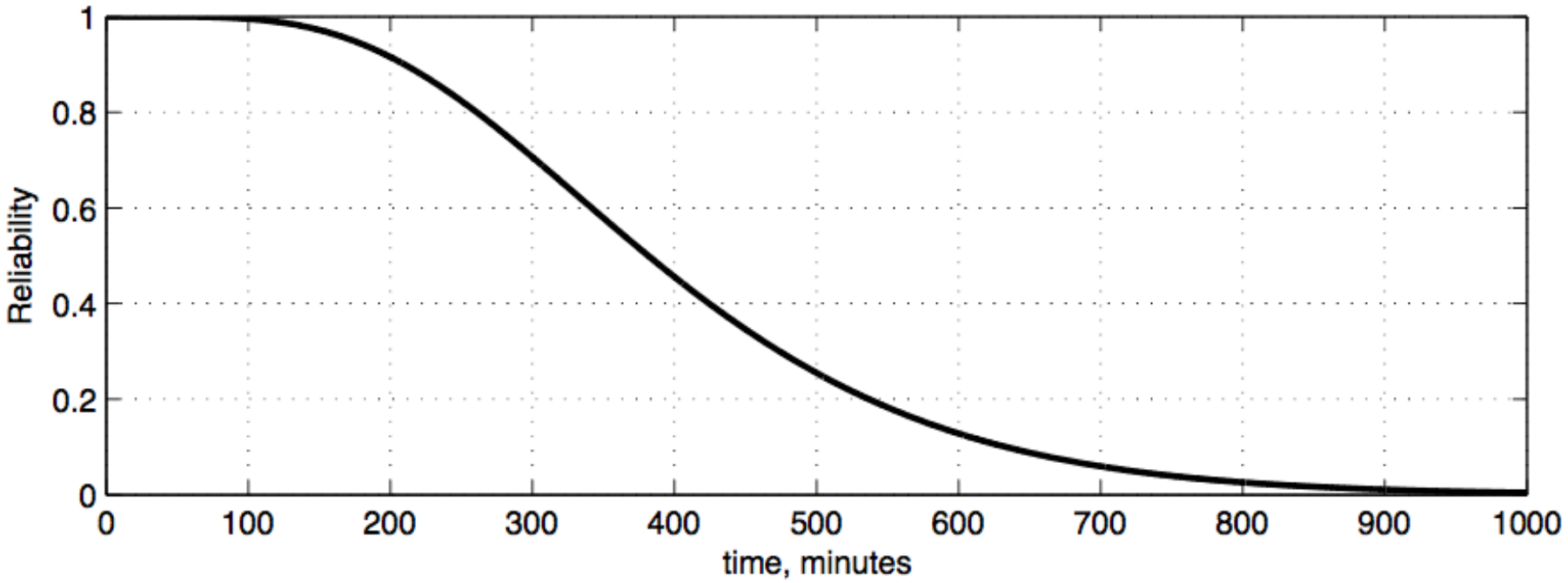
k-out-of-N:G

- Vi kan nå erstatte *p* med $e^{-t\lambda}$ og *q* med $1-e^{-t\lambda}$
- La oss bruke tallene fra Carlson et al, med en MTBF = 480 minutter.
- La oss videre anta at vi starter med en sverm på 10 roboter (N = 10)
- Svermen klarer å sverme hvis minst fem roboter fungerer (k = 5)

$$P(k,N,t) = \sum_{i=k}^{N} \binom{N}{i} (e^{-t\lambda})^i (1-e^{-t\lambda})^{N-i}$$

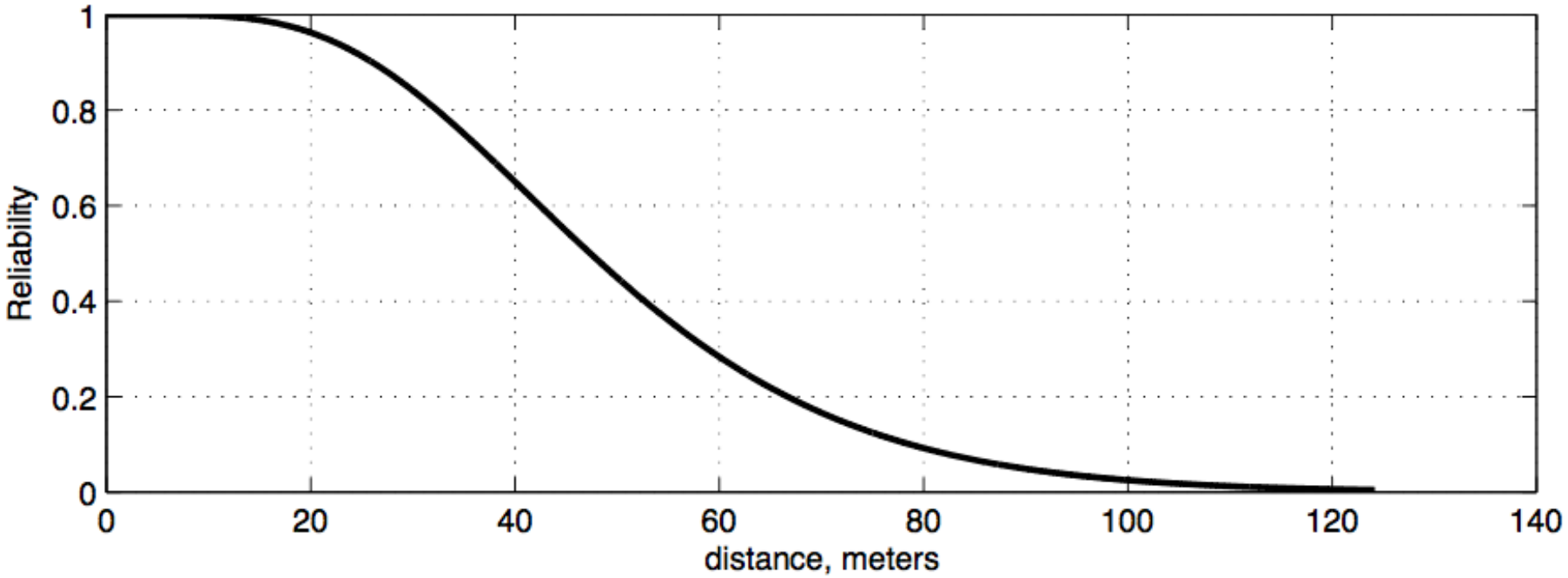
k-out-of-N:G

- Vi kan nå erstatte *p* med $e^{-t\lambda}$ og *q* med $1-e^{-t\lambda}$
- La oss bruke tallene fra Carlson et al, med en MTBF = 480 minutter.
- La oss videre anta at vi starter med en sverm på 10 roboter (N = 10)
- Svermen klarer å sverme hvis minst fem roboter fungerer (k = 5)



k-out-of-N:G

- Vi kan nå erstatte *p* med $e^{-t\lambda}$ og *q* med $1-e^{-t\lambda}$
- La oss bruke tallene fra Carlson et al, med en MTBF = 480 minutter.
- La oss videre anta at vi starter med en sverm på 10 roboter (N = 10)
- Svermen klarer å sverme hvis minst fem roboter fungerer (k = 5)



k-out-of-N:G

- Kan vi få svermen til å bevege seg vilkårlig langt, hvis vi legger til nok roboter? La oss vurdere tilfellet med delvis ødelagte roboter.

k-out-of-N:G

- Kan vi få svermen til å bevege seg vilkårlig langt, hvis vi legger til nok roboter? La oss vurdere tilfellet med delvis ødelagte roboter.
- Vi husker at når svermen hadde to delvis ødelagte motorer, klarte den bare å rive seg løs i 6 ut av 10 tilfeller.

k-out-of-N:G

- Kan vi få svermen til å bevege seg vilkårlig langt, hvis vi legger til nok roboter? La oss vurdere tilfellet med delvis ødelagte roboter.
- Vi husker at når svermen hadde to delvis ødelagte motorer, klarte den bare å rive seg løs i 6 ut av 10 tilfeller.
- Men når bare en robot er delvis ødelagt klarer svermen å rive seg løs i alle tilfellene.

k-out-of-N:G

- Kan vi få svermen til å bevege seg vilkårlig langt, hvis vi legger til nok roboter? La oss vurdere tilfellet med delvis ødelagte roboter.
- Vi husker at når svermen hadde to delvis ødelagte motorer, klarte den bare å rive seg løs i 6 ut av 10 tilfeller.
- Men når bare en robot er delvis ødelagt klarer svermen å rive seg løs i alle tilfellene.
- La oss derfor anta at svermen må 90% velfungerende roboter for å ikke stoppe opp.

k-out-of-N:G

- Kan vi få svermen til å bevege seg vilkårlig langt, hvis vi legger til nok roboter? La oss vurdere tilfellet med delvis ødelagte roboter.
- Vi husker at når svermen hadde to delvis ødelagte motorer, klarte den bare å rive seg løs i 6 ut av 10 tilfeller.
- Men når bare en robot er delvis ødelagt klarer svermen å rive seg løs i alle tilfellene.
- La oss derfor anta at svermen må 90% velfungerende roboter for å ikke stoppe opp.
- K er nå en funksjon av N (N * 0.9).

$$P(k, N, t) = \sum_{i=k}^{N} \binom{N}{i} (e^{-t\lambda})^i (1 - e^{-t\lambda})^{N-i}$$

k-out-of-N:G

- Kan vi få svermen til å bevege seg vilkårlig langt, hvis vi legger til nok roboter? La oss vurdere tilfellet med delvis ødelagte roboter.
- Vi husker at når svermen hadde to delvis ødelagte motorer, klarte den bare å rive seg løs i 6 ut av 10 tilfeller.
- Men når bare en robot er delvis ødelagt klarer svermen å rive seg løs i alle tilfellene.
- La oss derfor anta at svermen må 90% velfungerende roboter for å ikke stoppe opp.
- K er nå en funksjon av N (N * 0.9).
- Hva med *t* ?

Tiden det tar å reparere svermen

- Eksperimenter med 10 roboter og 1 delvis ødelagt robot viser at det tar i gjennomsnitt 14 minutter å reparere svermen.

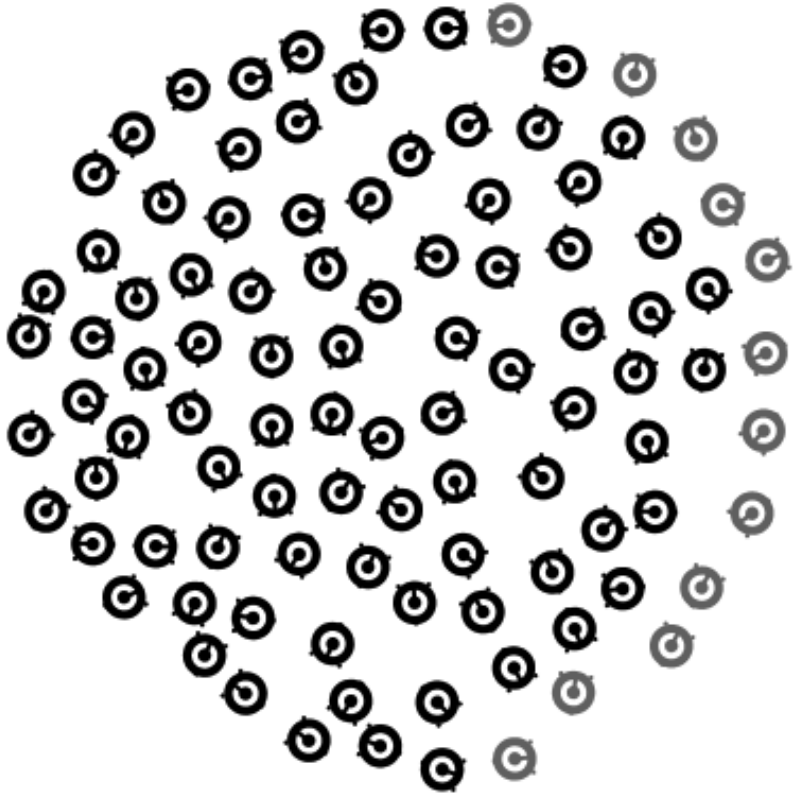
Tiden det tar å reparere svermen

- Eksperimenter med 10 roboter og 1 delvis ødelagt robot viser at det tar i gjennomsnitt 14 minutter å reparere svermen.
- Hva med en sverm med 100 roboter?

Tiden det tar å reparere svermen

- To forhold er avgjørende:

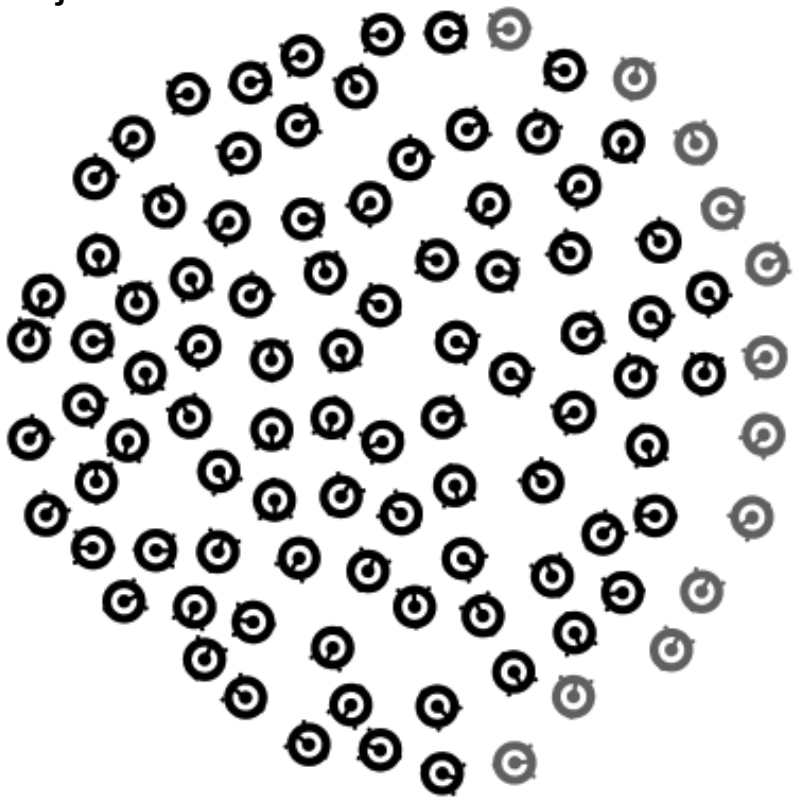




Tiden det tar å reparere svermen

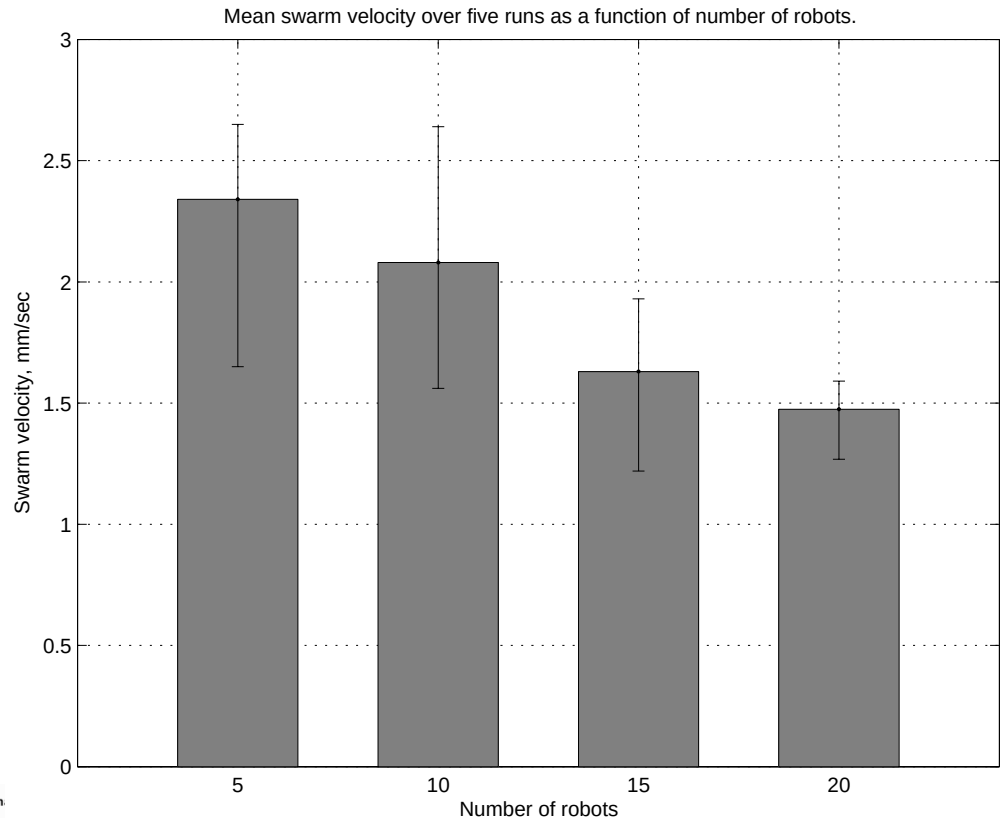
- To forhold er avgjørende:
- 1: Avstanden svermen må bevege seg for å bli reparert øker når antallet roboter øker. Avstanden er proporsjonal med $N^{1/2}$





Tiden det tar å reparere svermen

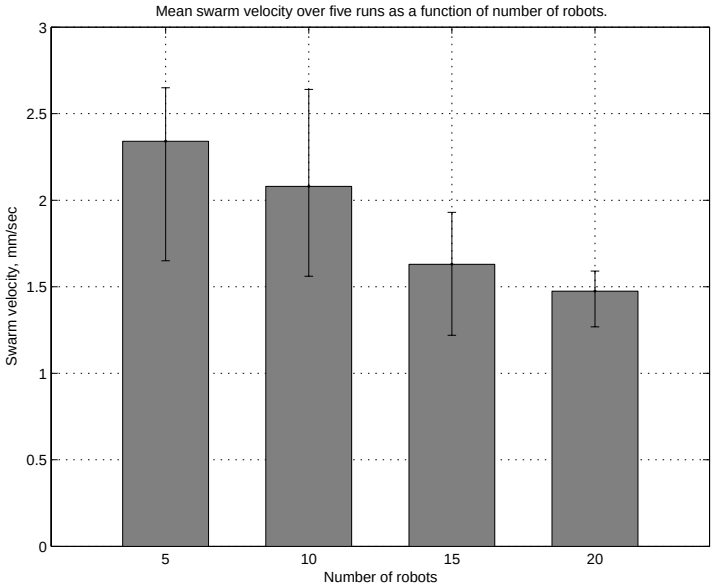
- To forhold er avgjørende:
- 1. Avstanden svermen må bevege seg øker.
- 2. Hastigheten svermen beveger seg med minker når det blir flere roboter. Hastigheten er proporsjonal med $N^{-1/2}$



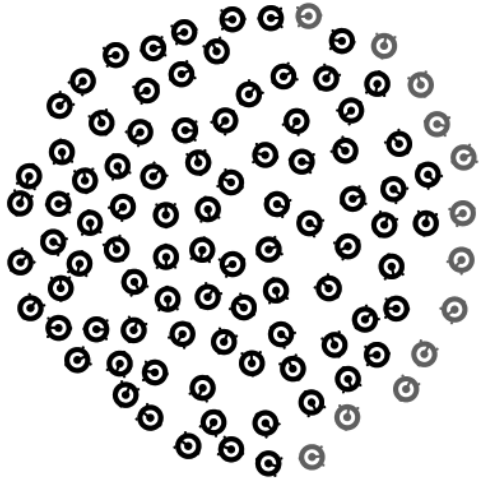
KONGSBERG PROPRIETARY – See Statement of Proprietary Inform

Tiden det tar å reparere svermen

- Avstanden er proporsjonal med $N^{1/2}$
- Hastigheten er proporsjonal med $N^{-1/2}$
- Tiden det tar å reparere svermen øker lineært med N!







Tiden det tar å reparere svermen

- Avstanden er proporsjonal med $N^{1/2}$
- Hastigheten er proporsjonal med $N^{-1/2}$
- Tiden det tar å reparere svermen øker lineært med N!
- Vi kan nå bruke k-out-of-N modellen med forskjellige verdier for N, og hvor k og t er gitt av N.

Tiden det tar å reparere svermen

- Avstanden er proporsjonal med $N^{1/2}$
- Hastigheten er proporsjonal med $N^{-1/2}$
- Tiden det tar å reparere svermen øker lineært med N!
- Vi kan nå bruke k-out-of-N modellen med forskjellige verdier for N, og hvor k og t er gitt av N.

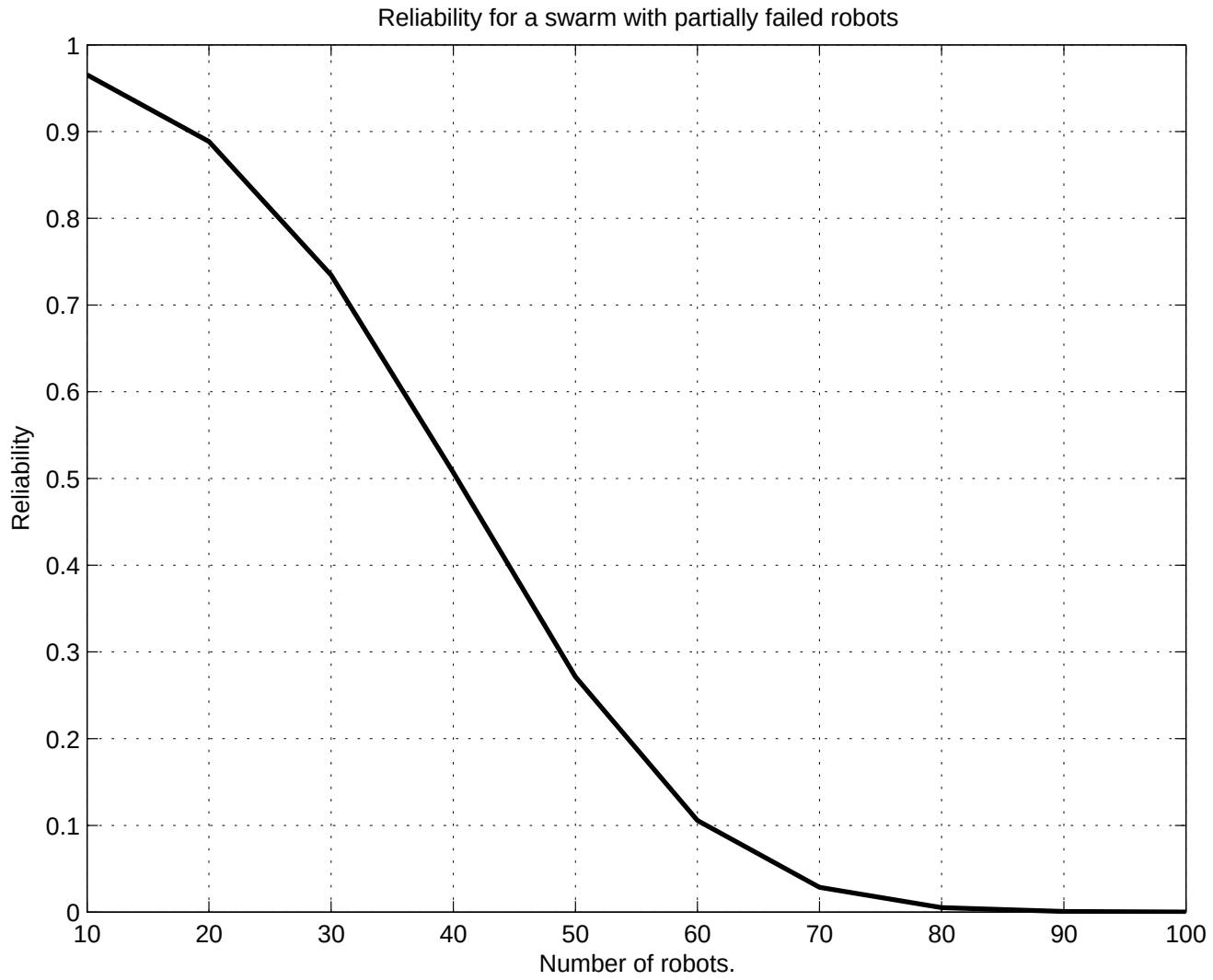
N	k	t
10	9	14 m.
20	18	28 m.
30	27	42 m.
40	36	56 m.
50	45	70 m.
60	54	84 m.
70	63	98 m.

Tiden det tar å reparere svermen

N	k	t
10	9	14 m.
20	18	28 m.
30	27	42 m.
40	36	56 m.
50	45	70 m.
60	54	84 m.
70	63	98 m.

$$P(k,N,t) = \sum_{i=k}^{N} \binom{N}{i} (e^{-t\lambda})^i (1-e^{-t\lambda})^{N-i}$$

Hvor stor sverm kan vi ha?



kongsberg.com



KONGSBERG

KONGSBERG PROPRIETARY – See Statement of Proprietary Information