# Synchronized Networks using Delayed Self-Reinforcement

Santosh Devasia
University of Washington

http://www.loydheath.com/
UniversityofWashington/Rainier-Vista/i-pMDDrLh

# Outline of talk

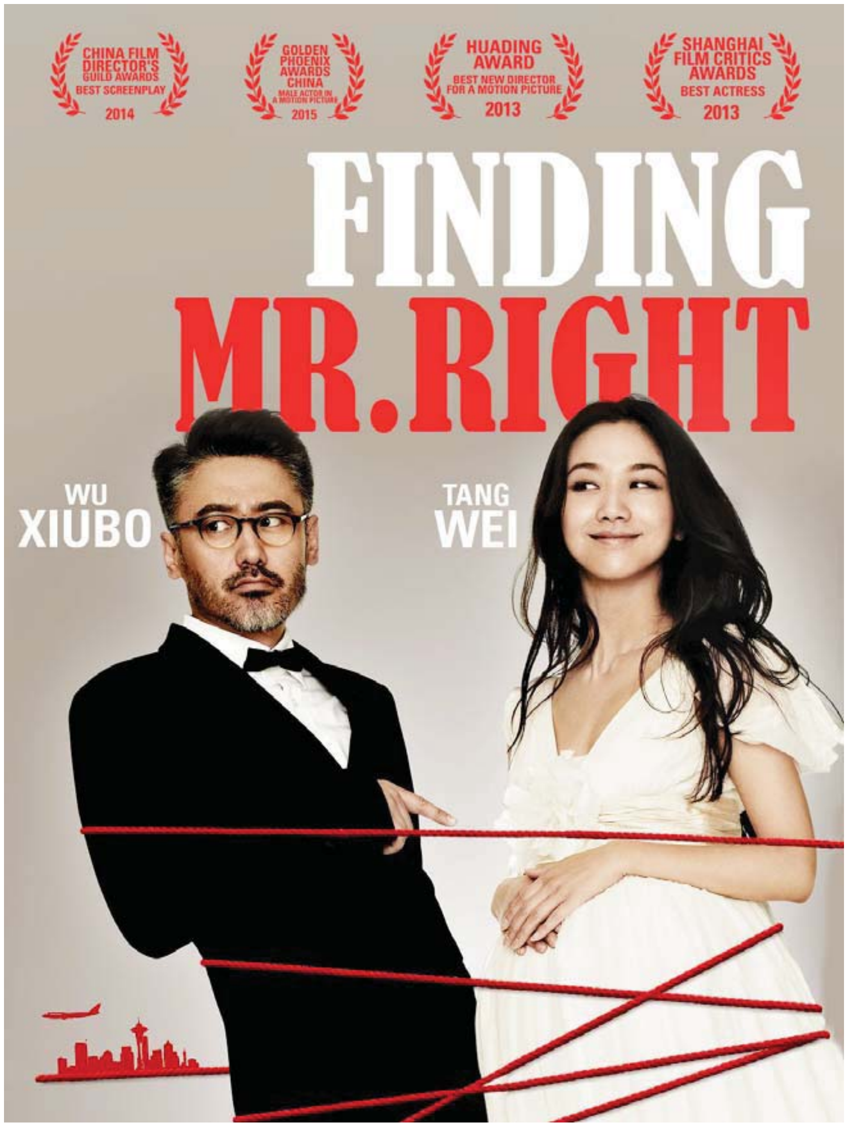
1. **Brief Intro to UW and BARC**
2. Motivation
3. Problem formulation (math)
4. Limits of current approaches
5. Proposed DSR approach
6. Results & Discussion
7. Ongoing efforts / Extensions
8. Conclusions

# Where is UW (Seattle)?

# What is Seattle is famous for?

- Rain and Coffee!

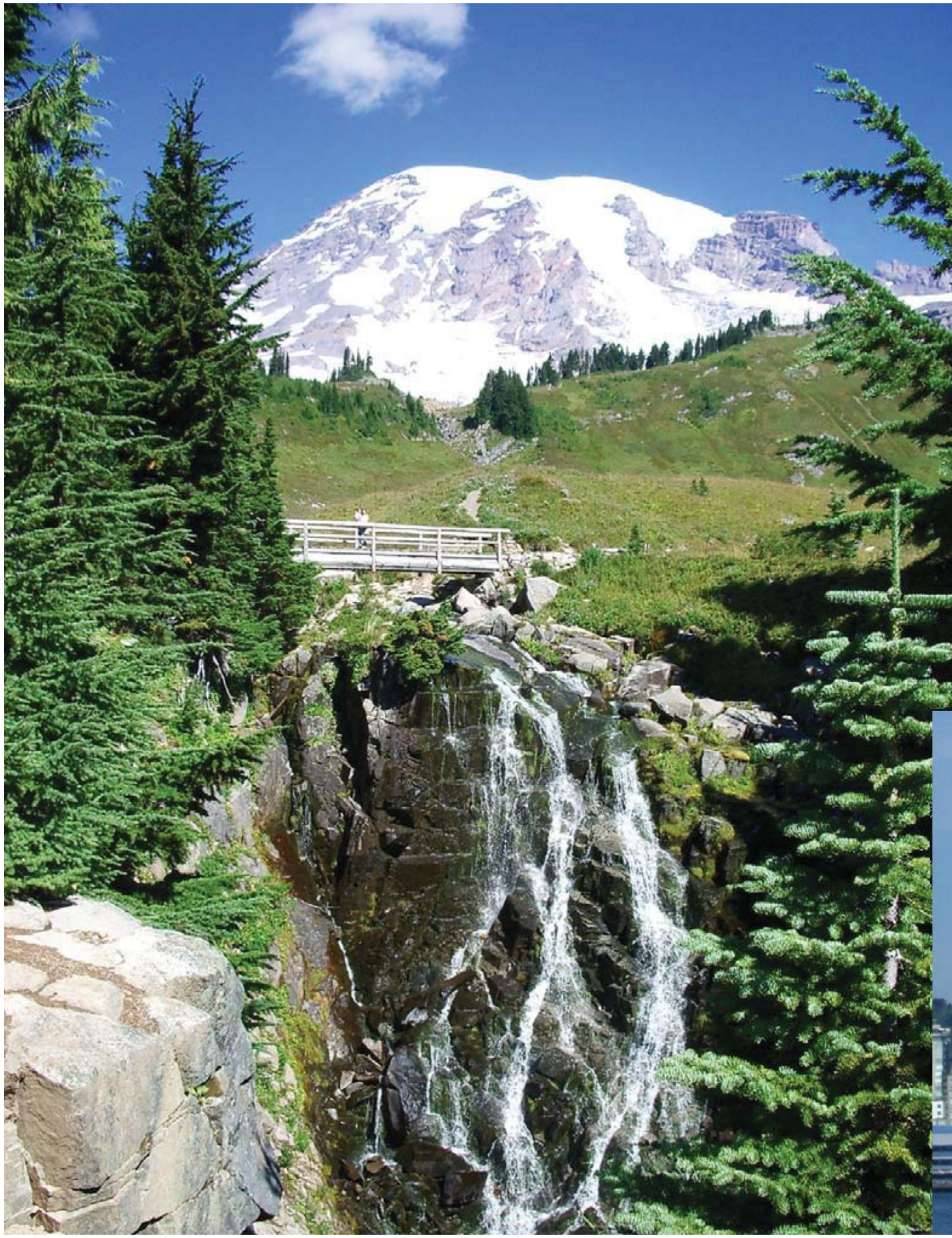
# What is Seattle is famous for?





- Rain and Coffee!
- Makes Seattle a romantic City

# Rain makes Seattle Scenic




# Other famous things in Seattle ?

amazon.com

STARBUCKS COFFEE


Blue Origin

WASHINGTON

BOEING

COSTCO WHOLESALE

ups

PACCAR

UNIVERSITY OF WASHINGTON 1861

LVX SIT

**Goal: UW link with these industries**

# Example: Boeing Advanced Research Center

- **Founded in 2014 (fall)**
- **Co-location space on campus**

- **Big Data/Machine Learning**
  - Reducing number of sensing points (sparse sensing)
  - Standardization of multiple parts
- **Robotics / Ergonomics and Safety**
  - Human-Robot collaboration, e.g., sanding
  - Shoulder fatigue modeling
  - Riveting and impact on humans
- **Composites**
  - Advanced Fiber Placement
  - Thermoplastics

**BARC Ribbon Cutting, January 2015**
UW President Michael Young, BCA CEO Ray Conner, Gov. Jay Inslee, Dean Michael Bragg

**Students work with Boeing Mechanics to make riveting safer**

# Why links to industry?

1. Education: Application-based learning; (students at all levels, BS, MS, PhD)
2. Students: Learn by doing. Highly motivated to work on these applied problems
3. Research: Leads to high-impact collaborative research
   + a source for set of rich problems with potential for strong impact

# U. of Washington at a Glance

- Founded in 1861
- *Students: 48,000*
- **#1 in** federal funds for research among US public universities
- **Ranked #10** by *U.S. News & World Global University rankings*
- Top **4 most Innovative** Universities in the world, Reuters


**Cherry Blossoms in UW**
**Spring-time in Seattle**

**Beautiful time to visit!**

# Welcome to visit Seattle/UW

- **General Chair** for
- Advanced intelligent Mechatronics (AIM) 2023 in Seattle
- You are most welcome to attend and come visit Seattle and UW

# Outline of talk

1. Brief Intro to UW and BARC
2. **Motivation**
3. Problem formulation (math)
4. Limits of current approaches
5. Proposed DSR approach
6. Results & Discussion
7. Ongoing efforts / Extensions
8. Conclusions

# Motivation

- So why another look at swarms… ?
- There is a lot of work in this already


Pond5.com, Shutterstock.com

# Observations do not match models

Video from **Information transfer and behavioural inertia in starling flocks**, Attanasi et. al., Nature Physics , 2014

# Observations do not match models

This study raised questions about validity of existing models (diffusion based)

Speed of orientation propagation proportional to time as in waves (not as in diffusion)

Video from **Information transfer and behavioural inertia in starling flocks**, Attanasi et. al., Nature Physics , 2014

# Motivation for renewed effort in swarming

1. Understand: why models do not match observations?
2. Improve: What model changes are needed to capture how swarms work in nature (esp. to match recent observations)
3. Apply: Could help to improve the design of similar engineered systems
   1. Platoons of self-driving cars
   2. UAVs (air, sea, land)
   3. Synchronized robots

# Outline of talk

1. Brief Intro to UW and BARC
2. Motivation
3. **Problem formulation (math)**
4. Limits of current approaches
5. Proposed DSR approach
6. Results & Discussion
7. Ongoing efforts / Extensions
8. Conclusions

# The research problem

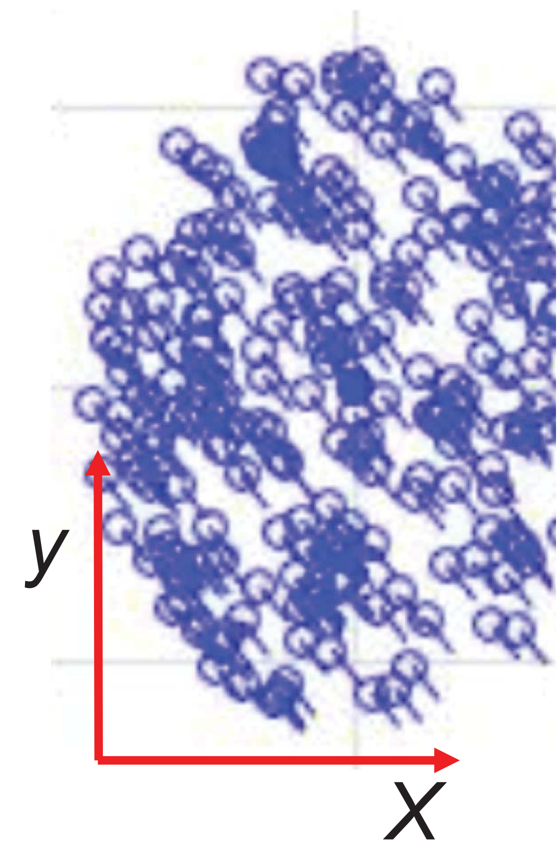
- Essentially,
  - How fast can response propagate through the network?
  - Can it be increased from current models?
  - Can we better match observations from nature?

- Will clarify this in the next few slides

# Standard swarming model

Classical models by
- Huth and Wissel, 1992. J. of Theoretical Biology,
- Vicsek, et. al., 1995. Phys. Rev. Lett.

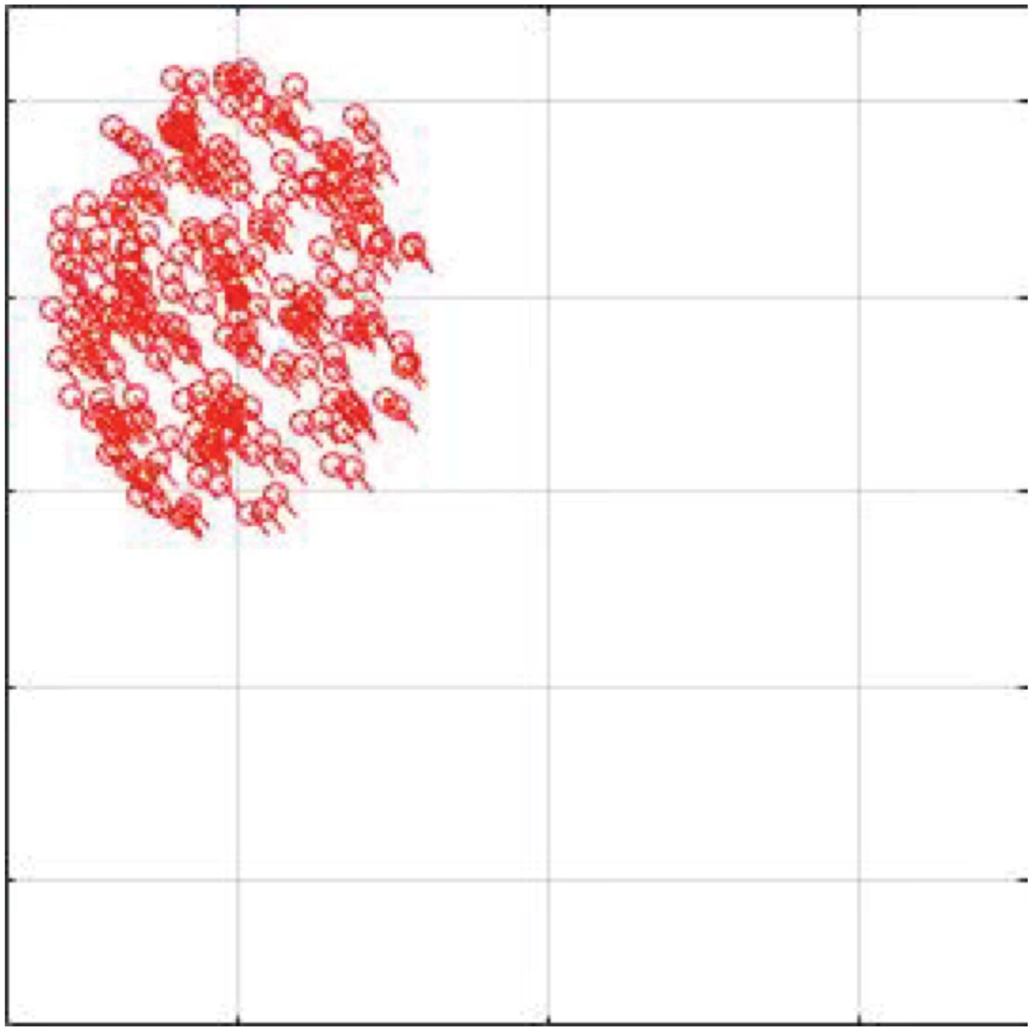
- Position of an agent is ($x_i$, $y_i$)
- Speed $v$ is constant
- Orientation $I_i$
- Agents update at time steps, $t_k = k\delta_t$ ($\delta_t$ for sensing, computing, actuation)
- $$x_i(k+1) = x_i(k) + v\delta_t \cos I_i,$$
  $$y_i(k+1) = y_i(k) + v\delta_t \sin I_i,$$
- Change orientation $I_i$ for maneuvers
- Orientation information needs to be propagated through network



# Loss of synchronization is a problem

Loss of orientation synchronization leads to loss of formation

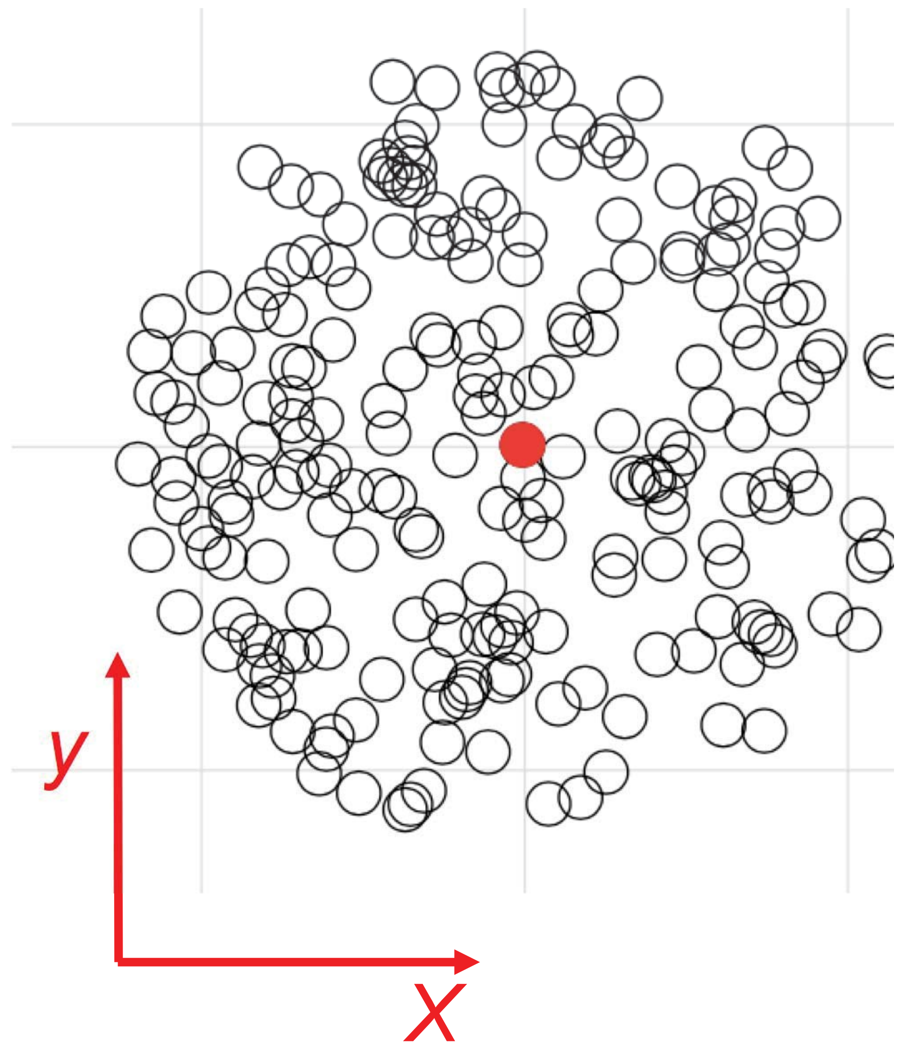
$$x_i(k+1) = x_i(k) + v\delta_t \cos I_i,$$
$$y_i(k+1) = y_i(k) + v\delta_t \sin I_i,$$



# What are limits to synchronization?

# Position changes (from before)

$$x_i(k+1) = x_i(k) + v\delta_t \cos I_i,$$
$$y_i(k+1) = y_i(k) + v\delta_t \sin I_i,$$



# How does the orientation change?

- Position changes specified earlier

$$x_i(k+1) = x_i(k) + v\delta_t \cos I_i,$$
$$y_i(k+1) = y_i(k) + v\delta_t \sin I_i,$$

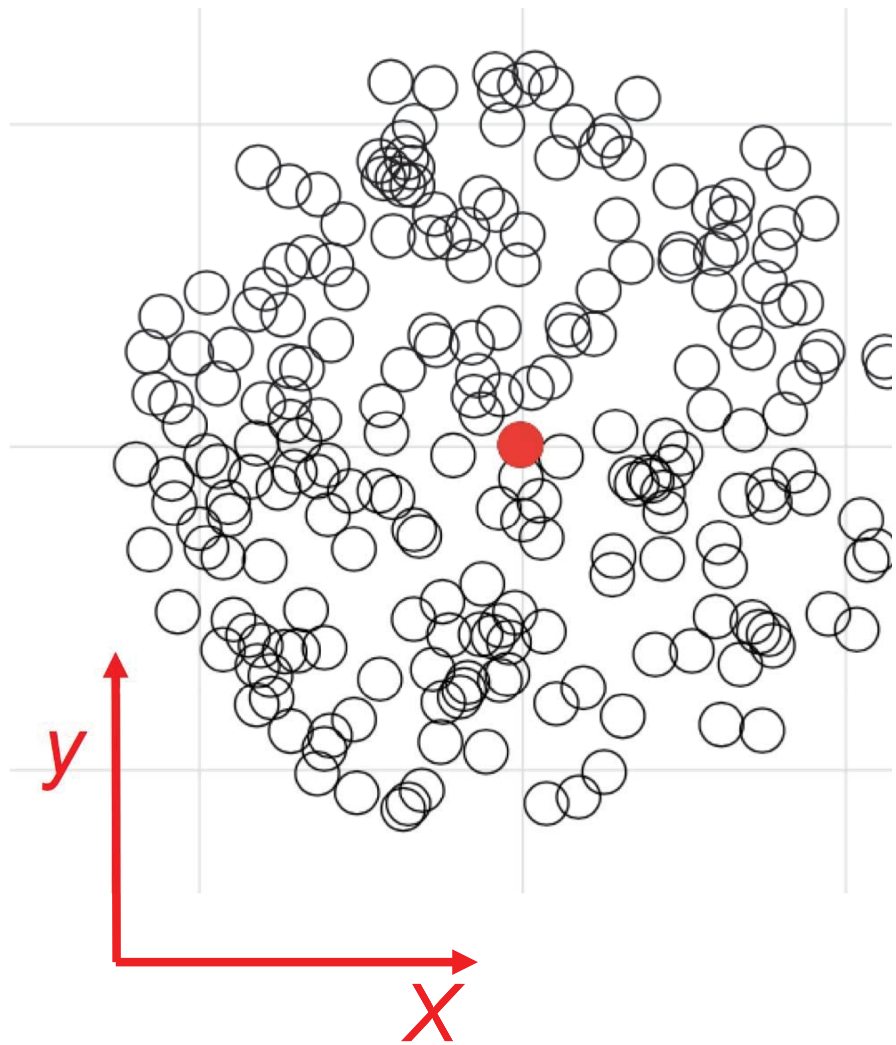
- How does the **orientation $I_i$** change?

$$[I_i(k+1) - I_i(k)] = -\gamma\Delta_i(k)\delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|}\sum_{j\in N_i}[I_i(k) - I_j(k)]$$

$$I_i(k+1) = I_i(k) - \gamma\Delta_i(k)\delta_t$$

where $\Delta_i$ is the average deviation from **neighbor j orientations,**
$|N_i|$ is the number of neighbors,
$\gamma$ is the alignment gain,



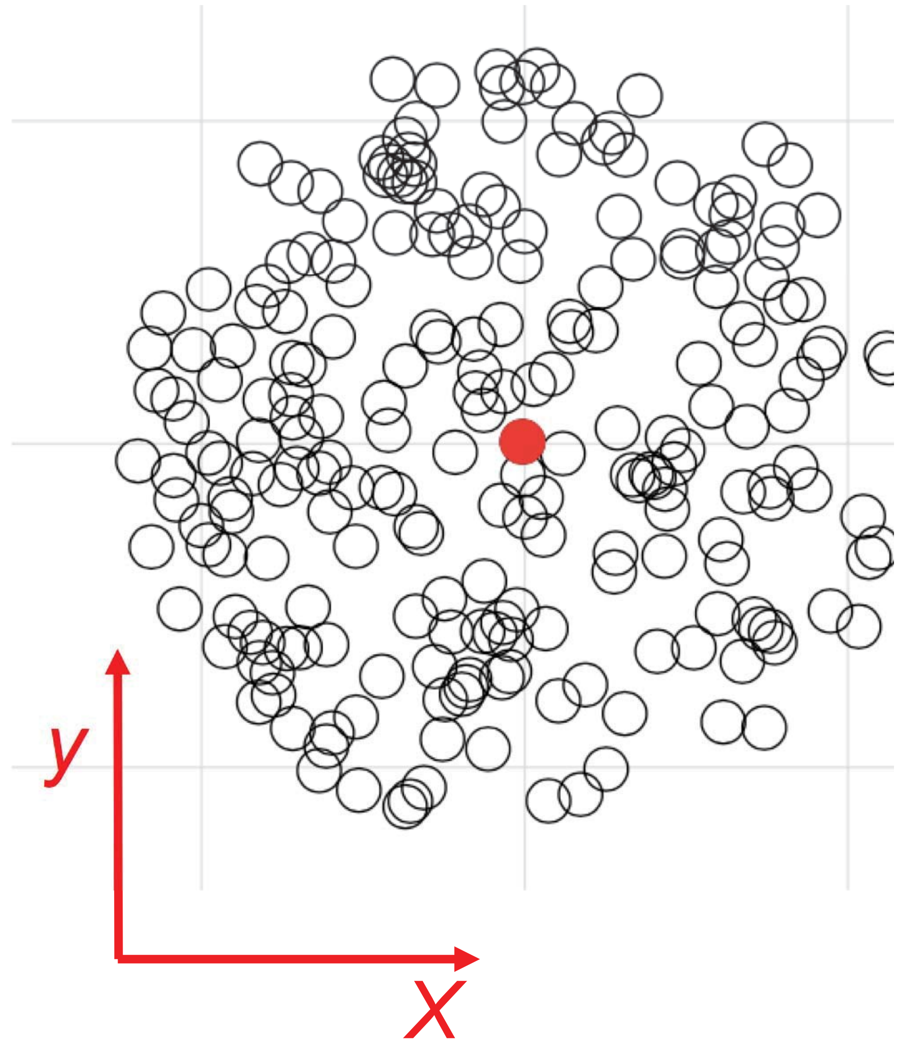
# From last page

- How does the orientation $I_i$ change?

$$I_i(k+1) = I_i(k) - \gamma\Delta_i(k)\delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|}\sum_{j\in N_i}[I_i(k) - I_j(k)]$$

where $\Delta_i$ is the average of neighbor orientations, $|N_i|$ is the number of neighbors, $\gamma$ is the alignment gain,



# Bottom line: Essentially a diffusion equation

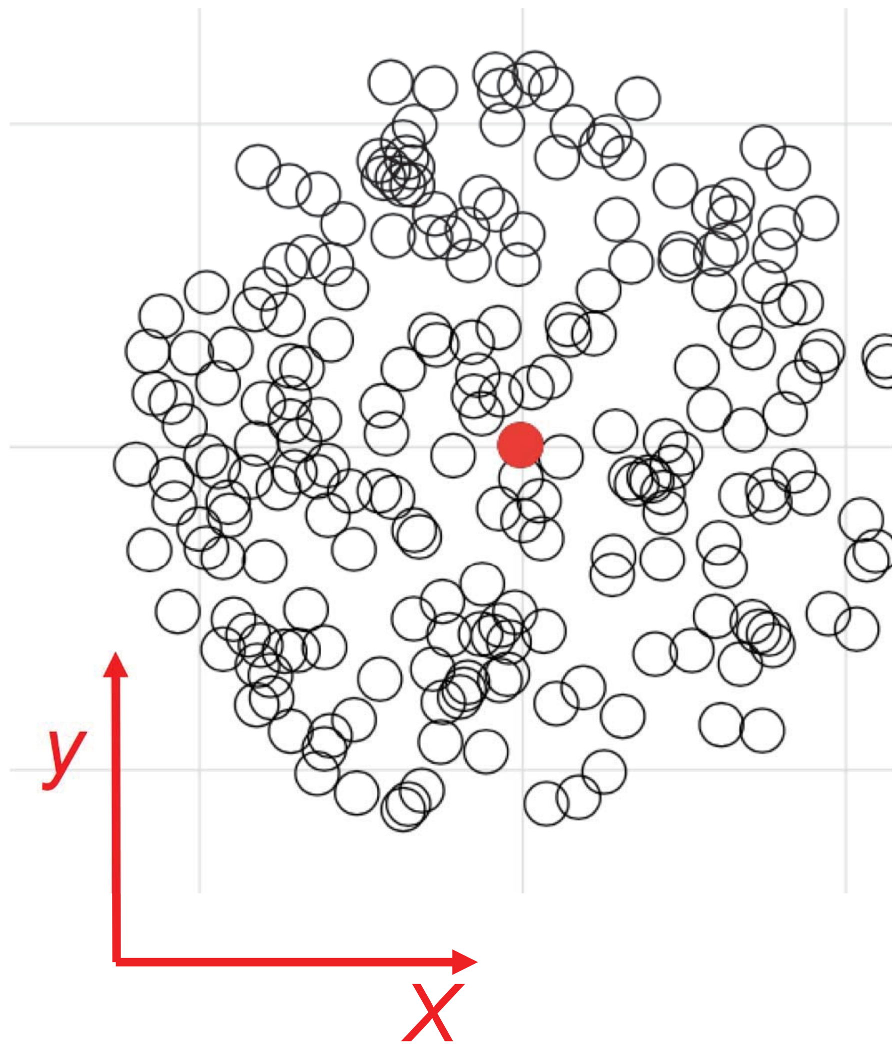
- How does the orientation $I_i$ change?

$$I_i(k+1) = I_i(k) - \gamma \Delta_i(k) \delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|} \sum_{j \in N_i} [I_i(k) - I_j(k)]$$

where $\Delta_i$ is the average of neighbor orientations, $|N_i|$ is the number of neighbors, $\gamma$ is the alignment gain,

$$\frac{\partial}{\partial t} I(t) = \gamma \frac{a^2}{2D} \nabla^2 I(t)$$




- a is average distance between agents
  D is the number of dimensions (e.g., 2)

# Diffusion: 1 Information spread rate

$$\frac{\partial}{\partial t} I(t) = \gamma \frac{a^2}{2D} \nabla^2 I(t)$$

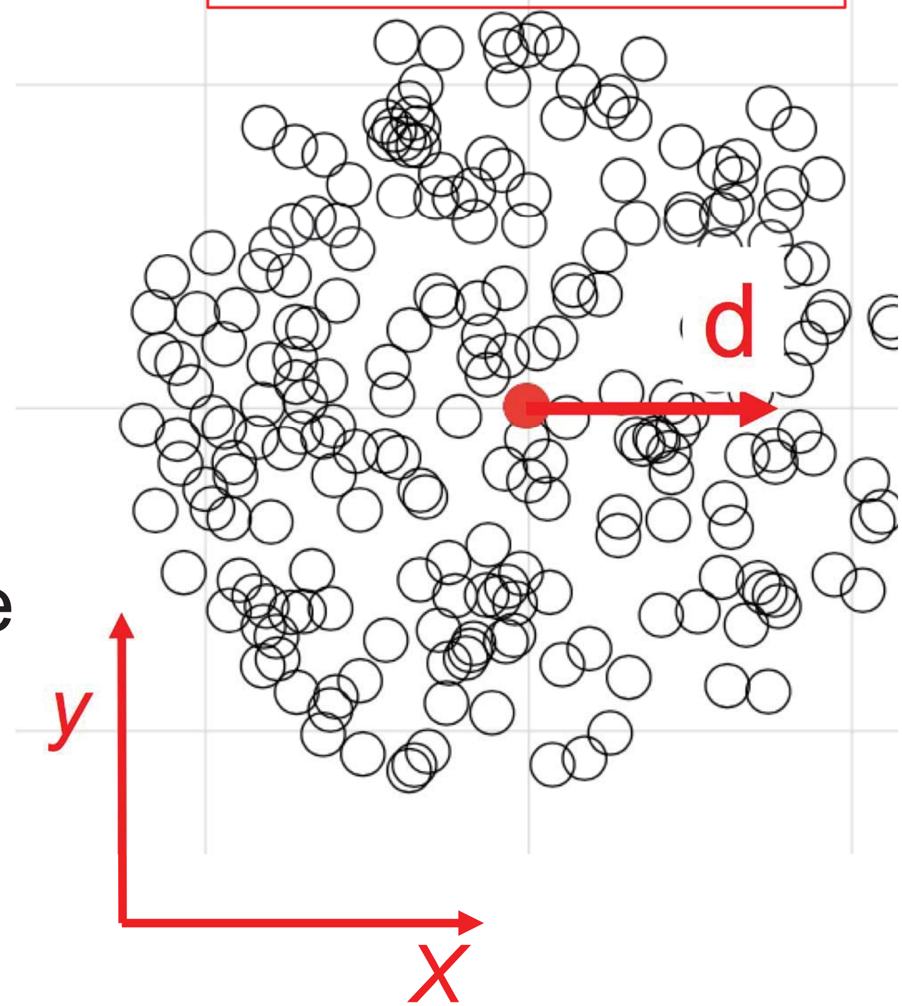
$$d \sim \sqrt{t},$$

Diffusion equation results in

d: Information travels distance

Ref:

- Attanasi et. al., 2014 Nature Physics



# Diffusion: 2 Information decay

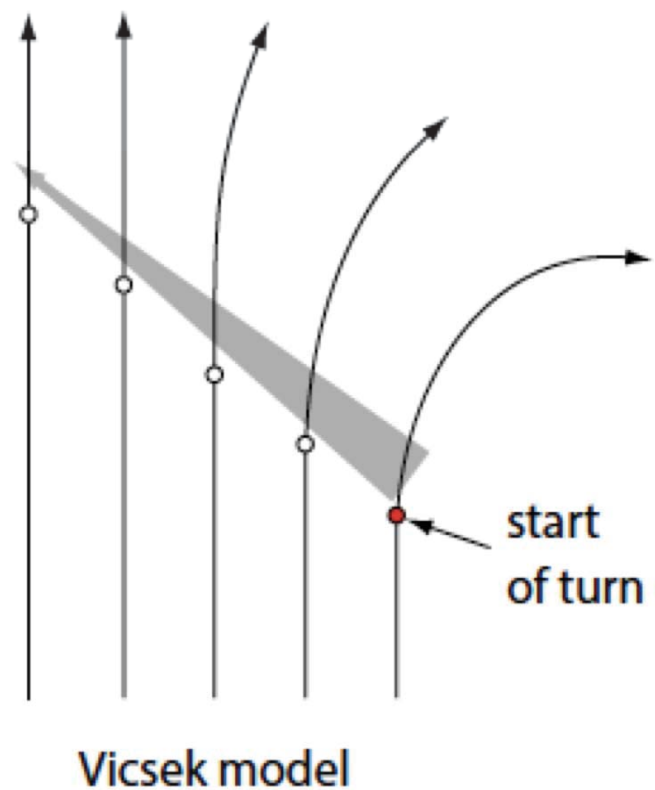
$$\frac{\partial}{\partial t} I(t) = \gamma \frac{a^2}{2D} \nabla^2 I(t)$$

Ref:
- Cavagna et. al., 2015 J. Stat Phys

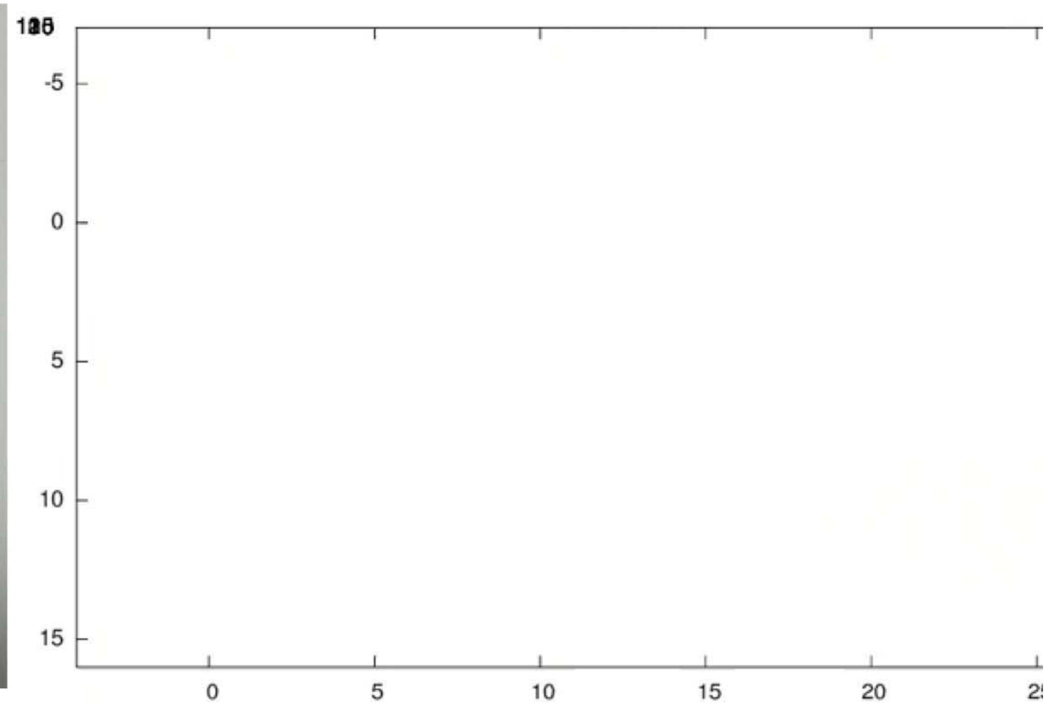
Response decays with distance from leader

Acceleration of turn decays with distance

Leads to loss of formation in traditional models!



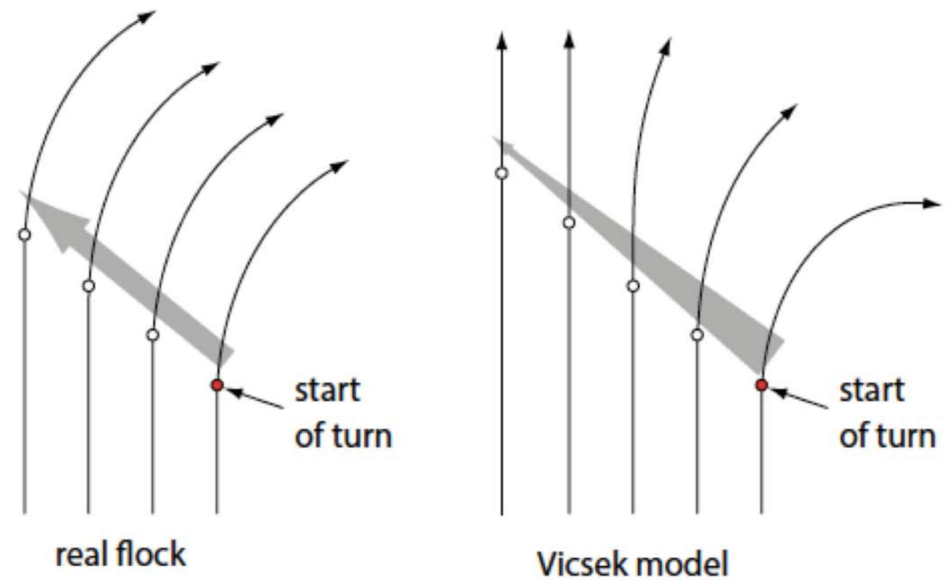
# Compare with recent observations




- Data from real flocks
- **Deviations found from traditional models**

By
- Attanasi et. al., 2014 Nature Physics

# Recent observations & problem



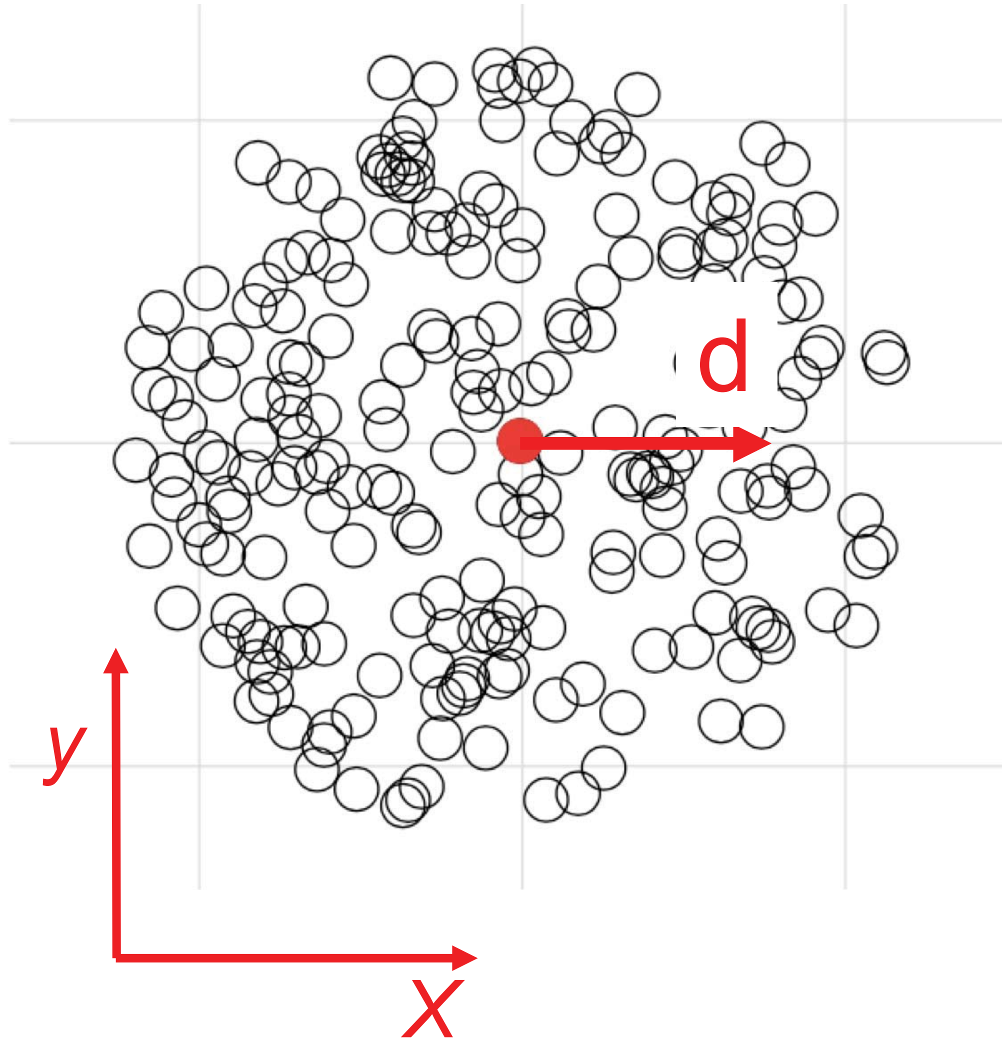

By
- Attanasi et. al., 2014 Nature Physics
- Cavagna et. al., 2015 J. Stat Phys

- **Three key differences/observations**
- 1. **No decay:** Turn information (max radial acceleration) does not decay with distance (as opposed to diffusion-type models)
- 2. **Wavelike**: Turn information spreads proportionally with time, wave-like (not square root of time predicted by diffusion models)
- 3. Re-orientation travels faster than neighbor realignment
- **Research problem: explain these differences**

# Outline of talk

1. Brief Intro to UW and BARC
2. Motivation
3. Problem formulation (math)
4. **Limits of current approaches**
5. Proposed DSR approach
6. Results & Discussion
7. Ongoing efforts / Extensions
8. Conclusions

# Other approaches to maintain formation



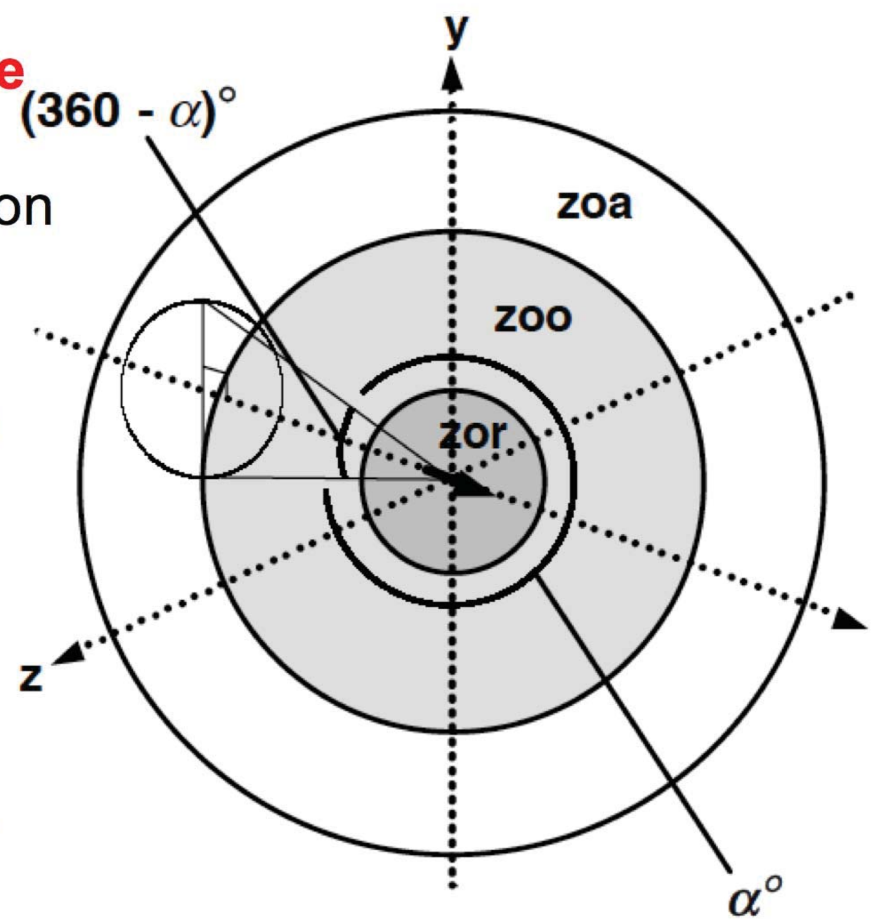
# 1. Distance dependent orientation

$$I_i(k+1) = I_i(k) - \gamma\Delta_i(k)\delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|}\sum_{j\in N_i}[I_i(k) - I_j(k)]$$

Example model by
- Couzin, et. al., 2002. J. of Theoretical Biology

- **Change policy depending on distance**
  **Repulsion**: align orientation $I_i$ to move away from individuals in zone of repulsion (inner most zone)
- **Attraction**: align orientation $I_i$ to move towards individuals in zone of attraction (outer most zone)
- **Orientation**: align orientation $I_i$ with individuals in zone of orientation
- Different weights to each action
- Action/neighbor could be randomized…



# 1. Distance dependent orientation

$$I_i(k+1) = I_i(k) - \gamma \Delta_i(k) \delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|} \sum_{j \in N_i} [I_i(k) - I_j(k)]$$

Example model by
- Couzin, et. al., 2002. J. of Theoretical Biology

- **Rep** awa repu
- **Attr** towa (out
- **Orie** indiv
- Diffe
- Acti

- However, recent findings in Attanasi et. al., 2014 Nature Physics indicate that
  Alignment information travels faster than neighbor realignment!
- Moreover, this also requires additional sensing
- So there might be other approaches to
  - improve information transfer rate and
  - help maintain formation
  - that are not dependent on such distance-based policies

# Approach 2: increase the alignment gain

# Current swarm-type network models

- In a swarm, each agent's update is given by

$$z_i[k+1] = z_i[k] + u_i[k]$$
$$= z_i[k] + \gamma \sum_{j=0}^{n} a_{ij}\left(z_j - z_i\right)$$

- $\gamma$ is the alignment strength
- $a_{ij}$ is positive only if agent j is a neighbor of agent i
- The zero[th] agent is a virtual source $Z_S$

# When connectivity is a given graph

$$z_i[k+1] = z_i[k] + u_i[k]$$
$$= z_i[k] + \gamma \sum_{j=0}^{n} a_{ij}\left(z_j - z_i\right)$$

- Rewrite in terms of the pinned Graph Laplacian K

$$\begin{aligned} Z[k+1] \quad &= Z[k] - \gamma K Z[k] + \gamma B Z_s[k] \\ &= \left(\mathbf{I} - \gamma K\right) Z[k] + \gamma B Z_s(k) \\ &= P Z[k] + \gamma B Z_s(k). \end{aligned}$$

- Convergence depends on Perron matrix P

# Standard convergence theory

- If gain $\gamma$ is sufficiently small and every agent is connected to the source agent through a path

$$\begin{aligned} Z[k+1] &= Z[k] - \gamma K Z[k] + \gamma B Z_s[k] \\ &= P Z[k] + \gamma B Z_s(k). \end{aligned}$$

- Contraction

$$[Z[k+1] - Z[k]] = P^k \left[ Z[1] - Z[0] \right] \to 0$$

- Leads to consensus

$$Z[k] \to K^{-1} B Z_d$$

$$K^{-1} B = \mathbf{1}_{n \times 1}.$$

$$Z[k] \to \mathbf{1}_n Z_d \text{ as } k \to \infty.$$

# Research issue: fast convergence

$$Z[k] \quad \to \mathbf{1}_n Z_d \ \text{ as } \ k \to \infty.$$

- Rate of convergence depends on
  - alignment gain $\gamma$ and
  - graph connectivity K

$$\begin{aligned} Z[k+1] \quad &= Z[k] - \gamma K Z[k] + \gamma B Z_s[k] \\ &= P Z[k] + \gamma B Z_s(k). \end{aligned}$$

- Large alignment gain $\gamma$ can result in fast response

# However there are limits on alignment gain

- Formally, let eigenvalues of pinned Laplacian K be

$$\lambda_{K,m} = m_{K,m} e^{i\phi_{K,m}}$$

- Then, the range of acceptable alignment gain is bounded

$$0 < \gamma \quad < \min_{1 \leq i \leq n} 2\frac{\cos(\phi_{K,i})}{m_{K,i}} = \overline{\gamma} < \infty.$$

- Details: Devasia, Indian Control Conference, 2019

# Other limits: bandwidth and input magnitude

- Increasing alignment gain $\gamma$ for fast response
  - Issue 1: requires larger inputs
  - Issue 2: makes the system is stiffer,
    the information update rate needs to be increased for stability.
    - However, update rate is limited by sensing/communication bandwidth.
- Max update rate (input bandwidth) as well as input size restrictions also limit the maximum convergence rate

# Outline of talk

1. Brief Intro to UW and BARC
2. Motivation
3. Problem formulation (math)
4. Limits of current approaches
5. **Proposed DSR approach**
6. Results & Discussion
7. Ongoing efforts / Extensions
8. Conclusions

# Proposed DSR approach

1. The approach
2. Does it require additional information from the network? (e.g., direct connection to source?)
3. Why it works? (why is it non diffusive?)
4. How to select the DSR parameters? ($\beta$)
5. Stability condition for graph networks

# Proposed DSR approach

- From Devasia, ASME J. of Dynamic Systems Measurement and Control, March 2019

# Proposed DSR approach

- Change from

$$[I_i(k+1) - I_i(k)] = -\gamma \Delta_i(k) \delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|} \sum_{j \in N_i} [I_i(k) - I_j(k)]$$

# Proposed DSR approach

- Change from

$$[I_i(k+1) - I_i(k)] = -\gamma \Delta_i(k) \delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|} \sum_{j \in N_i} [I_i(k) - I_j(k)]$$

- to

$$[I_i(k+1) - I_i(k)] = -\gamma \Delta_i(k) \delta_t + \beta [I_i(k) - I_i(k-1)]$$

# Proposed DSR approach

- Change from

$$[I_i(k+1) - I_i(k)] = -\gamma \Delta_i(k) \delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|} \sum_{j \in N_i} [I_i(k) - I_j(k)]$$

- to

$$[I_i(k+1) - I_i(k)] = -\gamma \Delta_i(k) \delta_t + \beta [I_i(k) - I_i(k-1)]$$

- where $\beta$ is the update gain on the delayed-self-reinforcement (DSR) using older update

# Proposed DSR approach

- Change from

$$[I_i(k+1) - I_i(k)] = -\gamma \Delta_i(k) \delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|} \sum_{j \in N_i} [I_i(k) - I_j(k)]$$

- to

$$[I_i(k+1) - I_i(k)] = -\gamma \Delta_i(k) \delta_t + \beta [I_i(k) - I_i(k-1)]$$

- where $\beta$ is the update gain on the delayed-self-reinforcement (DSR) using older update
- or

$$I_i(k+1) = I_i(k) - \gamma \Delta_i(k) \delta_t + \beta [I_i(k) - I_i(k-1)]$$

# Proposed DSR approach

- Change from

$$I_i(k+1) = I_i(k) - \gamma \Delta_i(k) \delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|} \sum_{j \in N_i} [I_i(k) - I_j(k)]$$

- to

$$I_i(k+1) = I_i(k) - \gamma \Delta_i(k) \delta_t + \beta [I_i(k) - I_i(k-1)]$$

- where $\beta$ is the update gain on the delayed-self-reinforcement (DSR) term in brackets
- delayed reinforcement term is referred to as the momentum term in gradient-based search algorithms

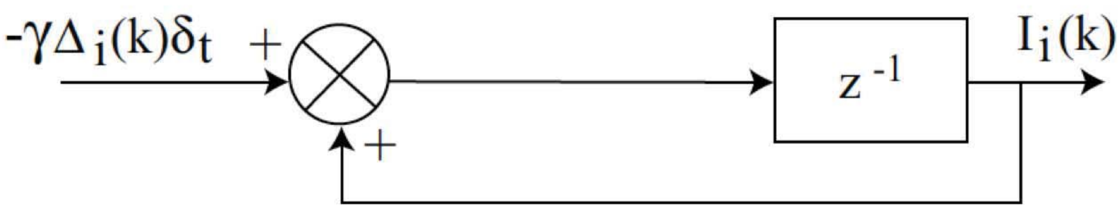

- Rumelhart, et. al., 1986, Parallel Distributed Processing, Vol. 1 . MIT Press, Cambridge, MA.
- Qian, 1999. "On the momentum term in gradient descent learning algorithms". Neural Networks,

# Proposed DSR approach

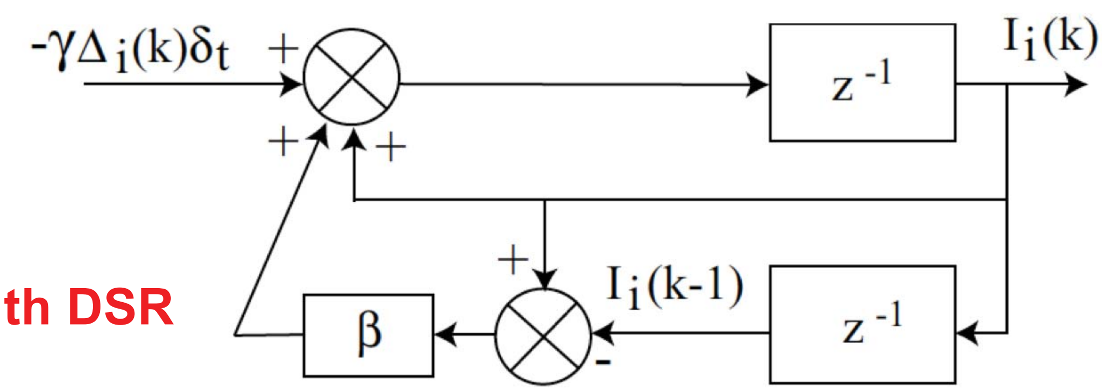
1. The approach
2. **Does it require additional information from the network? (e.g., direct connection to source?)**
3. Why it works? (why is it non diffusive?)
4. How to select the DSR parameters? ($\beta$)
5. Stability condition for graph networks

# Implementation for each agent

$$I_i(k+1) = I_i(k) - \gamma\Delta_i(k)\delta_t + \beta\,[I_i(k) - I_i(k-1)]$$




Without DSR




With DSR

- Update for each agent $I_i(k+1)$
- Uses a delayed version of previous input $I_i(k-1)$
- **No additional info from network. Just self reinforcement with prior info. Also, same update rate**

# Proposed DSR approach

1. The approach
2. Does it require additional information from the network? (e.g., direct connection to source?)
3. **Why it works? (why is it non diffusive?)**
4. How to select the DSR parameters? ($\beta$)
5. Stability condition for graph networks

# Why it works (Analysis)

$$I_i(k+1) = I_i(k) - \gamma \Delta_i(k) \delta_t + \beta \left[ I_i(k) - I_i(k-1) \right]$$

$$\left[ I_i(k+1) - I_i(k) \right] = -\gamma \Delta_i(k) \delta_t + \beta \left[ I_i(k) - I_i(k-1) \right]$$

# Rewrite the eqs.

$$I_i(k+1) = I_i(k) - \gamma\Delta_i(k)\delta_t + \beta\left[I_i(k) - I_i(k-1)\right]$$

$$\left[I_i(k+1) - I_i(k)\right] = -\gamma\Delta_i(k)\delta_t + \beta\left[I_i(k) - I_i(k-1)\right]$$

$$\begin{aligned}\frac{\beta}{\delta_t}\left\{\left[I_i(k+1) - I_i(k)\right] - \left[I_i(k) - I_i(k-1)\right]\right\} \\ + \quad \frac{1-\beta}{\delta_t}\left[I_i(k+1) - I_i(k)\right] \\ = -\gamma\Delta_i(k).\end{aligned}$$

# Take limit: small sampling time

$$I_i(k+1) = I_i(k) - \gamma\Delta_i(k)\delta_t + \beta\left[I_i(k) - I_i(k-1)\right]$$

$$\left[I_i(k+1) - I_i(k)\right] = -\gamma\Delta_i(k)\delta_t + \beta\left[I_i(k) - I_i(k-1)\right]$$

$$\frac{\beta}{\delta_t}\left\{\left[I_i(k+1) - I_i(k)\right] - \left[I_i(k) - I_i(k-1)\right]\right\} + \frac{1-\beta}{\delta_t}\left[I_i(k+1) - I_i(k)\right] = -\gamma\Delta_i(k).$$

$$\beta\delta_t\frac{\partial^2}{\partial t^2}I(t,X) + (1-\beta)\frac{\partial}{\partial t}I(t,X) = \gamma\frac{a^2}{2D}\nabla^2 I(t,X)$$

- Telegraphic transmission equations

# Can capture wavelike & diffusion

$$\beta\delta_t \frac{\partial^2}{\partial t^2} I(t,X) + (1-\beta)\frac{\partial}{\partial t} I(t,X) = \gamma \frac{a^2}{2D} \nabla^2 I(t,X)$$

- For $\beta \to 1$ (wavelike)

$$\frac{\partial^2}{\partial t^2} I(t) = \frac{\gamma a^2}{2D\delta_t} \nabla^2 I(t) = c^2 \nabla^2 I(t),$$

- For $\beta \to 0$ (diffusion)

$$\frac{\partial}{\partial t} I(t) = \gamma \frac{a^2}{2D} \nabla^2 I(t)$$

- Performance depends on DSR gain

# Proposed DSR approach

1. The approach
2. Does it require additional information from the network? (e.g., direct connection to source?)
3. Why it works? (why is it non diffusive?)
4. **How to select the DSR parameters? ($\beta$)**
5. Stability condition for graph networks

# Selection of DSR gain

$$\beta\delta_t \frac{\partial^2}{\partial t^2} I(t,X) + (1-\beta)\frac{\partial}{\partial t} I(t,X) = \gamma \frac{a^2}{2D} \nabla^2 I(t,X)$$

- Assuming a local graph (local in time since neighbors do not change very fast)

$$\frac{d^2}{dt^2} I(t) + \frac{(1-\beta)}{\beta\delta_t}\frac{d}{dt} I(t) = -\frac{\gamma}{\beta\delta_t} A I(t) + \frac{\gamma}{\beta\delta_t} B I_s(t)$$

- In Jordan form $A_J = T_A^{-1} A T_A$
- In modal form

$$\frac{d^2}{dt^2} I_J(t) + \frac{(1-\beta)}{\beta\delta_t}\frac{d}{dt} I_J(t) = -\frac{\gamma}{\beta\delta_t} A_J I_J(t) + \frac{\gamma}{\beta\delta_t} B_J I_s(t),$$

# Second-order system

- In modal form

$$\frac{d^2}{dt^2} I_J(t) + \frac{(1-\beta)}{\beta\delta_t} \frac{d}{dt} I_J(t) = -\frac{\gamma}{\beta\delta_t} A_J I_J(t) + \frac{\gamma}{\beta\delta_t} B_J I_s(t),$$

$$s^2 + \frac{(1-\beta)}{\beta\delta_t} s + \frac{\gamma\lambda_{A,i}}{\beta\delta_t} = 0$$

- Yields a second order equation
- $\beta$ can be chosen, e.g., to achieve critical damping of the main mode

# Optimal selection of DSR gain

- DSR gain selected for critical damping  to avoid overshoot

$$\beta^* = (1+2\gamma\delta_t\lambda_{A,i}) - \sqrt{(1+2\gamma\delta_t\lambda_{A,i})^2 - 1}$$
$$= \left(1+\frac{8\delta_t}{T_{s,i}}\right) - \sqrt{\left(1+\frac{8\delta_t}{T_{s,i}}\right)^2 - 1}.$$

# Settling with and without DSR

- DSR gain selected for critical damping to avoid overshoot

$$\beta^* = (1 + 2\gamma\delta_t\lambda_{A,i}) - \sqrt{(1 + 2\gamma\delta_t\lambda_{A,i})^2 - 1}$$
$$= \left(1 + \frac{8\delta_t}{T_{s,i}}\right) - \sqrt{\left(1 + \frac{8\delta_t}{T_{s,i}}\right)^2 - 1}.$$

- Settling time with DSR can be related to settling time without DSR – note: square root term

$$\hat{T}_{s,i} \approx \frac{5.8}{\zeta_i\omega_i} = \frac{5.8}{\omega_i} = 5.8\sqrt{\frac{\beta^*\delta_t}{\gamma\lambda_{A,i}}} = 2.9\sqrt{\beta^*\delta_t T_{s,i}}.$$

# Proposed DSR approach

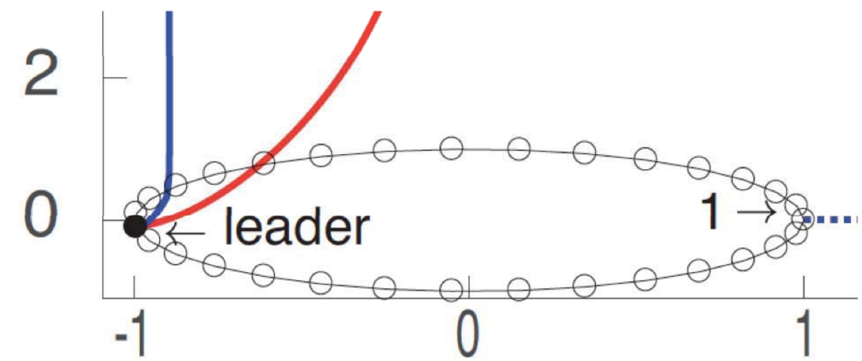
1. The approach
2. Does it require additional information from the network? (e.g., direct connection to source?)
3. Why it works? (why is it non diffusive?)
4. How to select the DSR parameters? ($\beta$)
5. **Stability condition for graph networks**

# Given a specific graph-based network

- How does the orientation $I_i$ change?

$$I_i(k+1) = I_i(k) - \gamma \Delta_i(k)\delta_t$$

$$\Delta_i(k) = \frac{1}{|N_i|}\sum_{j\in N_i}[I_i(k) - I_j(k)]$$

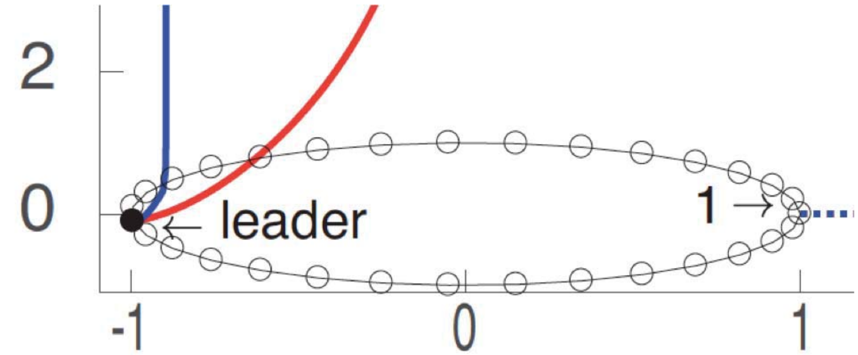



$$\begin{aligned} I(k+1) &= I(k) - \gamma K I(k) + \gamma B I_s(k) \\ &= P I(k) + \gamma B I_s(k) \end{aligned}$$

- Update K, B depends on the network graph

# Notion of DSR is similar

$$\begin{aligned} I(k+1) &= I(k) - \gamma K I(k) + \gamma B I_s(k) \\ &= P I(k) + \gamma B I_s(k) \end{aligned}$$




$$\begin{aligned} [I(k+1) - I(k)] &= -\gamma K I(k) + \gamma B I_s(k) \\ &\quad + \beta \left[ I(k) - I(k-1) \right], \end{aligned}$$

# Stability conditions based on graph

$$[I(k+1) - I(k)] \quad = -\gamma K I(k) + \gamma B I_s(k) + \beta\left[I(k) - I(k-1)\right],$$

$$-\left[1 - \frac{1}{2}\gamma\lambda_{K,n}\right] \quad < \quad \beta \quad < \quad 1$$

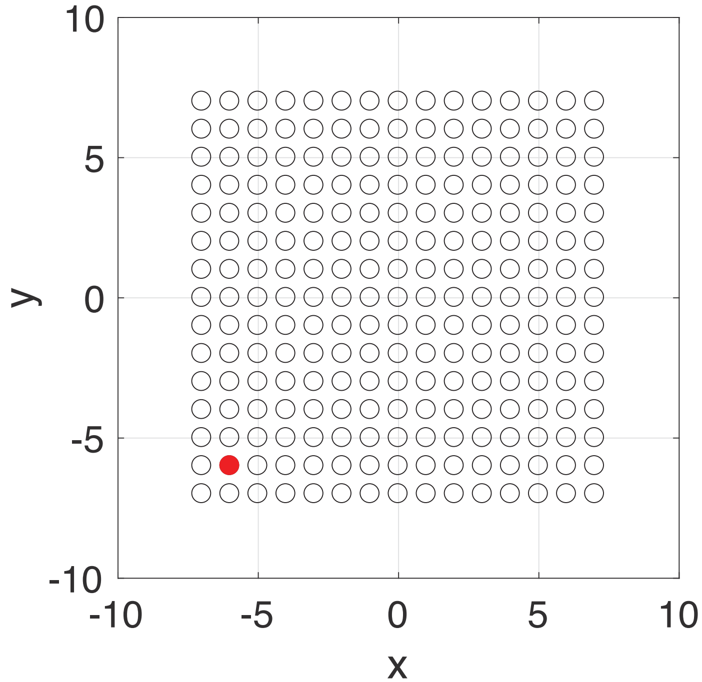
- **Main result: (can quantify stability explicitly)**
  If the original network is stable, then,
  the network is also stable if the DSR gain satisfies the above condition
- Devasia, IEEE Indian Control Conference, 2019

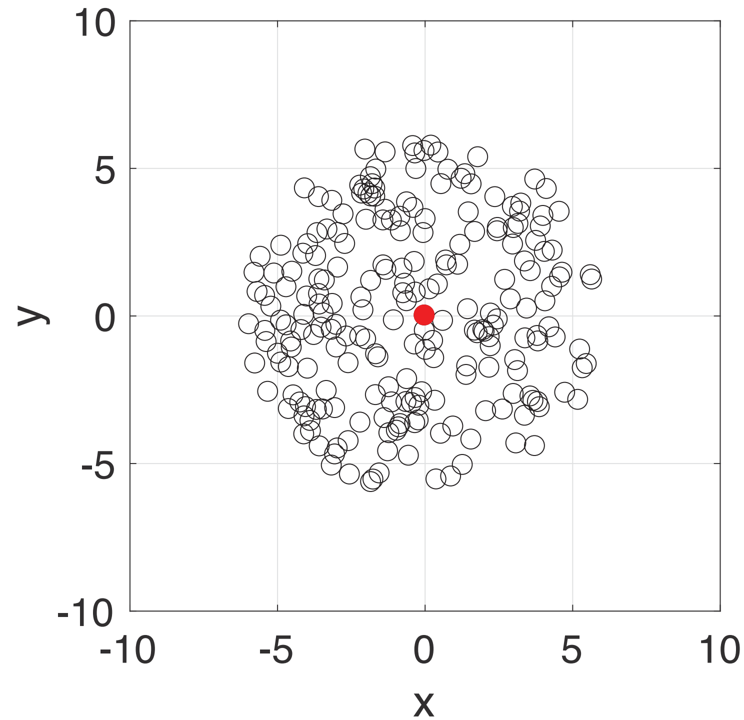
# Outline of talk

1. Brief Intro to UW and BARC
2. Motivation
3. Problem formulation (math)
4. Limits of current approaches
5. Proposed DSR approach
6. **Results & Discussion**
7. Ongoing efforts / Extensions
8. Conclusions

# Results (simulations)

1. **Two cases**: grid-like initial spacing and randomized initial spacing
2. **Comparisons:** with and without DSR

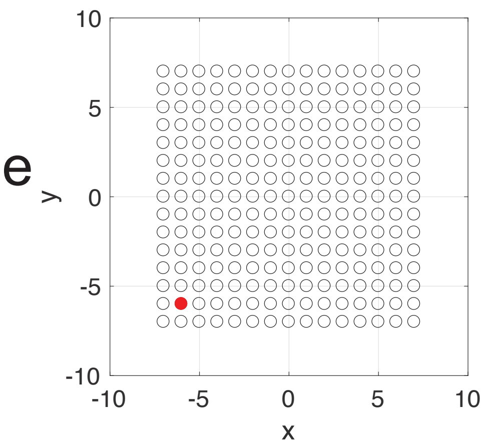


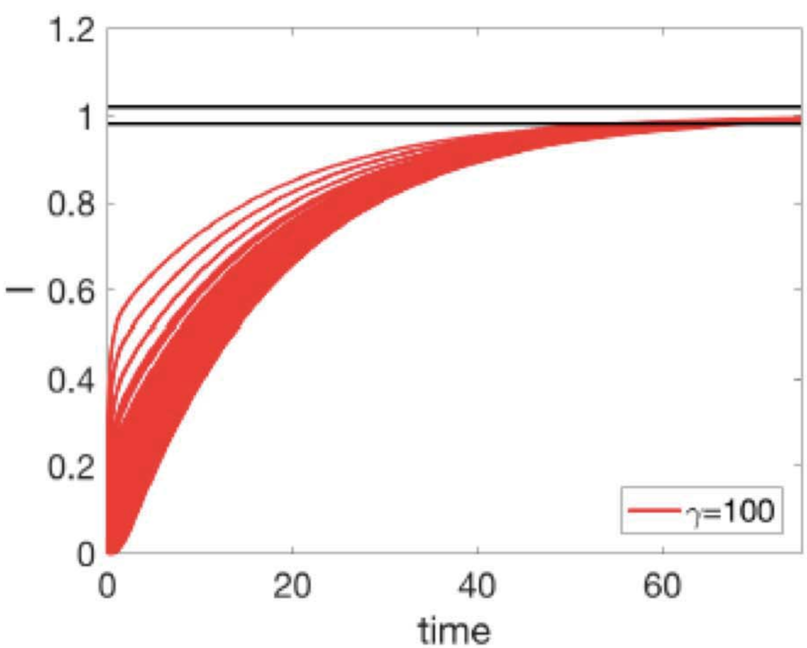



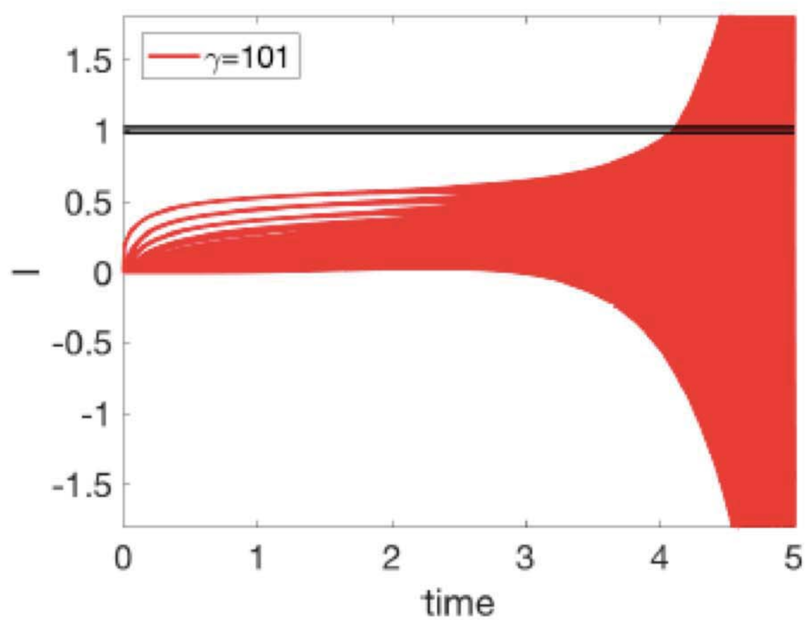

3. Leader shown in red; 125 agents

# Without DSR

1. Fix: Update time = 0.01s
2. Increasing alignment gain speeds response
3. Max alignment gain is about 100
4. With $\gamma$ =100, 2% settling time is 69s (without DSR)
5. Note that $\gamma$ =101 is unstable

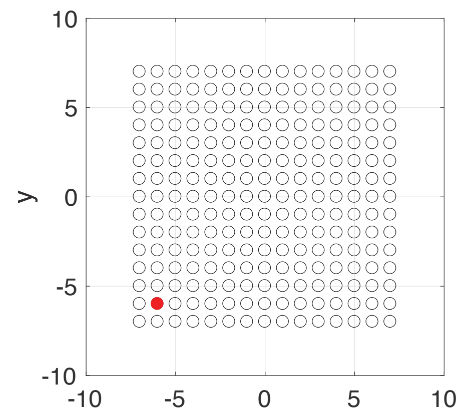


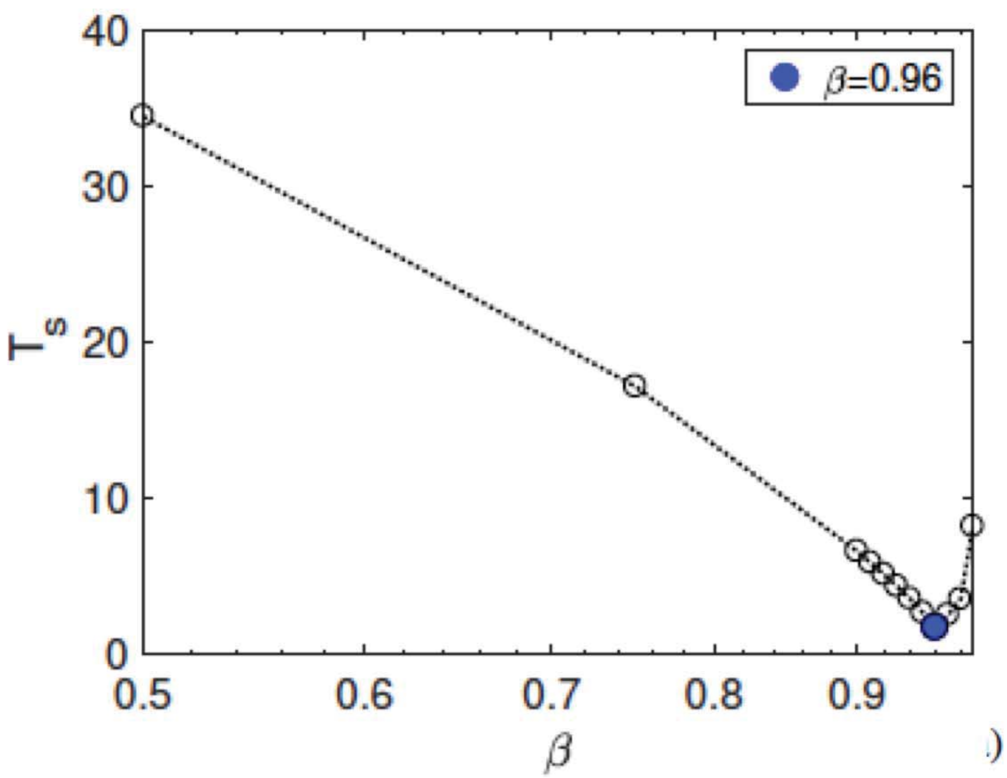



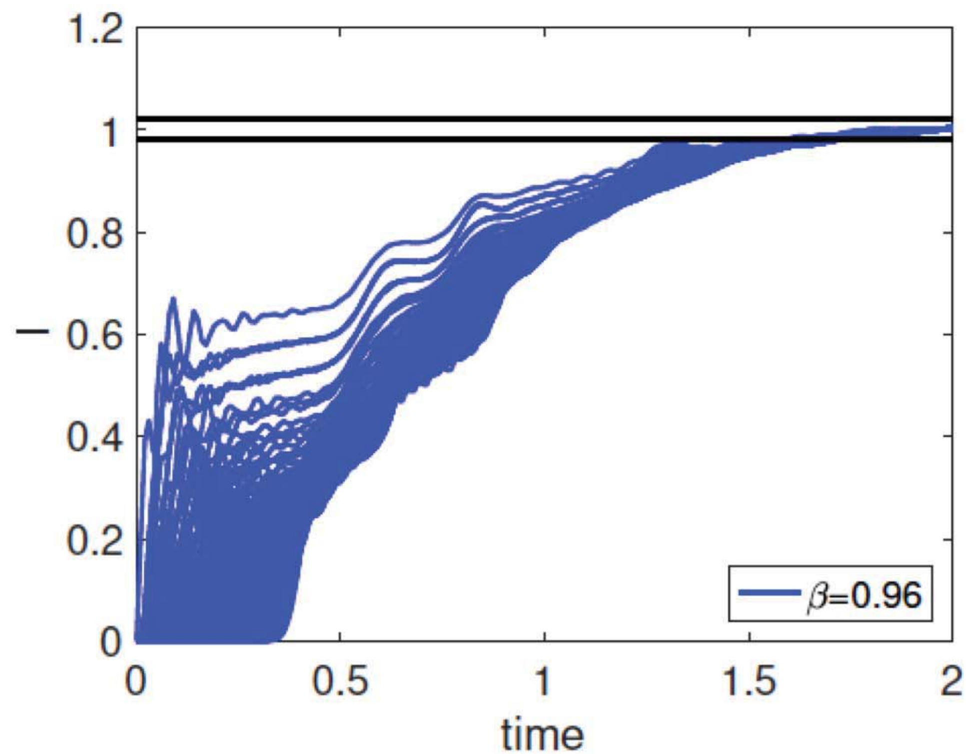



# Faster response with DSR

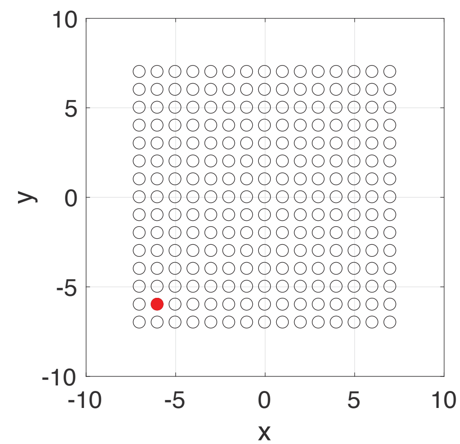
1. With same $\gamma$ =100, and update time = 0.01s
2. Optimal $\beta$ = 0.96 (numerical)
3. **Settling time reduces from 69s to 1.71s**
4. Prediction from optimal $\beta$ expression is 0.967
5. Close to numerical search of 0.96



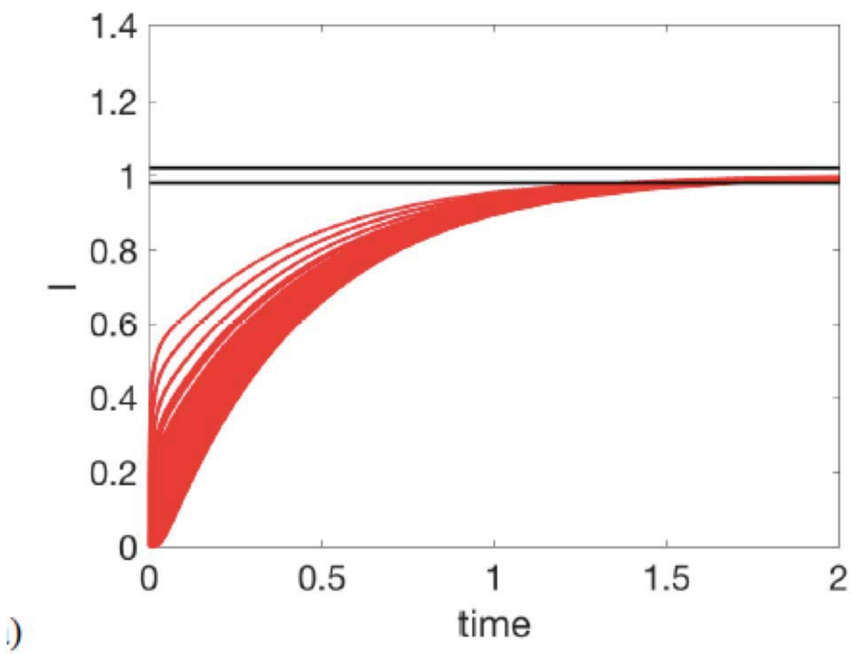



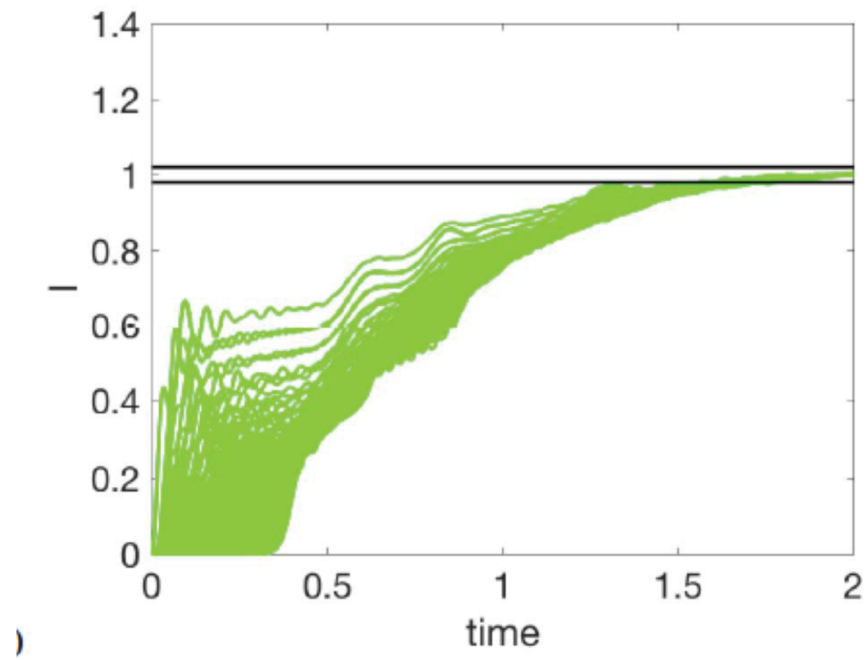



# Other methods (no DSR)



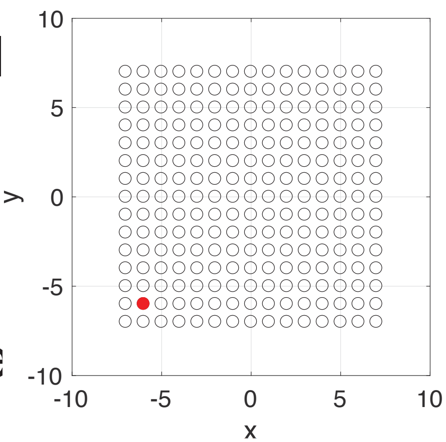

1. Left: increased alignment strength, $\gamma$ =4034, but update time needs to be $\delta_t$ =2.48 X $10^{-4}$ s for 1.71s settling time
2. Right: 2nd order superfluid model with update time $\delta_t$ =1.24 X $10^{-4}$ s
3. DSR only needs update time of $\delta_t$ = 0.01s
4. Bandwidth needs to be 100 time more for other methods

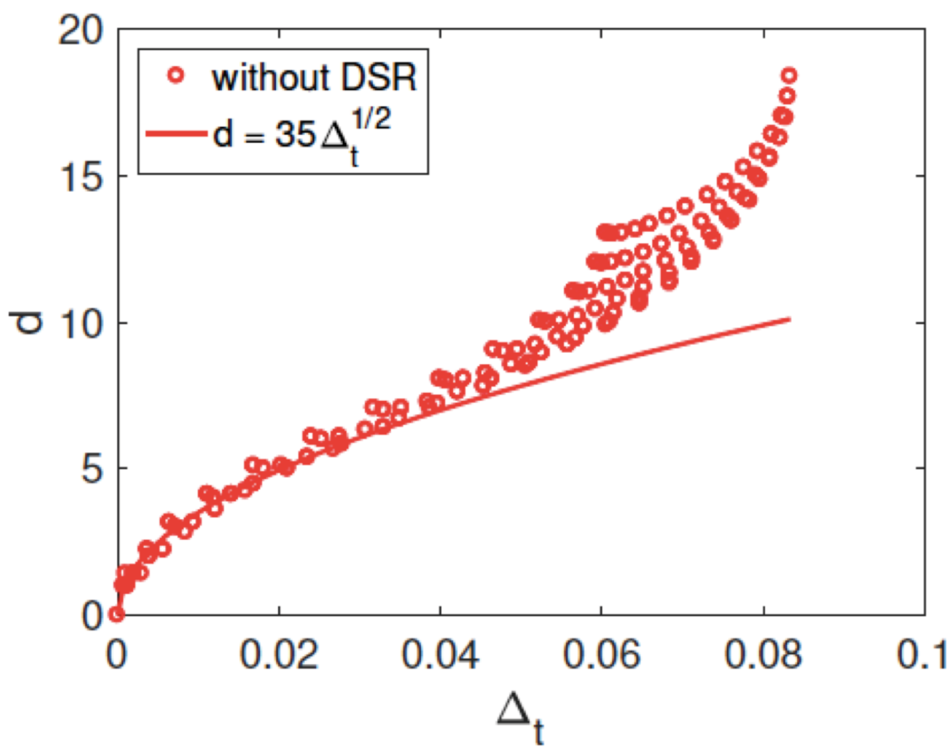


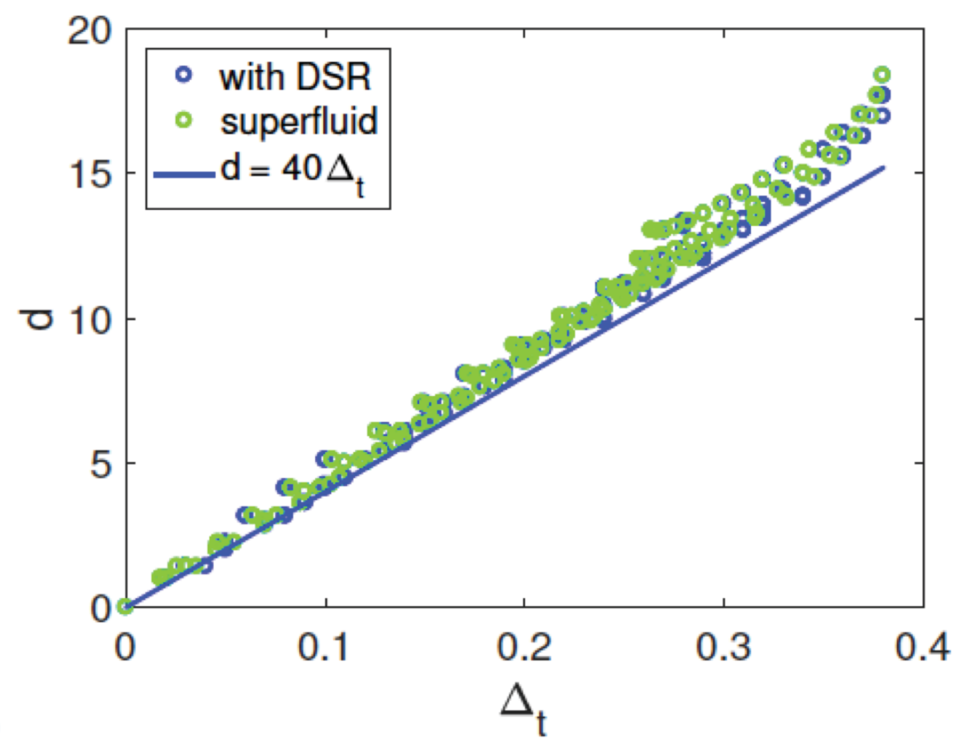



# Captures non-diffusive superfluid-like behavior

1. Left: without DSR info spread d is proportional to square root of elapsed time
2. Right: with DSR info spread d is proportional to elapsed time (as in superfluid model)
3. Spread rate (40m/s) matches theoretical value

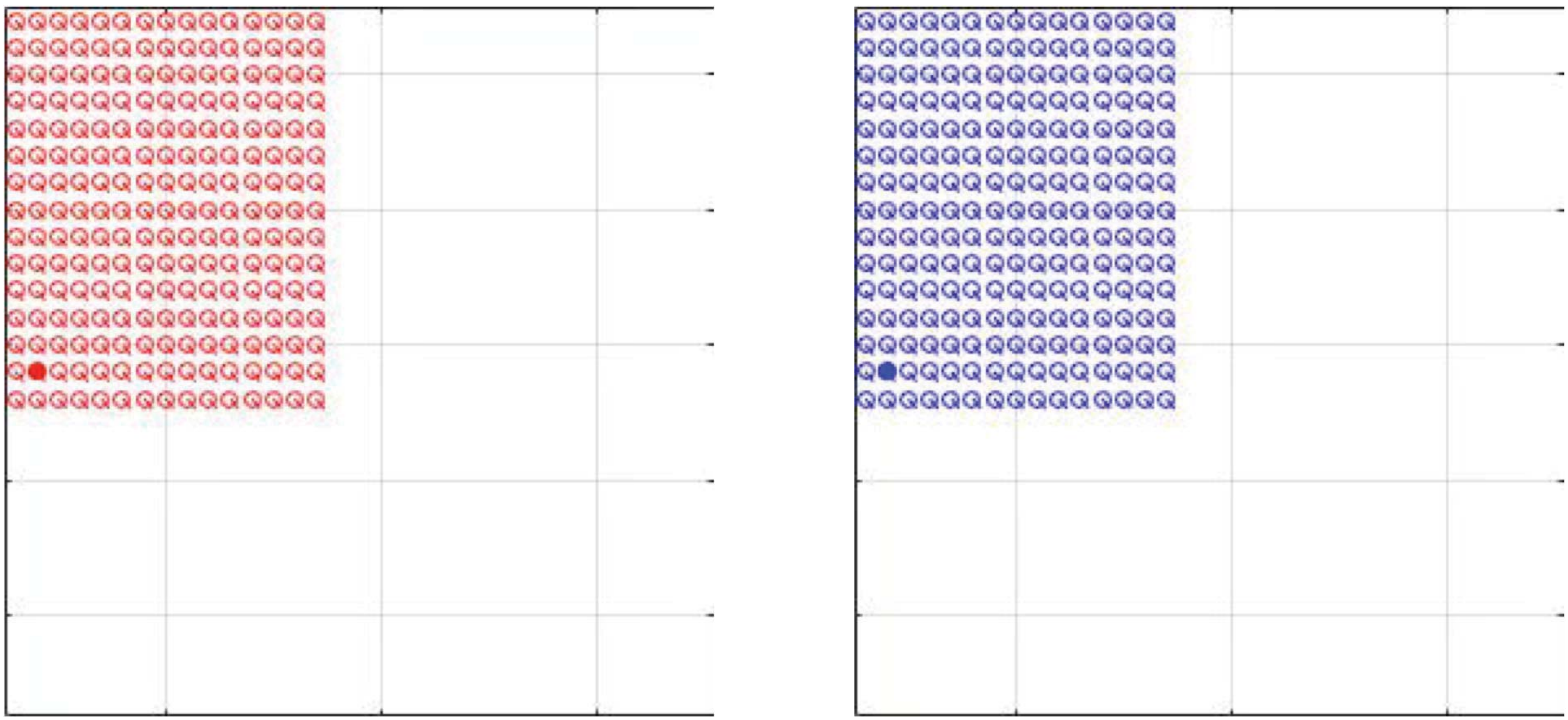



)

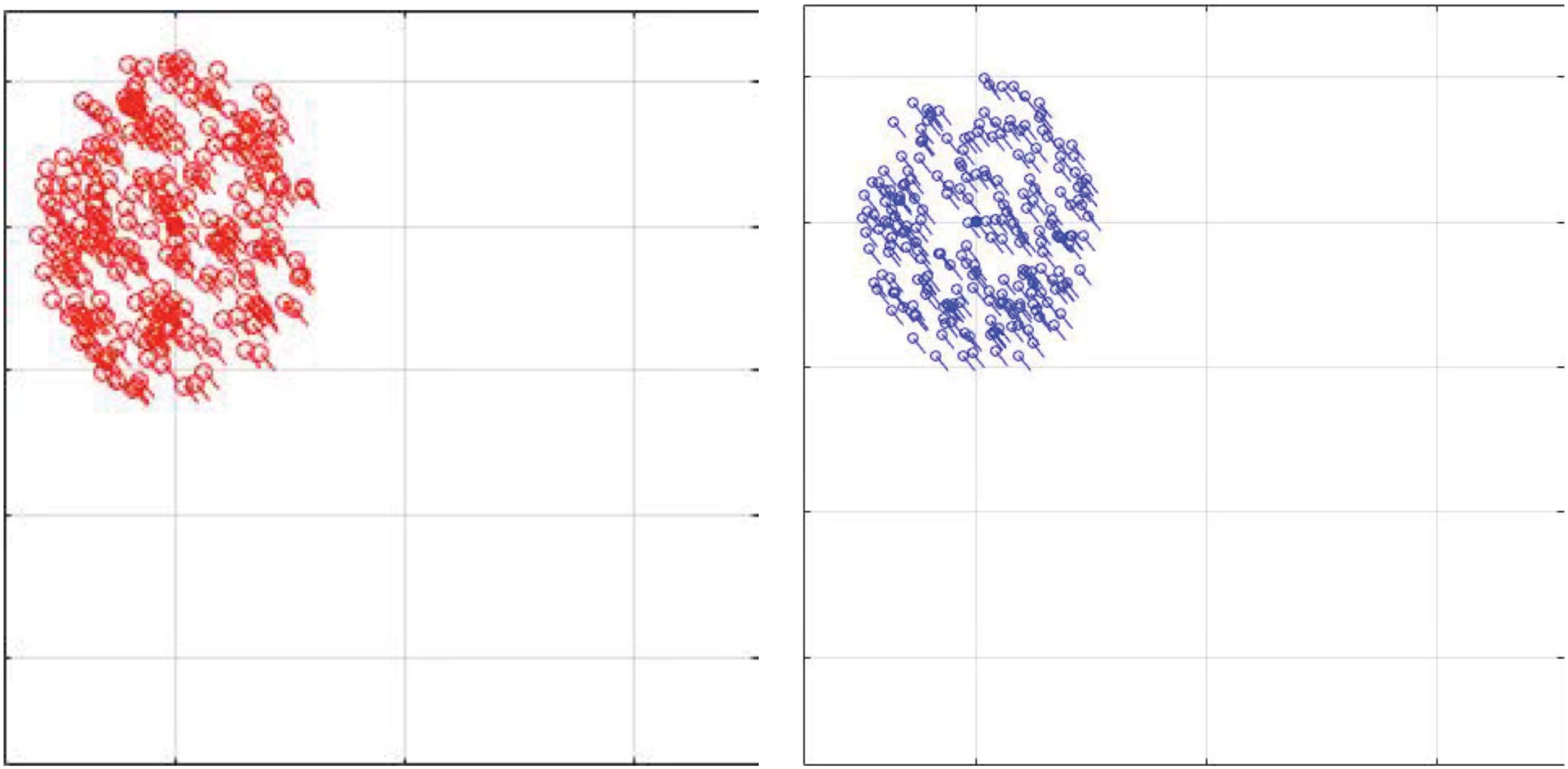
# Simulations with and without DSR



Without DSR With DSR

**DSR substantially improves performance**

# Simulations with and without DSR

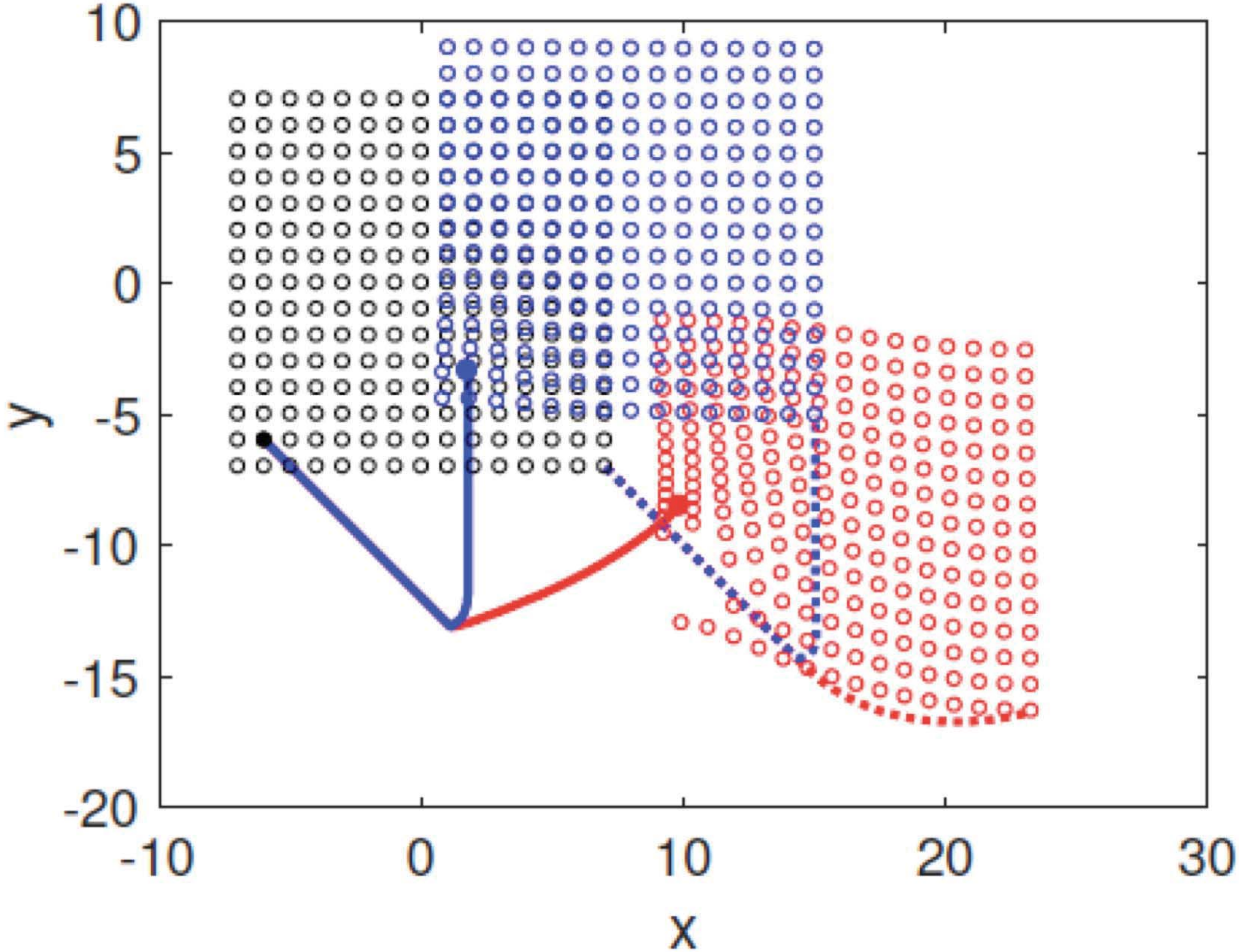


Without DSR With DSR

**DSR substantially improves performance**

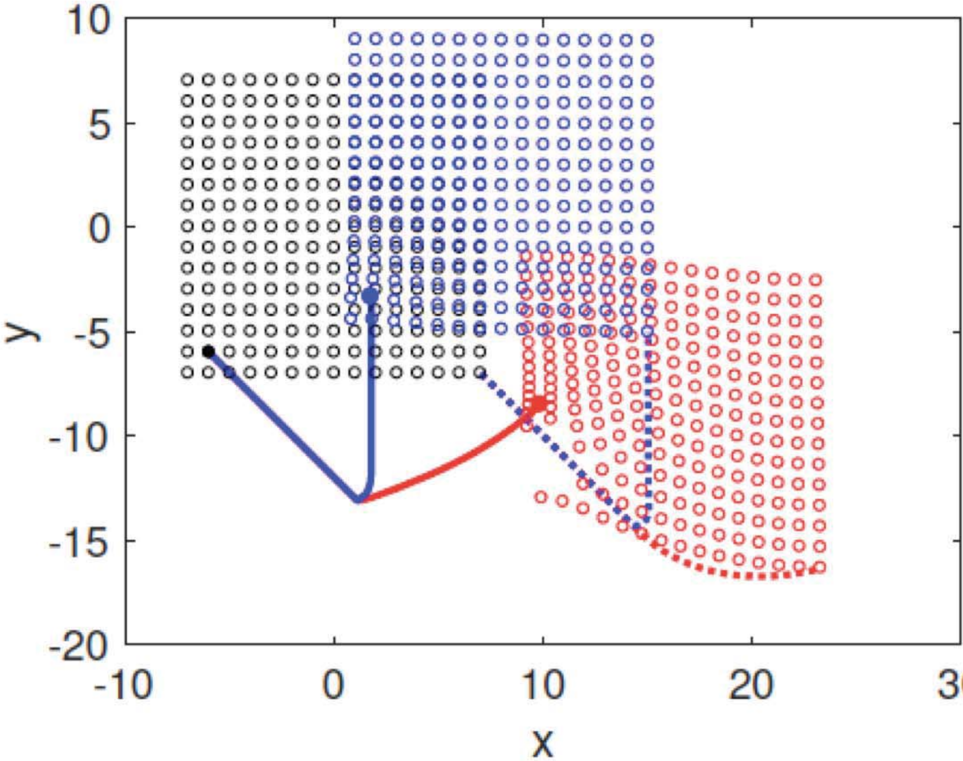
Tighter radius, faster turn maneuvers

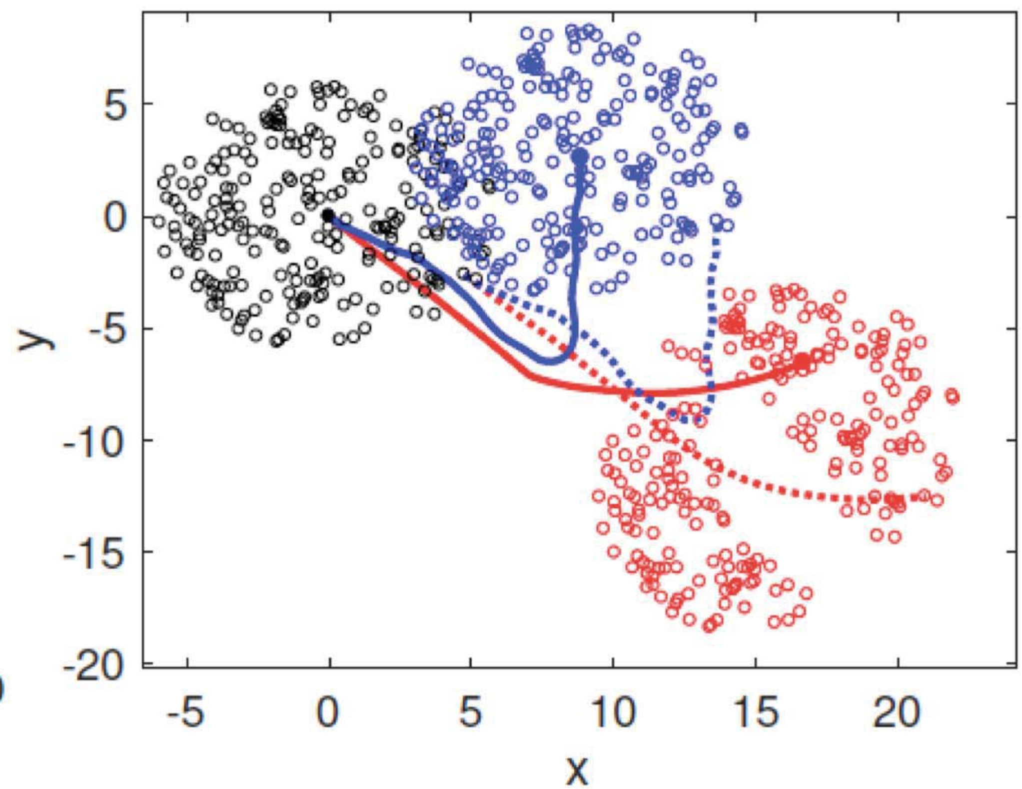
# DSR leads to parallel turns as in nature




Red: Without DSR , Blue: With DSR

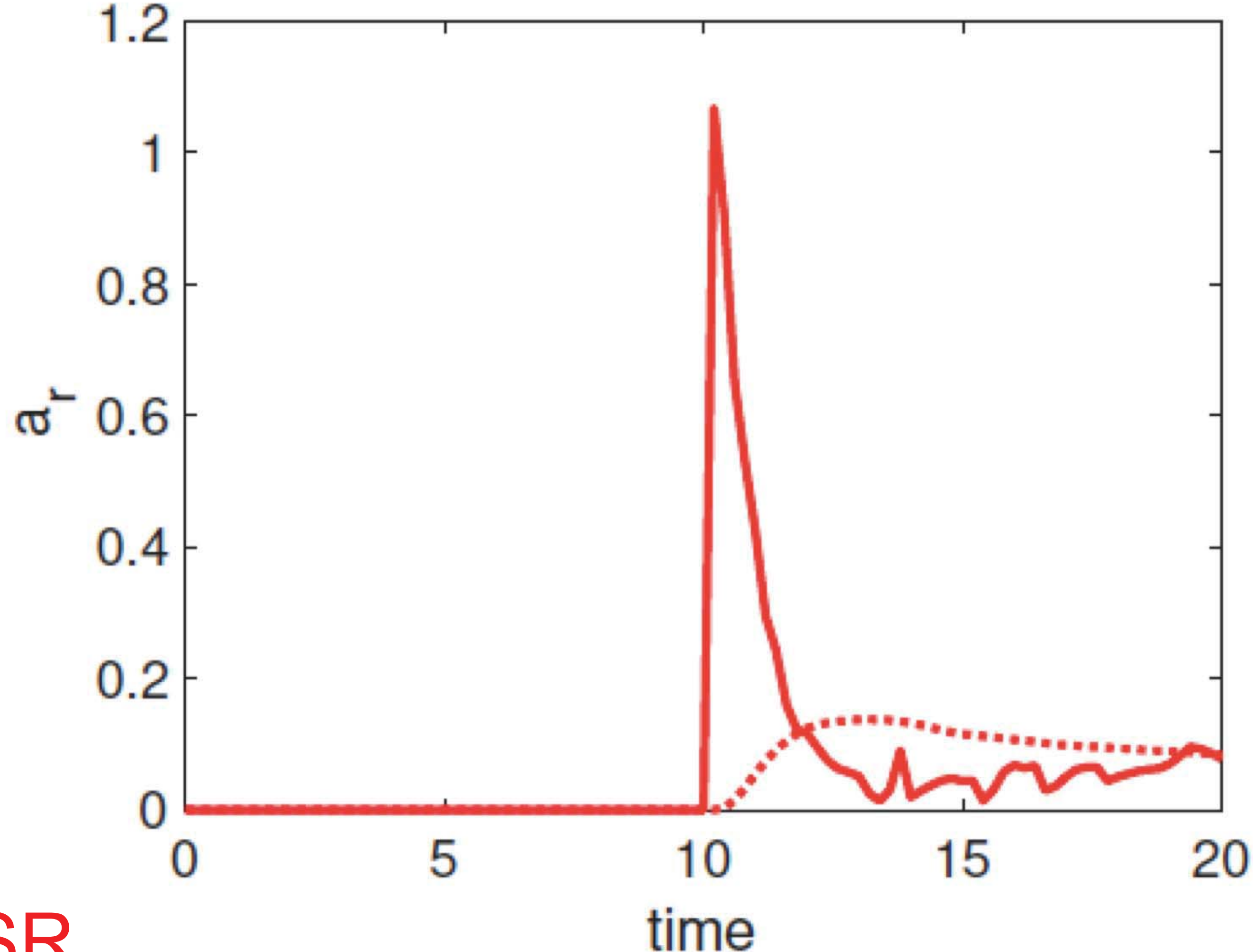
# Similar results for random ICs




Red: Without DSR , Blue: With DSR

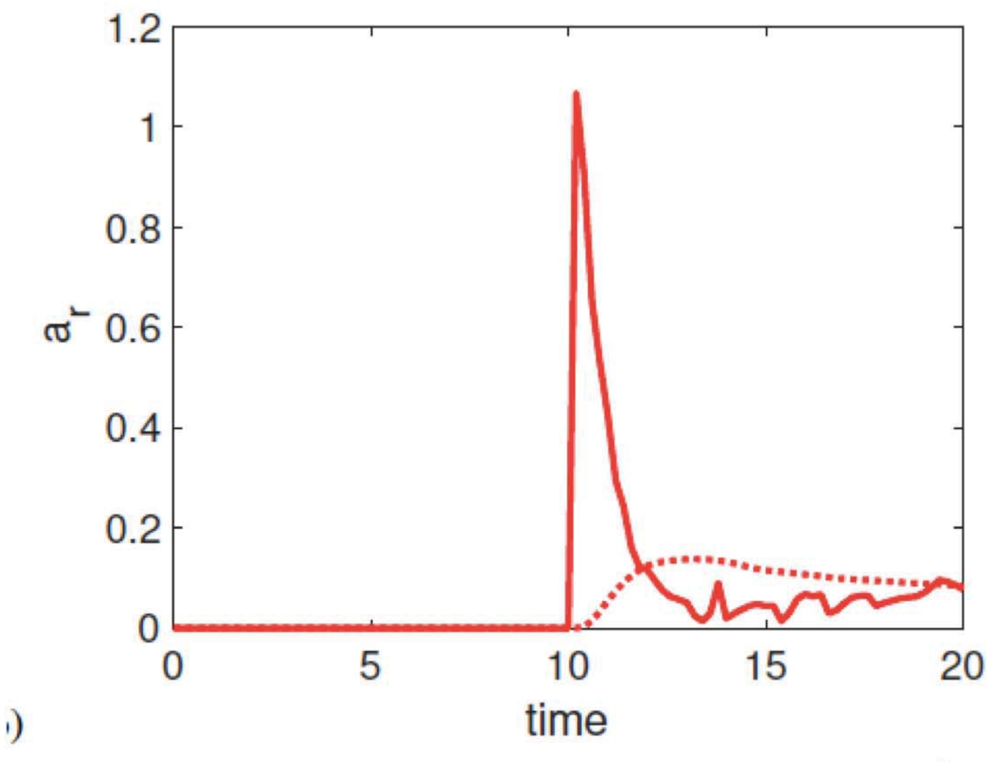
# Turn information decays across network

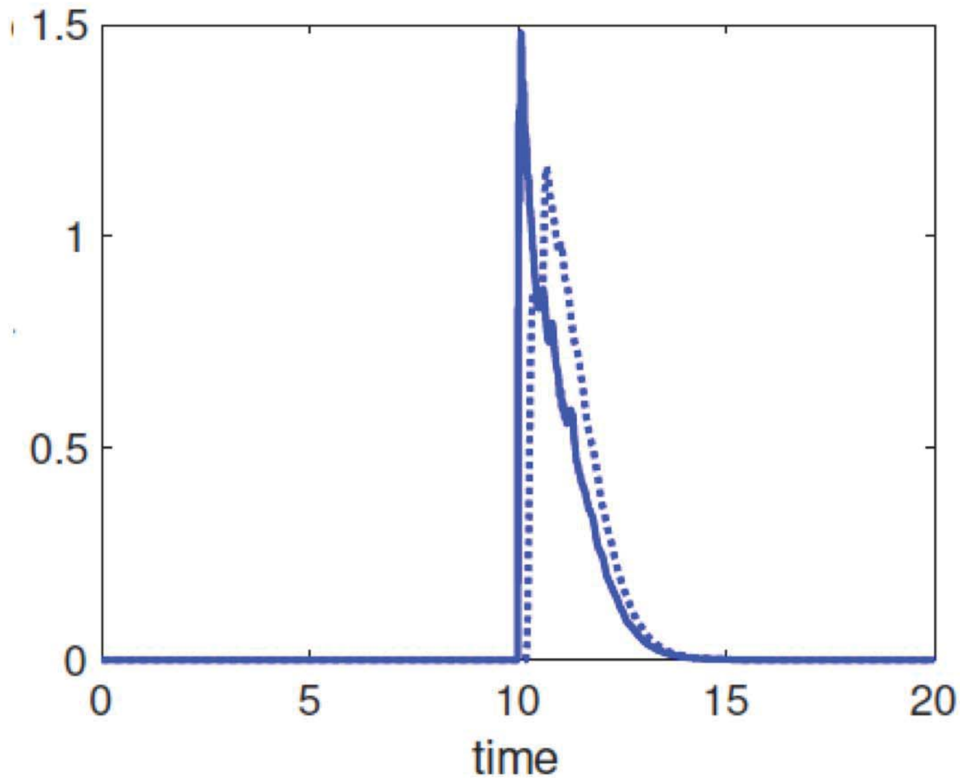



Without DSR

Solid: leader. and dashed follower further away

# Comparison:
# DSR reduces decay of turn information




Red: Without DSR , Blue: With DSR

Solid: leader and dashed follower further away

# Outline of talk

1. Brief Intro to UW and BARC
2. Motivation
3. Problem formulation (math)
4. Limits of current approaches
5. Proposed DSR approach
6. Results & Discussion
7. **Ongoing efforts / Extensions**
8. Conclusions

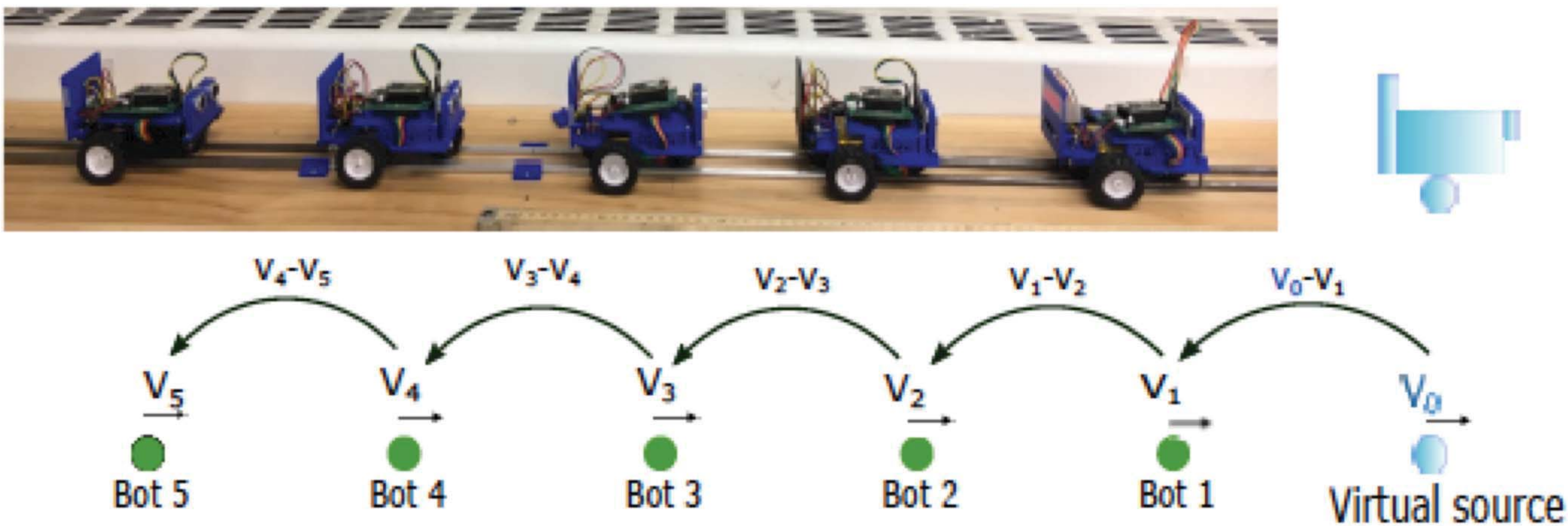
# Ongoing efforts / Extensions

1. **Experimental efforts**
2. General accelerated networks
3. Continuous-time domain

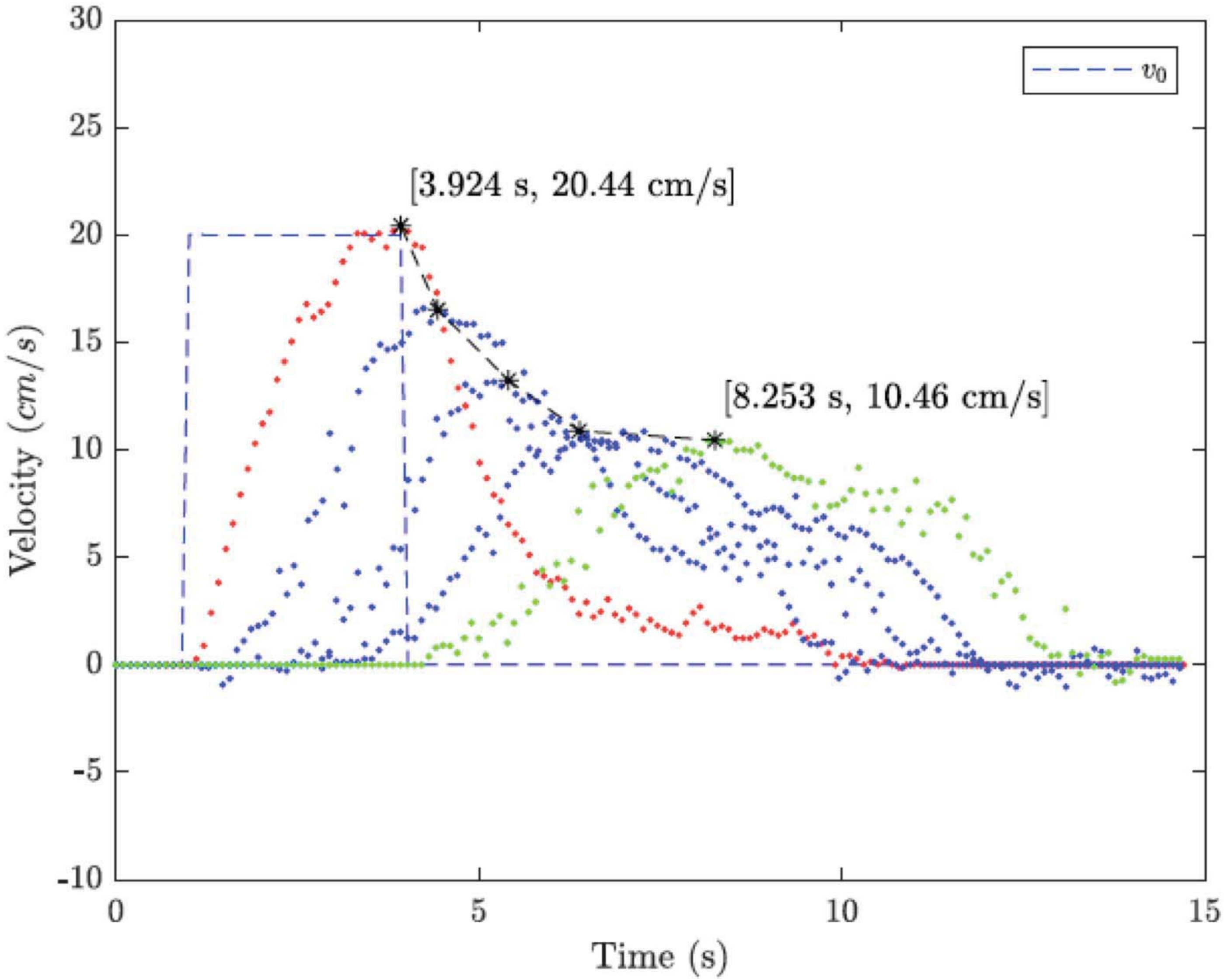
# Experimental efforts: platoons

Ref: Tiwari, Devasia, Submitted to ASME DSCC 2019

# Experimental efforts: platoons

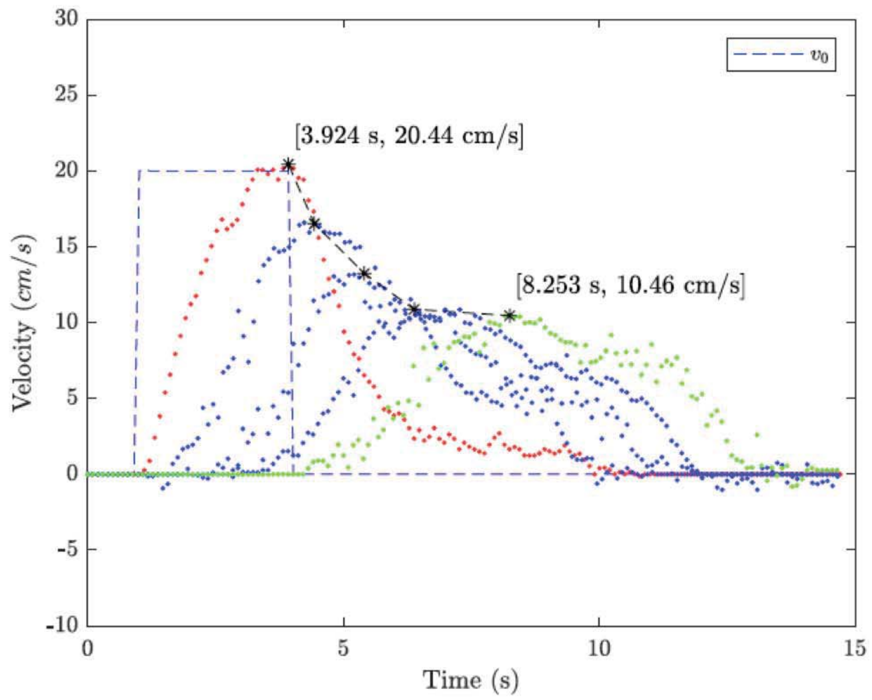


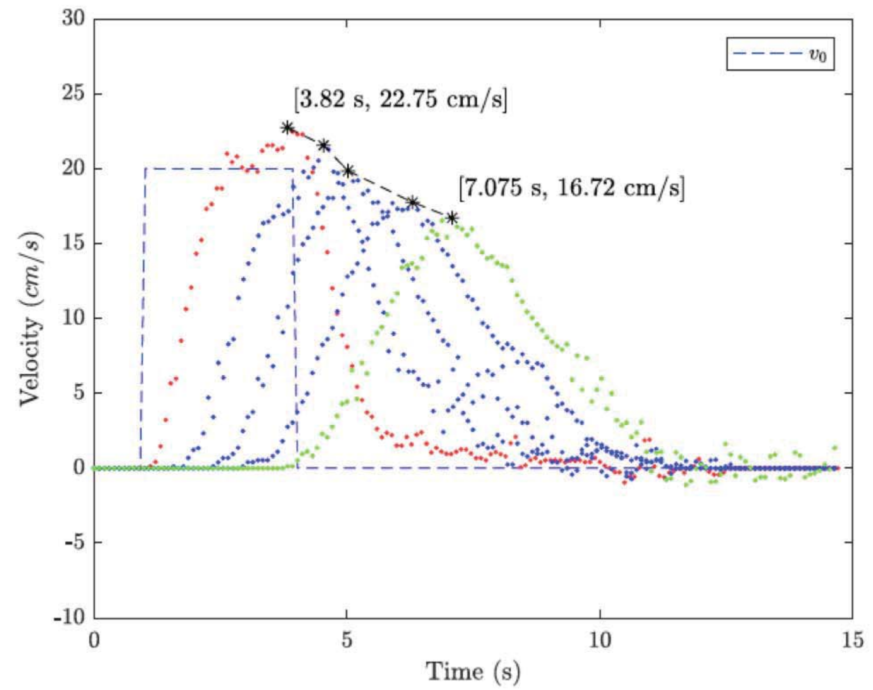
# Platoons (without DSR): damped




**Max velocity decays with distance from source**

# Velocity dispersion reduced with DSR




Without DSR With DSR

DSR improves cohesion of response

# Ongoing efforts / Extensions

1. Experimental efforts
2. **General accelerated networks**
3. Continuous-time domain

# Generalized DSR

- Update law can be considered as the gradient of a potential function, e.g., Richard Murray and R. Olfati-Saber

$$u(\hat{Z}) = -\frac{1}{2}\nabla\Phi_{\mathcal{G}}(\hat{Z}),$$

$$\Phi_{\mathcal{G}}(\hat{Z}) = \frac{1}{2}\sum_{i,j=1}^{n} a_{ij}\left(\hat{Z}_j - \hat{Z}_i\right)^2$$

$$u(\hat{Z}) = -\frac{1}{2}\nabla\Phi_{\mathcal{G}}(\hat{Z}) \;= -L\hat{Z}.$$

$$\begin{aligned}\hat{Z}[k+1] &= \hat{Z}[k] - \gamma\frac{1}{2}\nabla\Phi_{\mathcal{G}}(\hat{Z}) \\ &= \hat{Z}[k] - \gamma L\hat{Z}[k]\end{aligned}$$

# Accelerated Gradient Approach

- Modify the update $u(\hat{Z}) = -\frac{1}{2}\nabla\Phi_{\mathcal{G}}(\hat{Z}) = -L\hat{Z}.$

- to

$$\begin{aligned} u(\hat{Z}[k]) &= -\frac{1}{2}\nabla\Phi_{\mathcal{G}}\left\{\hat{Z}[k] + \beta\left(\hat{Z}[k] - \hat{Z}[k-1]\right)\right\} \\ &\quad + \beta\left(\hat{Z}[k] - \hat{Z}[k-1]\right) \\ &= -L\left\{\hat{Z}[k] + \beta\left(\hat{Z}[k] - \hat{Z}[k-1]\right)\right\} \\ &\quad + \beta\left(\hat{Z}[k] - \hat{Z}[k-1]\right). \end{aligned}$$

- Term outside is the momentum term (discussed earlier) and term inside is the addition by Nesterov

# Results in generalized update

$$\begin{aligned} u(\hat{Z}[k]) &= -\frac{1}{2}\nabla\Phi_{\mathcal{G}}\left\{\hat{Z}[k] + \beta\left(\hat{Z}[k] - \hat{Z}[k-1]\right)\right\} \\ &\quad + \beta\left(\hat{Z}[k] - \hat{Z}[k-1]\right) \\ &= -L\left\{\hat{Z}[k] + \beta\left(\hat{Z}[k] - \hat{Z}[k-1]\right)\right\} \\ &\quad + \beta\left(\hat{Z}[k] - \hat{Z}[k-1]\right). \end{aligned}$$

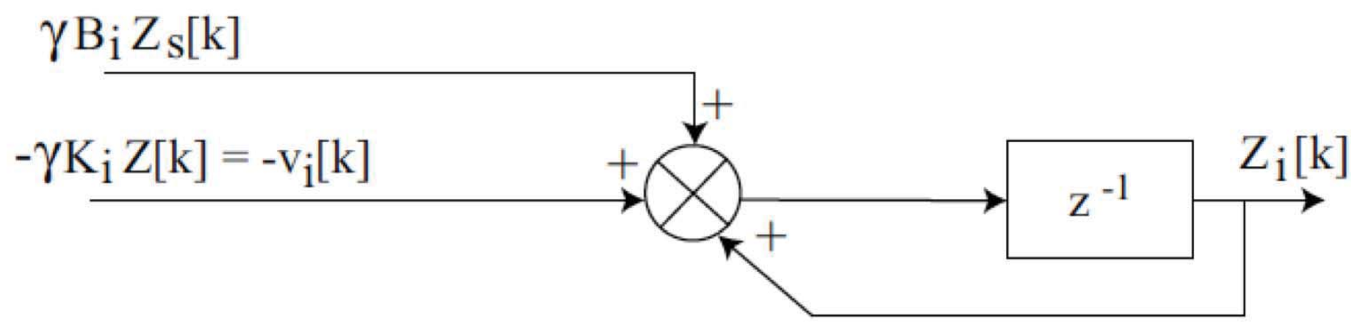
$$Z[k+1] \quad = Z[k] - \gamma K Z[k] + \gamma B Z_s[k]$$

$$\begin{aligned} Z[k+1] \quad &= Z[k] - \gamma K\left(Z[k] + \beta\left(Z[k] - Z[k-1]\right)\right) \\ &\quad + \beta\left(Z[k] - Z[k-1]\right) + \gamma B Z_s[k]. \end{aligned}$$

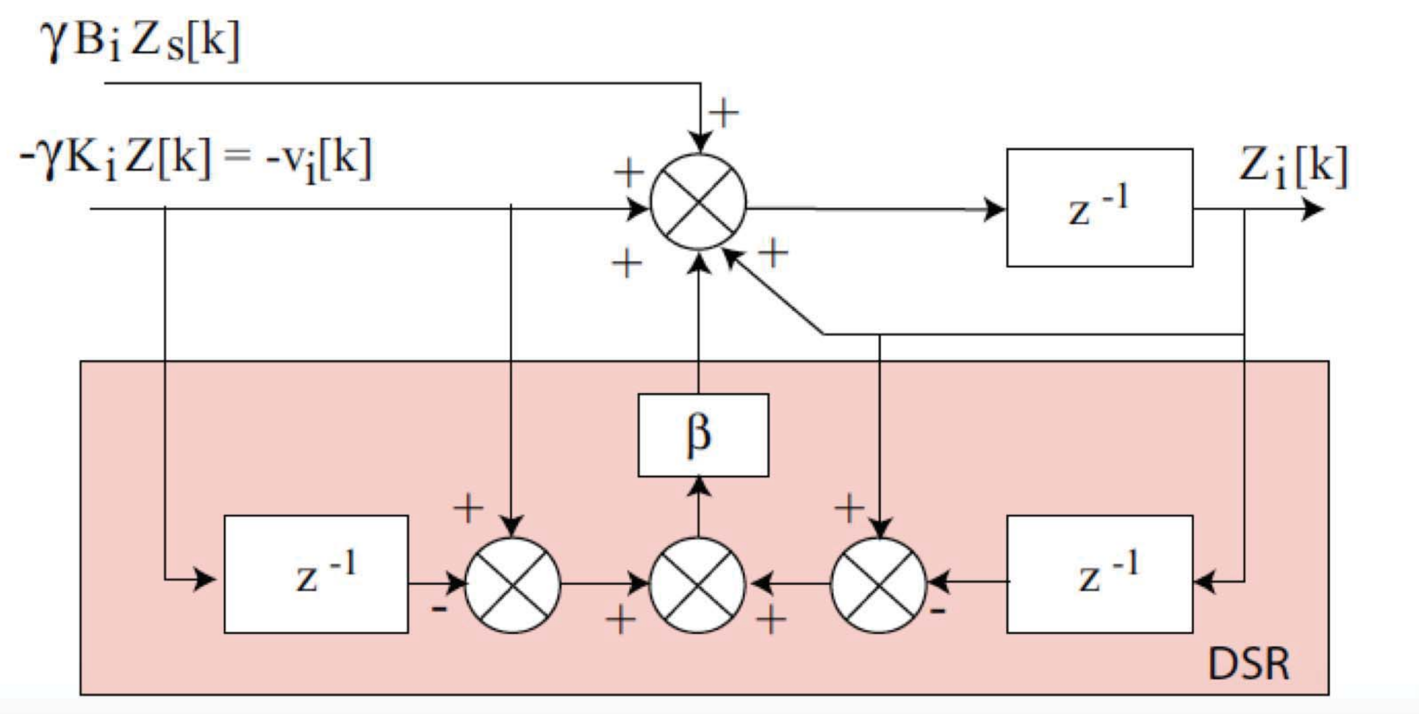
Ref: Devasia, IEEE ICPS, May 2019

# Implementation: still no network modification

- Old update




- Revised update



# Ongoing efforts / Extensions

1. Experimental efforts
2. General accelerated networks
3. **Continuous-time domain**

# For the continuous case

$$\dot{Z}(t) = -\alpha\beta KZ(t) + \alpha\beta Bz_s(t)$$

- Continuous-time networks
- K, B specify the network and access to source information
- Alpha is the alignment gain (as in the discrete version)

# Ideal Update

$$\dot{Z}(t) = -\alpha\beta KZ(t) + \alpha\beta Bz_s(t)$$

$$\dot{Z}(t) = -\alpha\beta KZ(t) + \alpha\beta Bz_s(t) + [I - \beta K]\dot{Z}(t)$$

- Effectively cancels the derivative on the left and leads to perfect tracking (ideally!)

$$\dot{Z}(t) = -\alpha Z(t) + \alpha \mathbf{1}_n z_s(t)$$

# Settling time tuning

$$\dot{Z}(t) = -\alpha\beta KZ(t) + \alpha\beta Bz_s(t)$$

$$\dot{Z}(t) = -\alpha\beta KZ(t) + \alpha\beta Bz_s(t) + [I - \beta K]\dot{Z}(t)$$

- Effectively cancels the derivative on the left and leads to perfect tracking (ideally!)
- Settling time tuned by update gain alpha

$$\dot{Z}(t) = -\alpha Z(t) + \alpha \mathbf{1}_n z_s(t)$$

# Delayed Self Reinforcement

$$\dot{Z}(t) = -\alpha\beta K Z(t) + \alpha\beta B z_s(t)$$

$$\dot{Z}(t) = -\alpha\beta K Z(t) + \alpha\beta B z_s(t) + [I - \beta K]\dot{Z}(t)$$

$$\dot{Z}(t) = \frac{Z(t) - Z(t-\tau)}{\tau}$$

- Derivative information couples to the entire network
- Hence not available
- Use DSR, which approximates the derivative
- Only uses given information from network

# Result: A delay differential equation

$$\dot{Z}(t) = -\alpha\beta K Z(t) + \alpha\beta B z_s(t)$$

$$\dot{Z}(t) = -\alpha\beta K Z(t) + \alpha\beta B z_s(t) + [I - \beta K]\dot{Z}(t)$$

$$\dot{Z}(t) = \frac{Z(t) - Z(t-\tau)}{\tau}$$

$$\dot{Z}(t) = U = AZ(t) + A_d Z(t-\tau) + B_d z_s(t)$$

$$A = -\alpha\beta K + \frac{1}{\tau}[I - \beta K]$$

$$A_d = -\frac{1}{\tau}[I - \beta K]$$

$$B_d = \alpha\beta B.$$

# Outline of talk

1. Brief Intro to UW and BARC
2. Motivation
3. Problem formulation (math)
4. Limits of current approaches
5. Proposed DSR approach
6. Results & Discussion
7. Ongoing efforts / Extensions
8. **Conclusions**

# Conclusions

- **Current swarm models**
  - Diffusion-like information spread:
  - 1) proportional to square root of time
  - 2) dampens out over space
- **Use of DSR**
  - wave-like information spread
  - Information spread proportional to time, with smaller decay
  - Increases synchronization (information transfer rate)
- **Bandwidth/Sensing/performance:**
  - DSR does not require additional sensing
  - DSR does not require increased bandwidth
  - DSR improve information transfer rate though the swarm
  - **DSR Improves settling time to new orientation from 69 s to 1.7 s**
- **So worth a try in your implementations!**

# Thank you

Swarm-like simulation with DSR