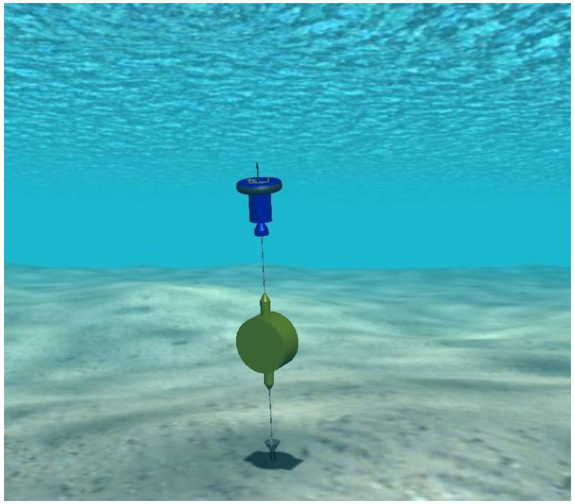
Adaptive Decentralized Control of Underwater Sensor Networks for Modeling Underwater Phenomena

Authors: Carrick Detweiler,Marek Doniec, Mingshun Jiang, Mac Schwager,Robert Chen, Daniela Rus

Speaker: Pradeep Kumar Ratnala

Modeling underwater phenomena

- Detecting and measuring the tidal front
- Chromophoric Dissolved Organic Matter (CDOM)
- An understanding of CDOM dynamics important for:
  - Remote sensing
  - Estimating light penetration
- Improved understanding of CDOM dynamics possible using sensor networks



http://www.subsea-tech.com

Underwater sensor networks

Understanding the dynamics of bodies of water → Need for sensor measurements over the full volume of water

- Challenge :
  - High density placement

Current systems

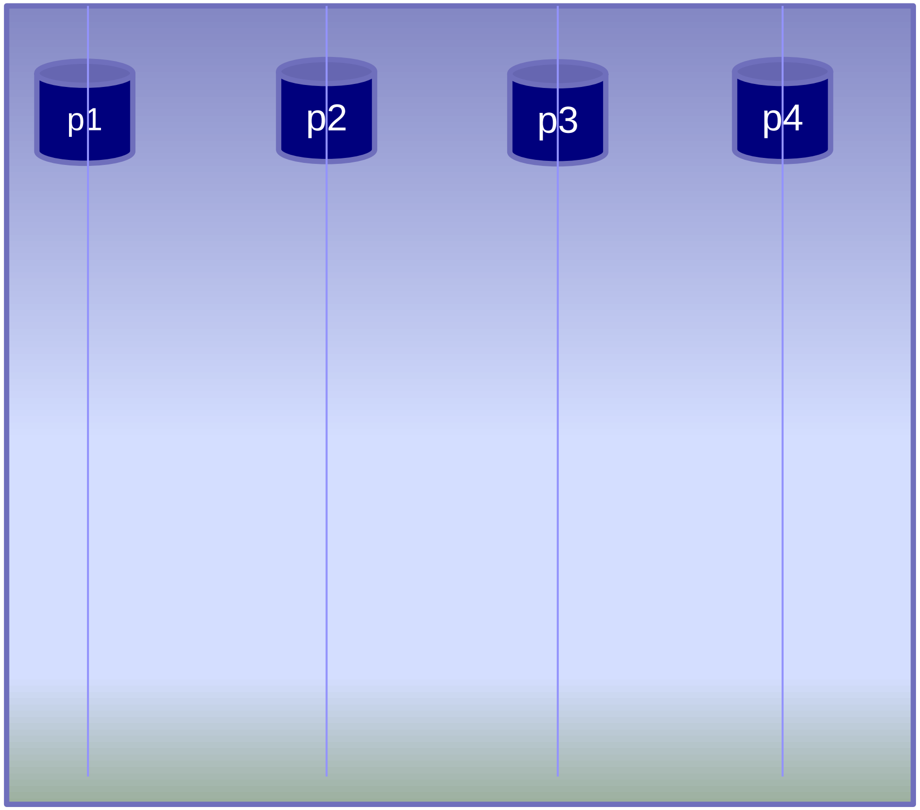
- Current systems:
  - Static sensor buoys
  - Ships/ROVs/AUVs
  - Water column profilers

- Problems:
  - Cost
  - Not adaptive
  - No Communication

→ Solving this requires algorithms and systems that enable adaptive and decentralized sensing

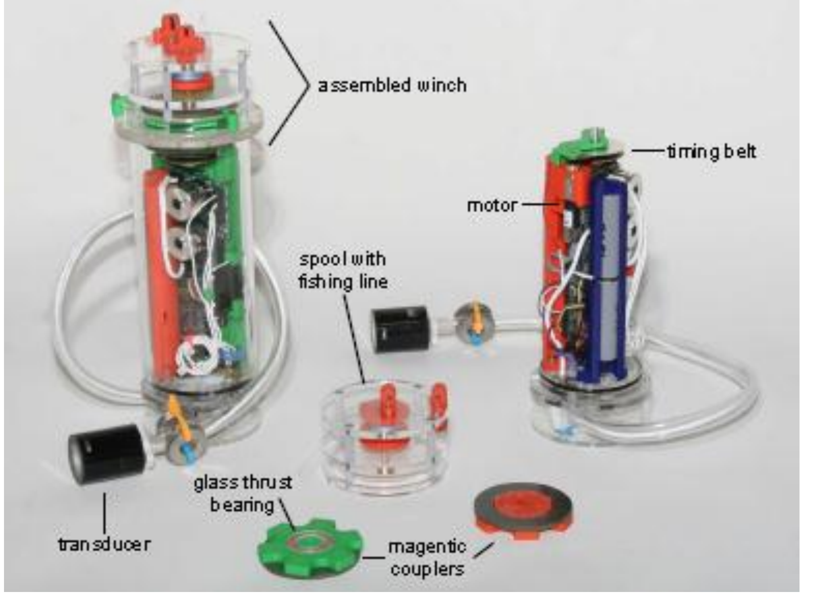
Dynamic depth adjustment algorithm

- Decentralized
- Adaptive
- Neighbor communication
- Runs online
- Converges to a local minimum



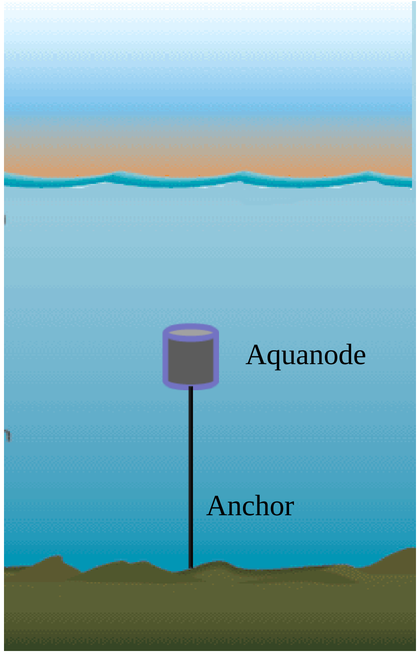
Underwater sensor network platform

- Base sensor node hardware – AQUANODE
  - ARM7TDMI processor
  - 40kB of RAM and 512kB on-chip flash
  - Pressure and Temperature sensors
  - 10W acoustic modem
  - Lithium-ion batteries (60 Whr of energy)



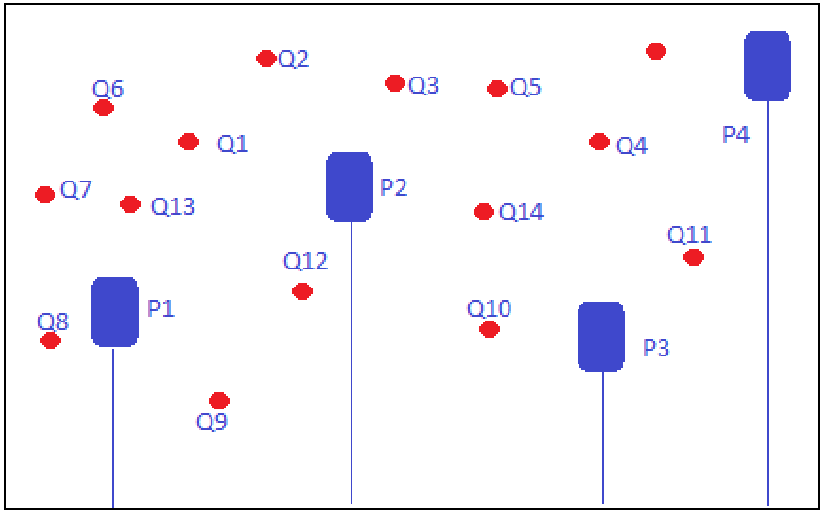
Underwater sensors - Depth adjustment

- AQUANODE extended with autonomous depth adjustment facility
- Anchored at bottom & float mid-water column
- Winch driven by a 1.5A motor controller
- Depth adjustment speed of 0.5 m/s



Decentralized control algorithm – Problem formulation

Given N sensors at locations $p_1 \ldots p_N$, and the set Q with all points in the region of interest, optimize their positions for providing the most information about the change in the values of all other positions $q \in Q$



Decentralized control algorithm – Objective function

- For the point of interest q_1 ,we want to position p_1 such that :
  $Cov(p_1,q_1)$ is maximized

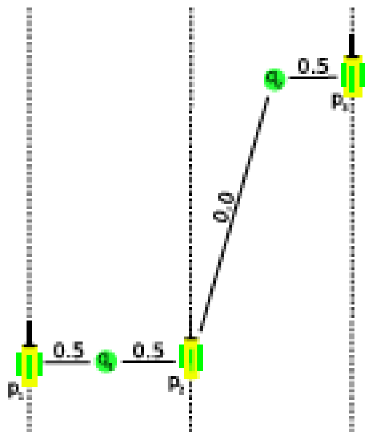
- For n sensors,

$$\arg\max_{p_i} \sum_{i=1}^{N} Cov(p_i, q_1)$$

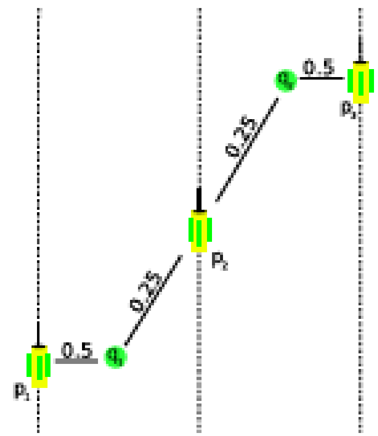
- For M points of interest,

$$\arg\max_{p_i} \sum_{j=1}^{M} \sum_{i=1}^{N} Cov(p_i, q_j)$$

Objective function



case A



case B

$$\arg\max_{p_i} \sum_{j=1}^{M} \sum_{i=1}^{N} Cov(p_i, q_j)$$

Minimize

$$\arg\max_{p_i} \sum_{j=1}^{M} \left(\sum_{i=1}^{N} Cov(p_i, q_j) \right)^{-1}$$

Total Cost Function:

$H(p_1 \ldots p_N) = \int g(q, p_1 \ldots p_N) dq + \sum_{i=1}^{n} \emptyset(p_i)$

General decentralized controller

- Goal is to minimize the objective function

$$\frac{\partial \mathcal{H}}{\partial z_i} = \frac{\partial}{\partial z_i} \int_Q g(q, p_1, \ldots, p_N)\, dq + \frac{\partial}{\partial z_i} \sum_{j=1}^{N} \phi(p_j)$$

$$= \int_Q -g(q, p_1, \ldots, p_N)^2 \frac{\partial}{\partial z_i} f(p_i, q)\, dq + \frac{\partial}{\partial z_i} \phi(p_i)$$

- Control input for each sensor

$$\boxed{\dot{p}_i = -k \frac{\partial \mathcal{H}}{\partial z_i}}$$

where k is some scalar constant

Decentralized control algorithm: Covariance models

- Multivariate Gaussian Model

$$F(p_i, q) = Cov(p_i, q))$$

$$= Ae^{-\left(\frac{(x_i-x_q)^2+(y_i-y_q)^2}{2\sigma_s^2}+\frac{(z_i-z_q)^2}{2\sigma_d^2}\right)}$$

- Model-based covariance:
  - Boston Harbor Model

Pseudo code

```
Procedure UPDATEDEPTH(p1…pN)
    integral<- 0
    for x=xmin to xmax do
            for y=ymin to ymax do
                    for z=zmin to zmax do
                            sum<-0
                            for i= 1 to N do
                                    sum+=F(p_i,x,y,z)
                            end for
                            integral += (-1/sum^2) * FDz(p_i,x,y,z)
                    end for
            end for
    end for
    delta = K * integral
    if delta > maxspeed then
            delta = maxspeed
    end if
    If delta < -maxspeed then
            delta = -maxspeed
    end if
    changeDepth(delta)
end Procedure
```

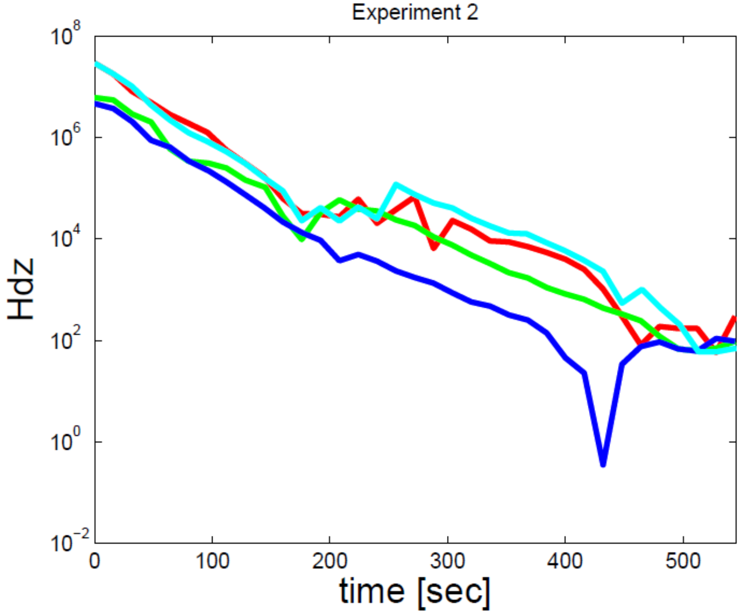
Simulation & experiments

- Matlab simulation
- Lab & Pool hardware experiments
  - Gaussian covariance model
  - Numerical covariance model
- River hardware experiment
  - Changing covariance

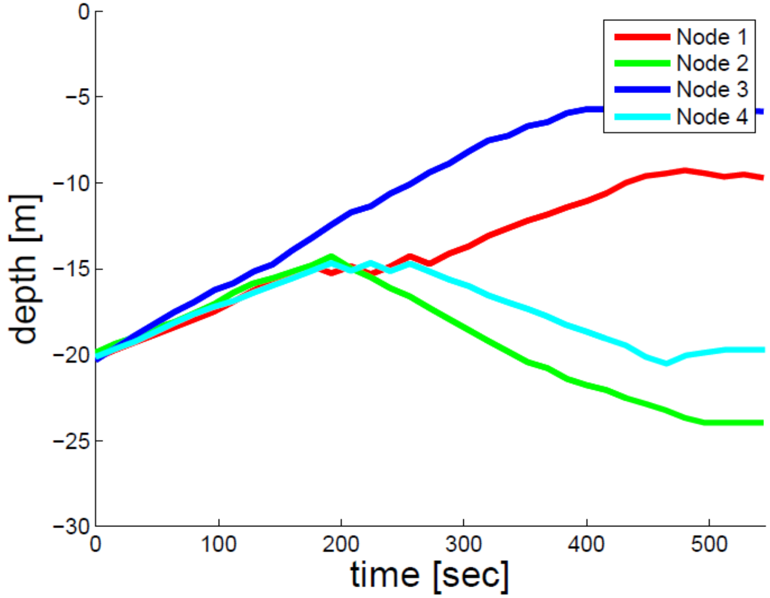
Results (lab & pool experiment)

	Node 0	Node 1	Node 2	Node 3
Bucket 1 Start Bucket 1 Final	10.0m 10.3m	10.0m 24.1m	10.0m 5.9m	10.0m 19.7m
Bucket 2 Start Bucket 2 Final	20.0m 19.8m	20.0m 5.9m	20.0m 23.8m	20.0m 10.2m
Bucket 3 Start Bucket 3 Final	3.7m 9.5m	7.8m 22.9m	12.2m 23.9m	15.9m 9.6m
Pool 1 Start Pool 1 End	10.2m 20.6m	9.9m 6.9m	10.1m 24.1m	9.8m 10.2m
Pool 2 Start Pool 2 End	20.0m 9.5m	20.1m 23.9m	20.3m 5.6m	20.1m 18.8
Pool 3 Start Pool 3 End	20.2m 9.6m	19.9m 24.0m	20.3m 5.8m	20.1m 19.7m

Results (lab & pool experiment) (II)

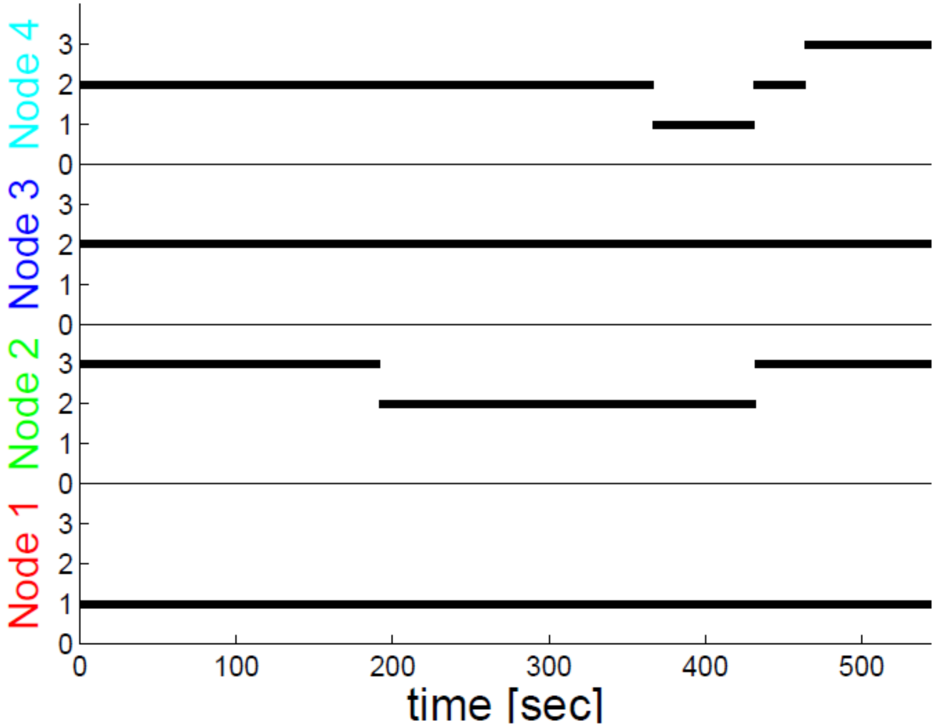


$\frac{\partial H}{\partial z_i}$ vs time

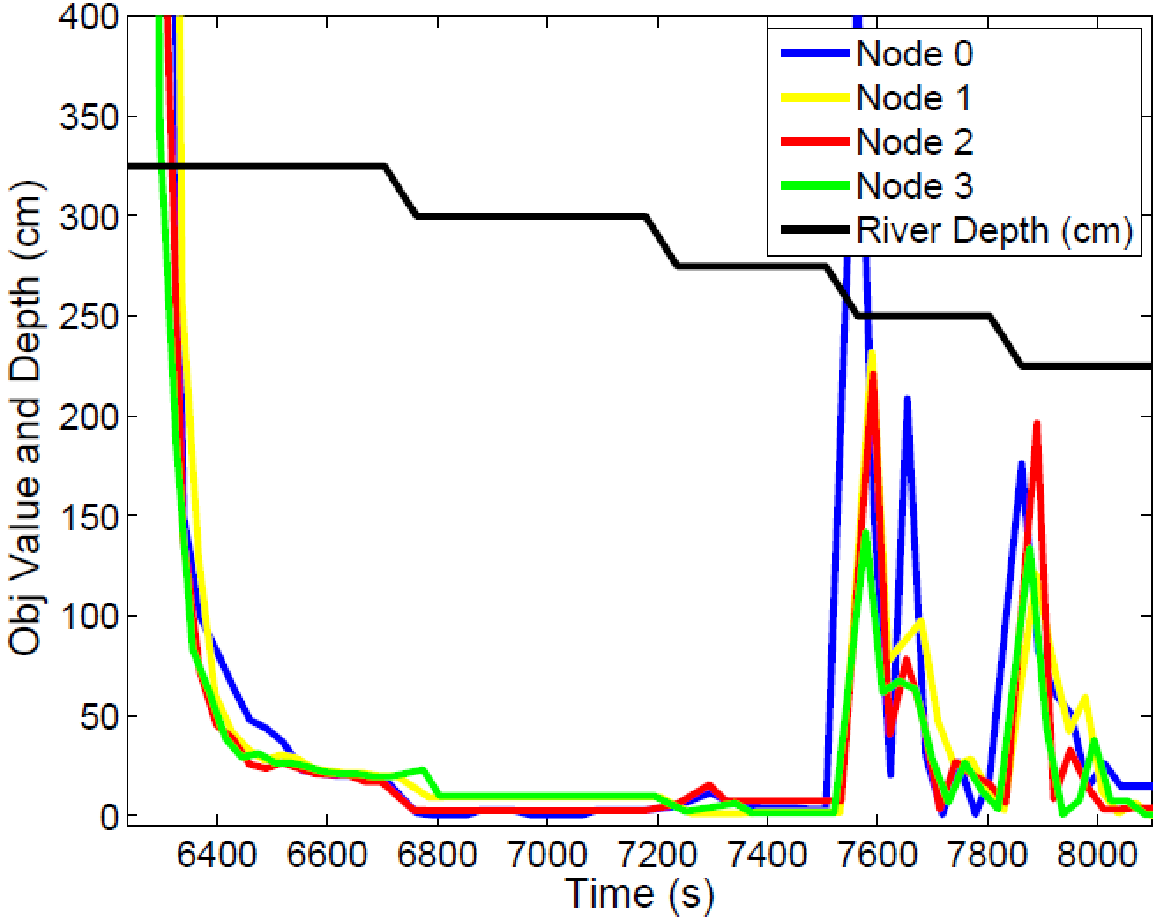


Communication performance

- Number of neighbors used to calculate the objective function

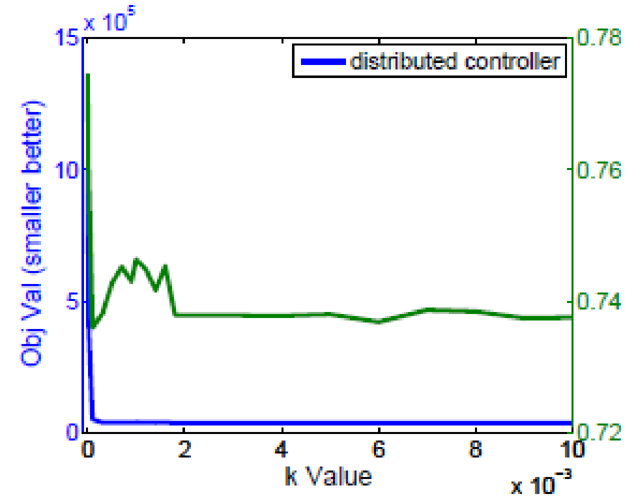


Results (River hardware experiment)

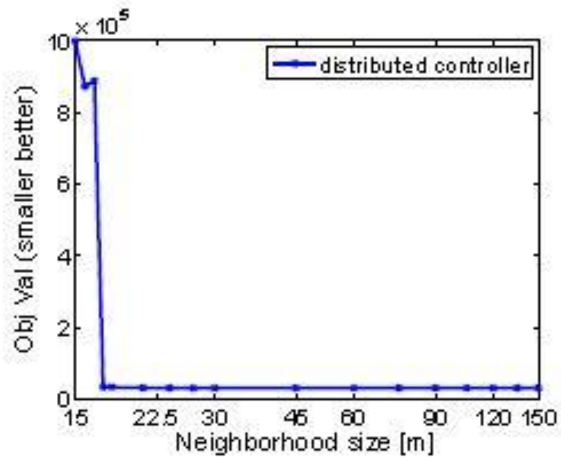


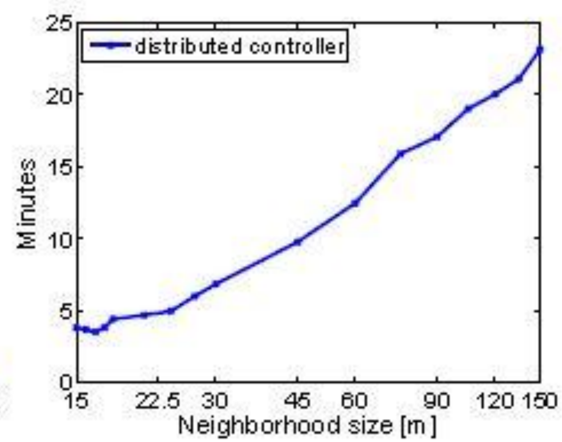
Parameter sensitivity

- Changing k



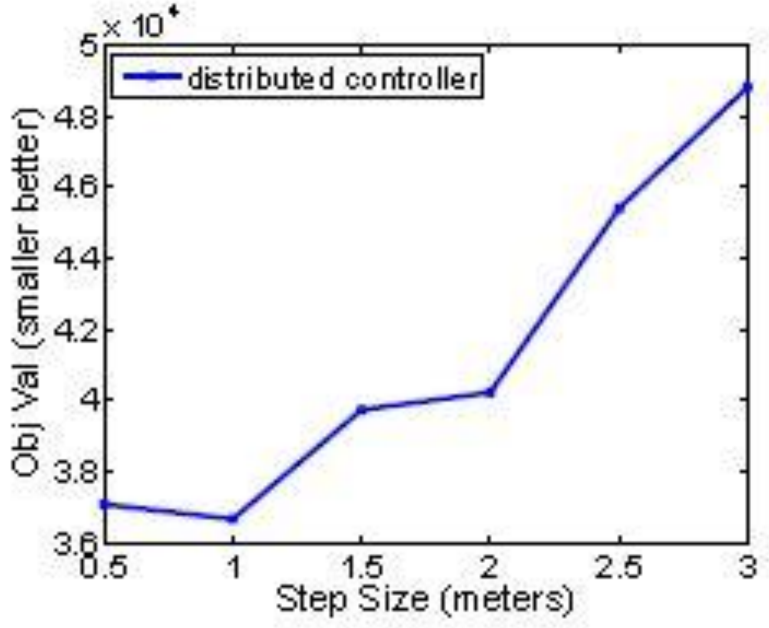
- Changing neighbourhood size

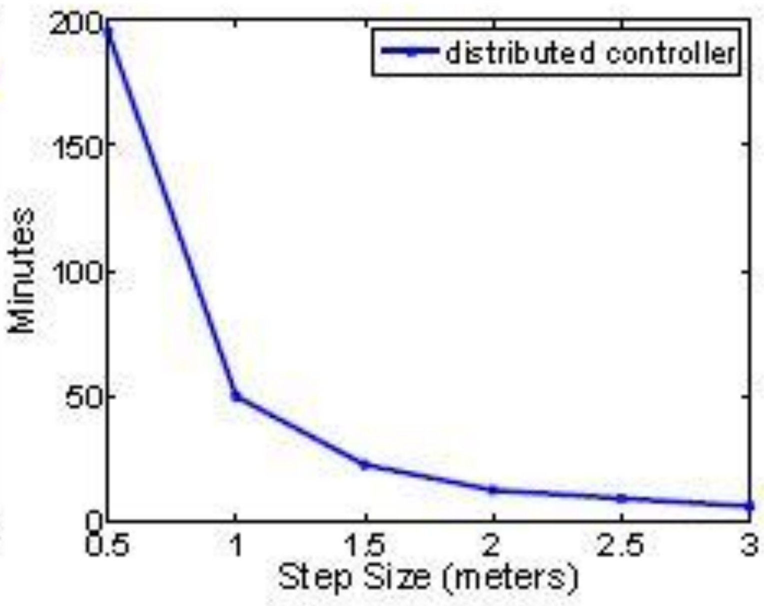




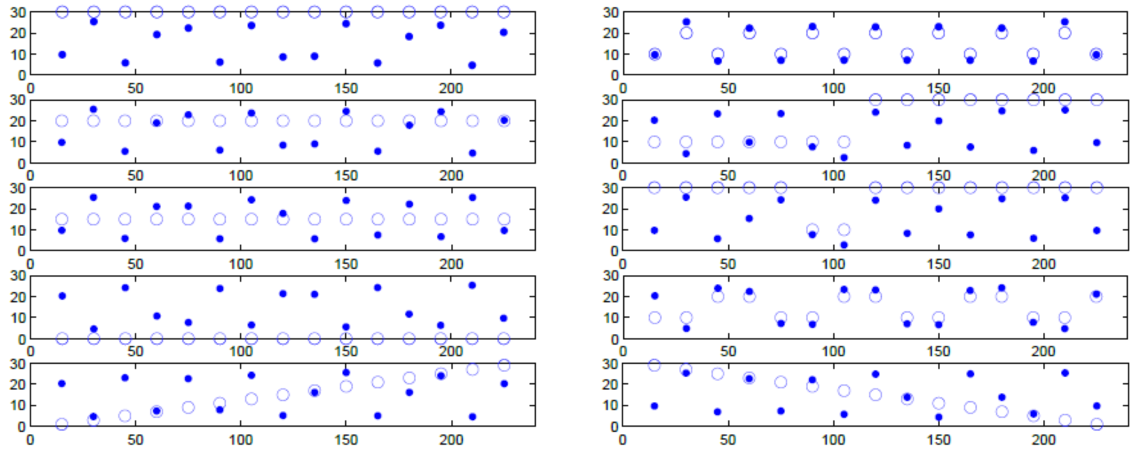
Parameter sensitivity(II)

- Changing grid size

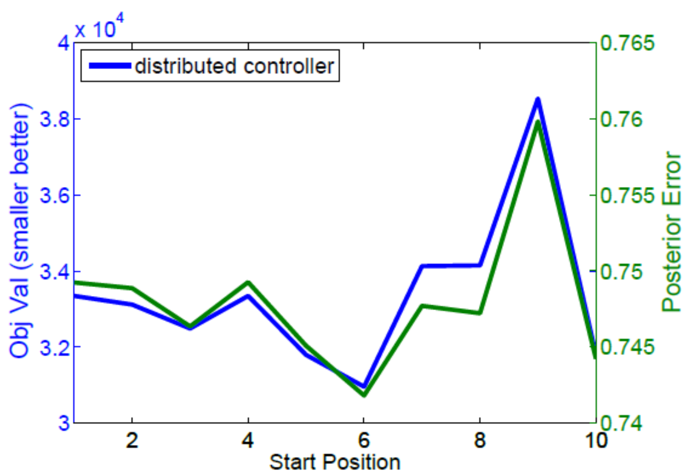




Positioning sensitivity



Start positions(circles) and final positions of the nodes (dots)



Conclusions

- Understanding dynamics of bodies of water requires sensing over full volume of water
- Gradient based decentralized controller
- Two covariance models
  - Multivariate Gaussian
  - Physics based hydrodynamic model
- Simulation & experiments, verifying the functionality