1

# techniques for target tracking by an autonomous cluster

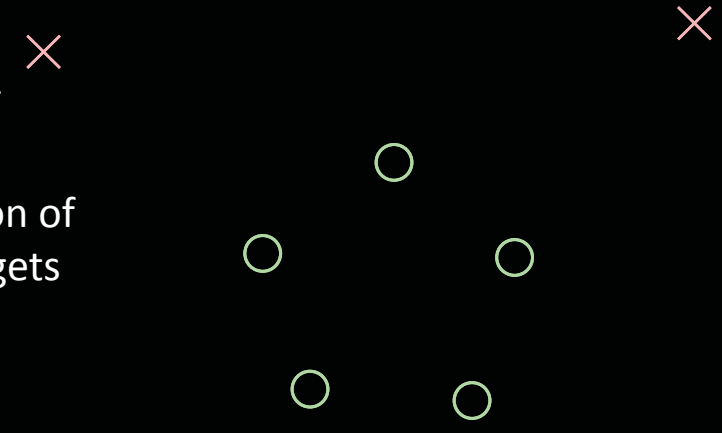
t. schneider
h. schmidt



2

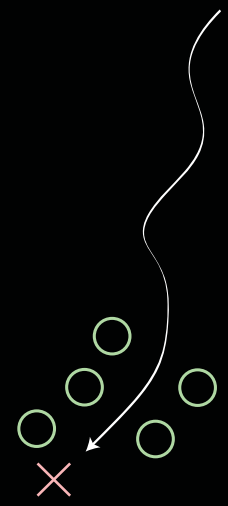
# problem

how can a cluster of vehicles (“friends”) best monitor a region of ocean and track targets that enter it?



# problem

if all the friends track
the first target to enter a region...





# problem

... no friends are left
in the region to track a
possible second target

5

# further considerations

low network bandwidth (acoustic comms) - friends must collaborate with minimal passed knowledge

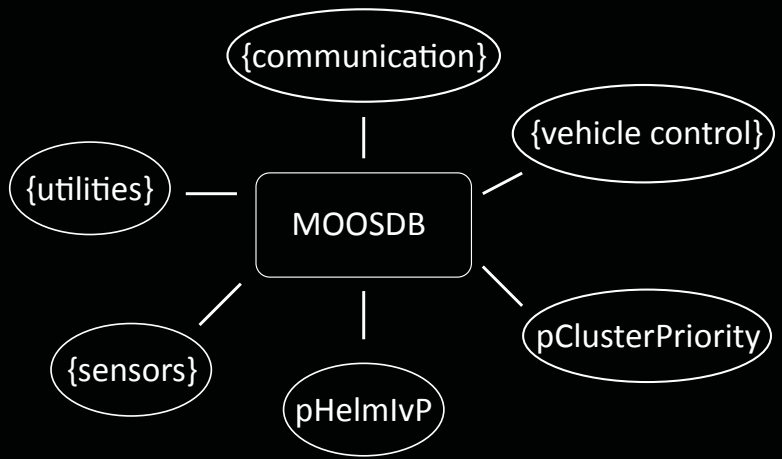
certain collaborating clusters may have different sensor capabilities than others - we will assume identical friends for now

6

# MOOS-IvP autonomy architecture

MOOS

- community of software processes that interact solely through a central database (MOOSDB)
- allows for rapid prototyping

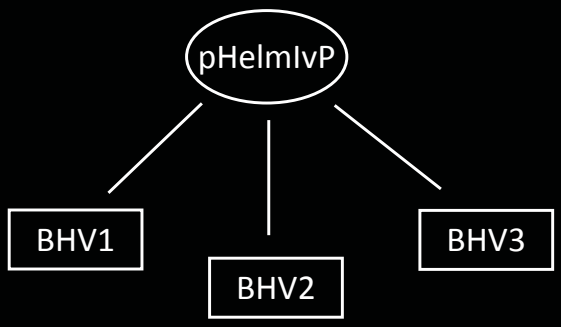


7

# MOOS-IvP autonomy architecture

IvP Helm (pHelmIvP)

- a single MOOS process that runs multiple behaviors
- each behavior expresses a "preference" (objective function) for speed, heading, depth
- pHelmIvP combines these "preferences" to form a single action for the vehicle




8

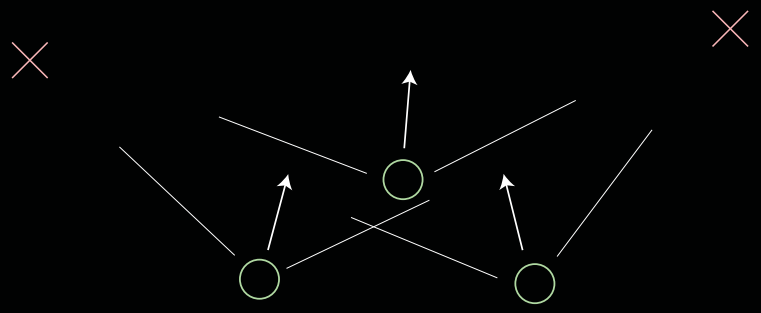
# solution

two behaviors and a new MOOS process

- BHVAttractor - expresses a desire to cut range to a target. an iteration of this behavior is run for each target.
- BHVRubberBand - expresses a desire to return to the initial formation position
- pClusterPriority - weights the strength of each BHVAttractor based on each friends proximity to the target

# BHVAttractor

with just this behavior, each vehicle attempts to minimize its distance to all the targets





# pClusterPriority

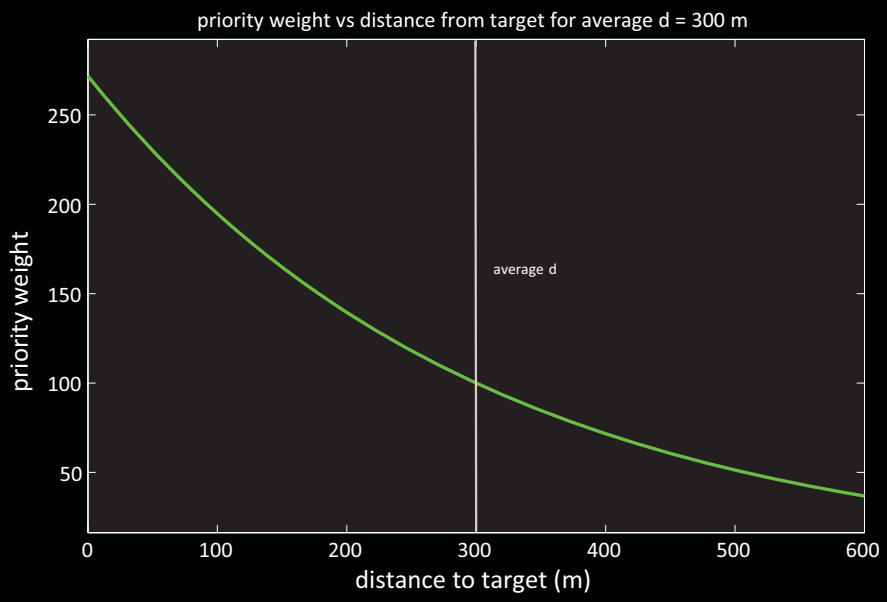
pClusterPriority weights each BHVAttractor by

$$A = A_0 e^{-\alpha(d-\overline{d})/\overline{d}}$$

where $A$ = priority weight of BHVAttractor
$A_0$ = normalizing constant ($A$ when $d = \overline{d}$)
$d$ = distance to target
$\overline{d}$ = average friends' distance to target
$\alpha$ = "strength" of decay

11

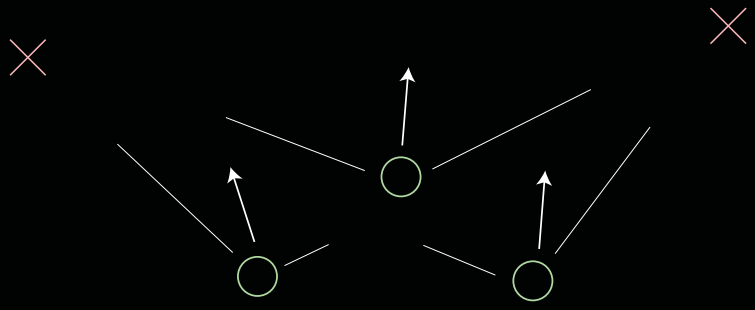
# pClusterPriority




12

# pClusterPriority

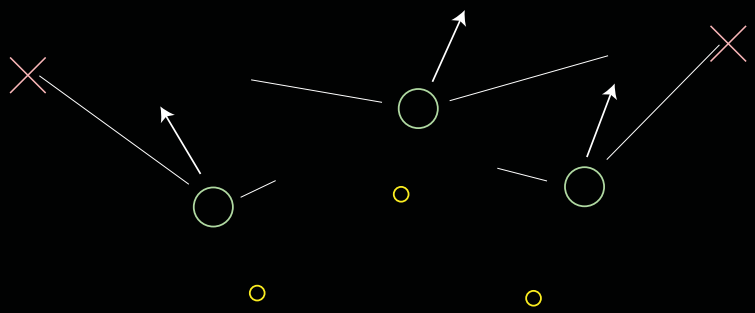
with this exponential decay of priority with respect to average distance, closer vehicles are given higher priority and the cluster splits intelligently



13

# pClusterPriority

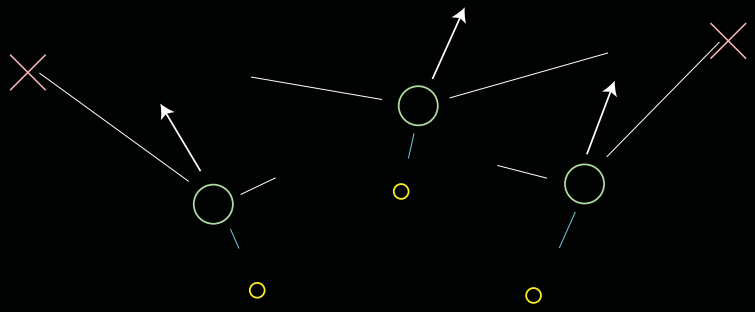
the priority weights are recalculated on each vehicle upon new friend and target position updates



14

# BHVRubberBand

this behavior increases in strength with distance from the cluster start point, ensuring the friends return eventually to their deployed region



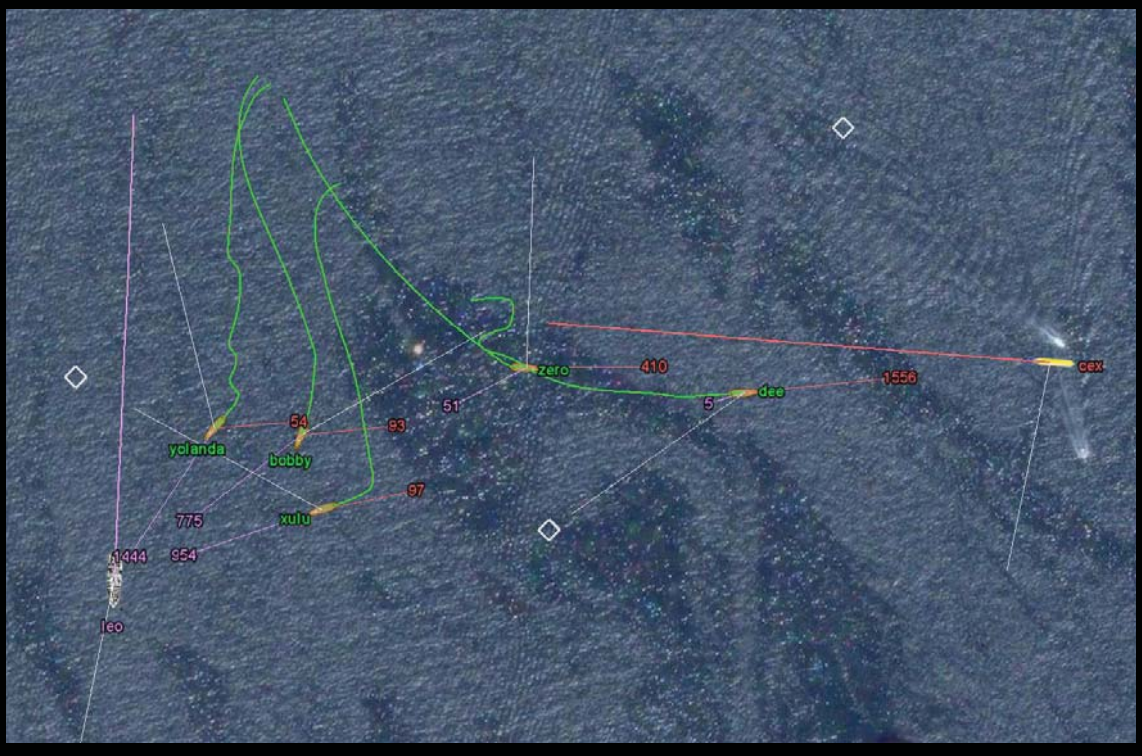
15

# test setup | simulation

- five friends - autonomous surface craft (kayaks - Bobby, Dee, Xulu, Yolanda, Zero).
- two targets - one ship (Leo) and one AUV (OEX)
- max speed - 1.5 m/s for kayaks, 5 m/s for Leo, 2 m/s for OEX

16

# test | simulation



17

# test setup | field trials | CCLNET08

- five friends - autonomous surface craft (real kayaks - Bobby, Dee; simulated kayaks - Xulu, Yolanda, Zero).
- three targets - one real AUV (OEX), two simulated AUVs (Macrura, Unicorn)

18

# test | field trials | CCLNET08

# future work

- adapt for use on AUVs (largely completed in simulation
- allow for weighting based on unique capabilities of cluster members
- improve real performance based on sea trial data



# acknowledgements


- collaborator / advisor: h. schmidt
- software: m. benjamin, p. newman, and others
- MIT / WHOI joint program