Multi-AUV Fault Tolerant Adaptive Sampling

P.B. Sujit
IIITD,
New Delhi - 110020
Email: sujit@iiitd.ac.in

Yanwu Zhang
Research Division, MBARI,
Moss Landing, CA - 95039, USA
Email: yzhang@mbari.org

A.J. Healey
Distinguished Emeritus Professor,
NPS, Monterey, CA - 93943
Email: healey@nps.edu

J.B. Sousa
ECE, University of Porto
Porto, Portugal - 4200365
Email: jtasso@fe.up.pt

***Abstract*—Autonomous underwater vehicles (AUVs) have been effectively used for investigating oceanographic processes which vary in space and time. For adequately capturing the spatial and temporal information of an oceanographic process at a desired resolution, adaptive sampling by multiple AUVs is often required. Although, multiple AUVs will allow quicker area coverage, occurrence of physical faults will create uncovered regions in the search space. The AUVs need to cooperate with one another and be fault tolerant. In this paper, we develop an adaptive sampling algorithm utilizing lane based sampling methodology. The width of lanes is adapted depending on the information acquired by the vehicles. We deploy the vehicles in opposite directions to allow periodic coordination and enable fault tolerance. Through simulations we show the capability of the developed algorithm to completely cover a region with minimal overlaps and fault tolerant to AUV physical faults.**

I. Introduction

Physical, biochemical and geological processes in the ocean are highly complex. Understanding oceanographic processes of various types and scales, e.g., water mass and heat fluxes, harmful algal blooms (HABs), is essential for maintaining sustained ocean health as they affect the human population through climate change, toxic plumes, fisheries, etc. Many dynamic processes are episodic phenomena with high spatio-temporal variability. Over the past decade, autonomous underwater vehicles (AUVs) and gliders are playing a pivotal role in studying oceanographic processes which vary in space and time [16, 8]. The "Lawn-mowing" pattern is the most often adopted AUV survey pattern. Bellingham and Willcox developed a quantitative AUV survey error metric which accounts for errors due to spatial under-sampling and temporal evolution of the surveyed oceanographic field [1, 14]. This pioneering effort provided a benchmark for designing and evaluating AUV sampling strategies. In order to better understand oceanographic processes, we need to design advanced sampling algorithms for the AUVs to cover a wide spectrum of spatio-temporal variations. For this, the AUVs must cooperate with each other in searching a region, detecting the phenomenon and mapping the complete event, and sampling the phenomenon with the desired resolution.

High spatio-temporal variability processes can be quickly detected and mapped using multiple AUVs. The AUVs' survey paths need to be designed so as to persistently sample and track the evolving process. Smith et al. [12] developed an intelligent path planning algorithm to track the process and collect data from the core of the process. After collection, data assimilation is done with the ROMS (Regional Ocean Modeling System) model and based on the updated predictions, a new mission is generated. Das et al. [6] developed a mechanism to determine a search region for sampling based on a combination of high frequency radar and MODIS satellite data. The selected region is then assigned to an AUV to sample using a lawn-mowing pattern. Both the approaches use a closed-loop model of prediction followed by data assimilation. However, both the above studies do not consider fault tolerance.

The emphasis of adaptive sampling is to adapt the sampling strategy based on observations, so that the future observations are effective and efficient for enabling better prediction of the oceanographic process [17]. Adaptive sampling has two main components: area coverage and high resolution sampling at the core of the process. Using multiple AUVs, these two components can be achieved effectively. Munafó et al. [9] designed a communication constrained AUV team approach to maximize the information gained by each agent in a static field. The AUVs sample only at certain locations and not continuously. Thus this approach is not suitable for spatio-temporal variable processes. Caiti et al. [2] developed a cooperative AUV path planning technique to estimate field flux while covering the entire area and maintaining agents in the communication range. Such constrains make all the platforms sweep along one particular direction. Fiorelli et al. [7] developed artificial potentials fields and a virtual leaders based cooperative control algorithm to adaptively sample a region by gliders. The control law requires the knowledge of all other agents' positions. The issue of area coverage is not addressed. The problem of area coverage and mapping an oceanographic the process in this area is different from the traditional problems of area coverage [4] and perimeter/boundary tracking [3, 13, 5]. In our problem these two problems are addressed jointly rather than separately.

Most of the work on multiple AUVs sampling assume that all the AUVs function perfectly throughout the mission. However, in reality, multiple AUVs deployments are susceptible to physical faults due to various issues like propeller getting stuck to a fishing net or seaweed, failure of actuators, etc. Due to physical faults, some of the AUVs are unable to perform the mission. Therefore one of the key requirements with multiple AUVs deployment is that the cooperative algorithms must be fault tolerant and achieve complete area coverage with minimal overlap. This issue of fault tolerant area coverage for sampling has not received adequate attention previously. In this paper, we develop an adaptive sampling algorithm that detects, maps and samples at the desired resolution within a given search region. The algorithm is fault tolerant to physical faults with minimal survey overlaps.

The developed adaptive sampling algorithm uses the lawn-mowing pattern for sampling. However, the width of the lane is adaptively changed to sample the process depending on the sensed information. In order to have fault tolerant cooperations between agents, we use the notion of periodic communication rather than communication constrained cooperation. To attain periodic coordination, every pair of agents are deployed in opposite directions traveling along the same lane. As the mission progresses, each pair meets at every second lane from the current lane as they are alternating their directions of travel. After meeting they move to the next lane and then depart away with opposite directions. During the meeting, they share the sensed information by which the next lane width is determined. In this fashion, coordination is achieved periodically, fault tolerance is achieved when one of the agent is at fault the other agent covers the track of the faulty agent. The algorithm is simple and scalable to a large number of agents. Yoon and Qiao [15] developed a similar algorithm with fixed lanes and assumed that agents can stop at the predefined meeting points, which is sometimes hard to realize in practice.

The rest of the paper is organized as follows. In Section II we describe the area coverage and fault tolerance problem taking realistic constraints into account for sampling. In Section III, the adaptive sampling algorithm is presented while the fault tolerance capability is described in Section IV. The developed algorithm is validated using simulations in Section V. We conclude in Section VI.

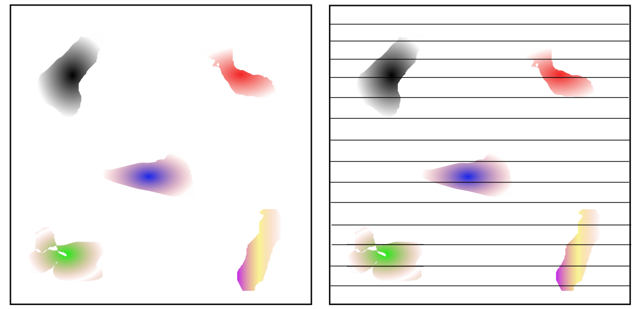
II. Problem statement

A. Problem

Consider a search space Ω having m number of processes that need to detected and mapped to a desired resolution. Each process is represented as $p_k, k = 1, \dots, m$. The search and sampling is carried out on 2D, that is, the AUVs cover the area and sample with constant depth. These process could be moving and spreading. For example, consider five processes of interest as shown in Figure 1(a). Assume that n AUVs are deployed for the mission. Each AUV is denoted as A_i and have kinematics constraints

$$\begin{aligned} \dot{x}_i &= v_i \cos(\psi_i) \\ \dot{y}_i &= v_i \sin(\psi_i) \\ \dot{\psi}_i &= \kappa(\psi_i^d - \psi_i) \end{aligned} \quad (1)$$

where, v_i is the speed of the vehicle A_i, ψ_i is the vehicle's heading and ψ_i^d is the desired heading. We assume that the speed of the AUVs can be different and hence the system is heterogeneous. Each AUV has limited communication range of r_c meters and carries a sensor s_i that measures the observation at the AUV current location $z_i = (x_i, y_i)$. Given the search space Ω and n AUVs, the problem is to detect all the processes, map them and sample to the desired solution under physical AUV faults and limited communication ranges.



(a) (b)

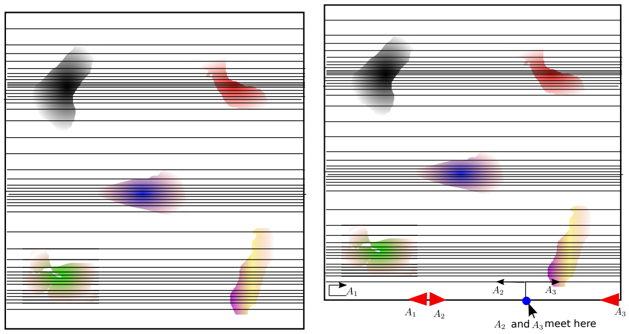
Fig. 1. (a) Five processes in a given search region (b) Fixed lane based sampling

B. Approach

There are three objectives for the problem: (i) complete coverage (ii) adaptive sampling and (iii) fault tolerance. A default fixed lane search and sampling satisfies only the first objective and not the others as shown in Fig. 1(b). Fixed lane search will miss the core regions and hence sampling may not satisfy the desired resolution. On the other hand, the lane size can be small enough that the complete coverage is carried out at high resolution. This will take longer duration to complete the mission and the process can move out of the search or even can disappear. In order to adaptively sample, we change the width of the lanes adaptively based on the sensed observations of the AUVs as shown in Figure 2(a). We can see that at the core of the process (area with high gradient) the lane width is small while in the areas where there is no process, the lane width length is modified to default value. This kind of adaptive strategy satisfies both objectives (i) and (ii) with quicker area coverage while sampling at higher resolution in the core regions. However, to enable fault tolerance to physical faults, we deploy the agents in opposite directions along the same lane. By deploying in opposite directions, the neighbor agents will meet periodically and thus enabling fault tolerance when one agent fails, the other agent can cover the region without leaving any gaps. For example, a three agent deployment is shown in Figure 2(b). The agent A_2 and A_3 will meet as the mission progress. While A_1 and A_2 will meet in the next lane. The new direction of agent A_1 after approaching the boundary and the direction of A_2 and A_3 after sharing information are shown in the figure. Detailed analysis of the lane width selection and fault tolerance are described in Section III and IV.

III. Adaptive sensing

The width of the lane must be adapted based on the strength of the observations. Initially all the vehicles are deployed on the first lane with opposite heading directions along the first lane. All the agents with odd id, that is, $A_i, i = 1, 3, 5, \dots$ will have heading direction from east to west, while the even agent ($A_i = 2, 4, \dots$) have direction of travel from west to east. The agents A_1 and A_n are the border agents as they approach the boundary of Ω every alternate lane. The direction of the agents



(a) (b)

Fig. 2. (a) Adaptive lane for sampling (b) Deployment of three agents and their initial directions of travel

is shown in Figure 2(b). For reference, the agents touching the boundary are called *boundary agents*, while the rest of agent in the middle of search space are called as *middle agents*.

The boundary agents have to change the lane when they meet the neighboring agents or when they approach the boundary. At the beginning of the mission, $t = 0$, let agent A_i be the boundary agent and it approaches the boundary at time t'. While traversing towards the boundary, it samples at some rate that is dependent on the sensor. For analysis, assume that the sampling rate of the sensor is s Hz and the sensed data is recorded into a vector D_i. We also assume that the time synchronization issues and delays in recording are either not present or taken care. When A_i approaches boundary, it has to determine the new lane width. The new lane width is determined between a low resolution (default fixed lane width, w_ℓ) and the minimum lane width defined by high resolution (w_h) sampling strategies. The lane widths w_ℓ and w_h are the bounds for the adaptive sampling strategy. Let $\bar{D}_i = \max D_i$, be the maximum sensed value of the process by A_i, then new lane width w_i is determined as

$$w_i = \begin{cases} w_h & \text{if } w_\ell e^{-\alpha \bar{D}_i} < w_h \\ w_\ell e^{-\alpha \bar{D}_i} & \text{Otherwise} \end{cases} \quad (2)$$

where $\alpha > 0$ is a constant. The agent A_i generates a new lane parallel to the current lane and travels towards the first location of the new lane at the boundary. After determining the lane width, the data in D is removed and new data recording begins. The new lane is represented by two waypoint W_i^r and $W_i^{\neg r}$, where W_i^r represents the lane termination point on the right side of boundary, while $W_i^{\neg r}$ represents the lane termination point on the left side of the boundary. Using the waypoints, the AUV updates its desired heading ψ_i^d as:

$$\psi_i^d = \begin{cases} atan2(y_i^r - y_i, x_i^r - x_i) & \text{if agent is going right} \\ atan2(y_i^{\neg r} - y_i, x_i^{\neg r} - x_i) & \text{if agent is going left} \end{cases} \quad (3)$$

When A_i is in the communication range of a neighboring agent A_j at time t'', then both A_i and A_j share $\bar{D}_i$ and $\bar{D}_j$ with each other. The agents compare the maximum sensed value of both the agents and determine the lane width based on the highest detected value as given in Algorithm 1. By modifying the lane width based on the cooperative information from neighbors will allow the agent with the highest sensing parameter to determine the lane width. Since, higher the sensing parameter value, lower the lane width, an agent whose sensing value is higher than the other agent will not have a low resolution lane width. Thus this, mechanism will enable sampling at high resolution in the detected areas.

Note that during comparison between two agents, we do not assume that both agents are sensing the same process. The processes detected by these agents could be different, however, their sensors and sensing parameter is the same. Therefore, the agent that has low sensing value may be subject to high resolution lane width but this will not reduce the process sensing accuracy. Once the new lane is determined, the agent A_i and A_j travel together to the next lane and then depart in opposite directions. While implementing this phenomena, we can consider the movement as a sequence of waypoints. When agents meet each other, they generate two waypoints: the first one that leads to the next lane, while the second one makes the agent follow the path defined by the lane.

Algorithm 1 Algorithm to adaptively select lanes

while Agent A_i is in motion **do**
  if Agent reaches boundary **then**
    Determine w_i using (2)
  else if Neighboring agent A_j is in the communication range **then**
    Share D_i and D_j
    if $D_i \geq D_j$ **then**
      Determine w_i using (2)
    else
      $\bar{D}_i \leftarrow \bar{D}_j$, Determine w_i using (2)
    end if
  end if
  Update agent positions using (1) and (3)
end while

The process to be detected can be located anywhere in the search space. However, to determine w_ℓ, we must assume some kind of process information to ensure that using w_ℓ, the vehicle will detect the process at least once, though not with desired resolution. Therefore, we assume that each process p_i has at least $2w_\ell$ height and width. We also assume that the process is slowly moving process ($< 30cm/sec$), and hence its speed is small compared to the AUV speed. Otherwise, the process may move out of the search space.

Lemma 1: Each and every unique process p_k will be detected by the agents

Proof: Assume the mission is carried out with a single vehicle. The vehicle moves along a defined lane either from left to right or right to left. The minimum height of $p_k \geq 2w_\ell$, and the maximum lane width for a vehicle is w_ℓ. Therefore, as the vehicle moves along a lane, it will intersect with the process and hence, it is detected. $\square$

IV. Fault Tolerance

During a mission, some of the agents may have faults. Since the agents have communication capability, we can assume that, if the neighboring agent is within the communication range, then it can share the information and update its status. However, it is not always the case. There can be situations where the vehicle is drifted away by currents after being at fault and unable to communicate or its transponders are not working to communicate. Therefore, to address different kinds of issues, we assume that if the agent has a fault then it cannot communicate. We devise a strategy by which the agents ensure the area is covered and adaptively reconfigure to the loss of the agent.

Agent faults will affect complete area coverage by leaving uncovered areas. We say that the area is completely covered, if there is no space $\Omega' \subset \Omega$, such that no vehicle has passed through Ω'. We consider two types of faults: border agent fault and middle agent faults. To show how fault tolerance is introduced into the system, we consider the scenario where three agents A_1, A_2 and A_3 are deployed with the assigned direction of travel as shown in Figure 3(a).

A. Boundary/Border agent faults

Consider the initial deployment of vehicles as shown in Figure 3(a). Assume that the border agent A_3 is at fault and it does not move. Then, A_2 will cover the region that A_3 was supposed to cover due to its direction of motion and will meet A_1 at location τ_{12} in the next lane as shown in Figure 3(b). Through, this simple example, we can see that if the border is at fault then middle agent or the other border agent will cover the region without generating Ω'.

Similarly, now consider A_1 is at fault, then agents A_2 and A_3 will meet each other at τ_{23} as shown in Figure 3(c). Depending upon the sensed value, the next lane width is determined. They both move up towards the next lane and depart in opposite directions. The agent A_2 approaches the boundary, determines next lane width and moves to the next lane. Since the agent A_1 has not started, the region in dashed rectangle as shown in Figure 3(c) is not covered. However, since we assumed that the minimum height of the process is $2w_\ell$, and A_2 passes through th e lane w_ℓ higher than the lane A_1 is located, it will detect the process. Thus, when there is a border agent fault, the agent in the middle will cover the region and ensure that all the processes are detected. Therefore, we can say that by deploying the agents in opposite directions, uncovered regions can be mitigated.

B. Middle agent faults

Consider the scenario as shown in Figure 3(d). Assume that the mission has been progressed upto $k-2^{th}$ lane where agents A_1 and A_2 meet at τ_{12}. The next lane is determined based on the sensed value. The direction of travel for the two agents is shown in the figure. Let agents A_2 and A_3 meet at τ_{23} on the $k-1$ lane. Since A_2 detected some process, the new lane is determined to be k' and the direction of travel for both the agents is shown in the figure. Assume that at τ_{23}, the agent A_2 is at fault after determining the new lane. The agent A_3 continues its determined lane as shown in the figure.

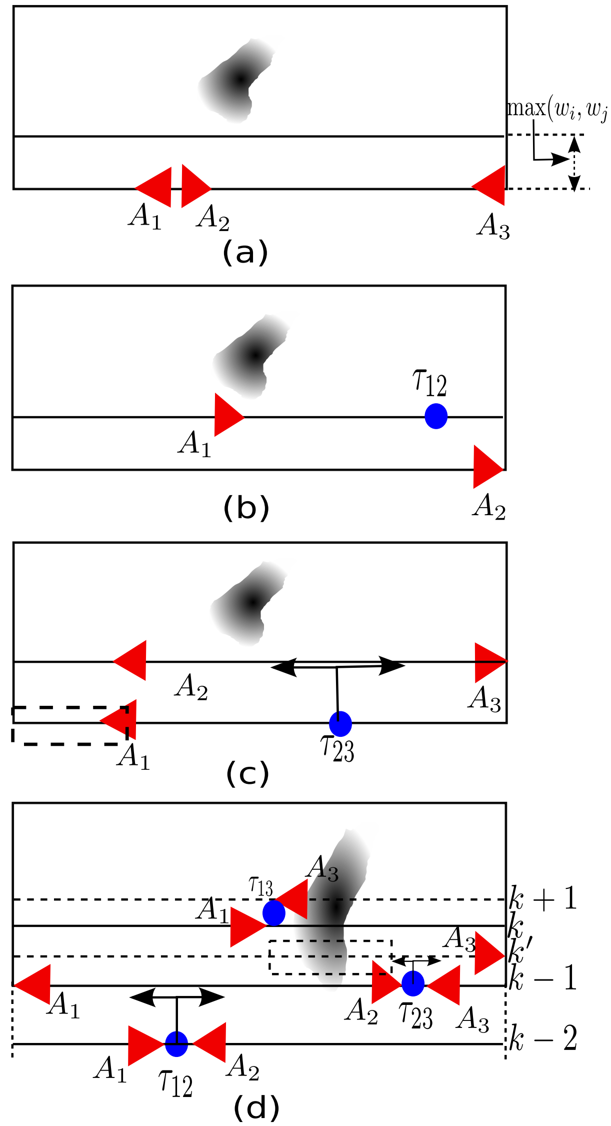


Fig. 3. (a) Initial locations of the vehicle and the direction of travel with the maximum lane width as the next search path (b) Situation when boundary agent A_3 failed at the initial location itself, then A_2 will cover A_3 search space and meet A_1 at τ_{12} (c) Situation where A_1 has failed and the region in rectangle is not covered (d) When middle agent A_2 fails, the rectangular region is not covered.

The agent A_1 after completing the lane $k-1$ will determine a new lane k, as shown in the figure. The lane k is above k' and A_1 will meet A_3 at τ_{13}. Note that τ_{13} is not in the same lane but both the agents are within the communication range. As A_2 could not move to complete its part of the coverage, the rectangular area shown in Figure 3(d) is uncovered. However, the lane $k+1$ taken by A_3 is at maximum w_ℓ units above lane k'. Since, we assume that the process is $2w_\ell$ units height, A_3 will detect any existing process. Hence, even when there is a middle agent fault, the notion of moving in opposite directions will enable fault tolerance.

Theorem 1: Assume that the agents fail sequentially, then the developed method of determining lanes as given in Algorithm 1 and deploying the agents in opposite direction is a fault tolerant method.

Proof: If the developed method is not fault tolerant then, there exists an uncovered space Ω' in the search space Ω. Lemma 1 shows that if there are no faults then the process will be detected. For Ω' to exist, there must be some agent faults, either a border agent or middle agent fault.

If there is a border agent fault, then in Section IV-A, we have shown that the agents cover the region under border agent fault. In Section IV-B, we have shown that the coverage is complete under middle agent faults. Since, coverage is complete under border and middle agent faults, and Ω' exists only under faulty conditions, the developed method is fault tolerant. □

V. Simulation results

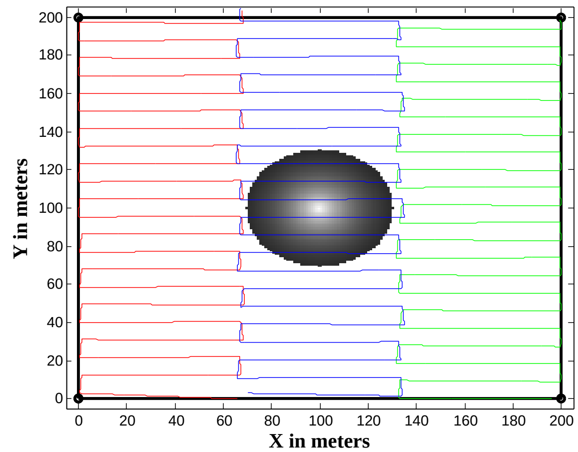
The proposed fault tolerant method is validated using simulations. Through simulations, we will show the following: coverage time analysis with low resolution, high resolution and the developed adaptive sampling method. Then we will analyze the effect of increasing the number of process on the coverage time and the effect of increase in number of faulty agents to coverage and sampling.

A. Example

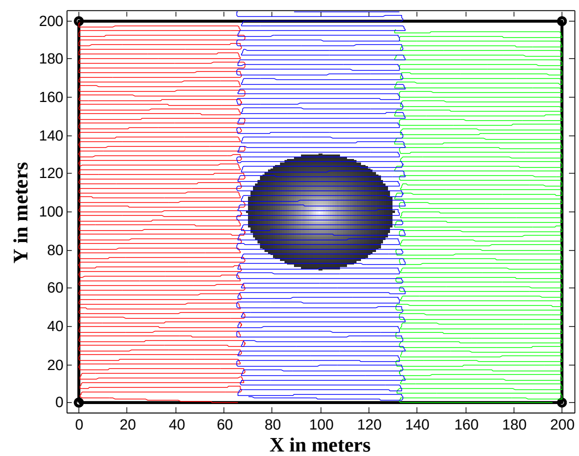
Initially, we will show the performance of the developed algorithm using an example. We considered a scenario, where a process is stationary in the middle of a $200m \times 200m$ search space as shown in Figure 1(b). The process has a radius of $30m$ located at $(100, 100)$ and the value of the process is dependent on the distance between the agent and the (100,100). The value is determines as $D_i(t) = e^{-\beta d_i}$, where $d_i = ||z_i - (100, 100)||$ and $||.||$ is the Euclidean norm. The vehicles speeds are constant at 3knots.

Figure 4(a) shows the trajectories of the three vehicles starting from $z_1 = (66.67, 0), z_2 = (69.67, 3), z_3 = (197, 0)$, respectively using the default fixed lane width (low resolution sampling (LRS)) of $10m$. From the figure, we can see that the vehicles observe the process for a short duration. Next, we used high resolution sampling (HRS) for the vehicle. The Figure 4(b) shows the trajectories of the vehicle sampling the process for longer duration and hence receive more information. Figure 5 shows the vehicle trajectories using the developed adaptive lane selection algorithm (AS). We can see that even in this strategy, the sampling of the process is high but not as high as HRS. However, in adaptive sampling, we can see that when there is no process, it uses LRS and changes the lane width adaptively to increase the information of the process.

In order to compare the performance of the three strategies, we record the sensed data $D_i(t)$ for each vehicle over the entire mission. We then determine information captured as $F = \sum_i \sum_{t=0}^{T} D_i(t)$. From the strategies, we know that HRS achieves the best performance. Therefore, we consider



(a)



(b)

Fig. 4. (a) Area coverage and process detection using default fixed width lanes of $10m$ (b) Area coverage and process detection using high resolution fixed width lanes of $3m$

information obtained by HRS as the base strategy and hence its value is 1 in Figure 6(a). The percentage information that AS captured was 76% of HRS while LRS captured only 28% of the HRS information.

Further to determine the time taken to accomplish the mission, we recorded the mission time of each strategy and the Figure 6(b) shows the time taken by each strategy. In this case, we considered LRS strategy as the base strategy and hence its value is 1. The AS takes 38% more time than LRS while HRS takes 330% more than LRS. Thus from Figures 6(a) and 6(b) we can see that the developed adaptive sampling strategy captures good amount of information while allowing the missing to be completed quickly.

B. Effect of increase in number of processes

For a given number of agents, HRS determines the upper bound on the coverage as well as to achieve maximum infor-

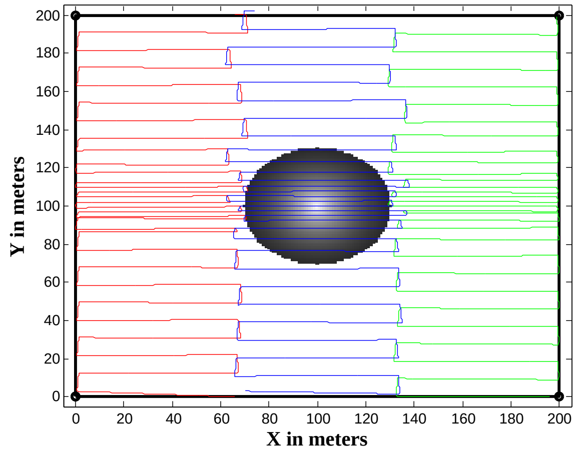
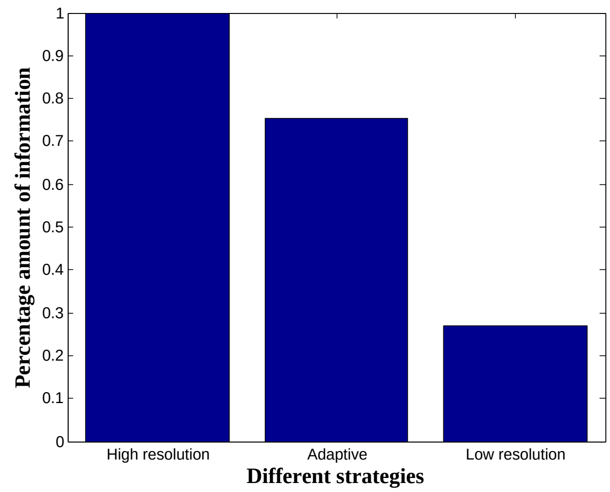
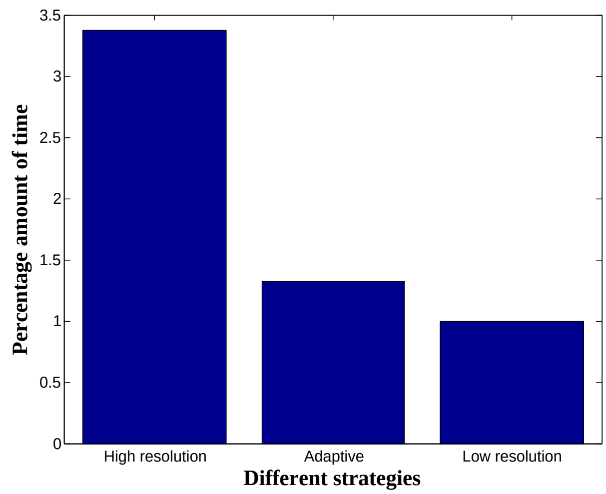


Fig. 5. Area coverage and process detection using adaptive lane width



(a)



(b)

Fig. 6. (a) The percentage information gathered using different techniques assuming high resolution strategy as the baseline (b) The time taken by each strategy to complete the mission

mation about any kind of process. With increase in number of processes, the agents have to carry out more number of high resolution sampling lanes. In the worst case, if there are large number of processes, then the agents will perform a HRS even though they are using AS. Any further increase in number of processes will not affect the sampling and coverage time.

C. *Effect of faulty agents*

The presence of faulty agents will reduce the coverage time and the sampling information about the process. We will first show through an example the effect of faulty agent on the performance and then show the trajectories of the vehicles for border and middle agent failures.

Consider the same scenario as given in Section V-A with the same initial locations. At time $t = 295$, the border agent A_1 fails to operate. This scenario is shown in Figure 7(a). From this time onwards, only agents A_2 and A_3 carry out the sampling and coverage actions. From the figure, we can see that the two agents cover the area effectively and adaptively sample the process.

We next considered the middle agent failure as shown in the Figure 7(b). The fault in A_2 causes the width of some region to be more than the default lane width as shown by an ellipse in Figure 7(b). The region is of height $15m$ and the minimum process height is $20m$, therefore if a process was existing in this region, then the remaining agents will detect it. From the figure we can see that both the agents cover the region and the process quite well.

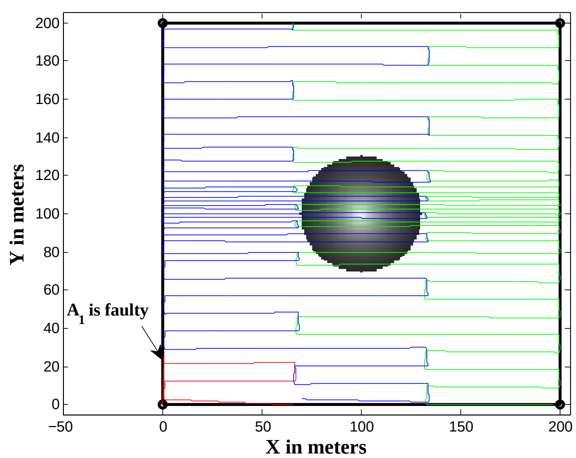
The adaptive lane sampling and coverage algorithm not only performs well for static process but also for moving scenarios. Consider the scenario in Figure 8, where the process is moving at $20cm/sec$. From the figure, we can see that as the process moves, the agents perform high resolution sampling at the core of the process and track it properly. Thus AS is useful for both static and moving processes as well.

With increase in number of faulty agents, the time to accomplish the mission increases. Figure 9 shows the effect of increasing the number of faulty agents for a 3 agent simulation. It is natural that with increase in number of faulty agents, the effective number of agents performing the mission decreases. Hence, the time taken to accomplish the mission increases. The same effect is shown in the figure.

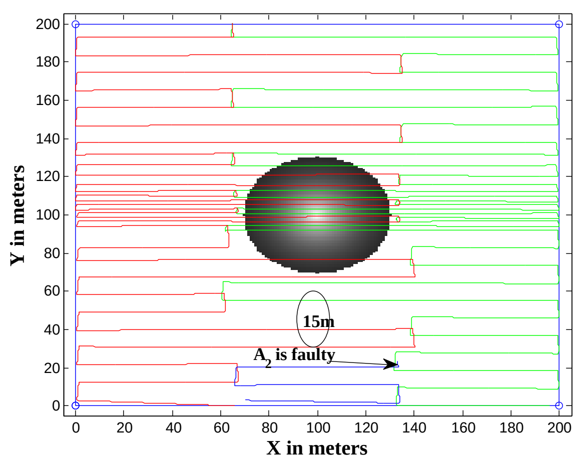
D. *Discussions*

In this section, we will discuss justifications for several assumptions in the paper.

1) 2D sampling: Through simulations we have shown that the developed adaptive sampling strategy performs better sampling than fixed lane width sampling strategy and is fault tolerant. However, typical sampling of processes in the ocean is carried out in 3D. That is given a lane, the agent performs a YO-YO maneuver in $x-z$ coordinate frame and y coordinate



(a)



(b)

Fig. 7. (a) Agent A_1 becomes a faulty agent at time $t = 295s$ (b) Agent A_2 becomes a faulty agent at time $t = 295s$

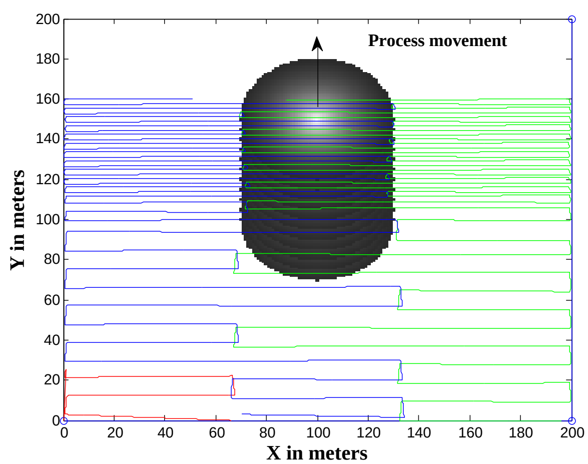


Fig. 8. Area coverage and process detection for a moving process with A_1 as the faulty agent

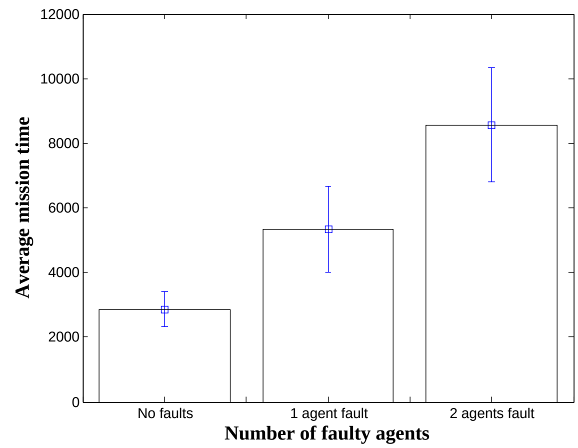


Fig. 9. Effect of increase in number of faulty agents

is constant defined by the $lane$. Here, we assume that the agents move in a horizontal plane. The developed strategy need to be modified to be used in ocean sampling applications. However, there are several sampling strategies in river that use constant depth for which the developed algorithm suits pretty well. For example, the sampling strategy developed in [10, 11] uses constant depth sampling for pollution parameters in a river. However, for the sampling strategy to be used for oceanography applications, 3D trajectories need to be accommodated. This work will be carried out in the future.

2) Minimum process width: Note that, unlike image based sonar, the sensor carried by the AUV to detect biophysical process is point based. That is, it samples at a single point. Therefore, to search the space within reasonable amount of time, virtual sensor footprint is assumed in the form of minimum lane width. Typically, in the ocean, the process of interest are in kilometers and the minimum sampling lanes are of the order of tens of meters. Hence, an assumption of $2w_\ell$ is a reasonable assumption for sampling applications. However, even if we assume a minimum lane length of w_ℓ, still we could achieve coverage by considering several modifications. This aspect will be investigated further.

3) Speed variations: The simulations were carried out using constant AUV speed. However, the area coverage algorithm will work for heterogeneous speeds. However, the speeds between the agents should not vary too much. Further work needs to be done to determine tight bounds on the maximum variation in agent speeds.

4) Process speed: In the simulations we assumed that the process speed in $30cm/s$. However, the maximum proceed speed that the process can travel depends not only on the agent speed, but on a combination of agent speed and size of the search space. An upper bound of the process speed needs to be derived.

VI. Conclusions

We presented a simple algorithm for multiple AUVs performing sampling missions in the ocean. The algorithm has two main features: fault tolerance and adaptive sampling based on the sensor observations. Through analysis and examples we have shown that the algorithm is fault tolerant to agent faults. The algorithm can handle border agent and middle faults equally well. The results show that the performance of the adaptive sampling strategy developed in this paper performs close to the high resolution sampling strategy in terms of information captured, while it performs close to the low resolution sampling strategy in terms of time taken to accomplish the mission. Thus, the developed method achieves good performance in terms of information and mission time.

REFERENCES

[1] J.G. Bellingham and J.S. Willcox. Optimizing auv oceanographic surveys. *Proceedings of AUV'96*, pages 391–398, Monterey, CA, June 1996.

[2] A. Caiti, G. Casalino, A. Munafo, and A. Turetta. Cooperating auv teams: Adaptive area coverage with space-varying communication constraints. In *OCEANS 2009 - EUROPE*, pages 1 –7, may 2009. doi: 10.1109/OCEANSE.2009.5278255.

[3] D.W. Casbeer, D.B. Kingston, R.W. Beard, and T.W. McLain. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, 37(6):351–360, 2006.

[4] H. Choset. Coverage for robotics–a survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1):113–126, 2001.

[5] J. Clark and R. Fierro. Mobile robotic sensors for perimeter detection and tracking. *ISA transactions*, 46 (1):3–13, 2007.

[6] J. Das, K. Rajan, S. Frolov, F. Py, J. Ryan, D.A. Caron, and G.S. Sukhatme. Towards marine bloom trajectory prediction for auv mission planning. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4784–4790. IEEE, 2010.

[7] E. Fiorelli, N.E. Leonard, P. Bhatta, D.A. Paley, R. Bachmayer, and D.M. Fratantoni. Multi-auv control and adaptive sampling in monterey bay. *Oceanic Engineering, IEEE Journal of*, 31(4):935–948, 2006.

[8] N.E. Leonard, D.A. Paley, R.E. Davis, D.M. Fratantoni, F. Lekien, and F. Zhang. Coordinated control of an underwater glider fleet in an adaptive ocean sampling field experiment in monterey bay. *Journal of Field Robotics*, 27(6):718–740, 2010.

[9] A. Munafo, E. Simetti, A. Turetta, A. Caiti, and G. Casalino. Autonomous underwater vehicle teams for adaptive ocean sampling: a data-driven approach. *Journal of Ocean Dynamics*, 61:1981–1994, November 2011.

[10] M. Rahimi, R. Pon, W.J. Kaiser, G.S. Sukhatme, D. Estrin, and M. Srivastava. Adaptive sampling for environmental robotics. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, pages 3537–3544. IEEE, 2004.

[11] M. Rahimi, M. Hansen, W.J. Kaiser, G.S. Sukhatme, and D. Estrin. Adaptive sampling for environmental field estimation using robotic sensors. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3692–3698. IEEE, 2005.

[12] R.N. Smith, M. Schwager, S.L. Smith, B.H. Jones, D. Rus, and G.S. Sukhatme. Persistent ocean monitoring with underwater gliders: Adapting sampling resolution. *Journal of Field Robotics*, 28(5):714–741, 2011.

[13] S. Susca, F. Bullo, and S. Martínez. Monitoring environmental boundaries with a robotic sensor network. *Control Systems Technology, IEEE Transactions on*, 16(2):288–296, 2008.

[14] J.S. Willcox, J.G. Bellingham, Y. Zhang, and A.B. Baggeroer. Performance metrics for oceanographic surveys with autonomous underwater vehicles. *Oceanic Engineering, IEEE Journal of*, 26(4):711–725, 2001.

[15] Seokhoon Yoon and Chunming Qiao. Cooperative search and survey using autonomous underwater vehicles (auvs). *Parallel and Distributed Systems, IEEE Transactions on*, 22(3):364 –379, march 2011. ISSN 1045-9219. doi: 10.1109/TPDS.2010.88.

[16] Y. Zhang, A.B. Baggeroer, and J.G. Bellingham. Spectral-feature classification of oceanographic processes using an autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 26(4):726–741, 2001.

[17] Y Zhang, J.G. Bellingham, J.P. Ryan, J. Harvey, R. McEwen, and M. Godin. Adaptive sampling in ocean observation. *Invited talk at the Second International Workshop on Seafloor Observation,*, Shanghai, China, November 2011.