The 5th Annual IEEE International Conference on
Cyber Technology in Automation, Control and Intelligent Systems
June 8-12, 2015, Shenyang, China

An Adaptive Scheme of Sharing Compressed Flow Information Among Networked Underwater Gliders

Shijie Liu*†, Jiancheng Yu*, Aiqun Zhang* and Fumin Zhang‡
*Shenyang Institute of Automation, Chinese Academy of Sciences
Shenyang,110016, P.R.China.
Email: {liushijie,yjc,zaq}@sia.cn
†University of Chinese Academy of Sciences
Beijing, 100049, P.R.China.
‡School of Electrical and Computer Engineering, Georgia Institute of Technology
Atlanta, Georgia 30332, USA
Email: fumin@gatech.edu

***Abstract*—Underwater glider fleets are often used cooperatively for large-scale ocean observation. A key requirement for underwater glider operation is to reduce the amount of information that needs to be shared among gliders over communication links. This paper presents a method to estimate the quality of communication link between underwater gliders sharing their information. The representation of high flow region that gliders should notice is compressed using the support vector data description (SVDD) method. The ratio of compression of SVDD is optimized to match the link quality and SVDD error estimation. The compressed data is then communicated among gliders for a distributed path planning algorithm. We show that such an adaptive compression method can provide desired accuracy.**

I. INTRODUCTION

Underwater gliders play important roles in ocean sampling [1] due to its long-endurance, low-cost and reusability. Gliders travel at relatively low speed comparing to the speed of ocean current. Gliders should also avoid to surface in areas with heavy ship traffic. These requirements post challenges for path planning algorithms.

For large scale environmental monitoring, it is often preferred to use a fleet of gliders that cooperate with each other to increase the quality of data collected [2]. Cooperative path planning, however, remains a novel research direction for marine robots. Our previous work [3] proposed a distributed approach for cooperative path planning using a reduced amount of information. The reduction, which was achieved by a method based on the support vector data description (SVDD) [4], extracts boundaries of flow regions with high speed that gliders should avoid. The reduced information is then communicated among neighboring gliders. In reality, the underwater acoustic communication link or the satellite communication link, the two major communication methods used by gliders are subject to time-varying disturbances. The reduced information may not be able to be reliably transmitted. In this work, we consider the changing link quality of the communication channel and adjust the ratio of reduction according to both the link quality and the fidelity of representation.

Link quality estimation plays a key role in wireless sensor networks. It has an overall impact on the network performance and also affects the design of higher level protocols [5]. Quality of service (QoS) represents the overall performance of a communication network seen by the users of the network, hence are directly related to networked robotics [6]. Underwater networking reprints a fast growing area that is developed from underwater communication. An effective protocol to transmit data packets underwater is called Stop and Wait (S&W), which is a type of Automatic Repeat reQuest(ARQ) scheme. The transmitter sends a packet and then waits for a response, called an acknowledgement, from the receiver. If this acknowledgement does not arrive within a predefined time, or a negative acknowledgement arrives, a packet will be retransmitted. The efficiency of the conventional protocol can be improved by transmitting a group of packets[7], and the optimal packet size can be determined as a function of QoS that is related to range and bit rate.

In this paper, we first propose a QoS estimation method for the communication links used by underwater gliders. Then, we improve the data compression scheme in [3] to incorporate the estimated QoS. The compression ratio is determined by solving an optimization problem that is constrained by the estimated QoS. The compressed data is then shared among the gliders for distributed path planning. Our method enables the gliders to adaptively adjust the compression ratio to match the QoS of communication links. This allows us to investigate the quality of path planning under the influence of unreliable communication.

The paper is organized as follows. In section II, we briefly review the problem of distributed path planning and the need to find a compressed representation of flow field. In section III, we set up the problem for this paper. The adaptive compression scheme is proposed in section IV . Section V demonstrates the simulation results of distributed path planning and section VI provides conclusion and discussion.

II. BACKGROUND INFORMATION

We consider a team of K gliders $G_1, G_2, \cdots, G_K$ deployed into the ocean at different locations and need to coordinate their paths for certain task. For this paper we focus on planning

---

This work is supported by the National Natural Science Foundation of China (Grant No. 61233013) and the State Key Laboratory of Robotics at Shenyang Institute of Automation (Grant No. 2014-Z02).

978-1-4799-8730-6/15/$31.00 © 2015 IEEE

the planar trajectories for the gliders. We assume that each glider has knowledge about the depth-averaged flow within a small patch around itself. We can discretize the patch into $m \times n$ grid points indexed by (i, j). Consequently, the depth-averaged flow velocities in this patch can be represented by two $m \times n$ matrices $\mathbf{U} = \{u_{ij}\}_{m,n}$ and $\mathbf{V} = \{v_{ij}\}_{m,n}$, where u_{ij} denotes flow speed in the east/west direction at gridpoint (i, j) and v_{ij} denotes flow speed in north/south direction at grid point (i, j). A local flow map can then be represented as $\mathbf{F} = [\mathbf{U}(x, y), \mathbf{V}(x, y)]$ in this patch.

We consider glider G_1 planing a path from its starting position $\mathbf{r_b}$ to a destination $\mathbf{r_d}$, as illustrated by Fig. 1. G_1 has its local flow map, $\mathbf{F}_1$, shown in Fig. 1. In this local patch, $\mathbf{F}_1$ is sufficient to plan an accurate path for G_1. However, G_1 does not have information about the region out of this patch even though G_1 has to avoid regions with strong currents against its desired motion outside its patch. This path planning problem can be solved by allowing gliders to share their knowledge about the flow over communication links. But transmitting all the depth averaged flow data within a glider patch over communication links is difficult and even unachievable. Therefore, the problems are to determine what information is necessary to be shared and how to reduce the amount of the information for cooperative path planing

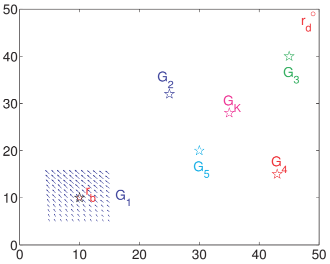


Fig. 1. Glider G_1 only has local information in a small patch for planning a path from $\mathbf{r_b}$ to $\mathbf{r_d}$.

The flow information can be reduced using support vector data description (SVDD) [4] to represent the boundaries of regions with high flow. The SVDD method produces representation of a dataset by searching for the smallest hypersphere that contains as many target points (grid points with strong flow) as possible but does not include outlier points (grid points with weak flow). Consider a set of training data $\{\mathbf{x}_i, i = 1, 2, \ldots, N\}$ that contain both target points and outlier points. SVDD finds Lagrangian multipliers α_i^* that maximize

$$L = \sum_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{1}$$

The optimization solutions will contain a large number of α_i^* with 0 value. The target points $\mathbf{x}_i$ with the corresponding $\alpha_i^* > 0$ are called *support vectors*. In the 2D plane, the support vectors determine a circle that separates the target points from the outlier points, which can be viewed as a representation of the boundary of the regions of high flow.

The boundary of regions with strong flow usually has irregular shape. By replacing the inner product $(\mathbf{x}_i \cdot \mathbf{x}_j)$ by a Gaussian kernel function $K_G(\mathbf{x}_i, \mathbf{x}_j) = \exp(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2})$ [4], support vectors can be computed to maximize the Lagrangian

$$L = 1 - \sum_i \alpha_i^2 - \sum_{i \neq j} \alpha_i \alpha_j K_G(\mathbf{x}_i, \mathbf{x}_j) \tag{2}$$

Once a set of support vectors have been determined. They can be used to decide wether a grid point is inside or outside of the boundary. Let S be the set of support vectors. A test point $\mathbf{z}$ is considered inside the boundary when the following inequality is satisfied

$$K_G(\mathbf{z}, \mathbf{z}) - 2 \sum_s \alpha_s^* K_G(\mathbf{z}, \mathbf{x}_s) + \sum_{s,k} \alpha_s^* \alpha_k^* K_G(\mathbf{x}_s, \mathbf{x}_k) \leq R^{*2} \tag{3}$$

where $\mathbf{x}_s, \mathbf{x}_k \in S$.

Let the boundary determined by G_1 be Ω_1. Then G_1 will communicate all support vectors that represent Ω_1 to G_2 for G_2 to plan its path. The parameters that need to be transmitted by G_1 to G_2 include: the width of the Gaussian kernel σ, the set of support vectors S, and the corresponding value of the Lagrange multipliers. Then G_2 will be able to judge whether one point $\mathbf{z}$ is within Ω_1 or not by using the set of support vectors S and the corresponding Lagrange multipliers that represent Ω_1 as in equation (3).

The fidelity of SVDD is of great importance. There are two types of representation errors due to the reduction of information by SVDD. The first type is the ***Target Rejection Error***, represented by e_1, indicates how many target points are rejected. The second type is the called ***Outlier Acception Error***, represented by e_2, indicates how many outlier points are accepted. These errors can be adjusted by adjusting the width σ in the Gaussian kernel K_G. Our previous work [3] determines σ to minimize the following cost function

$$E = (e_1 + e_2) + \frac{N_{\mathrm{SV}}}{N}. \tag{4}$$

where N_{SV} is the total number of support vectors generated by the SVDD, and N is the total number of grid points.

III. PROBLEM FORMULATION

We consider the situation when one glider, say G_2 tries to determine a set of support vectors to be transmitted to glider G_1. If the communication link between G_1 and G_2 are unreliable, then the information may be corrupted during the transmission. But G_1 may suppose that the information it received is correctly transmitted and reconstructs the high-flow regions in the blank zone based on these information. In the worst situation, corrupted information might lead G_1 into dangerous flow.

The communication load between G_1 and G_2 depends on how many support vectors are needed by G_2 to represents the boundary of the region with strong flow. The number of support vectors can be adjusted by the width σ in the Gaussian kernel K_G. To see the effect of σ on the number of the support vectors, let us consider a very small σ. In this case

$$K_G(\mathbf{x}_i, \mathbf{x}_j) = \exp(\frac{-(\mathbf{x}_i - \mathbf{x}_j)^2}{\sigma^2}) \simeq 0$$

when $i \neq j$. Then Equation (2) becomes $L = 1 - \sum_i \alpha_i^2$ which is maximized when all $\alpha_i^* = \frac{1}{N}$. In this situation, all the data points in the cluster become support vectors, and no compression is achieved. For a very large σ, $K_G(\mathbf{x}_i, \mathbf{x}_j) \simeq 1$, Then equation (2) becomes $L = 1 - \sum_i \alpha_i^2 - \sum_{i \neq j} \alpha_i \alpha_j$ which is maximized by letting only one $\alpha_i^* = 1$. Hence, only one data point will be selected to represent the entire cluster. We deduce that σ should be determined by considering the quality of the communication link between G_1 and G_2. If the link quality is high, then more support vectors can be used. If the link quality is low, then we should reduce the number of support vectors.

We consider three main components in quality estimation of the link between G_1 and G_2:

- Packet Delay Variation. Packet delay is the time it takes to transmit a packet from source to destination. Packet delay time includes queuing time (related to packaging time of transmitter), propagating time (related to distance and medium), processing time(related to quality of receiver) and transmitting time (related to relay stations, if exist). Under standard communication condition, nominal packet delay time across certain distance can be calculated before hand. The variation represents the deviation between actual packet delay and the nominal packet delay.
- Packet Loss Rate. Packet loss rate equals the number of packets lost during transmission divided by the total number of transmitted packets. Packet loss occurs due to congestion and broken communication link caused by node failures or blocks.
- Bit Error Rate (BER). Different from packet loss rate that the entire packet has been lost, bit error rate is the number of bit errors divided by the total number of bits in a received packet. Bit errors indicate the number of received bits that have been altered due to noise, interference, distortion or bit synchronization errors.

Our research goal is to determine a proper σ which will balance the representation error e_1 and e_2 while the resulted payload is below a safe threshold of the estimated communication link. In particular, we formulate a constrained optimization problem as follows:

$$\min_{\sigma} \quad E = e_1 + e_2 \tag{5}$$

under the constraint

$$\text{RLQ} \leq Q_t \tag{6}$$

where RLQ represents the Required Level of Quality for sending packets with certain size under the communication link and Q_t is a predetermined threshold.

IV. LINK QUALITY ESTIMATION

In general, the RLQ of a communication link to transmit n_d data bits can be modeled by the following equation:

$$\text{RLQ} = \eta(\kappa_1 \cdot \text{Var}(t_d) + \kappa_2 \cdot P) \tag{7}$$

The parameters κ_1 and κ_2 are positive regulating factors that can be chosen by design. t_d represents the packet delay and $\text{Var}(t_d)$ represents the variation in the packet delay. P represents the bit error rate. And η is a function of both the bit error rate and the packet loss rate. We will discuss how to determine these parameters.

A. Estimation of the Parameters

For G_2 to measure Packet Delay Variation, a nominal time delay t_n for transmitting a packet of certain size over an established link is needed. Under standard communicating condition, t_n can be theoretically calculated. The transmitter should process a n bits packet before transmitting it. Let this processing time be $t_s = n \cdot T_s$ while T_s is the time to process one bit data for transmitting. The time for receiver to process this packet is $t_r = n \cdot T_r$ while T_r is the time to receive and check one bit. Thus a nominal delay time for transmitting a packet is

$$t_n = 2 \cdot (t_s + t_r + t_p)$$

where $t_p = d/c$ is the propagating time from leaving the transmitter to arriving the receiver. d is the distance between the transmitter and the receiver, c is the nominal speed for a signal propagating in a medium. If the communication is over an underwater acoustic link, c equals about 1500 m/s. If G_2 measures m delay time $\{t_{d_i}\}$, the variance of packet delay can be calculated by

$$\text{Var}(t_d) = \frac{1}{m} \sum_{i=1}^{m} (t_{d_i} - t_n)^2 \tag{8}$$

However, the nominal time delay t is rarely known in practice since the distance between the two gliders is unknown.

The protocol we propose to evaluate the packet delay variation, the packet loss rate and the bit error rate is adapted from Stop and Wait (S&W) scheme. This protocol requires that the transmitter, say G_2, stops and waits for a acknowledgement response after sending a handshaking message. After a link has been built, G_2 sends G_1 a testing packet which contains n bits. In this n bits data, n_h is allocated for the header bits, n_d is allocated for data bits and n_t is allocated for the tail bits. The n_d data bits contains a predefined sequence which is known to both the transmitter and the receiver. When G_1 receives the packet, it checks the bit error and records the bit error rate as BER_s, also called *sending BER*. Immediately, G_1 starts a new test packet with its n_t tail bits filled with the BER_s as a responding packet to send back to G_2. After receiving this response, G_2 checks the bit error rate BER_r, also called *returning BER*, and reads BER_s together to estimate the bit error rate. Let the probability of bit error be p which can be a function of BER_r and BER_s, mathematically

$$p = f(BER_r, BER_s) \tag{9}$$

Then the total average bits error probability is

$$P = 1 - (1 - p)^{n_d} \tag{10}$$

Here we only concern the error of data bits and assume that the bit error occurs independently.

If G_2 fails to receive the response, then either G_1's response is lost or G_1 does not receive the packet at all. When G_2 sends a packet, no matter the response is received or not, we say that G_2 performs a *one-round test*. This one-round test is successful if the response of the packet is received and failing if the response is not received within a predefined time interval. If failure, a new packet is retransmitted by G_2 thus starts a new one-round test. When the response is finally received even after several reties or the maximum limit of retries has reached, we say a *full round test* has been performed. A full-round test may contain many one-round tests. If a one-round test fails and the packet has been retransmitted n_w times with t_w being the time interval between the retransmissions, then a delay of $n_w t_w$ occurs. Let T_d be the time of successfully transmitting one packet at the last one-round test. Then the full-round test takes a total time given below

$$t_d = T_d + n_w \cdot t_w,\ 0 \leq n_w \leq n_m \tag{11}$$

where n_w denotes how many times G_2 does not receive a response in a full-round test and indicates the Packet Loss Rate. n_m is the maximum limit of retries.

Suppose the handshaking protocol performs k full-round tests, then G_2 will measure a set of $\{T_{d_i}\}$ and a set of $\{t_{d_i}\}$, $i = 1 \cdots k$. We evaluate t_n by

$$t_n = \frac{1}{k}\sum_{i=1}^{k} T_{di} \tag{12}$$

The variance of packet delay $\mathrm{Var}(t_d)$ is then evaluated using equation (8).

For each one-round test, we just want to correctly transmit the n_d data bits. Time for processing n_d bits is $n_d \cdot T_s$ while the nominal time spent is t_n. Considering the bits error, the effective time for transmission of the data bits would be $n_d \cdot T_s \cdot (1-P)$ where P is the total average bits error rate in (10). The efficiency of the transmission then can be described by

$$\eta = \frac{n_d \cdot T_s \cdot (1-P)}{t_n} = \frac{n_d \cdot T_s}{t_n} \cdot (1-p)^{n_d} \tag{13}$$

This efficiency η takes the bit error probability p into consideration as well as the communication load n_d. The efficiency would be improved if n_d increases and p decreases.

Until now, all parameters in the RLQ equation (7) have been estimated. If a link has constant $\mathrm{Var}(t_d)$ and P, then larger n_d leads higher efficiency η, but requires higher RLQ. If either the $\mathrm{Var}(t_d)$ or P becomes larger, then the RLQ for sending a packet with the same size becomes higher.

Fig. 2 illustrates that the RLQ related to the number of data bits n_d. We assume that the CPU works at 10MHz e.g. $T_s = 1/10^7 s$. The average nominal time delay $t = 0.5s$, the packet delay variance $\mathrm{Var}(t_d) = 0.01$ and the bit error rate $p = 10^{-4}$. The regulating parameters are selected by $\kappa_1 = 1/8T_s$ and $\kappa_2 = 1/10T_s$. When the link is under poorer condition, the RLQ for transmitting packet with same size becomes higher. Fig. 3(a) shows that the RLQ along with the range of the bit error rate p evaluated by equation (9). The information bits in this case is selected as $n_d = 300$. The others are identical to those in Fig. 2. We can see that with increasing of p, the quality is deteriorating. RLQ changing with the time delay variance $\mathrm{Var}(t_d)$ is shown in Fig. 3(b). The bit error rate p is set to be 10^{-4}.

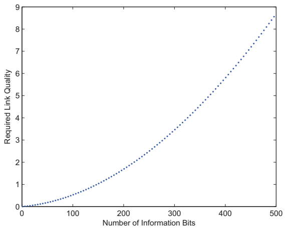
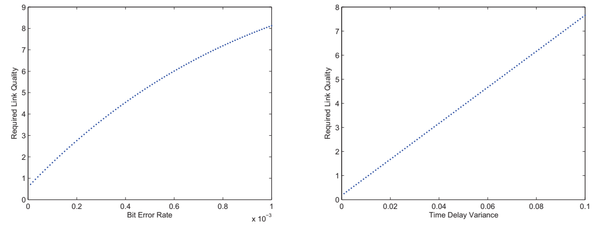


Fig. 2. The Required Level of Quality for transmitting varying n_d bits under certain communication condition.



(a) RLQ vs. p.

(b) RLQ vs. $\mathrm{Var}(t_d)$.

Fig. 3. The Required Level of Quality for transmitting certain n_d data bits under different communication condition.

B. Determine the Compression Level

We now come back to the optimization problem in (5). The constraint

$$\mathrm{RLQ} \leq Q_t$$

needs to be satisfied. RLQ is related to the number of data bits n_d. Let c be the number of bits for coding one number. There are N_{SV} two dimensional support vectors plus one parameter for σ, the width of Gaussian kernel, to be coded. Therefore,

$$n_d = c(1 + 2N_{\mathrm{SV}}). \tag{14}$$

Hence RLQ is now a function of N_{SV} determined by σ. The representing errors e_1 and e_2 in the optimization problem (5) are also determined by σ if the testing points are preselected.

It is difficult to solve this problem directly. Instead, we pursue approximation solution using numerical method. In our model, the RLQ is increasing with varying n_d under certain communication condition. So that the preset Q_t would claim that the transmitting data bits n_d should be less than a certain n_d^*. The number of support vectors related to n_d^* is denoted by N_{SV}^*. If the link quality is not considered, or no constraints existed in the optimization problem, there is a σ which leads minimal error E, denoted by σ_{op}. The number of support vectors resulted by σ_{op} is N_{SV}^{op}. We compare N_{SV}^{op} with N_{SV}^* to see whether the global optimal σ_{op} can be selected. If N_{SV}^{op} satisfies that $N_{\mathrm{SV}}^{op} < N_{\mathrm{SV}}^*$, which means the resulting communication load is within safe threshold, then σ_{op} is chosen. Otherwise, $N_{\mathrm{SV}}^{op} > N_{\mathrm{SV}}^*$, indicates that packet of such size would be corrupted during the upcoming communication. For sake of safety, we should minimize the number of N_{SV} to contain less communication load. The σ should be increased to minimize N_{SV}. While the increased σ leads larger error than minimal E. The optimal σ in this situation should be the one its resulting $N_{\mathrm{SV}} < N_{\mathrm{SV}}^*$ and the same time the error E is minimal.

V. PATH PLANNING SIMULATION USING REDUCED INFORMATION

After receiving the support vectors and weights from all other gliders, G_1 now has sufficient knowledge to plan paths using the fast marching algorithm [3]. G_1 interprets the information into knowledge about the outside blank zone. It checks whether a grid point is within an area with strong flow or not using the support vectors S and corresponding α^* using Equation (3).

We assume that five gliders G_1,G_2,G_3, G_4 and G_5 work in a rectangular region in the ocean that is discretized into 50 by 50 grid points. The initial positions are $G_1(14, 15)$, $G_2(18, 34)$, $G_3(25, 20)$, $G_4(39, 15)$ and $G_5(39, 39)$ respectively. The (simulated) spatially distributed currents in this region are generated by the ocean current model used in [8]. Within their own patch, gliders G_2,G_3, G_4 and G_5 pick out the grid points with velocity greater than $v = 0.22$ while G_1 is moving at speed $v_g = 0.25$. The selected high-flow regions are denoted by Ω_2, Ω_3, Ω_4 and Ω_5 respectively, shown in Fig. 4.

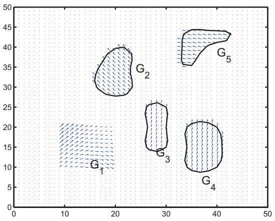
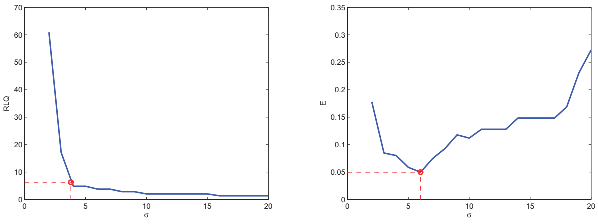


Fig. 4. SVDD results on the high-flow regions in each glider's local patch.

Each glider uses SVDD to compress the region with higher flow. The Gaussian kernel width σ is determined to satisfy equation (5) and constraint (6). We set parameters in link quality estimation $T_s = 1/10^7$s, $t = 0.5s$, $\mathrm{Var}(t_d) = 0.001$ and $p = 10^{-4}$. The regulating parameters are selected as $\kappa_1 = 1/8T_s$ and $\kappa_2 = 1/10T_s$. Let c in equation (14) be 30. Take Ω_5 for instance, we set $Q_t = 6.3$, the consequence is that the N_{SV} should be smaller than a threshold. We directly plot out the relationship between σ and RLQ, as shown in Fig. 5(a). The error estimation E change with σ is plotted in Fig. 5(b). Constrained by Q_t, the σ should larger than 4.2.



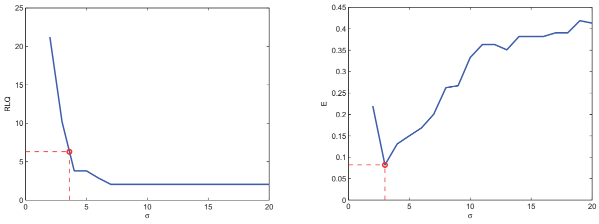
(a) RLQ vs. σ.

(b) Error E vs. σ.

Fig. 5. For Ω_2, the optimal σ could meet the requirement of safety link quality Q_t.

From Fig. 5(b), we can see that the least $E = 0.0499$ occurs when $\sigma = 6$. Thus in this situation, the optimal σ equals 6. For Ω_3, see Fig. 6(a), the σ should be larger than 3.6. Comparing with Fig. 6(b), the optimal σ should be 4 although the global minimum is located at $\sigma = 3$.

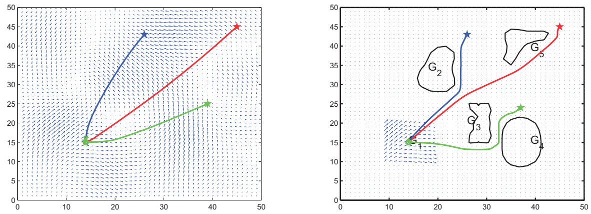
The distributed path planning result is shown in Fig. 7(b), we label this situation as "distributed". For comparison, we assume that G_1 knows the flow information of the entire region, this situation is labeled as "global". The global path planning results is shown in Fig. 7(a). We can see that in this situation the glider G_1 would leverage the high-flow region.



(a) RLQ vs. σ.

(b) Error E vs. σ.

Fig. 6. For Ω_3, the optimal σ could not meet the requirement of safety link quality Q_t. A sub-optimal σ is chosen.

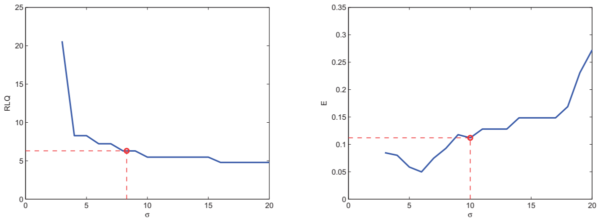


(a)

(b)

Fig. 7. (a) Global path planning; (b) Distributed path planning.

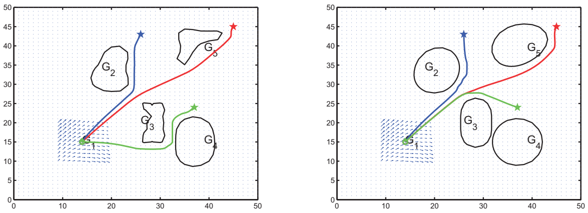
If the quality of a link is under lower level, in simulation, we change the time delay variance p from 10^{-4} to 10^{-2}, then sending same size packet would need higher required link quality. The RLQ with varying σ is shown in Fig. 8(a). If the safety threshold is required to keep to be 6.3, the σ should be larger than 9.2. Then the optimal σ moves to 10. Fig. 9 shows the paths planed with lower-level link quality comparing with the higher quality results. The planner of G_1 recognizes and rebuilds the high-flow region by checking whether a gridpoint is accepted by the SVDD mapping. If accepted, the flow on this point is signed to be the average flow together with this SVDD. When the σ becomes large, the description would accept lot of points that are not originally belong to the high-flow region Ω. Therefore the rebuilt regions in Fig. 9(b) are larger than the Fig. 9(a). Under poorer communication condition, more grid points are viewed as obstacles. The paths deviate from the optimal path.



(a) RLQ vs. σ.

(b) Error E vs. σ.

Fig. 8. For Ω_3, the global optimal σ could not meet the requirement of safety link quality Q_t. The chosen σ is a local one.



(a) Large Q_t, small σ (b) Small Q_t, large σ

Fig. 9. Paths planed with different σ.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents a novel approach to adjust the parameter of SVDD which is used to reduce information shared among underwater gliders for path planning based on link quality estimation. It is shown that the SVDD method with properly determined parameter σ can effectively represent the boundary of the regions with strong current. Simulation results show that the link quality estimation works well to determine the ratio of reduction to balance the accuracy of shared information with the quality of the communication link. Our future work will explore possibilities of representing the direction of the strong flow so that some high flow regions can be leveraged by path planning.

REFERENCES

[1] J. C. Yu, A. Q. Zhang, W. M. Jin, Q. Chen, Y. Tian, and C. J. Liu, "Development and experiments of the sea-wing underwater glider," *China Ocean Engineering*, vol. 25, pp. 721–736, 2011.

[2] 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*, vol. 27, no. 6, pp. 718–740, 2010.

[3] S. Liu, J. Yu, A. Zhang, and F. Zhang, "Cooperative path planning for networked gliders under weak communication," in *Proceedings of the International Conference on Underwater Networks & Systems*. ACM, 2014, p. 5.

[4] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[5] N. Baccour, A. Koubaa, L. Mottola, M. A. Zuniga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 4, p. 34, 2012.

[6] E. Troubleyn, I. Moerman, and P. Demeester, "Qos challenges in wireless sensor networked robotics," *Wireless personal communications*, vol. 70, no. 3, pp. 1059–1075, 2013.

[7] M. Stojanovic, "Optimization of a data link protocol for an underwater acoustic channel," in *Oceans 2005-Europe*, vol. 1. IEEE, 2005, pp. 68–73.

[8] D. Chang, X. Liang, W. Wu, C. R. Edwards, and F. Zhang, "Real-time modeling of ocean currents for navigating underwater glider sensing networks," in *Cooperative Robots and Sensor Networks*. Springer, 2014, pp. 61–75.