ISSN 2278-3091

Volume 4, No.2, March - April 2015

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/ijatcse/static/pdf/file/ijatcse01422015.pdf

Autonomous Surface Vessels Navigation Using Cluster Space Control Technique in Multi-Robot System

R.Latha [1], K.Vijayalakshmi[2]

[1]Asst prof of MCA,Vel tech High tech Engineering College,Chennai-62,India,rlatha80@gmail.com

[2] PG Scholar,Vel tech High tech Engineering College ,Chennai-62,India,viji.sumathi1993@gmail.com

ABSTRACT

Multi-robot systems control over many advantages over a single-robot managing system, including redundancy, coverage and flexibility. One of the important technical considerations in fielding multi-robot systems for real-world applications is the coordinating the different units. The cluster based controlling technique using simplified specification and monitoring of the movement of mobile multi-robot systems. Our previous work has established this approach and has experimentally verified its use for land-based systems consisting of 2-4 robots and with varying implementations ranging from automated movement control to human-in-the-loop piloting. In this paper, we plan the design and fabrication of a new low-cost autonomous surface vessel (ASV). The technical system includes a multi-robot system capable of autonomous navigation using the cluster control technique. It also adds a centralized controller, currently connected via a shore-based computer that wirelessly receives ASV data and relays using commands. Using the different cluster space control approach, these drive commands allow a robot to remotely drive a two-ASV cluster or to specify that the two ASVs maintain formation with a third robot. The resulting multi-ASV clusters can be translated, and resized depending on the needs of a particular application. Experimental results demonstrating these abililities of robots to provide, and plans for future work are discussed.

Keywords: Cluster, robot, ASV**.**

1. INTRODUCTION

Over the past 20 years, advances in GPS-based sensing, wireless communications, and control systems have enabled the development of automated boats for a wide range of applications ranging from environmental monitoring to physical manipulation. With new advances in multi-robot systems, multi-boat fleets have become possible. Such systems have potential benefits such as providing redundancy and reconfigurability, improving coverage /availability/ throughput,enabling spatially diverse functionality, supporting graceful constitution and degradation, and allowing the fusion of physically distributed sensors and actuators [1]. The Massachusetts Institute of Technology's SCOUT system is one of the best-known multi-ASV systems.

Consisting of up to four robotic kayaks at once, the SCOUT fleet has been used to demonstrate advanced navigation techniques as well as the provision of support services, such as communications and navigation support, for other marine assets. The U.S. Naval Academy has demonstrated the use of automated tugboat fleets and swarm navigation techniques in order to move other ships. And at Carnegie-Mellon University, researchers have demonstrated the use of two of their OASIS ASVs for tele supervised sensing of aquatic phenomena such as harmful algae blooms.

In particular, we are interested in a subset of such applications in which the controlled spatial distribution of tightly interacting robots provides enhanced performance. To address this challenge, we have developed a novel, control theoretic formation control approach, the cluster space control strategy, which allows specification, monitoring and control of formation attributes as if the formation was a virtual kinematic mechanism. On top of this formation control layer, we are developing an application specification layer that allows a user to specify application- or task-specific requirements in order to direct the operation of the multi-robot system. We have explored this cluster space control framework through considerable conceptual development experiments that have included land rovers, aerial robots, and marine surface vessels.

2. RELATED WORK

In geographic routing, the forwarding decision of each node is based on the locations and path of the node's one-hop neighbors and location of the packet destination as well. A forwarding node therefore needs to maintain these two types of locations. Many works, e.g. Quorum System have been proposed to discover and maintain the location of destination. Then, the maintenance of one-hop neighbors' location has been often neglected. Some geographic routing schemes, e.g. [2], simply assume that a forwarding node knows the location of its neighbors. While others, e.g. [3], uses periodical beacon broadcasting to exchange neighbors' locations. In the periodic beaconing scheme, individual node broadcasts a beacon with a fixed beacon at different interval. If a node does not hear any beacon from a neighbor for a certain period of time interval, called neighbor break of interval, the node considers this neighbor has moved away from the radio range and removes the outdated neighbor from its neighbor list. The neighbor break of interval is multiple times of the beacon interval. Heissenbuttel et al. [4] have showed that periodic beaconing can cause the inaccurate local topologies in highly mobile robot networks, which leads to performances degeneration, e.g. repeated packet loss and longer interruption. The authors discuss that the outdated entries in the neighbor list is the major source that decreases the performance. They proposed several simple optimizations that adapt beacon interval to node mobility or traffic load, including distance-based beaconing, speed-based beaconing and reactive beaconing. We discuss these three schemes in the following.

In the range-based beaconing, a node send a beacon when it has moved a given distance d. The node removes an outdated neighbor if the node does not hear any beacons from the neighbor while the node has moved more than k-times the distance d, or after a maximum time-out of 5s. This approach therefore is adaptive to the node mobility, e.g. a faster moving node sends beacons more frequently and backward. However, this approach has two complication. First, a slow node may have many outdated neighbors in its neighbor list since the neighbor time-out interval at the restrict node is extended. Second, when a fast moved node passes by a slow node, the fast node may not catch the slow node due the infrequent beaconing of the slow node, which decreases the recognised network connectivity.

In the speed-based beaconing, the beacon interval is dependent on the node speed. A node determines its beacon interval from a predefined range [6] with the exact value chosen being inversely proportional to its speed. The neighbor time-out interval of a node is a multiple k of its beacon interval. Nodes piggybacked their neighbor time-out interval in the beacons [7]. A receiving node compares the piggybacked time-out interval with its own time-out interval, and selects the lesser one as the time-out interval for this neighbor. In this way, a restrict node can have short time-out interval for its fast neighbor and therefore eliminate the first problem presented in the range-based beaconing. However, the speed(acceleration-based beaconing still suffer the problem that a quick node may not detect the slow nodes. In active beaconing, the beacon creation is trig- gered by data packet transmissions. When a node has a packet to send, the node first broadcasts a beacon request packet(package). The neighbors overhearing the request packet respond with beacons. Thus, the node can build an detailed local topology before the data transmission. However, this process is introdused prior to each data transmission, which can lead to excessive beacon broad-casts, particularly when the traffic load in the network is high.

The robot strategy proposed in this work dynamically adjusts the beacon update intervals based on the mobility dynamics of the nodes and the forwarding patterns in the network. The beacons transmitted by the nodes contain their current position and speed. Nodes estimate their positions systematically by employing linear kinematic equations based on the parameters announced in the last announced beacon. If the predicted location is different from the existing location, a new beacon is broadcast to inform the neighbors about changes in the node's mobility characteristics. Note that, an accurate representation of the local topology is particularly desired at those nodes that are responsible for forwarding packets. Hence, robot seeks to increase the frequency of beacon updates at those nodes that overhear data packet transmissions. As a result, nodes involved in promotes packets can build an enriched view of the local topology. There also exist some geographic routing protocols that do not need to maintain the neighbor list and therefore can avoid position updates, e.g. IGF [13}. These protocols are com- monly referred to as beacon-less routing protocols. The main ideal is that, the forwarding node broadcasts the data packet to all its neighbors who then distributed decide which node relays the packet. Normally, in these protocols, after receiving a packet, each neighbor sets a timer for relaying the packet based on some metrics, e.g., the distance to the destination. The neighbor that has the smallest timer will expire first and deliver the packet. By overhearing the delivered packet, other neighbors can erase their own timers and ensure that no duplicate packet is transmitted. Hence, the beacon-less routing Protocols can avoid excessive position updates and are particular suitable for networks where the topology is highly dynamic, e.g. in wireless sensor network where nodes periodically switch on and off (to save energy consumption)[20].

3. PROPOSED WORK

We consider that all robot nodes in the networks generate data packets periodically. Each packet generated by the sensor nodes should be reach to the sink within the dead line of each packet. Here we consider a mobile robot to do this data collection from sensor through some particular rendezvous points within the dead line of those packets. Here our aim is to plan a travel for the cluster according to the delay, utility and the total length of the travel. The following modules are constructed as shown below.

3.1 Analyzing the cluster details

3.2 Setting less hop count transmission

3.3 Select robot as cluster

3.4 Find and collect data from clustered robots

3.5 Handover the data to BS.

3.1 Analyzing the cluster details

Handover the data to base station robot when robot within the transmission coverage area of robots. The robots which are located in the range of Basestation it transforms all the information to the base station with minimum hops.

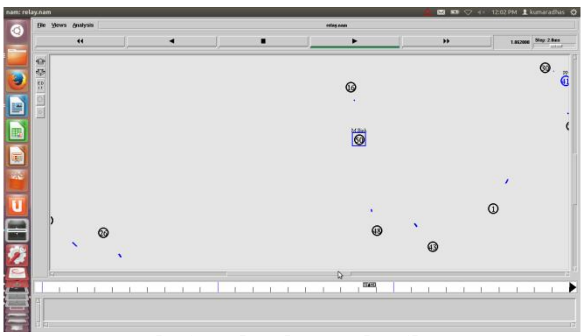

Figure 1: **Analyzing the cluster details**

3.2 Setting Less Hop Count Transmission

Multi-hop routing, packets have to knowledge multiple relays before reaching the base station. Minimizing energy consumption on the forwarding path does not necessarily prolong network lifetime as some popular sensors on the path. So to avoid the difficulty in multi-hop routing we are setting the less hop count transmission.

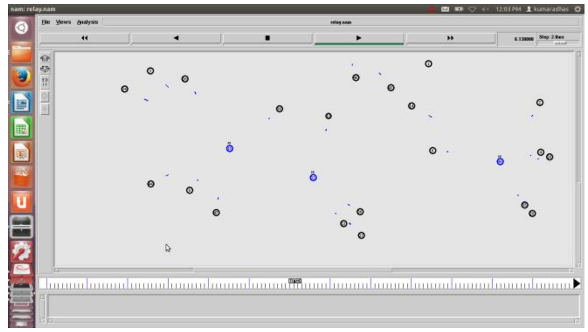

Figure 2: Setting Less Hop Count Transmission

3.3 Select robot as cluster

A subset of robots will be selected as the cluster, each aggregating the local data from its affiliated robots within a certain number of relay hops. These clusters will temporarily cache the data and upload them to the mobile robot when it arrives. The clusters can simply be a subset of robots in the network or some other special devices, such as storage robots with larger memory and more battery power.

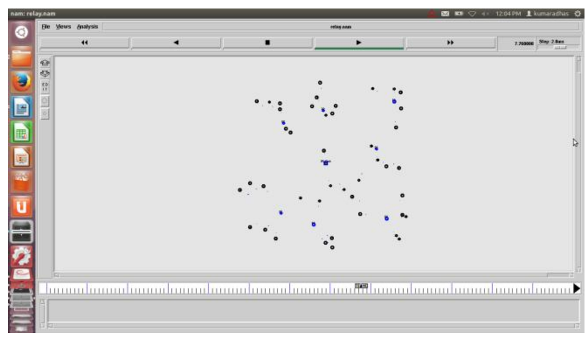

Figure 3: Select robot as cluster

3.4 Find and collect data from cluster robots

Since the mobile robot has the freedom to move to any location in the sensing field, it provides an opportunity to plan an optimam tour for it. Our basic idea is to find a set of specialized nodes referred to as clusters in the network and determine the tour of the mobile collector by visiting each cluster in a specific sequence. When the mobile robot arrives, it polls each cluster to request data uploading. And then upload the data to mobile robot.

3.5 Handover the data to BS

A cluster uploads data packets to the mobile robot in a single hop. The mobile robot starts its tour from the base station, which is located either inside or outside the sensing field, collects data packets at the clusters and then returns the data to the base station. Finally mobile robot handover the data to base station.

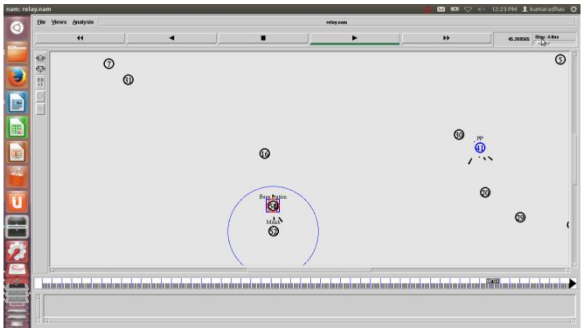

Figure 4: Handover the data to BS

4. SIMULATION

In this section, it describes the simulation tool and various parameters chosen for simulation. The various performance metrics used to compare the performance of mobile sink moving in different methods. The speeds are uniformly chosen between the minimum and maximum speeds set to 0 m/s and 20 m/s, respectively. When the node reaches its destination, it stays there for a certain *pause time*, after which it chooses another random destination point and repeats the process. The simulation ends for 100s. The data traffic is generated by Constant Bit Rate (CBR) sessions initiated between the source and destination. All the nodes are assumed to have the same amount of battery capacity with full energy at the beginning of the simulation and initial energy of each node is 0.5 J. Transmitting power and Receiving power of each node are some 0 mW and 0.335mW respectively.

4.1 Energy Efficiency (EE)

Total received data measured in Bytes/Joule at the end of simulation.

EE=Total received data (bytes)/total consumed energy (Joules).

If battery capacity of node reaches zero then node will die.

The lifetime of network can be included in many process:

1. It may be defined as the time taken for K% of the nodes in a network to die
2. It might be the time appropriated for the first node to die.
3. It can also be the time for all nodes in the network to die.
4. The comparisons of our work are shown below.

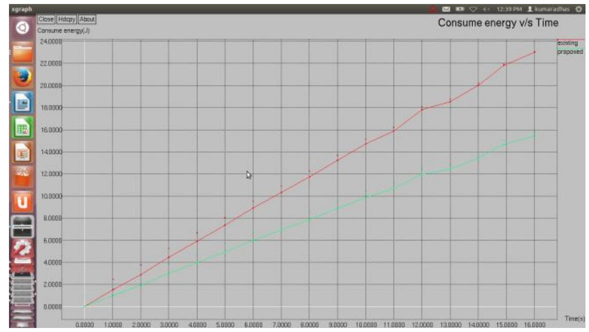

Figure 5: X-graph 1

The comparison of energy consumption is shown in above Fig. It depends on overhead and load, the path failure mainly depends on due to lack energy of any one node or RP on selected path that sink moves.

4.2 Throughput

In wireless sensor networks, such as transmission or packet radio, network throwout is the average of successful message delivery over a communication channel. This data may be passed over a physical or logical link, or send through a certain network node. The throughput is usually measured in data packets per time slot. It can be computed as maximum throughout, maximum hypothetically throughput, maximum sustained throughput, peak throughput(throwout), normalized throughput(throwout) and so on.

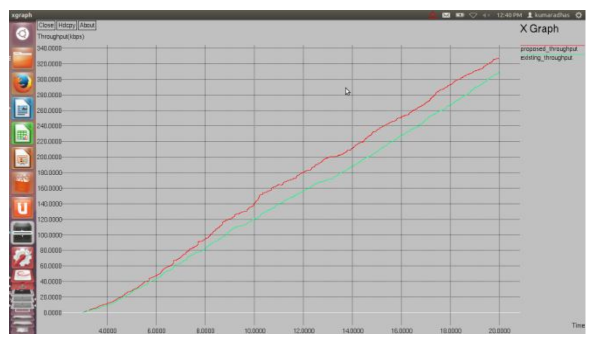

Figure 6: X-Graph 2

The comparison of throughput is shown in above Fig. It depends on the rate of data packet flow in a particular channel that successfully delivered the data packet.

Throughput = File Size / Transmission Time (bps)

5.CONCLUSION AND FUTURE WORK

In our work, we have done clustered, the method for controlling the movement of a robot in a network. Robot selects the set of clusters from the robotic network that the energy consumption of robots is reduced and to prevent the formation of energy holes while assure sensed data are collected on time. We also compare clusters against existing schemes in terms of the difference between robot node energy consumption. Our simulation results show that clusters uniformly distributes energy consumption and better than existing methods. As a future work, we plan to enhance our approach to include data with different delay requirements. This means a mobile sink is required to visit some sensor nodes or parts of a network more frequently than others while ensuring that energy usage is minimized, and all data are gather within a given deadline. Moreover, we plan to extend clustered robot to the multiple mobile sinks/rovers case. Also we generate emergency signal generation from sensor nodes that can be reached to sink through multi hop communication of clusters.

REFERENCES

[1] C. Kitts and M. Egerstedt, “Design, Control and Applications of Real- World Multirobot Systems.” IEEE Robotics and Automation Magazine, v 15, n 1, March 2008, p. 8.
[2] J. Curcio, J. Leonard, and A. Patrikalakis. "SCOUT - a low cost autonomous surface platform for research in cooperative autonomy," Proceedings of MTS/IEEE OCEANS, vol 1, 2005, pp. 725- 729.
[3] J. Curcio, et al. “Experiments in Moving Baseline Navigation using Autonomous Surface Craft”, Proceedings of MTS/IEEE Oceans, vol 1, 2005, pp. 730-735.
[4] J. Esposito, M. Feemster, and E. Smith, "Cooperative manipulation on the water using a swarm of autonomous tugboats," IEEE International Conference on Robotics and Automation, May 2008, pp.1501-1506.

[5] J.M. Dolan, et al. “Cooperative aquatic sensing using the telesupervised ocean sensor fleet.” Proceedings of Remote Sensing of the Ocean, Sea Ice, and Large Water Regions, vol 7473, 2009 , pp. 1-12.
[6] C. A. Kitts and I. Mas, “Cluster space specification and control of mobile multirobot systems,” IEEE/ASME Transactions on Mechatronics, vol. 14, no. 2, April 2009, pp. 207–218.
[7] I. Mas and C. Kitts. “Obstacle Avoidance Policies for Cluster Space Control of Non-Holonomic Multi-Robot Systems.” IEEE/ASME Transactions on Mechatronics. In Press.
[8] I. Mas and C. Kitts. “Model-Based Nonlinear Cluster Space Control of Mobile Robot Formations.” Multi-Robot Systems, Trends and Development, T. Yasuda (Ed), INTECH, Ch 4.
[9] I. Mas, S. Li, J. Acain, and C. Kitts. “Entrapment/escorting and patrolling missions in multi-robot cluster space control.” Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, Oct 2009, pp. 5855-5861.
[10] P. Mahacek, I. Mas, O. Petrovic, J. Acain and C. Kitts. “Cluster Space Control of a 2-Robot System as applied to Autonomous Surface Vessels.” Proceedings IEEE/MTS Oceans Conference, Quebec City, Canada, pp. 1-5, Sep 2008.
[11] P. Mahacek, I. Mas, O. Petrovic, J. Acain, and C. Kitts. “Cluster space control of autonomous surface vessels.” Marine Technology Society Journal, v 43 n 1, 2009, pp. 13-20.
[12] P. Mahacek, I. Mas and C. Kitts. “Cluster Space Control of Autonomous Surface Vessels Utilizing Obstacle Avoidance and Shielding Techniques.” Proceedings IEEE/OES Autonomous Underwater VehiclesConference, Monterey CA, pp. 1-5, Sep 2010.
[13] P. Mahacek, C. Kitts and I. Mas. “Dynamic Guarding of Marine Assets through Cluster Control of Automated Surface Vessel Fleets.” Submitted to IEEE/ASME Transactions on Mechatronics.
[14] E. Fiorelli, N.E. Leonard, P. Bhatta, D. Paley, R. Bachmayer, and D.M. Fratantoni, Multi-AUV Control and Adaptive Sampling in Monterey Bay. Proceedings of IEEE Autonomous Underwater Vehicles 2004: Workshop on Multiple AUV Operations. Sebasco, ME. June 2004.
[15] V. Howard. “A Study of Gradient Climbing Technique Using Cluster Space Control of Multi-Robot Systems.” M.S. Thesis, Department of Mechanical Engineering, Santa Clara University, Santa Clara, CA,January 2011.
[16] T. Adamek. “Cluster Space Gradient Contour Tracking for Mobile Multi-robot Systems.” M.S. Thesis, Department of Mechanical Engineering, Santa Clara University, Santa Clara, CA, December 2010.
[17] P. Mahacek, T. Adamek, V. Howard, K. Rasal, C. Kitts, B. Kirkwood, and G. Wheat. “Cluster Space Gradient Tracking – Control of Multi- Robot Systems.” Proceedings of the ASME Information Storage and Processing Systems Conference, Santa Clara CA, June 2011. In Press.
[18] G. Okamoto, C. Chih-Wei, and C. Kitts. “Beamforming Performance for a Reconfigurable Sparse Array Smart Antenna System via Multiple Mobile Robotic Systems.” Proceedings of the SPIE - The International Society for Optical Engineering, v 7706, pp 77060-71. April 2010.