Open access • Journal Article • DOI:10.1109/JSYST.2010.2040225

Consensus Control for a System of Underwater Swarm Robots — Source link

Matthew Joordens, Mo Jamshidi

Institutions: Deakin University, University of Texas at San Antonio

Published on: 17 Feb 2010 - IEEE Systems Journal (IEEE)

Topics: Robot control, Swarm robotics, Automatic control, Mobile robot and Robot

Consensus Control for a System of Underwater Swarm Robots

Matthew A. Joordens, *Member, IEEE*, and Mo Jamshidi, *Fellow, IEEE*

***Abstract*—The control of a swarm of underwater robots requires both a control algorithm and a communication system. Unfortunately, underwater communications is difficult at the best of times and so large time delays and minimal information is a concern. The control system must be able to handle a large number of robots without a master control, i.e., a decentralized control approach. This paper describes Consensus control as a way to decentralize. Consensus control allows each robot to know the final goal and then to decide, based on the position of the other robots, its best move to help achieve the goal.**

***Index Terms*—Robots, underwater vehicles.**

I. Introduction

THE control of a multivehicle system stems from the work being done at the Autonomous Control Engineering (ACE) Laboratory at the University of Texas, San Antonio (UTSA). This work is in systems of systems (SoS). The idea is that different systems can be made to cooperate. These systems could be robotic, automation or even human. The ACE lab is currently looking at taking systems of different types of robots (land, air, and sea). This paper concentrates on the underwater realm.

Further, this paper looks at the control of a swarm of underwater robots using consensus control. There are a variety of methods that can be used to control a swarm, such as consensus, receding horizon and spatio-temporal. [1]–[7] The bulk of these methods used the principles of consensus control as main control method. Consensus provides a broad framework and the guide to the control system allowing other methods to be used within it. Most control methods used predictive control [1]–[5] which is required here as well. These methods, however, are mainly focused on maintaining a formation of agents, whereas the focus of this paper is the cooperative control to complete a task. Consensus control gave the flexibility required by allowing other methods within it.

II. Background

The oceans of the world are, even today, a great unknown. Recreational divers can only dive to depths of up to 40 m and then only for a few minutes. Commercial/technical divers do not venture much below 300 m.[8] This is largely because of the pressure of the water at depth. To go below this depth submersibles are used. They may be manned or unmanned.

Manuscript received February 04, 2009; revised August 27, 2009; accepted August 27, 2009. Date of publication February 17, 2010; date of current version April 07, 2010. This work was supported in part by Deakin University and University of Texas, San Antonio, Autonomous Control Engineering.

M. A. Joordens is with the School of Engineering, Deakin University, Geelong 3217, Australia (e-mail: matthew.joordens@deakin.edu.au).

M. Jamshidi is with the Department of Electrical and Computer Engineering, University of Texas at San Antonio, San Antonio, TX 78249 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSYST.2010.2040225

Fig. 1. Military SoS (courtesy of the Bureau of Industry and Security, U.S. Department of Commerce).

As an example, both types of vehicles were used to explore the Titanic, which is in 3840 m of water [9]. At this depth, the pressure of the water is 385 bar [8] ! There are not many vessels built to withstand that pressure.

Even in the shallower waters most sea based human endeavour is limited to harbors, oyster farms, oil rigs, and research. Most of the work done by underwater vehicles/robots is limited to visual inspection. But what if we can do more?

Autonomous underwater vehicles (AUVs) are unmanned, untethered, self-propelled platforms.[10] AUVs have the potential to revolutionize our access to the oceans to address the critical problems faced by the marine community such as underwater search/rescue [11] and mapping, climate change assessment, underwater inspection, marine habitat monitoring, shallow water mine counter measures [12], and scientific studies in deep ocean areas. Recent trends in AUV technology are moving towards reducing the vehicle size and improving its deployability to reduce the operational costs. This will make it possible to create swarms of robots to operate and perform tasks that would be difficult for a single robot.

III. Systems of Systems

SoS may be a new area of research but the idea is quite old. The defence force has been interested ever since fighting began. An army is comprised of many different systems, be they the old cavalry, foot soldiers and pikemen to the modern infantry, tanks, planes and ships, (Figs. 1 and 2). The army that was often victorious was the one that could control these separate systems as one coordinated system of systems [13]. Today, modern technology has allowed this type of SoS to work very effectively.

Modern communications allow each system to know what the other systems know and then the different systems can make informed decisions towards the goal of the whole system.

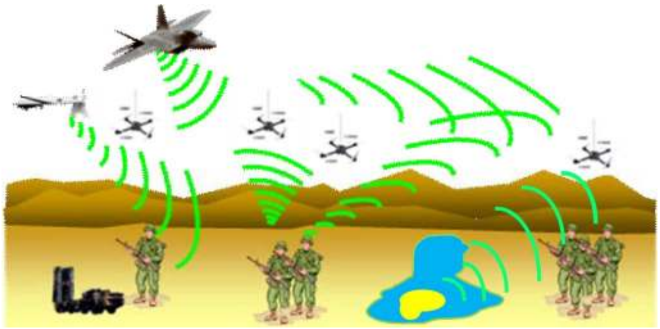

Fig. 2. SoS with water, infantry, and air units.

IV. Consensus Control

This is what consensus control is based on. In an ideal world, each system knows exactly what every other system knows at the same time [14], [15]. This is normally impossible but consensus control attempts to get close to this [16]–[18]. It is a strategy of disseminating knowledge to multiple units which, in this case are robots. By giving or sharing information, all units have the same knowledge and each robot can form an opinion of the action to take. Thus there is no central command robot or master control. It tries to work much like a football team. The team has a goal, which is to get the ball to one end of the field. There is no central control; all the players have a single overall goal, but to make that goal a reality each player will decide upon its own sub goal.

Consensus control shares to required knowledge and then lets the individual units formulate their own plan

V. Possibilities for Consensus Control

A. Ship Inspection

In today's world of terrorism many new security measures must be taken. One such terrorist threat is limpet mines on the hulls of large ships, such as oil tankers. The mines are very easy to place but, because of the size of the vessel, are hard to detect. If a mined ship got into a harbor and then was destroyed, the damage would be catastrophic. One needs to inspect the ship before it gets into the harbor.

There is one opportunity to do this and that is when the pilot is transferred to the ship just outside of the harbor. Using a single inspection robot to inspect the hull would take too long. This is a very dangerous job for divers and would still take too long. But what about a swarm of consensus controlled robots? They could be thrown over the side of the pilot boat, each inspect a small portion of the ship, knowing where each other robot is inspecting. This would very quickly put a total picture together of the ship's hull and any abnormalities could be detected quickly.

B. Undersea Harvesting

There are many resources at the bottom of the oceans that are too hard and expensive to mine of harvest. One case is manganese nodules. These nodules can be found strewn over the sea bed either too deep or too widely distributed for divers to collect. However, a swarm of robots, each knowing where the other robots are, may well be able to harvest this resource [13].

VI. Robot Types

So what sort of robots should be used? The main difference between the robots is if they are tethered or not.

A. Tethered

A tethered robot allows for easy and fast communications. This is desirable when a large amount of information, such as video data, is to be shared.

They almost have unlimited power, as the power is supplied through the tether. This means that the robots can be strong and fast units.

They have off board intelligence, normally on a computer on land or the mother ship. This allows large and powerful computers to be used which will be able to easily handle the large amounts of information.

On the other hand, the robots range is limited by the length of the tether and there is a constant concern about entangling the tether. A tangle means that a diver must retrieve the robot, other robots must be used to retrieve it or it must be cast off and lost.

B. No Tether

This type of robot can have a longer range.

It also does not have to worry about any tether drag reducing it efficiency.

Entanglements are also not a concern giving this robot greater freedom of movement.

On the other hand, this robot must be self contained. Thus, it must carry its own power supply. To conserve this power it must move slowly and cannot be very powerful.

It also has its computer on board, necessitating a smaller, less powerful control unit.

But the biggest difficulty is communications. As radio is very poor in water, either low wave length or sonar must be used. This makes the communications very slow and error prone. Only small amounts of data can be transmitted.

C. So Which Robot?

Both types of robots have their advantages and disadvantages. With consensus control however, one does not have to choose between the two. The robots without tethers can be used as scouts. They can range far away from the mother ship and guide that ship to any points of interest where their tethered cousins can do the heavy work. Consensus control gives the goal, but the individual robots can decide what to do based on their knowledge from the other robots and their own special abilities.

Shown in Fig. 3 are the two robots that the author is working with. The tethered VideoRay and the author's own home built tetherless robot [19].

VII. Communications

Underwater communications can be implemented in numerous ways including acoustic propagation, fiber-optic communications, and radio modems. Fibre-optic or any cable based systems can be ruled out immediately. Anything that requires a tether means that only a few robots can be used. A swarm of robots with a swarm of tethers will get tangled up very quickly [20].

Fig. 3. A possible pairing, the author's tetherless robot working in cooperation with a tethered VideoRay.

As detailed in [21], acoustic propagation faces lots of problems compared to radio modems. These problems are mainly due to very limited bandwidth, large signal propagation time and overload on the receiving antenna by local transmit power levels (Near and Far problem). The limited bandwidth implies that the use of multi-channels techniques is very limited. The near and far problem occurs when an acoustic unit may not transmit and receive at same time because of local transmit power levels. Large propagation delays involved in acoustic propagation are in the range of seconds. All of these factors lead to a decision to choose some alternative technology to communicate effectively between the AUVs. Thus it was decided to try using radio modems for communication.

The radio modem chosen was the Zigbee module. Zigbee is a low-power wireless communication technology and international standard protocol for the next- generation wireless networking. It reduces the data size and allows for lower-cost network construction with simplified protocol and limited functionality. Zigbee uses the [22] MAC layers and PHY layers defined by IEEE® 802.15.4, which is the shortest distance wireless communication standard for 2.4 GHz. The benefits of Zigbee are robustness, simplicity, low-power consumption, and mesh networking [23] 802.15.4 provides a robust foundation for Zigbee, ensuring a reliable solution in noisy environments. Features such as channel assessment and channel selection help the device to pick the best possible channel, avoiding other wireless networks such as Wi-Fi. Message acknowledgment helps to ensure that the data is delivered to its destination. The ability to cover large areas using routers is one of key features of the Zigbee network and helps to differentiate it from other technologies [24]. Mesh networking can extend the range through routing and it also has self healing capability that increases reliability of the network by re-routing a message in case of node failure. Finally, multiple levels of security ensure that the network and data remain intact and secure.

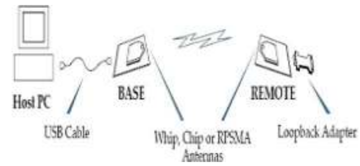

Fig. 4. Experimental setup [25].

All this is very good with one flaw. The module's frequency of 2.4 GHz is best absorbed by water. (Microwave ovens use a frequency of 2.45 GHz for just this reason.) Therefore the range of such a system is probably too small for this application. There has not been any previous work on the use of Zigbee modules in an underwater environment as far as the author can determine. The small size of these modules which would allow easy placement inside a robot and a small antenna requirement where too tempting to ignore.

A rough experiment was set up to test the possibility of using Zigbee modules. Two modules were set up in pipes with a closed end. The pipes were dipped into the water and RS232 cables ran up out of the pipes to a computer. This experiment indicated that the system could communicate up to 3 m. The open pipes were a concern but still this warranted a more controlled experiment.

The experiment with the Zigbee modules was carried out in a 3-m-deep swimming pool to verify the signal strength and to see range of attenuation affected by the signal based on distance between transmitter (base) and receiver (remote) and depth of base and remote. Each time 15 packets of information is sent from base to the remote. Base, remote and the experimental setup is shown in Fig. 4, with the antennas immersed in the water connected to the shielded modules with shielded cable.

This experiment gave a range of over 8 m. This indicated that something was wrong. The shielding was insufficient. After a few more experiments it was determined that the correct setup was to shield all equipment, to impedance match the antenna cable and to physically ground the shielding to an earth stake. Using this setup it was found that the modules can only communicate underwater to a range of 0.15 m.

Whilst this meant that 2.4 GHz modules could not be used, there was not much hope for them anyway, it set the experimental method to be used to investigate other, lower, frequencies. The ACE lab at UTSA is currently working on a multiband, low-frequency system to overcome these problems.

Until the right frequency is determined the authors will use 300-Bd acoustic modems. This is a worst case scenario and means that only about 30 bytes can be transmitted per second. If significant control can be achieved with this communications system, then faster system will be that much easier.

VIII. Pose

The next problem in consensus control is knowledge of the robot's pose, or position and orientation. Information about the robots position, the direction it is facing, and other factors such as its velocity are all important for the other robots to be aware of.

A. Dead Reckoning

So, how is this information determined? GPS, that wonderful navigation system used by most systems in the world today does not work underwater. The main method of navigation underwater is dead reckoning. This system determines the vessels orientation, its velocity and time taken to estimate its new position. Large vessels like submarines can use large accurate gyroscopes to determine this information. In the small units, inertial measurement units (IMUs) are used. The problem with this system is that any errors that occur (and they will occur) will accumulate.

B. TriTech Navigation

One possible system that is being used by the author is the Tritech MicroNav system. This system uses a transponder on each robot to be tracked. It has one buoy in the water and, it can give a bearing and range to each transponder using a sonar ranging system. It can convert the bearing and range into X, Y, and Z coordinates with itself being at coordinates (0,0,0). By placing a GPS unit on the Buoy, it can even give GPS locations for the transponders.

The MicroNav can track up to 16 different transponders at a rate of 4 per second. Unfortunately, the author has so far only managed a rate of 1 per second and these are prone to positional errors. This information then needs to be shared with the robots so that they know where they are. The slow rate of positional information is not that much of a concern as the communications system used cannot work any faster anyway.

It was decided to use the MicroNav system to give the robots a known position to work from where the MicroNav is at the origin (0,0,0).

IX. Setup

Due to the communications problems, the author choose to use the commercial VideoRay robots for this part of the research. The VideoRay is a small tethered robot controlled from an on shore computer. The MicroNav tracking was then to be used to allow the robots to get a fix on their position whenever the MicroNav had that information available. At other times, the VideoRay only has a compass and a depth sensor to guide it.

The author was performing the experiments in a pool to have a controlled environment. Unfortunately, the MicroNav could not perform well in the confined space of a pool. There were too many reflections off the pool walls for the MicroNav to deal with. Therefore, the author had to start with simulation and then move to a lake or ocean at a later date.

X. VideoRay SDK

The robots may know where they are now but they still need to be controlled autonomously. The author first looked at the current software for the VideoRay which is a good product built in LabVIEW. The author, however, needed to handle more than one VideoRay and possibly other robot types and needed autonomous control. It also needed to interface to the navigation system, sonar units, inertial measurement units (IMUs), and other possible sensors. The author decided then to write his own software. As the author would need various different control packages, it was decided to write a software development kit (SDK) for the VideoRay to make the control code easier to write. This gives one the ability to quickly generate code that can control the VideoRay.

XI. Multiple Vehicle Control

For research purposes, it was easier to control several vehicles on one computer. This made the simulation of communications between the vehicles easy to perform.

For the purposes of looking at consensus control a swarm of robots was needed. The author had a "swarm" of two VideoRays. Thus more VideoRays were needed. It was decided then to simulate more VideoRays.

To do this, a VideoRay was filmed moving next to a scale and the film was then analysed to determine its accelerations and velocities at different thruster powers. The allowed the VideoRays simulation software to be tested against the real robot and thus the author could create as many VideoRays in the software as required. By driving the VideoRay at various power levels, the acceleration and maximum speeds at those levels was determined. This information was enough to develop a simulator accurate enough to combine virtual VideoRays with real ones. The simulator can handle two real VideoRays or include them as part of the virtual swarm.

The real VideoRays can then cooperate with the virtual ones and thus determine how well they cooperate and how accurate the simulations are.

XII. Consensus

In order to test the consensus control, a simple task was provided. The swarm of robots were to patrol a square path defined by four waypoints. Ten simulated VideoRays were produced all at different depths to avoid collisions.

As can be seen in Fig. 5, the robots are very haphazard to the patrol. Bunching up in places and leaving other areas sparse.

To create a better patrol the robots should be evenly spaced. For this to happen, consensus control demands that each robot must have the same overall goal, know where the other robots are, and hence share its position and then decide on its own action.

There are four steps required to set up consensus control [17].

A. Cooperation Objective

It must be determined what constitutes cooperation. In this case the distance between each of the robots must be the same. Hence

$$J = \frac{p}{n} - \frac{\left(\sum_{i=2}^{n} VR_1 - VR_{i-1}\right)}{n} \tag{1}$$

where $VR_i - VR_{i-1}$ is the distance between two VideoRays, one following the other, and where

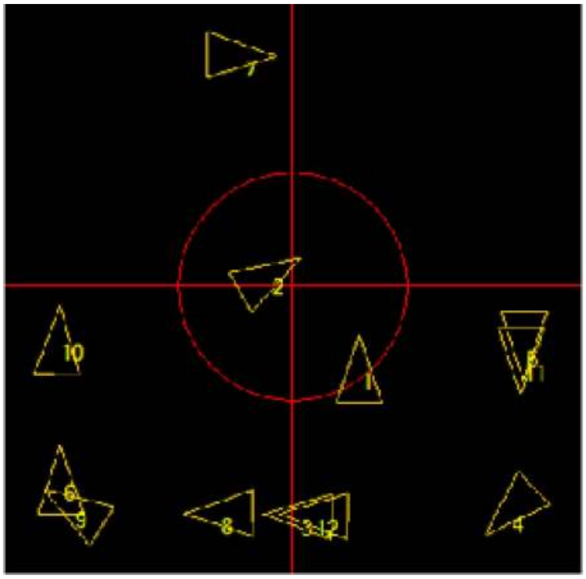

Fig. 5. VideoRays patrolling a square without consensus control (The two middle robots are the real VideoRays sitting on the bottom).

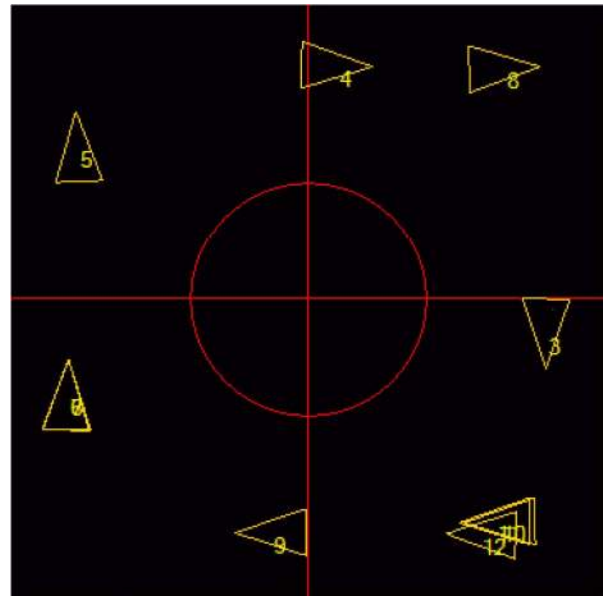

Fig. 6. VideoRays patrolling a square with consensus control.

p total distance of the path;

n number of VideoRays;

J cooperation constraint.

In this case, cooperation is said to be achieved when the distance between all the robots (VideoRays) is equal to the total length of the path divided by the number of robots in which case J would be zero. To allow some tolerance one can say that when $J < \epsilon$ then the robots have achieved ϵ-cooperation, where ϵ is the error margin allowed.

B. Information Requirements

To achieve cooperation, it must be determined what information is to be shared.

In this case that information in each robots location and its identification, a unique number to define each robot.

C. Centralized Strategy

Next, a centralized strategy is identified to obtain the required goal.

In this case, each robot much be sped up or slowed down to maintain its position in the patrol and to maintain the equal distances between each robot.

D. Consensus Building

Now the centralized strategy is broken down so each robot can make its own decisions to achieve the common goal.

In this case each robot can determine the number of robots present (or the number it thinks are present if communications are bad). It knows the total length of the patrol path, and so can calculate the distance that it must stay behind the robot in front of it.

As can be seen from Fig. 6, the consensus control has given better control to the robots that are not using any sensors, just the location information being shared. There is still some bunching up due to creating some time delay in the communications system.

XIII. Communications Time Delay

Using 300-Bd communications means that very little information can be sent and only very slowly. Each robot will share its positional information. The information packet sent consists of the robots ID, its X, Y and Z coordinates and its heading. The robot with the smallest ID number sends its information first. With the act of sending goes the power to transmit again. All robots receive this information and the robot with the next highest ID gains the power to transmit its information which it does as soon as possible. Once the robot with the highest numbered ID transmits, no robot can send. The lowest numbered ID robot waits for an allocated period of time and, if nothing is received, starts the process again. This method can be used if the robots are sequentially numbered and are all close enough to each other to ensure that they will all receive every transmission. This method had a break in transmission after the highest robot has transmitted, but no time is lost in acknowledgement transmissions.

Even so, the time delay is very large and hard to deal with even in this fixed topology system.[26] Because of the large time delay, we will not consider it in the consensus algorithms as many others have done for smaller delays [16], [27], [28] but, instead attempt to correct the data.

The time delay means that each robot knows the position of the other robots sometime in the past, but it does not know its current position. As seen in the last run, there is a bunching up of robots as each robot has invalid positional information about the other robots. There are actually two time delays, the time it takes to communicate and the positional data that gets more inaccurate as time passes until it is updated.

The first delay is very small compared to the second delay and has been studied in various papers [18], [29].

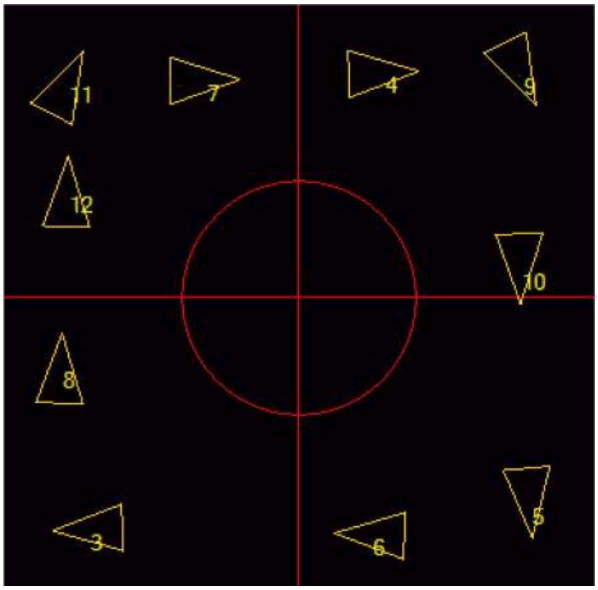

Fig. 7. VideoRays patrolling a square with consensus control using position prediction.

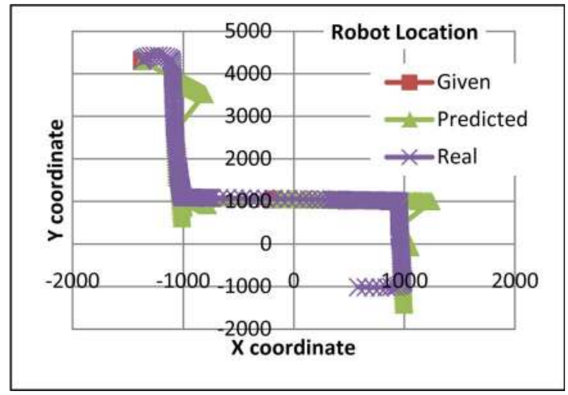

Fig. 8. Plot of robot (Z coordinate is constant and therefore ignored).

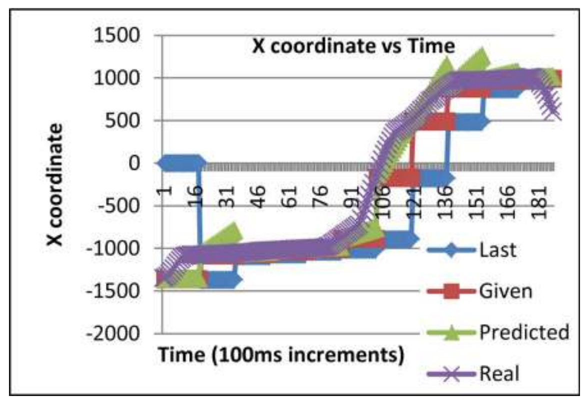

Fig. 9. X coordinate of robot over time.

The second delay can be as large as 2 s. The robots used can move up to 600 mm in 2 s or almost two robot lengths! This is the delay that needs to be addressed. We can try to predict the other robots paths [30]. Various control methods look at disruption to the communications system and the need to use prediction [1]–[5]. All of these studies, however, assumed that the robots had the sensor capability to determine the other robots positions without any communications. This was not the case here as the VideoRays only had a compass and a depth sensor. The camera on the VideoRays was not being used as it was decided that murky water could easily render this option useless.

To predict the current position, each robot timestamps each packet of information as it is received and keeps the last two packets of information about each other robot. Linear extrapolations are then done to predicted or estimate the current location of each robot as follows:

$$[P'_n] = [P_n] + ([P_n] - [P1_n]) * \frac{(T - t_n)}{(t_n - t1_n)} \tag{2}$$

where

N	robot's ID;
P'	estimated X, Y, Z coordinates of robot n ;
P	last known position of robot n;
$P1$	next-to-last known position of robot n;
t	time of P ;
$t1$	time of $P1$;
T	current time.

Using this prediction/estimation approach, a further run was performed as seen in Fig. 7.

As can be seen, the distribution of the robots is much more even. Let us look closer at one of the robots as seen by the robot behind it.

Fig. 8 shows the movement of one robot finding its first waypoint and then starting to move around the square of four waypoints. It shows the given or last known positions, the predicted plot, and the actual plot. The predicted plot looks terribly wrong in places while the given positions always look right. This is deceiving however. The given positions are always right but they are only valid at certain points in time. A better way to look at this is by looking at each coordinate over time.

Figs. 9 and 10 show the X and Y coordinates of a robot as seen from the robot following it. The given and last given plots show the stepwise nature of the information sent by the robots. As can be seen, the given positions are initially accurate but then become more inaccurate compared to the real position of the robot. The predicted plot is only truly accurate when the robot has been moving in a straight line, after all, a linear extrapolation was used, but in most cases the predicted position is more accurate then the given position—see Table I.

Table I shows quite large errors, particularly in the Y axis. This begs the question: Even with the improvement in error, how did the predicted approach work?

A review of the data saw that there were a few very large errors, mainly while turning corners. A better examination of the errors is to look at 95% of the errors, removing the largest errors, see Table II.

Table II shows that most of the errors were much more reasonable with about a 7% improvement using the prediction model.

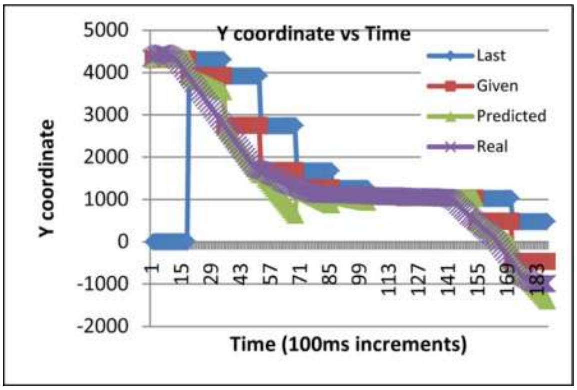

Fig. 10. Y coordinate of robot over time.

TABLE I
% Average Errors in Given and Predicted Positions

	Average Error in Given Position	Average Error in Predicted Position
X Axis error	33.7%	18.8%
Y Axis error	105.9%	37.2%

TABLE II
95% of Errors, (5% of Largest Errors Removed)

	Average Error for 95% in Given Position	Average Error for 95% in Predicted Position
X Axis error	17.8%	10.6%
Y Axis error	19.3%	10.8%

The larger errors were not being used long enough to destroy the advantages of the predicted model.

A more accurate prediction could be made if the last three or more given positions were kept and higher order extrapolations were made. The linear extrapolation however was a significant improvement over straight consensus control and higher order extrapolations would only be used if the time delays were much larger.

XIV. Simulated and Real

With the simulated task complete, the real VideoRays were introduced. Both the simulated and real VideoRays were to do the task, patrolling a path, together. The MicroNav was to track the real robots and feed their positions into the simulation.

The problem that cropped up here was that the MicroNav did not work in the pool environment. The walls of the pool caused too many reflections for the system to work.

Because of this, it was decided to use the simulator tracking system itself to track the real robots. The only sensors that the VideoRays had were a depth sensor and an electronic compass. The tracking system knows the power level that the real robot is using and its orientation from the on board compass. By using dead reckoning, it could estimate the robots position just as it tracked the simulated robots. This was not the ideal situation as it is an open control system. The robots do not know exactly where they are without the feedback of the MicroNav system. The first test run included the two real robots with the 10 simulated robots.

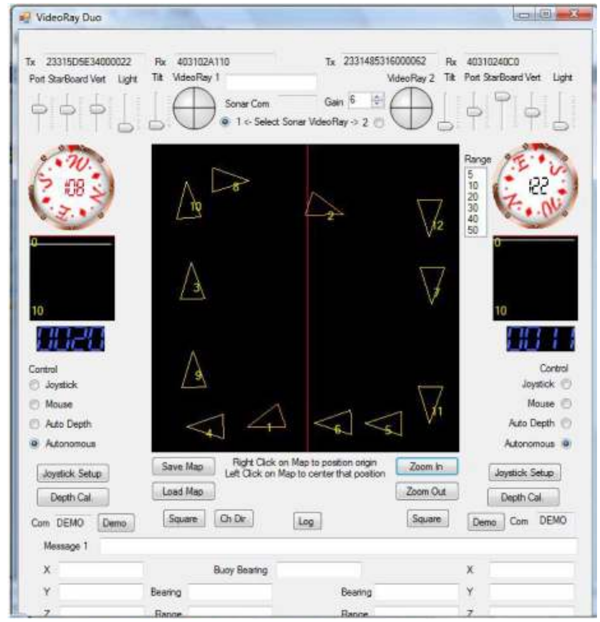

Fig. 11. Simulation run with the two real VideoRays.

The real robots are numbered 1 and 2. The side panels on each side of the simulation section Show the details of the real robots. VideoRay 1 is on the left. The items displayed from top to bottom for both robots are:

- Tx—The serial control string sent to the VideoRay;
- Rx—The serial string received from the VideoRay;
- power/thrust levels of the thrusters (a centred slider is zero thrust, up is forward/upward thrust and down is the reverse);
- a compass (a red bearing is negative.);
- a depth graph and readout (VideoRay 1 is at 2 m and the other at 1.1 m);
- a control selection—the robots are set to Autonomous control.

The lower section displays the information that the MicroNav is sending. It is blank as the MicroNav was not being used here.

The center top section allows mouse control via trackballs and sonar selection when a VideoRay is equipped with a sonar unit. Neither was equipped here.

To judge how well the control system worked the VideoRays were filmed and then broken down into a series of images in a similar fashion to time lapse photography. One image is shown in Fig. 12.

The resulting images were overlaid on each other to see motion of the Videorays. Over that the route, that each VideoRay took was traced, VideoRay 1 in black. Then the reference squares were traced in red.Fig. 13

The misalignment of the reference squares is due to a number of factors. The compasses on the VideoRays were out by about 8 degrees to each other. The camera was at about a 30–degree angle to the water and the VideoRays were at different depths. The deeper VideoRays route looks smaller and higher up on the image due to this.

Note however that the VideoRays end up where they started. This indicates that the simulator system was able to control the

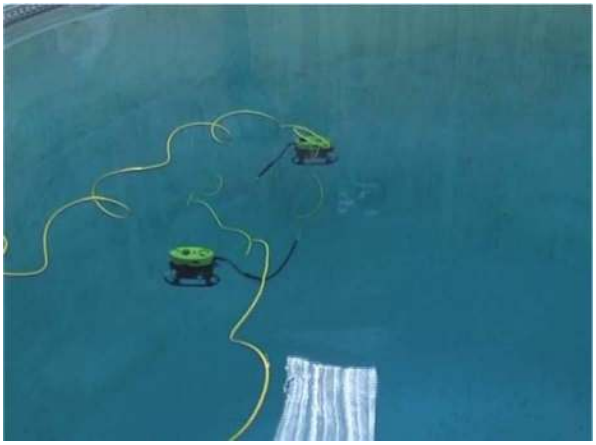

Fig. 12. Two VideoRay patrolling in a pool.

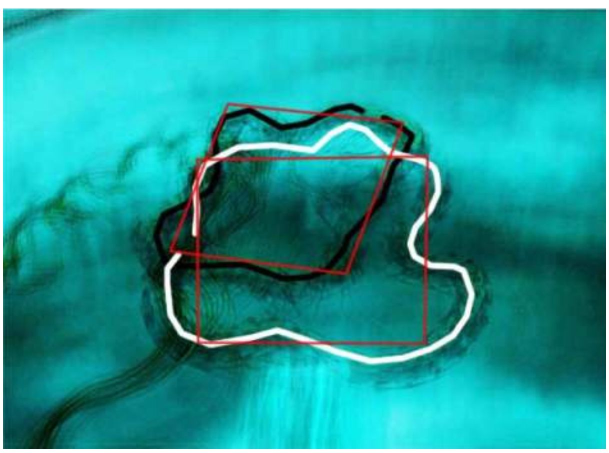

Fig. 13. Route taken by each VideoRay.

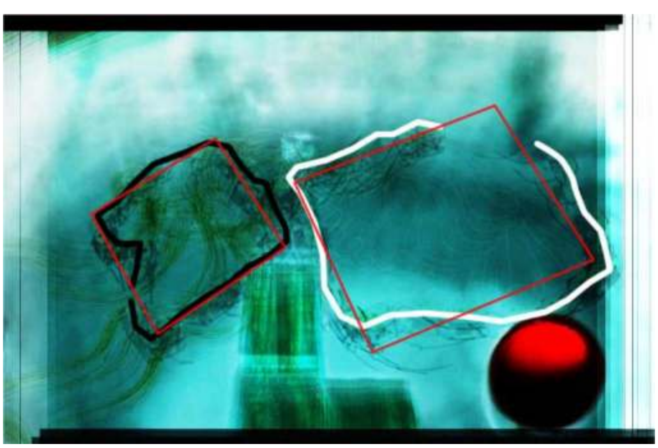

Fig. 14. Route of two VideoRays with bearing correction.

VideoRays well, even in an open system, and hence, that the simulator gives a good representation that of how the VideoRays move.

It was also noted that even though the paths closed, they were erratic. This was due to a lag in the compass reading of up to 2 s. To overcome this, the previous two compass readings were used to extrapolate a new reading closer to the true compass bearing. The results are seen in Fig. 14.

The VideoRays motion was still erratic in places but displayed a smoother route overall.

In this trail, the two VideoRays were not started in the same reference square as the author, manually controlling two VideoRays and watching the software, had more than a little difficulty getting both VideoRays to the right starting spot at the same time to begin the trial. This did not matter however as the control system was controlling a square patrol path and assumed that the real VideoRays were in the right square.

This image, Fig. 14, also shows how the handheld camera's motion was corrected for. The Red ball at the bottom left was a fix buoy and as each image was overlaid on the next one, the image was adjusted so that the buoy was in the one position.

Finally, looking at Fig. 11, it can be seen that the consensus control system effectively incorporated both the real VideoRays and the simulated ones ensuring that they all followed the patrol route and did not interfere with each other.

XV. Future Work

Now that real robots can be controlled along with simulated ones, new scenarios can be tried. For example, in a rescue situation, the simulated robots can be used to search and the real ones can affect the rescue. A system with real time delays can tested and previous theoretical work, such as [21] can be applied to it.

XVI. Conclusion

In an underwater robotic swarm environment, consensus allows a distributed control over the robots. The most important factor in achieving control is the communication rate which can be very slow as can the time between information updates. To counteract this, a prediction/estimation algorithm must be employed. A simple linear extrapolation will perform the estimation with sufficient accuracy to allow the consensus control to be effective.

Further, real robots can be incorporated with the simulation and will allow more algorithms to be tested with real systems.

References

[1] E. Franco, L. Magni, T. Parisini, M. M. Polycarpou, and D. M. Raimondo, "Cooperative constrained control of distributed agents with nonlinear dynamics and delayed information exchange: A stabilizing receding-horizon approach," *IEEE Trans. Autom. Control*, vol. 53, no. 1, pp. 324–338, Feb. 2008.

[2] Z. Qu, J. Wang, R. A. Hull, and J. Martin, "Cooperative control design and stability analysis for multi-agent systems with communication delays," in *Proc. 2006 IEEE Int. Conf. Robotics and Automation*, 2006, pp. 970–975.

[3] F. Stulp, M. Isik, and M. Beetz, "Implicit coordination in robotic teams using learned prediction models," in *Proc. 2006 IEEE Int. Conf. Robotics and Automation*, Orlando, FL, 2006.

[4] G. Ferrari-Trecate, L. Galbusera, M. P. E. Marciandi, and R. Scattolini, "Model predictive control schemes for consensus in multi-agent systems with single- and double-integrator dynamics," *IEEE Trans. Autom. Control*, vol. 54, no. 11, pp. 2560–2572, Nov. 2009.

[5] H. Jongho and H. J. Kim, "Nonlinear model predictive formation flight," *IEEE Trans. Syst., Man, Cybern. A: Syst., Humans*, vol. 39, no. 5, pp. 1116–1125, Sep. 2009.

[6] R. O. Abel, S. Dasgupta, and J. G. Kuhl, "One step ahead quadratic receding horizon distributed control of multiagent formations," in *Proc. IEEE Int. Conf. Control Applications*, 2007, pp. 569–574.

[7] J. Choi, J. Lee, and S. Oh, "Swarm intelligence for achieving the global maximum using spatio-temporal Gaussian processes," in *Proc. American Control Conf.*, 2008, pp. 135–140.

[8] , D. Richardson, Ed., *The Encyclopedia of Recreational Diving*. Santa Margarita, CA: PADI, 2008.
[9] R. Ballard*, Titanic: The Last Great Images*. Philadelphia, PA: Running Press, 2008.
[10] P.-P. B. E. An, B. Baud, T. Carlson, A. Folleco, and T. J. Tarn, "Multiple communicating autonomous underwater vehicles," in *IEEE International Confrefence of Robotics Automation*, 2004, pp. 4461–4465.
[11] L. C. Y. Zhang, "A distributed protocol for multi-hop underwater robot positioning," in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, Shenyang, , 2004, pp. 480–483.
[12] J. B. Curtin, J. Catipovic, and D. Webb, "Autonomous oceanography sampling network," *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.
[13] T. Shaneyfelt, M. Joordens, K. Nagothu, J. Prevost, A. Kumar, M. Ghazi, and M. Jamshidi, "Control and simulation of robotic swarms in heterogeneous environments," in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics*, 2008, pp. 1314–1319.
[14] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.
[15] W. Ren, R. W. Beard, and E. M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. American Control Conf.*, Portland, OR, 2005, pp. 1859–1864.
[16] T. Namerikawa and C. Yoshioka, "Consensus control of observer-based multi-agent system with communications delay," in *Proc. SICE Annu. Conf.*, Tokyo, Japan, 2008.
[17] W. Ren and R. W. Beard*, Distributed Consensus in Multi-Vehicle Coorerative Control*. London, U.K.: Springer-Verlag, 2008.
[18] R. O. Saber and R. M. Murray, "Consensus protocols for networks of dynamic agents," in *Proc. 2003 American Control Conference*, Denver, CO, 2003.
[19] M. Joordens, "Design of a low cost underwater robotic research platform," in *Proc. IEEE SoSE 2008 Conf.*, Monterey, CA, 2008.
[20] K. Nagothu, M. Joordens, and M. Jamshidi, "Communications for underwater robotics research platforms," in *Proc. IEEE Int. Conf. System of Systems Engineering*, Monterey Bay, CA, 2008, pp. 1–6.
[21] A. Speranzon, "Coordination, Consensus and Communication in Mult-Robot Control Systems," Ph.D. dissertation, Royal Inst. Technol., Stockholm, Sweden, 2006.
[22] M. B. Shuaib, F. Sallabi, and A. Lakas, "Co-existence of Zigbee and WLAN, a performance study," in *Proc. IEEE Int. Conf. Wireless and Optical Communications Networks*, Pomana, CA, 2006, pp. 1–6.
[23] Y.-C. H. J.-S. Lee, "Itri zb node: A zigbee/ieee 802.15.4 platform for wireless sensor networks," in *Proc. IEEE Int. Conf. System, Man, and Cybernics*, 2006, pp. 1462–1467.
[24] Z. Alliance*, Zigbee Specifications Version 1.0*. San Ramon, CA: Zigbee Alliance, 2006.
[25] *Digi*. : , 2008.
[26] W. Yang, A. L. Bertozzi, and X. Wang, "Stability of a second order consensus algorithm with time delay," in *Proc. IEEE Conf. Decision and Control*, Cancuun, Mexico, 2008, pp. 2926–2931.
[27] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
[28] D. P. Spanos, R. Olfati-Saber, and R. M. Murray, "Dynamic consensus on mobile networks," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, 2005.
[29] Li X.-S and H.-J. Fang, "Optimal weights for consensus of networked multi-agent systems," *Inform. Technol. J.*, vol. 8, no. 1, pp. 77.82–77.82, 2009.
[30] J. Lavaei, A. Momeni, and A. G. A. Aghdam, "A model predictive decentralized control scheme with reduced communication requirement for spacecraft formation," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 2, pp. 268–278, Mar. 2008.

Matthew A. Joordens (M'07) received the B.Eng. degree from Ballarat University, Ballarat, Australia, in 1988 and the M.Eng. degree in virtual reality from Deakin University, Melbourne, Australia, in 1996.

He began his career in industrial control technology, designing control systems to automate various different industrial processes. For five years, he designed microprocessor based control systems for companies such as Ford, Pilkington Glass, Webtek, and Blue Circle Southern Cement. He then moved to Deakin University and wrote their first electronics units. Using his industrial experience, he designed one of the first Australian engineering degrees in mechatronics that still runs at Deakin—Mechatronics and Robotics. He has been with Deakin University for 17 years, where he is currently a Lectures in digital electronics, microcontrollers, robotics and artificial intelligence. He is presently researching underwater swarm robotics in the U.S. and Australia.

Mr. Joordens is a Fellow of the Institution of Engineers, Australia.

Mo M. Jamshidi (S'66–M'71–SM'74–F'89) received the B.S. degree from Oregon State University, Corvallis, in June 1967 and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign in June 1969 and February 1971, respectively. He holds three honorary doctorate degrees from Azerbaijan National University, Baku, Azerbaijan, the University of Waterloo, Waterloo, ON, Canada and the Technical University of Crete, Crete, Greece.

He has been an Advisor, an IPA, or special government employee with NASA (ten years), U.S. AFRL (nine years), and DOE (six years) since 1984. In 1999, he was a NATO Distinguished Professor in Portugal in the area of intelligent systems and control. He was the Founding Director of the Center for Autonomous Control Engineering (ACE) at the University of New Mexico (UNM), and moved the Center to the University of Texas in early 2006. In the summer of 2009, he was a Distinguished Fellow of the U.K. Royal Academy of Engineering at Cardiff University, Wales, U.K. Currently, he is the Lutcher Brown Endowed Chaired Professor at the University of Texas System at San Antonio. He has been the Director of the International Consortium on System of Systems Engineering (ICSoS) since 2006 and is Regents Professor Emeritus of Electrical and Computer Engineering at UNM and an Honorary Professor at Deakin University, Melbourne, Australia (2009–2012). He has over 600 technical publications, including 63 books and edited volumes. Six of his books have been translated into at least one foreign language.

Dr. Jamshidi is the Founding Editor/Co-Founding Editor or Editor-in-Chief (EIC) of five journals. He is EIC of the new IEEE SYSTEMS JOURNAL (inaugurated in 2007) and Founding EIC of the *IEEE Control Systems Magazine*. He is currently on the Board of Governors of the IEEE Society on Systems, Man, and Cybernetics and the IEEE Systems Council. He is a member of the Russian Academy of Nonlinear Sciences and Foreign Member of the Hungarian Academy of Engineering. He is a recipient of the IEEE Centennial Medal and IEEE Circuits and Systems Distinguished Member Award. In 2005, he received the IEEE Systems, Man, and Cybernetics (SMC) Society's Norbert Weiner Research Achievement Award, and in 2006, received the IEEE SMC Distinguished Contribution Award and a Distinguished Alumni in Engineering Award from Oregon State University. He is a Fellow of ASME, AAAS, NYAS, TWAS and an Associate Fellow of AIAA.