failure is to repeat the task a finite number of times.

The second kind of failure is when sensors yield conflicting data. One example might be the case of the magnetic contact switches in the dock which might be telling us that the dock is both open and closed at the same time. In this case a decision tree is followed to resolve such conflicts in the best a priori manner possible.

There are other specific failures that we considered. A typical deep water deployment for instance, the Labrador Sea deployment being an extreme case in point, the most vulnerable part of the mooring is the surface expression due to wind, sea state, etc. This issue is addressed at two levels, mechanically the tether from the surface expression to the subsurface buoy is designed to be the weakest part of the system and from an algorithmic viewpoint the dock was designed to continue functioning in the absence of the surface expression. The major tasks that the surface expression serves including - providing GPS time, providing sensor data and remote access to human operators are all non critical for autonomous operations. Thus if the surface expression were to cease functioning the dock and the vehicle would be synchronising their clocks locally based on the time being read by the dock and the mission schedule could only be triggered by two of the normal mechanisms discussed earlier.

One possible failure scenario is the possibility of a missing vehicle when the dock wakes up. Should the dock wake up and discover that the vehicle is not in the dock and not on a mission, it has the ability to send an abort sequence via the acoustic modem to the vehicle. If the vehicle were to receive an abort sequence its first action would be to unlatch (in case it was latched onto some other part of the dock or mooring) and then to try and home to the dock. After sending the abort sequence the dock also turns on the homing beacon, a task usually carried out out only by the vehicle, to bypass the possiblity that the AUV was not able to turn on the homing beacon.

Another possible scenario is that the acoustic modem, the primary communication link between the dock and the vehicle were to fail to send data back to the dock. The major impact in this case would be that the dock would have no way of knowing when the vehicle had returned from its mission. To guard against this eventuality the dock implicitly waits for a user specified interval past the time that the vehicle was expected back before forcing the alignment carriage to be be run up the pole. Thus if a vehicle was present the magnetic switches that are used to check for alignment of cores would determine the presence or absence of a vehicle.

Finally, in the case of an irrecoverable lockup on the dock, a mechanism exists for sending a reset sequence to the dock computer that would cause it to reboot. By manipulating the files on the dock we can also replace source and configuration files on the dock as well as recompile code remotely

VI. Conclusions

In this paper we have described the implementation of a complete docking methodology for an auotnomous ocean sampling network. In a future publication we hope to report on our ongoing efforts in using this methodology for extended deployments of this system.

Acknowledgements

We would like to thank members of the MIT Sea Grant Underwater Vehicle Laboratory; members of the WHOI Buoy Group, and members of the WHOI DSL docking group, the WHOI Acoustic Telemetry Group for their help in different aspects of the work described in this paper.

References

1. Curtin, T., Bellingham, J., Catopovic, J., Webb, D. "Autonomous Oceanographic Sampling Networks", Oceanography, vol. 6, no. 3, pp 86-94, 1993.

2. Bellingham, J.B., Goudey, C.A., Consi, T.R., et al. "A Second Generation Survey AUV", *Proceedings of the 1994 IEEE Conference on AUV Technology,* Cambridge, MA, 1994.

3. Singh, H., Bowen, M., Hover, F., LeBas, P., Yerger, D. “Intelligent Docking for an Autonomous Ocean Sampling Network”, *Oceans 97 MTS/IEEE Conference,* Halifax, Canada, 1997

4. Singh, H., Catipovic, J., Eastwood, R., Freitag, L., Henriksen, H., Hover, F., Yoerger, D. "An Integrated Approach to Multiple AUV Communications, Navigation, and Docking", Proceedings of the Oceans ’96 MTS/IEEE Conference, pp 59-64, 1996.

the time had the dock and vehicle idle. Trading off this idle time with a need to check the health of the dock as well as log the data from the sensors on board the dock we settled on a cycle such that the entire system would come alive once every two hours before shutting down, if a mission was not scheduled for the next time period. The typical time to boot up, make contact with the subsystems and transfer data between them was of the order of two minutes. With the entire suite of sensors running on the dock it the dock uses 35W of energy. However in the sleep state it uses only 100mW.

III. Mission Schedule

There are currently three mechanisms for scheduling a mission. It may be scheduled a priori, downloadable from operators on shore, or it may be triggered based on sensor input on the dock. The a priori mission schedule is an ascii file that resides on both the dock station and the vehicle and contains the time to start a mission, the name of the mission and its duration. This allows the dock to prepare for launching the vehicle at the right time as well as lets it know when to expect the return of the vehicle. Additionally this is a fall back mechanism should communication between remote users and the dock fail for any reason. The downloadable mission schedule is identical to the a priori mission schedule except that the ascii mission file has been received from remote operators via satellite and sent to the vehicle via the dock. A program on the dock also monitors sensors located on the dock and may make a decision to start a vehicle mission based on its input. This protocol has a mechanism built in so that it may be armed or disarmed by users on shore.

IV. Dock, AUV and Remote Interactions

Various hardware mechanisms exist for remote users to interact with the dock and the AUV. All interactions take place along a chain from the user via satellite(Inmarsat C or Argos) or local RF link to the surface expression and from there via a four wire redundant RS-485 interface to the dock and eventually via inductive coupling between the dock and the AUV. There also exists acoustic communication links between the dock and the vehicle.

Every two hours the vehicle wakes itself up a few minutes before the hour and goes through a self check, the results of which it stores locally. On the hour the dock wakes up and communicates to the vehicle via ftp. It goes to the local data directory on the vehicle and checks for the existence of a specific file which it obtains. contained in this file is a running list of mission data files to be transferred, at which point the dock initiates the FTP session to transfer the data files, estimates how long the transfer will take and put a lock file on the vehicle. After the transfer is completed the lock file is removed. The lock file also serves the purpose of synchronizing the time between the AUV and the dock as it contains information about when it was created on the dock and a process on the vehicle is constantly checking for its existence locally and thus has a local time stamp associated with when it appeared on the vehicle. The data from the AUV consists of archival data that is logged on the dock, highly compressed data that is transferred back to remote users via InmarsatC, as well as vehicle status information that the dock concatenates with its own status information into a bit stream suitable for Argos transmission.

As part of its wakeup cycle the dock also communicates with the surface expression. It queries the surface expression for sensor data, for time because the surface is set to GPS time, as well as for any incoming messages that the surface may have received via its inmarsat link. The incoming messages may be of several types. They may consist of mission files which the dock keeps a local copy before passing onto the vehicle, they may be specific files destined for the dock, or they may be files that are specific to the vehicle. The dock transmits a very compressed data packet for transmission over the argos link as well as short snapshots of status and data for transmission over the Inmarsat C link. An acoustic communications link also exists between the dock and the AUV. This serves to keep the dock informed about vehicle position when the AUV is on a mission as well as lets the dock know when the AUV thinks it is latched onto the docking pole.

V. Reliability

One of the biggest problems with the development and use of such a large complex system, is the overall reliability of the individual subsystems. This is further compounded by the need in this case for totally autonomous operations. We now discuss some of the most likely failure mechanisms and how we deal with them.

There are two basic kinds of failures that we need to consider. The first is when a task fails to complete. An example of such a task might be when the motor driving the carriage on the dock fails to successfully align the inductive cores on the vehicle and the dock. The normal mechanism for dealing with such a

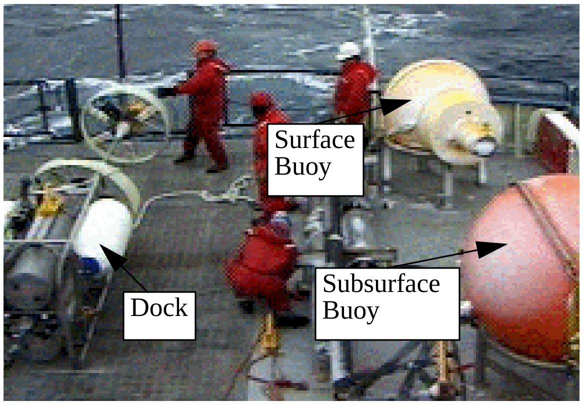

Figure 2. Docking System Components

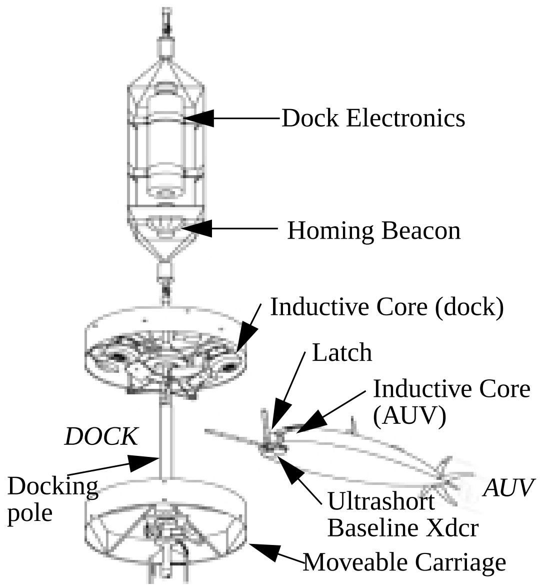

Figure 3. The Docking System

II. Algorithmic Structure

The basic algorithmic structure is described in the flowchart given in Figure 4. After an AUV latches onto the dock, the dock detects the presence of the AUV, runs its actuator to align the inductive core on the AUV with the inductive core on the dock before beginning a transfer and archive of the AUV's data. After transmitting the news of the successful return of the AUV to the dock as well as a suitably compressed representative set of the scientific data to remote users via a satellite link on the surface, the dock begins the longer process of recharging the AUV's batteries. Once the vehicle is ready, the dock waits until it is time to send the vehicle on its next mission. While the vehicle is on its mission the dock tries to track the vehicle until it safely returns and this cycle continues.

While this flow chart does at a simplistic level capture the essence of all the tasks that are required of the dock it does not address a number of issues which impact the actual implementation. These issues include the power budget associated with a deployment that lasts several months, the reliability of the entire system in the event of subsytem failure, the protocols for dock, AUV interactions and the precise nature of remote access for users onshore. In the succeeding sections we look at each of these issues in turn, and how they impacted our design.

The actual implementation of the flow chart was carried out by means of a finite state machine model that considered the dock to be in one of four states. The four states are associated with powering on the dock, the AUV being deemed ready for a mission, the AUV out on a mission and the AUV back on the dock being serviced with respect to data transfer and or battery charging. This finite state machine approach allows us to check consistency over time as well as raises flags if we transition between two unacceptable states.

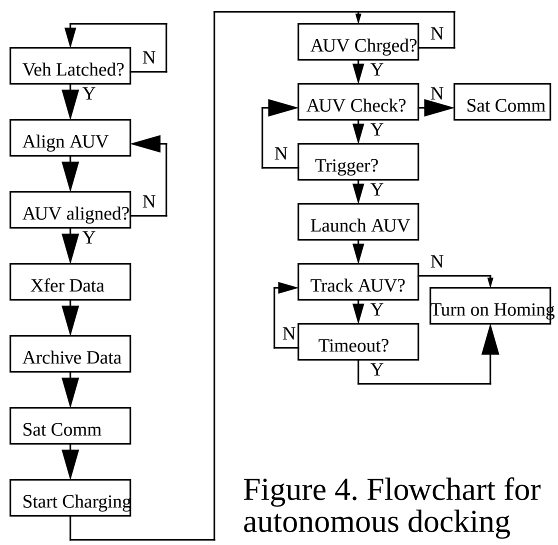

Figure 4. Flowchart for autonomous docking

Power for the dock was supplied by a collection of ~5000 alkaline D-cells. These were arranged in two housings connected to the dock controller. The computing engine on the dock is an i486 based PC-104 system. If we consider typical mission scenarios such as the one outlined for the Labrador Sea the requirements were for running vehicles for six twelve-hour-long missions. Considering that the overall deployment was to last for a period of over four months it becomes obvious that the majority of

An Intelligent Dock for an Autonomous Ocean Sampling Network

Hanumant Singh, Steve Lerner, Keith von der Heyt

Department of Applied Ocean Physics and Engineering
Woods Hole Oceanographic Institution
Woods Hole, MA 02543

Bradley A. Moran

Sea Grant Underwater Vehicles Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

In this paper we examine the issues associated with the design, at an algorithmic level, of a dock capable of autonomous or near autonomous operations in tending to and serving as a data and power repository for Autonomous Underwater Vehicles. The emphasis in this application is on long term deployments in the deep ocean at remote sites, as part of an Autonomous Ocean Sampling Network. The system consists of a full ocean mooring with a surface expression for satellite and other communications which is connected to the dock located at depth. The docking system itself is an amalgamation of electronic and mechanical systems designed to allow the vehicle to approach from any direction while serving the needs to transfer data from the AUV and to charge its batteries. On the algorithmic level, we utilize a finite state machine which based on input from its sensors ensures that the AUV is securely docked; decides and reacts to a vehicle that requires charging; tracks it when it is out on a mission; grabs the data that the AUV collects during its mission; while providing the mechanism for interacting with a person far removed from the remote site to learn about vehicle status and command high level missions.

I. Introduction

One concept of an Autonomous Ocean Sampling Network (AOSN)[1] envisions a number of autonomous underwater vehicles deployed at a remote site on oceanographic moorings. The mooring needs to autonomously serve the needs of the AUV and remote users as well as have enough intelligence to continue functioning on it own volition. Specifically, the mooring serves as a repository for vehicle data, as a charging station for in our case the MIT Odyssey IIB class AUV[2], as well as providing access to users on shore through a satellite link based in the surface expression of the mooring. Our basic mechanism for homing into the dock based on an acoustic beacon has been described elsewhere[4][5]. The dock consists of a pole to which the vehicle attaches via a latch in the vehicle's nose as illustrated in figure 3. In this paper in particular, rather than the mechanical design, we focus on the algorithmic design of the software and the issues that dictate our choice for autonomously accomplishing the docking tasks described above.

An example configuration of the entire mechanical system, in this case for a deployment in Labrador Sea, is shown in Figure 1 with some of the major components on the deck of the R/V Knorr before being deployed in the Labrador Sea shown in Figure 2.

This work was supported by a sub-contract from the MIT Sea Grant program through contract# N00014-95-1-1316.

.

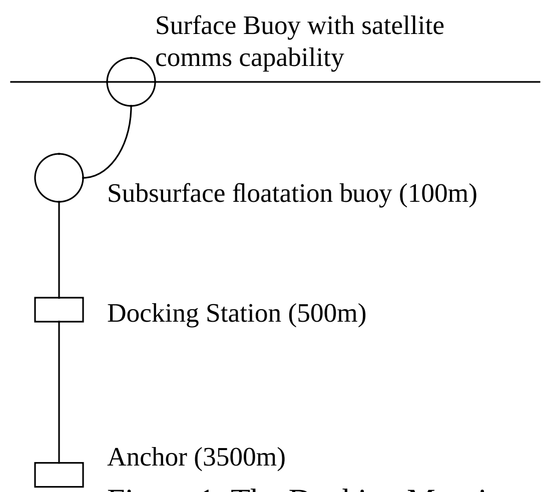

Figure 1. The Docking Mooring