Shallow Water Stationkeeping of an Autonomous Underwater Vehicle: The Experimental Results of a Disturbance Compensation Controller

J.S. Riedel
Center for AUV Research
Naval Postgraduate School
Monterey, CA 93943
jsriedel@me.nps.navy.mil

ABSTRACT

The continual development of computer technology has enabled the expansion of intelligent control into the field of underwater robots, where potential uses include oceanographic research, environmental monitoring and military mine countermeasures. With the naval focus shifting to operations in the littorals, and the need to lower cost of operations, tetherless autonomous vehicles are now being proposed for use in very shallow water minefield reconnaissance. These areas are dominated by a highly energetic environment arising from waves and currents. Motion control in such an environment becomes a difficult task and is the subject of this work.

The main objective of this paper is to show that intervention tasks performed by intelligent underwater robots are improved by their ability to gather, learn and use information about their working environment. Using a new generalized approach to the modeling of underwater vehicles, which directly includes disturbance effects, a new Disturbance Compensation Controller (DCC) is proposed. The DCC, employing onboard vehicle sensors, allows the robot to learn and estimate the seaway dynamics. This self-derived knowledge is embedded in a non-linear sliding mode control law which allows significantly improved motion stabilization. The performance of the DCC has been experimentally verified in Monterey Harbor using the NPS Phoenix AUV.

I. INTRODUCTION

This paper will discuss the development and employment of a real-time disturbance compensation controller (DCC) which will allow an AUV to dynamically position itself in the presence of waves. The paper will begin with an overview of the DCC, followed by a discussion of an asynchronous Extended Kalman Filter for state and disturbance estimation. This nonlinear estimator is critical to the DCC performance since the Sliding Mode Controller (SMC) requires full state feedback, and not all states are measurable. In addition, the EKF provides the controller with a smoothed estimate of the unmeasured fluid particle velocity which is used to compensate for the wave induced disturbance.

Next, through the design and implementation of an asynchronous simulator, which realistically models the vehicle dynamics, the sensors including noise and the sensor processes, the DCC is tuned and the achievable performance is demonstrated. Lastly, it is shown experimentally, that the DCC is stable and allows the NPS Phoenix AUV to hold position in the surge direction, while subjected to ocean waves in Monterey Bay.

A. DCC Overview

The design of the disturbance compensation controller can be looked at as an optimization problem since there are competing goals. First, since the design requirement is to minimize position error in the presence of disturbances, a high gain control is desirable. Using high gain control, the system becomes sensitive to measurement noise and uncertainty, thereby causing the gain to be reduced to maintain stability.

An estimator is needed to provide the unmeasurable states to the controller, and to filter the sensor noise thereby improving the systems performance. Here, the requirement is to accurately track the signal, again requiring a high filter gain, while smoothing the noise, (a low gain). As with the controller, trade-offs must be made.

The over all goal is to develop a combined controller/estimator which, when implemented, will enable the vehicle to maintain position while using noisy sensor information. The output of this system, for implementation, is a commanded voltage that is sent from the DCC process to the real-time execution computer, without excessive lags to ensure stability. A mathematical description to the above problem is given below, with a block diagram of the DCC in provided in Figure 1.

$$\begin{aligned}
&\textit{State:} && x^T = [X, u_r, F]; \quad d = u_f \\
&\textit{System:} && \dot{x} = f(x,n,d); \quad y^T = [x, u_r, u_g,] = Cx + Dd \\
&\textit{Disturbance:} && \dot{x}_f = Ax_f + v \quad u_f = Cx_f \\
&\textit{Control law:} && n = smc(\hat{x}, \hat{u}_f, x_{com}) \\
&\textit{Estimator:} && [\hat{x}, \hat{u}_f] = EKF(f(x,n,d), A, y, n)
\end{aligned} \tag{1}$$

B. State And Disturbance Estimation

There are many methods available to estimate states and disturbances in practice today. A few of these include the Luenberger Observer [1] and the Kalman Filter [2] for linear systems, and the Sliding Mode Observer [3], the "Rajamani" Observer [4] and the Extended Kalman Filter [5] for nonlinear systems. Each method has both pros and cons depending on the application. For this work, an Extended Kalman Filter was chosen since a relatively accurate vehicle

model is available, and since the disturbance is stochastic in nature.

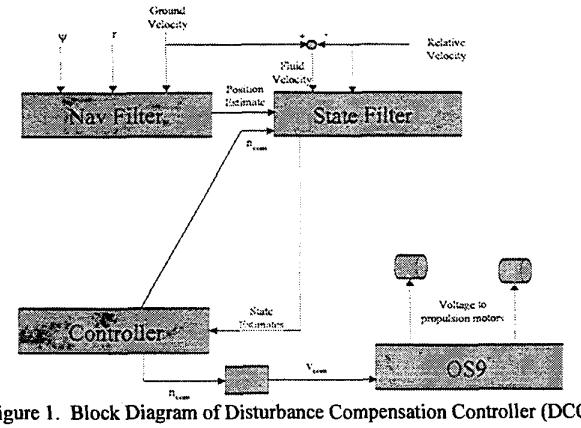

Figure 1. Block Diagram of Disturbance Compensation Controller (DCC)

Kalman filtering is the process of recursively updating an estimate of systems states based upon measurements corrupted by noise. The system state is a collection of variables that describe the dynamics of a system, and in this case they are position, relative velocity and propeller thrust, of which only relative velocity is measurable.

System state are updated with knowledge of system dynamics (vehicle model), measurement dynamics (measurement model), system noise (modeling uncertainty) and measurement noise (measurement errors). The system model is not perfect in describing the dynamics of the vehicle and will contain a certain amount of uncertainty, called system noise. There is also some uncertainty associated with each measurement taken. This uncertainty can be composed of both random white noise and a bias. Measurements which cannot be directly obtained, such as fluid velocity, are related to measurements which are directly obtainable, such as relative velocity and ground velocity, in the measurement model. Recursively updating means the Kalman filter does not need to keep record of all past measurements, only the most recent ones.

II. MODEL AND FILTER DEVELOPMENT

Using a three state surge model [6] and a four state AR model for the wave dynamics [7], an augment state and disturbance model was formed, and used as the basis of an EKF. This model allows the disturbance to be treated as an additional state, where the vehicle states and disturbance estimates are filter outputs. The augmented vehicle and disturbance model is given by,

$$
\begin{aligned}
\dot{x} &= u_r + u_f \\
\dot{u}_r &= \alpha u_r |u_r| + F \\
\dot{F} &= \frac{-1}{\tau} F + \frac{\gamma}{\tau} u_r |n| + \frac{\beta}{\tau} n|n| \\
\dot{x}_{w,1} &= x_{w,2} = u_f \\
\dot{x}_{w,2} &= \dot{u}_f = x_{w,3} \\
\dot{x}_{w,3} &= a_1 x_{w,1} + a_2 u_f + a_3 x_{w,3} + a_4 x_{w,4} \\
\dot{x}_{w,4} &= v \\
y &= [x, u_r, u_g]^T
\end{aligned}
\qquad , (2)
$$

where the AR coefficients are found using the an online adaptive wave model.

A. Kalman Filter Algorithm

Using standard design techniques [2], the filter was developed and implemented using the following algorithm. First, the system model matrix $\boldsymbol{A}$, system noise matrix $\boldsymbol{Q}$, measurement matrix $\boldsymbol{C}$, measurement noise matrix $\boldsymbol{R}$, and the error covariance matrix $\boldsymbol{P}$ are initialized to appropriate values. The error covariance matrix is a can be thought of as a level of uncertainty in the state vector. Then the state vector, error covariance and measurement vector are propagated one time step using the model.

When the new measurement is received, and innovation error is calculated based on the difference between the measured values and the estimated values. Using the propagated error covariance, measurement noise matrix and measurement matrix, a gain is determined for the state vector and error covariance update. This process of propagating and updating is repeated through out the length of the vehicle mission. This recursive algorithm, in discrete form is given by,

$$
\begin{aligned}
\Phi_{k/k-1} &= \left. \frac{\partial f(x_{aug}, n)}{\partial x_{aug}} \right|_{\hat{x}_{k-1/k-1}} \\
\hat{x}_{k/k-1} &= \Phi_{k/k-1} \hat{x}_{k-1/k-1} \\
P_{k/k-1} &= \Phi_{k/k-1} P_{k-1/k-1} \Phi^T_{k/k-1} + Q \\
G_k &= P_{k/k-1} h_k^T [h_k P_{k/k-1} h_k^T + R]^{-1} \\
\hat{x}_{k/k} &= \hat{x}_{k/k-1} + G_k [y_k - h_k \hat{x}_{k/k-1}] \\
P_{k/k} &= [I - G_k h_k] P_{k/k-1}
\end{aligned}
\qquad , (3)
$$

where Φ represents the system dynamics, and $\boldsymbol{h}$=$\boldsymbol{C}$ since the measurements are linear in the state. The continuous linearized matrices for this particular design are given as,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2c\hat{u}_r sign(\hat{u}_r) & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{\beta}{\tau}|n_{com}| & \frac{-1}{\tau} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & -4 & -6 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4)$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

1. Asynchronous Data Processing

In the preceding discussion, the data contained in the measurements was assumed to be received at the same time with equal intervals through out the mission. In reality, all measurements are not received at the same rate, therefore, the EKF design must allow for this asynchronous sampling rate. In the Phoenix AUV, the vehicle control loop runs at 8 Hz, while the RDI DVL runs at 2 Hz, and the SonTek ADV at 6 Hz. See [7] for more details on these sensors. The main data acquisition process samples the sensor processes at the same frequency as the control loop, however, if the sensor has not yet updated, the data acquisition process records the value of the previous time step. The filter allows for the varying measurement rates by using a dynamic switching of the measurement matrix, C. The measurement matrix basically uses a zero-order hold on the measurement channel that has not been updated, and propagates the state using the previous measurement.

III. TUNNING OF THE DCC

Using the filter design from the previous section, and the sliding mode controller described in [6] an asynchronous simulator was developed for design validation. The simulator contains the non-linear vehicle dynamics, asynchronous sensor models with measurement noise, seaway dynamics and the DCC. Using this simulator as a design tool the DCC was adjusted to achieve an optimum design. The gains in both the controller and filter were adjusted so that performance requirements were met.

The stability performance of the estimator is shown through simulation, see Figure 2, since there are no formal proofs to determine the stability of combined nonlinear estimators and controllers. As seen, the error covariance levels all converge indicating a stable nonlinear filter design. Some of the covariance levels may appear to be "too high" giving the feeling that the filter is not properly designed, however, design decisions must be made to ensure that the filter lags are no too excessive, and that the estimator tracks well.

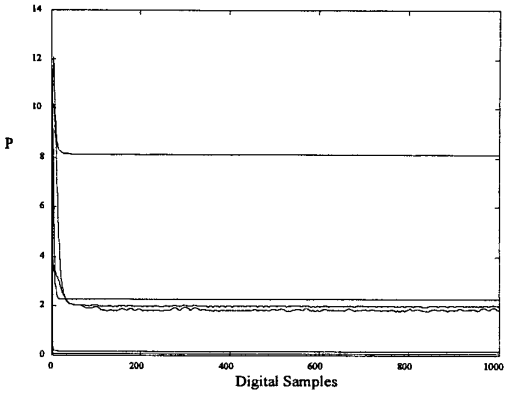

Figure 2. DCC Error Covariance Evolution

IV. INITIAL IN-WATER TESTING

Using short missions, the DCC was adjusted to achieve acceptable performance. These runs were performed on March 25, 1999, in Monterey Harbor. Of concern, was the amount of noise that was resident on the ADV sensor. This noise was far beyond the level which the vendor advertised. Using the design results from the simulations, the DCC was implemented in the Phoenix AUV. Figures 3-5 display initial results. As seen, the filter tracks the signals extremely well, including the noise.

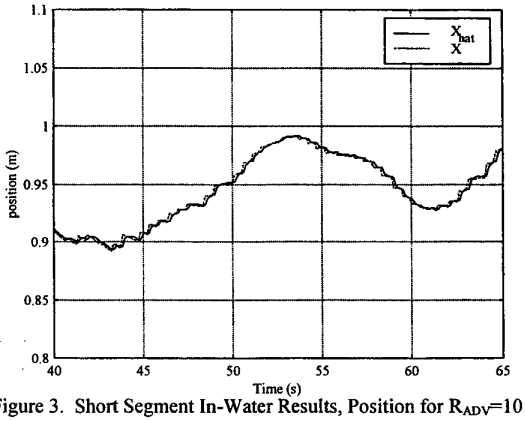

Figure 3. Short Segment In-Water Results, Position for R_{ADV}=10

This tracking of the noise has significant detrimental effects to the propulsion system as seen in Figure 5. The noise had been transmitted into the controller resulting in severe oscillation in the propeller response. These oscillations eventually lead to mechanical failure of the propulsion system shafting due to the shearing of connecting pins.

Using the information obtain during this set of runs allowed the filter gains to be adjusted to eliminate the transmission of sensor noise into the controller. Using linear design techniques, the combined controller filter transfer function from ADV input to propeller output was formed.

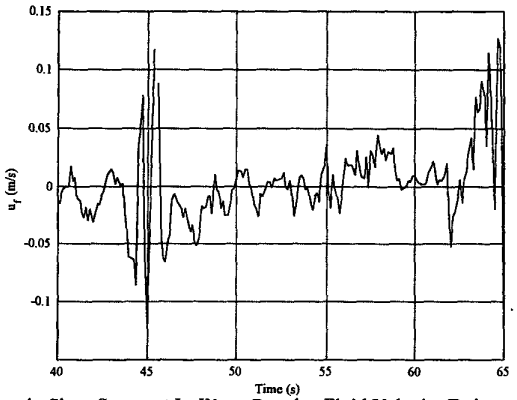

Figure 4. Short Segment In-Water Results, Fluid Velocity Estimate for R_{ADV}=10

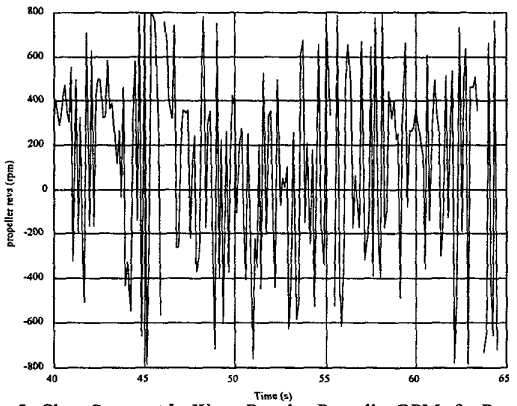

Figure 5. Short Segment In-Water Results, Propeller RPMs for R_{ADV}=10

By adjusting the level of the measurement noise parameters, attenuation of the noise into the control system was accomplished, and the response bandwidth of the controller was increased. This improvement in frequency response will reduce the propeller oscillations, thereby minimizing the chance of mechanical failure of the propulsion system. Using the new design values, the DCC was again tested in Monterey Harbor. The results of this testing are shown in Figures 6-8.

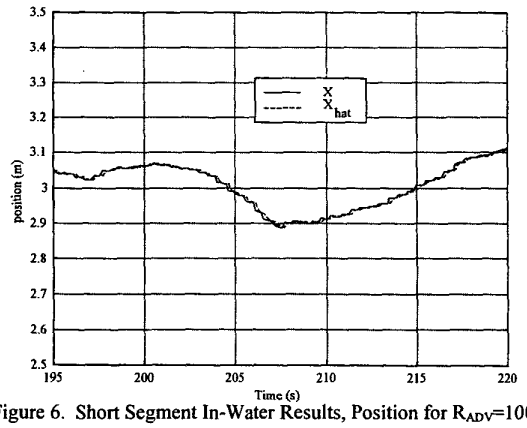

Figure 6. Short Segment In-Water Results, Position for R_{ADV}=100

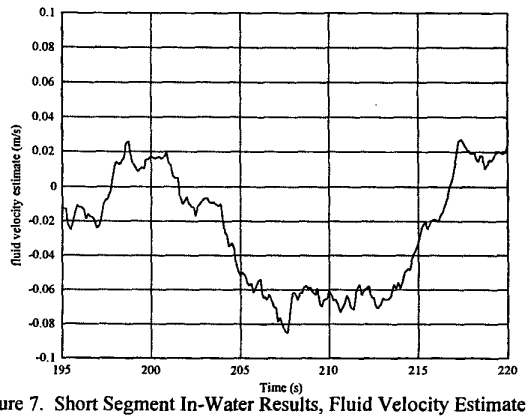

Figure 7. Short Segment In-Water Results, Fluid Velocity Estimate for R_{ADV}=100

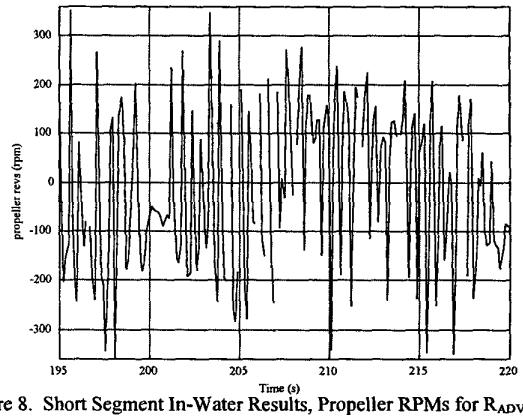

Figure 8. Short Segment In-Water Results, Propeller RPMs for R_{ADV}=100

As the result of the tuning of the DCC, the performance has improved dramatically. As before, the DCC maintains position extremely well, with a much reduced propeller response. Comparing the magnitude of the estimated fluid velocities between the two designs, Figures 4 and 7, it can be seen that for the same magnitude of input disturbance, position response has remained unchanged, but propeller response has reduced increasing the life of the propulsion system.

V. SOFTWARE IMPLEMENTATION

The implementation of this control process is unique since it is split between the two CPUs installed in Phoenix. The NPS AUV uses a Pentium based PC-104 running QNX and a GESPAC Card Cage running OS9 for mission control and execution. The DCC requires information from both processors, connected by Ethernet sockets, to compute and pass the commanded propeller RPMs to the execution level.

The control architecture presently running in Phoenix is based on shared memory processes. The PC-104 computer runs a "main" process that controls the synchronization of the data sharing, while the GESPAC clock controls the real-time control features. The two-processors use the shared memory as the common data buffer, controlled by semaphores to ensure the information transfer is consistent

with the clock speed. A graphical representation of this description is shown in Figure 9. As seen in the graphic, for the DCC implementation, all needed process are run in the PC-104 with the only purpose for the GESPAC is to send the commanded voltages to the propulsion motors.

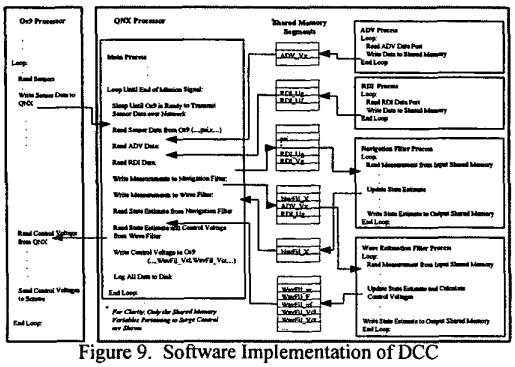

Figure 9. Software Implementation of DCC

A block diagram of the DCC implementation in the Phoenix AUV was given in Figure 1. This diagram represents the melding of the software and the hardware in the vehicle. The ground velocity is from the RDI, the relative velocity from the ADV and ψ, r from the directional gyro.

VI. EXPERIMENTAL VALIDATION OF THE DCC

The DCC was tested in Monterey Harbor between the months of March and May 1999. During this time, the Phoenix was held under surge control for over 90 minutes, during various runs, without a drive off. Table 1 provides a sample of the runs conducted during the validation of the controller.

Defining a measure of performance, the disturbance rejection ratio (*DRR*), as the ratio of standard deviation of the vehicle ground velocity to the standard deviation of the fluid velocity, the ability of an AUV to reject disturbances for different conditions and control designs can be compared. Referring to the DRR definition, for perfect disturbance cancellation the DRR will be equal to zero, while for designs where the control input excites the vehicle, [7], the DRR will be greater than one. For each operating point, the standard deviation of the propeller response is normalized by the maximum propeller revolutions, n_{max}.

Table 1 indicates that excellent disturbance rejection was achieved, even for the short runs where only limited statistical information was recorded. The tests showed that even when the vehicle was disturbed by a source other than the fluid velocity, it was able to return to the commanded position in a stable fashion.

A series a plots, Figures 10-13, show the results of one of the validation runs. This run was conducted on April 22, 1999 in Monterey Harbor. The Phoenix was commanded to a navigational position of 0 meters in the longitudinal direction. As the results indicate, the vehicle behaved as expected. The standard deviation of the positional error was 9.6 cm with ground velocity standard deviation of 1.5 cm/s.

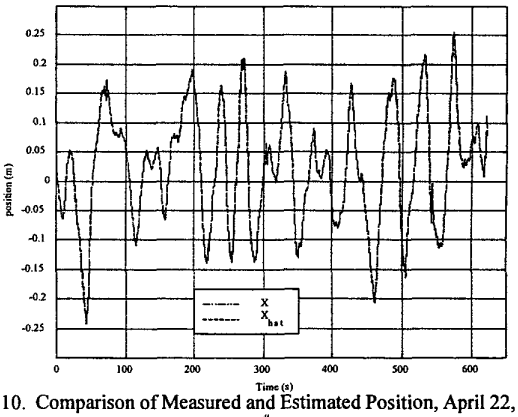

Figure 10. Comparison of Measured and Estimated Position, April 22, 1999, Run#3

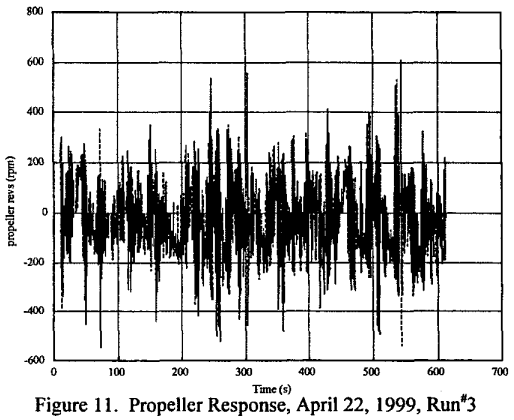

Figure 11. Propeller Response, April 22, 1999, Run#3

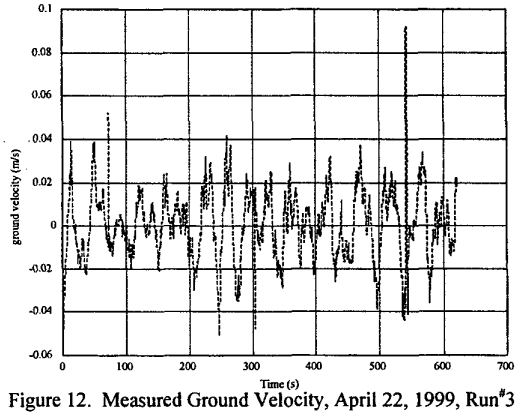

Figure 12. Measured Ground Velocity, April 22, 1999, Run#3

This run was the most interesting of the validation runs conducted. At the beginning of this run, it was noticed that the starboard shaft was not turning. Even with this propulsion system casualty, the vehicle was able to hold position and the controller did not go unstable. With only one shaft turning the effective input gain for the vehicle was

reduced by 50%. Operations of this nature indicate a very robust design. It can be seen in Figure 11, that there is a small increase in propeller revolutions around the 50 second point of the run. Data analysis indicated that this was approximately when the starboard shaft failed. Investigation into the cause of the shaft failure determined that a universal joint in that shafting had worked loose.

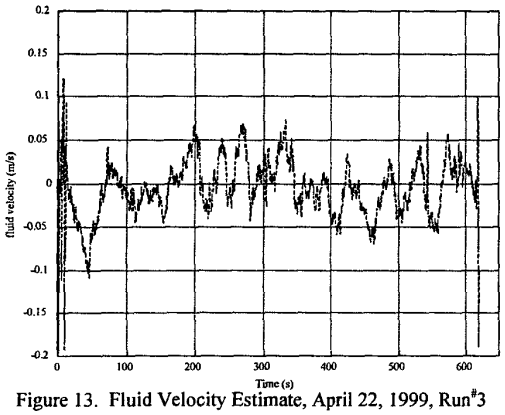

Figure 13. Fluid Velocity Estimate, April 22, 1999, Run#3

As a graphical representation of the performance expected for the DCC a various simulations ware conducted, using the asynchronous simulator discussed earlier, with the estimated fluid velocity obtained during this run (April 22, 1999, run# 3) as the disturbance. The gains on the DCC were varied to produce a position response verses propeller response curve. The actual experimental results, presented in Table 1, were superimposed on the theoretical curve obtained from simulation. These results are shown in Figure 14. As seen, the experimental and theoretical results are very close indicating a physically realistic simulator.

The comparisons displayed in Figure 14, yield insight into the achievable performance of the DCC. It indicates that there is a limit to the amount of disturbance rejection that is physically realizable. This limit is controlled by ability of the propulsion system to produce the needed input to maintain position without excessive oscillations. The excessive oscillations have a detrimental effect of the life of the propulsion system.

As a note, the short runs, displayed in Figure 14, were conducted with a controller gain parameter of $\eta = 100$, a high gain. If the length of these runs were extended, these points would move closer to the curve as with the other runs displayed.

Up to this point, the only results presented are for the Phoenix maintaining position to the origin, the point which the run was initiated. Questions arise as to how effective the controller is in dealing with transients. This question may be answered by referring to Figure 15. This figure depicts the transient response of the Phoenix for the various DCC gains presented in Figure 14. As the figure indicates, the controller deals with transients extremely well.

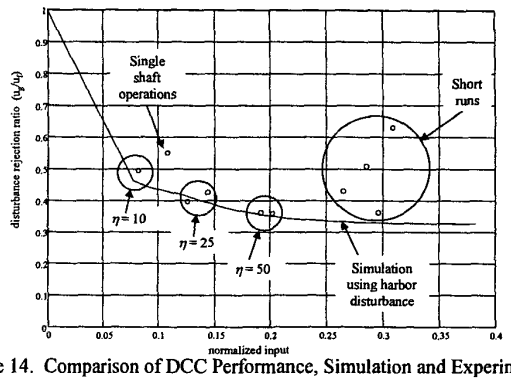

Figure 14. Comparison of DCC Performance, Simulation and Experimental

The responses displayed in Figure 16 are for the regulator solution. What is meant by this, recalling that the SMC formulation requires kinematically consistent position, velocity and acceleration, is that no command inputs, other that position were used. In doing this, it is expected that the vehicle will over shoot and oscillate around the commanded position consistent with some settling time.

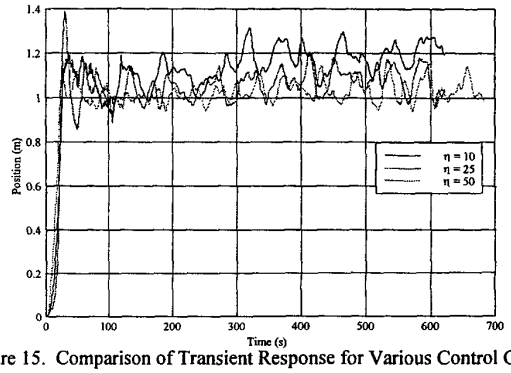

Figure 15. Comparison of Transient Response for Various Control Gains

With these transient responses come some issues with regard to operational implementation. If the goal is to position the vehicle close to, but with out touching, an object, some means of predicting the transient must be available. A resulting question that needs to be answered is; Does the developed simulator, which, based on the comparison in Figure 14, accurately predict the transient response? By comparing the results of the experimental runs and the simulated results, for the same disturbance input and DCC design, see Figure 16, the question can be answer, "yes".

As seen in this plot, the simulated results accurate reflect the measured transient response of the Phoenix. The steady state response, however, does not match. The reason for this is two-fold. First, the Phoenix, for recovery reasons, is maintained approximately two-pounds buoyant. This weight and buoyancy mismatch cause additional excitation forces resulting from the wave induced fluid accelerations. Since the fluid acceleration cannot be measured, this additional excitation force is difficult to replicate in simulation yielding errors between the real and simulated

response. Second, the experimental results are measured from a 6DOF rigid body, where as the simulator results come from a 1DOF surge model. The coupling effects from the surge-pitch dynamics will effect the comparison.

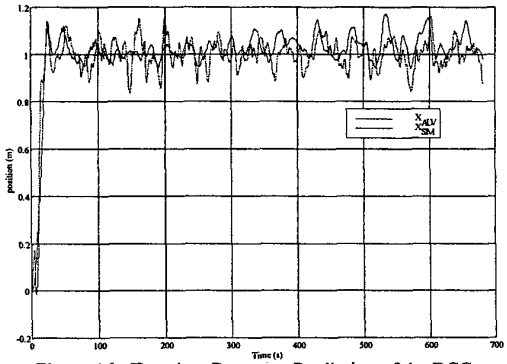

Figure 16. Transient Response Prediction of the DCC

VII. SUMMARY

The design, implementation and validation of a new Disturbance Compensation Controller (DCC) has been presented. The results indicate that by using a properly tuned system, the ability of intelligent underwater robots to perform intervention tasks are improved by their ability to gather, learn and use information about their working environment. Although no formal proof for stability is available, asynchronous simulations have demonstrated that the DCC is stable and provides acceptable tracking and estimation of state and disturbance inputs. The work has validated that the development and implementation of a real-time embedded Disturbance Compensation Controller (DCC) for small AUVs, and provided a new technology showing that it is possible to use underwater vehicles for station-keeping tasks in shallow water.

ACKNOWLEDGMENT

I would also like to recognize the financial support of the Office of Naval Research (Mr. Tom Curtain) under contract No. N0001498WR30175.

REFERENCES

1. Ogata, K., *Modern Control Engineering*, Prentice-Hall, 1990.
2. Gelb, A., *Applied Optimal Estimation*, MIT Press, Cambridge, MA, 1974.
3. Canudas de Wit, C. and Slotine, J.-J.E., "Sliding Observers for Robot Manipulators," *Automatica*, v. 27, n.5, 1991, pp. 859-864.
4. Rajamani, R., "Observers for Lipschitz Nonlinear systems," *IEEE Transactions on Automatic Control*, v43, n 3, March 1998, pp. 397-401.
5. Healey, A.J.; An, E.P.; and Marco, D.B.; "On Line Compensation Of Heading Sensor Bias For Low Cost AUVS," *Proceedings of the IEEE Symposium on Autonomous Underwater Vehicle Technology*, Cambridge, MA, August 1998.
6. Riedel, J.S., and Healey, A.J., "Shallow Water Station Keeping Of AUVs Using Multi-Sensor Fusion For Wave Disturbance Prediction And Compensation," *IEEE Proceedings of Oceans '98*, Nice, France, September 1998.
7. Riedel, J.S., *Seaway Learning And Motion Compensation In Shallow Waters For Small AUVs*, Ph.D. Dissertation, Naval Postgraduate School, Monterey, CA, June 1999.

Date	Run#	Length	*DRR*	σ_n/n_{max}	comments
4/2/99	4	4 min	0.3624	2.96	high gain, short run
	5	4 min	0.6324	3.08	high gain, short run, vehicle physical disturbed
	6	4 min	0.4312	0.265	high gain, short run
	8	4 min	0.5090	0.285	high gain, short run
4/22/99	3	10 min	0.5508	0.108	high gain, single shaft
5/25/99	6	10 min	0.3620	0.192	medium-high gain, ADV noise problem
	8	10 min	0.3978	0.126	medium-low gain, ADV noise problem
	9	10 min	0.4957	0.083	low gain, ADV noise problem
	11	10 min	0.3587	0.202	medium-high gain, ADV noise problem
	12	10 min	0.4276	0.144	medium-low gain, ADV noise problem

Table 1 Sample Summary of DCC Validation Runs