1994

Tactical / Execution Level Coordination for Hover Control of the NPS AUV II using Onboard Sonar Servoing

Healey, A.J; Marco, D.B.; McGhee, R.B.; Brutzman, D.P.;
Cristi, R.; Papoulias, F.A.; Kwak, S.H.

https://hdl.handle.net/10945/44828

Tactical /Execution Level Coordination for Hover Control of the NPS AUV II using Onboard Sonar Servoing

A. J. Healey, D. B. Marco, R. B. McGhee, D. P. Brutzman, R. Cristi, F. A. Papoulias
S. H. Kwak (Loral Defense Systems Inc.)
Naval Postgraduate School
Monterey
California, 93943

Abstract

This paper describes recent work with the NPS AUV II vehicle in the further development of the execution level software to incorporate hover control behavior in the NPS hover tank. Of particular interest is the use of the ST 1000 and ST 725 high frequency sonars to provide data about the environment. Thus positioning can be accomplished without the use of beacons. Motion behaviors may be instituted that include diving and pitch control under thruster power; heading control at zero speed; lateral and longitudinal positioning, as well as the automatic initiation of filters as needed when a new target is found. A simple task level language that will be used to direct tactical level output to a port in communication with the execution level software will be given.

Introduction

For several years the Naval Postgraduate School has been engaged in development of advanced control technology for unmanned vehicles that will have useful roles to play in the future actions of the US Navy. One of these roles is in the support of mine warfare missions involving search and find operations with lethal targets. While some have contended that there is a need for pure autonomy of power and control, it is our contention that, in control, only some level of autonomy will be needed and high level (low bandwidth / acoustic) communications with the vehicle will indeed be desired. Building an ever increasing level of automatic capability into a vehicle is of interest to us. In particular, under a new NSF grant, we are also concerned with the ease of reconfiguration of control software code as missions or vehicle capabilities change. To that end, we have defined a tri-level software architecture comprising Strategic, Tactical, and Execution levels. The Strategic level, using Prolog as a rule based mission specification language, tells the vehicle what to do next. Error recovery procedures from failures in the mission tasks or the vehicle subsystems are included in the rules of the Strategic level. The Tactical level, written in an object oriented code (using CLIPS-Ada in simulation versions to date), uses the methods of the objects to decided how to implement what is done next. The Execution level then commands the vehicle subsystems to institute behaviors that correspond to those commanded. Communication from the Tactical level to the Execution level takes place through a single (parallel) port. In the hierarchical design of the control system, the master scheduler in the Tactical level - the Officer of the Deck (OOD) [1], is the only object communicating with the Execution level software through that designated port so that *conflicts in behaviors* can not occur. (cf. Brutzman [2])

In new experiments with the NPS AUV II we have extended the flight control experiments conducted and reported at the prior conference [3]. We have now developed the thruster control

0-7803-1808-0/94 $4.00

behavior of the vehicle. Experiments with coordinated actions between the high frequency sonar and position control of the vehicle have now been conducted. Controlled sweeping of the sonar and positioning the vehicle to and from a wall have indicated that it may be possible in the near future to use these sonars to drive a vehicle to a target or between obstructions.

This paper will describe the results of these experiments and outline a shorthand language using key word scripts describing behaviors and set point values for response modes that are performed by the vehicle somewhat similar to the Task Based Control scheme of Wang [4].

System Overview

The NPS AUV II shown in Figure 1, has been recently outfitted with a Datasonics PSA 900 sonar at 200 KHz. to derive altitude above bottom signals, and two TRITECH high frequency sonars that are mechanically scannable through 360 degrees. The ST 1000 sonar is a 1 degree pencil beam profiling sonar that is best suited to measurement of the time of flight for the first strong return. The ST 725 sonar is a 20 degree high beam 2.5 degree wide that also may be mechanically scanned through full 360 degrees, or reduced sector, to capture a wider image of obstacles in front of the vehicle. This sonar returns intensity as a function of range in bins for any given heading.

These devices may be addressed for control purposes through a serial port where certain key ASCII characters are used to perform operational functions such as '*send one ping and analyze the return structure*' (either range to first return or intensity in a series of range bins), or '*turn by one step*'. By performing a sequence of such commands the sonar head may be made to *self center*, *continuously rotate while pinging and return the data stream,* or to *sweep over a defined sector and return the data stream.*

Additionally, the vehicle has now been equipped with cross body thrusters. Two vertical thrusters are for heave and pitch control, and two transverse thrusters are for heading and lateral movement control, in addition to the two propulsion motors at the stern and eight fin surfaces for flight control.

In previous work [5], waypoint following in a transit phase of a mission was demonstrated where competent behaviors were demonstrated and composed of

a) Forward_Speed,
b) Fin_Steering
c) Fin_Depth_Control
d) Waypoint_Following
e) Bottom_Following

These behaviors were implemented with a)-c) running simultaneously, but subsumed by d); and c) subsumed by e).

Control for waypoint following was accomplished in the Execution level with digital control algorithms running at 0.1 sec. update rate. Now, new behaviors, are enabled using active control of thrusters. These are,

d) Submerge_and_Pitch_Control
e) Heading_Control,
g) Longitudinal_Positional_Control,
h) Lateral_Speed_Control
i) Lateral_Positional_Control_on_Heading,
j) Center_Sonar,
k) Ping_and_Get_Sonar_Range (each sonar),
l) Ping_and_Get_Sonar_Range_and_Step_One(each sonar),
m) Step_Sonar (no ping, each sonar),
n) Initiate_Filter_For_Sonar_Range (Needed For Smoothed Range and Range Rate Estimation),
o) Reinitialize_Filter,

Most of these behaviors needs a given subset of actuators to be active under the operation of either an open loop command or a feedback control law. This means that vertical thrusters may be used via control laws to control depth as well as pitch, and lateral thrusters to control heading as well as lateral position and side slip speed. In combination with propulsion motors, behaviors including Submerge_and_Pitch_Control , Longitudinal_Speed_Control and Longitudinal_Position_Control, as well as Heading_Control, may be commanded reliably. Or, Heading_Control and Submerge_and_Pitch_Control; and virtually any multiple combination of a) to o) above that would not cause a conflict of actuator control or sensor usage can be commanded.

Simultaneous activation of combinations of these behaviors are instituted using a simple script of flags and set points that are a way of implementing the 'reaction plans' of Bonnasso [8], and the Task Based Control of Wang[4]. While the work of [8] has developed GAPPS rules that are more like our Strategic level rules, but, in the end would provide mode commands to vehicle servos, our work is developed around a rule based control to sequence mission related tasks [Byrnes,[6]], where the middle level of a tri-level architecture will generate the scripts required to produce in the vehicle the requisite behavioral action. We also believe as in Wang et. al. [4] that the key to successfully developing underwater robots is to have a fully functional execution level first with attention paid to the proper functioning of the servo loops addressing the actuators and sensors as keyed to the behaviors that must be capably instituted. Thus the behaviors a) through o) are stably implemented in the NPS AUV II vehicle through attention to appropriate digital control loops in the Execution level.

At the time of writing, all code runs on the OS-9 operating system in 'C' language and all sensors are queried at each pass through the control timing loop.

The Tactical level / Execution level interface is defined[Brutzman, [2]] and is comprised of

1) a serial port through which the Gespac 80386 card running DOS writes script packets,

2) a parallel port through which sensor data is written to the data manager,

It follows that every time step (0.1 sec.), script packets are read into the OS-9 side and are available as *flags and set points* through which behaviors a) to o) or combinations of a) to o) may be activated or suppressed in the Execution level software.

Figure 2 gives an example of a script of flags and setpoints to enable the vehicle to submerge to 4 feet of depth at a heading of 90 degrees. Then to Ping_and_Get_Sonar_Range from the first sonar, to start a filter to identify range and range rate on that target, and perform Longitudinal Positioning to 2 feet from target, and to

Submerge_and_Pitch_Control	4
Heading_Control	90
Ping_and_Get_Sonar_Range	1
Initiate_Filter_For_Sonar_Range	
Longitudinal_Position_Control	2

Figure 2 Example of a Behavior Based Script for Tactical/Execution Level Interface.

One of the crucial elements of the Tactical level software is to order the behaviors in such a way that conflicts are resolved before new commands are sent. Clearly it would not be sensible to call for a Heading_Control to 90 degrees and expect the sonar to get range to a target while the vehicle is in motion. The sequencing of task orders should also be keyed to a suitably defined event to ensure the previous behavior has settled out before committing to the next. There is therefore an event based sequencing element to the task sequencing which could, by suitable definition, be keyed to time delay. To test out this type of behavior based sequencing, a mission is read into memory that includes key times as well as behaviors to be initiated although with a full tactical level the scripts would be passed in real time across the parallel port.

Sonar Management and Control

Keeping the vehicle stationary and sweeping the ST 1000 sonar through 360 degrees has enabled us to acquire maps of the scene within which the vehicle operates. Also, for the behaviors listed above, sonar management can be effected at the execution level. For instance, to find a target we take the point of view of submarine officers that until you have three consecutive pings returning range consistently, there is no target out there. What this means here is that we command a heading on the sonar, tell it to ping and return range three times, and if three consecutive range data are close to each other, we associate a target with that fact. At that point, we start a Kalman Filter (constant gain) that will smooth the range data and estimate range rate. A third order filter is used as in,

Range dynamics model:

$$\mathbf{x}_{k+1} = \mathbf{\Phi x}_k + \mathbf{q}_k$$
$$y_k = \mathbf{Hx}_k + \nu_k$$

with the update of estimate from

$$\hat{\mathbf{x}}_{k+1} = \mathbf{\Phi}\hat{\mathbf{x}}_k + \mathbf{K}(y_k - \mathbf{H}\hat{\mathbf{x}}_k)$$

Use of the innovation $(y_k - \mathbf{H}\hat{\mathbf{x}}_k)$in the filter is key and for values larger than a defined threshold (say 0.3m), we can say that the filter has lost lock on the target, at which time, the filter update is ignored and the estimate propagated uncorrected. If several consecutive pings do not give innovation values within limits, then reacquiring of a target may become necessary. In our experiments to date, it has been possible to acquire a target because the walls constrain the environment, and there is always a wall to use. In open ocean, reacquiring a target would only be possible through initiating a new sweep motion of the sonar.

One of the recommendations arising from our more recent work will be that a 'sonar manager' be built into the middle level of the software architecture as it becomes clear that adaptive changes to the sonar setting will be a necessary feature of practical sonar management. For instance, at close range, good results cannot be obtained when sonar gains are high. Sonar gain (signal strength) must be tailored to the range of returns in addition to the usual TVG function, (see Figure 3 for instance).

Execution Level Software Structure

The structure of the Execution level software is illustrated by Figure 4 which indicates that it is composed of software at the hardware interface (software drivers) as well as software for vehicle control. After initialization of power systems and sonars, and the basic driver settings the PIA card pins that control the on/off feature of power supplies, thruster power, screw power, and sonar power, a simple timing loop is entered and reentered at a fixed update rate (in our case 0.1 sec.) during which the following must take place,

1. reading the sensors,
2. reading the port for behavior based command flags and set points,
3. selecting appropriate code proceedures to compute and send control values to actuators,
4. writing data to memory, and
5. checking time for any time based events and waiting for the next timing interrupt to maintain integrity of the digital control loop,

Specific control laws as built into callable modules of code will be describe later, and are easily selected according to the communication flags, provided that they exist in the first place.

Communications Language for Behavior Specification

The Tactical level / Execution level is now working to the point that behavior scripts can be sent to the OS-9 side from a serial port (or from memory) and thus a form of supervisory control or task based control (TBC) could now be implemented. Wang has mentioned the possibility of using finite state machines (FSM) rather than textural rule based languages as in Byrnes [6] to specify behaviors (tasks) in conducting a mission. What we are discussing here however, is not how to organize a mission, but rather the specifics of how behavior requirements are communicated to the execution level. For this purpose we propose first that competent behaviors of the AUV be available through well designed servo loops and robust control laws acting in a constant update rate digital control loop.

The activation of any given behavior is then turned on by a flag with a parameter for the command value or command generator to be used. Based on the truth of any flag, one of many available 'C' modules in the Execution level code is used as appropriate for that required behavior.

We see that it makes sense to send behavior commands that linguistically convey the meaning of the behavior as closely as possible as in those chosen above in a) through o). This will help to enforce code readability and possibly to make code modules transferable between vehicles that could exhibit common behaviors. In our case, the Tactical level object "OOD"[1] is responsible for constructing the scripts that are sent to the parallel port to communicate with the execution level processor. The flags are read and interpreted by the execution level in the same sense that they were transmitted and used to select appropriate controls.

Modes of behavior could be changed as fast as the update rate of the Execution level control loop.

Experiments Conducted and Results

We have recently conducted experiments in the NPS Hover Tank, a 20' by 20' by 6' deep tank where the NPS AUV II was required to demonstrate several thruster controlled behaviors including a mission where the time based sequence of multiple coordinated behaviors was accomplished. The first of these behaviors was to submerge to a required depth and keep a particular heading.

The combination of the Submerge_and_Pitch flag with set points of 4 and 0, indicate that the vertical thrusters must be activated by a control law involving combined depth and pitch control such as;

Submerge using Rate Limit and Pitch Control

```
$\dot{z}_{com} = K_z z_e$

if(abs($\dot{z}_{com}$) >= $\dot{z}_{max}$)
{
    sat($\dot{z}_{com}$) = $\dot{z}_{max}\left(\frac{\dot{z}_{com}}{abs(\dot{z}_{com})}\right)$
}
else
{
    sat($\dot{z}_{com}$) = $\dot{z}_{com}$
}
```

$$V_{bvt} = K_w(sat(\dot{z}_{com})) - K_\theta \theta_e + T_{d\theta}\dot{\theta}$$

$$V_{svt} = K_w(sat(\dot{z}_{com})) + K_\theta \theta_e - T_{d\theta}\dot{\theta}$$

with the result as shown in Figure 5. Very smooth control is achieved with even a simple P/D control law. We contrast the results with those of JASON whose vertical thrusters have been claimed to be responsible for the inducing of vertical plane limit cycling whereas our thrusters perform well in both steady state as well as transient response [Cody, [7]] in spite of the use of a linear law using square law thrusters. Input linearization and sliding mode control laws are the next step for comparison of closed loop dive performance.

We have instituted a rate limit feature to our control law, shown in the block diagram in Figure 6, because of the requirement in may cases to come to depth without overshoot.

Heading Control Law

The Heading_Contol flag with set point of 90 degrees will turn the vehicle to 90 degrees with the following P/D control law,

$$V_{blt} = K_{\varphi}(\varphi_e + T_{d\varphi}\dot{\varphi})$$

$$V_{slt} = -K_{\varphi}(\varphi_e + T_{d\varphi}\dot{\varphi})$$

Figure 7 shows the results of three separate responses to indicate the level of repeatability in the motion produced. Again, while these results are not optimized and only the result of PD controllers, we expect to gain even more precision using more sophisticated techniques such as sliding mode control [9] model based maneuvering[10], and command generator tracking.

Vehicle Longitudinal Motion Control Results

Evidence of the need to manage the sonar signal strength is shown in Figure 3 where the response of the vehicle in closing to a target shows that continuous pinging at close range tends to break up the range signal. After 70 seconds, the loss of signal quality causes a motion disturbance, whereas when the sonar gain is lowered (Figure 8), the response is smooth and steady. The P/D control used was identical for each stern propeller,

$$V_{ls} = V_{rs} = K_X(X_e + T_{dX}\dot{X})$$

Vehicle Lateral Motion Control Keeping Heading Results

Experiments to servo laterally to a wall keeping heading under control without forward motion were conducted. The response of the lateral range as depicted by the ST 1000 sonar is given as well as the thruster behavior in Figure 9. What we show is the ability to side slip and lock onto a wall. This behavior is reasonable well controlled in spite of the simple control law which will be improved in the near future. We can now demonstrate that wall following and keeping depth and heading can be done with reasonable precision. The effects of current and wave action of course are not present.

$$V_{blt} = V_{slt} = K_Y(Y_e + T_{dY}\dot{Y})$$

Time Base Sequence Control Of Complex Behaviors

In performing a complex set of behaviors, we have demonstrated that it is possible to order a sequence of tasks using time as a distinguishing event so that the vehicle can execute an example mission comprised of

1. Submerge_and_Pitch_Control at 0 pitch angle to 4 feet depth at Heading _Control , 90 degrees
2. Ping_and_Get_Sonar_Range at 0 degrees to establish range to target.
3. Initiate_Filter_For_Sonar_Range,
4. Longitudinal_Positional_Control to 2 feet from target,
5. Longitudinal_Positional_Control to 4 feet from target without Reinitialize_Filter
6. Heading_Control to 270 degrees (a full 180 degrees rotation of the vehicle)
7. Ping_and_Get_Sonar_Range to new target (opposite end wall)
8. Initiate_Filter_For_Sonar_Range for new target

9. Longitudinal_Positional_Control to 2 feet from target

10 Submerge_and_Pitch_Control at 0 pitch angle to 0 feet depth (surface) at Heading _Control, 270 degrees

These maneuvers have been accomplished in sequence as can be seen on the video part of this paper.

Conclusion

The conclusion to date of our work has indicated that complex behavior can be coordinated through tactical level scripts communicating to the execution level software commanding competent behaviors of the vehicle that are defined and well controlled through careful attention to the digital control design. In this mode, the designer of the scripts acts as the Strategic level of our tri-level software architecture previously reported.

The independent coordination of sonar for range finding on a bearing, or for imaging over a particular sector of bearings is being used to derive motion commands for the vehicle resulting in smooth vehicle motion results in an underwater environment free from current and wave action. Time delays in processing sonar data can be a difficult problem and is under research. We would anticipate, however, that in the near future, sonar based relative navigation without the use of beacons could be possible in structured or feature rich scenes.

Acknowledgment

The authors wish to recognize the financial support of the National Science Foundation under Grant No.BCS-9306252 as well the Naval Surface Warfare Center, Annapolis Division., the Naval Explosive Ordnance Disposal Technology Center, and the Naval Postgraduate School Direct Research Foundation, all partly contributing to the conduct of this work,

References

[1] Byrnes, R., Kwak, S. H., McGhee, R. B., Healey, A. J., Nelson, M. L.,"Rational Behavior Model:An Implemented Tri-Level Multilingual Software Architecture for Control of Autonomous Vehicles" Proceedings of the 8th. UUST, Held at University of New Hampshire, Durham, NH. September 27-29, 1993 pp. 160-179

[2] Brutzman, D. P., Kanayama, Y., Zyda, M. J.," Integrated Simulation for Rapid Development of Autonomous Underwater Vehicles", Proceedings of IEEE Oceanic Engineering Society _Symposium on Autonomous Underwater Vehicles, AUV-92_ Washington D.C., June 2-3, 1992.pp. 3-10

[3] Healey, A. J., Marco, D. B., " Experimental Verification of Mission Planning by Autonomous Mission Execution and Data Visualization using the NPS AUV II." Proceedings of IEEE Oceanic Engineering Society _Symposium on Autonomous Underwater Vehicles, AUV-92_ Washington D.C., June 2-3, 1992.

[4] Wang, H., Marks, R. L., Rock, S. M., Lee, M. J.," Task Based Control Architecture for an Untethered Unmanned Submersible" Proceedings of the 8th. UUST, Held at University of New Hampshire, Durham, NH. September 27-29, 1993 pp. 137-149

[5] Healey, A. J., Marco, D. B., "Slow Speed Flight Control of Autonomous Underwater Vehicles: Experimental Results with NPS AUV II" _Proceedings of the 2nd. International Offshore and Polar Engineering Conference,_ San Francisco, July 14-19 1992.

[6] Byrnes, R. B. "The Rational Behavior Model: A Multi Paradigm, Tri-Level Software Architecture For Control Of Autonomous Vehicles", Ph.D.

Dissertation, Naval Postgraduate School, Monterey CA. March 1993

[7] Cody, S. "Step and Triangular Wave Input Response for Small Tunnel Thrusters" MSME Thesis Naval Postgraduate School, Monterey, CA. Dec. 1992

[8] Bonasso, R. P., Barrat, J. "A Reactive Robot System for Find and Visit Tasks in a Dynamic Ocean Environment", Proceedings of the 8th. UUST, Held at University of New Hampshire, Durham, NH. September 27-29, 1993 pp. 69-80

[9] Healey, A. J., Lienard, D., "Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles", <u>IEEE Journal of Oceanic Engineering</u> Vol. 18, No. 3, July 1993 pp. 1-13

[10] Healey, A. J., "Model Based Maneuvering Controls for Autonomous Underwater Vehicles". <u>Trans. ASME Journal of Dynamic Systems, Measurement and Control</u> Vol. 114, No. 4, Dec. 1992 pp. 614-622

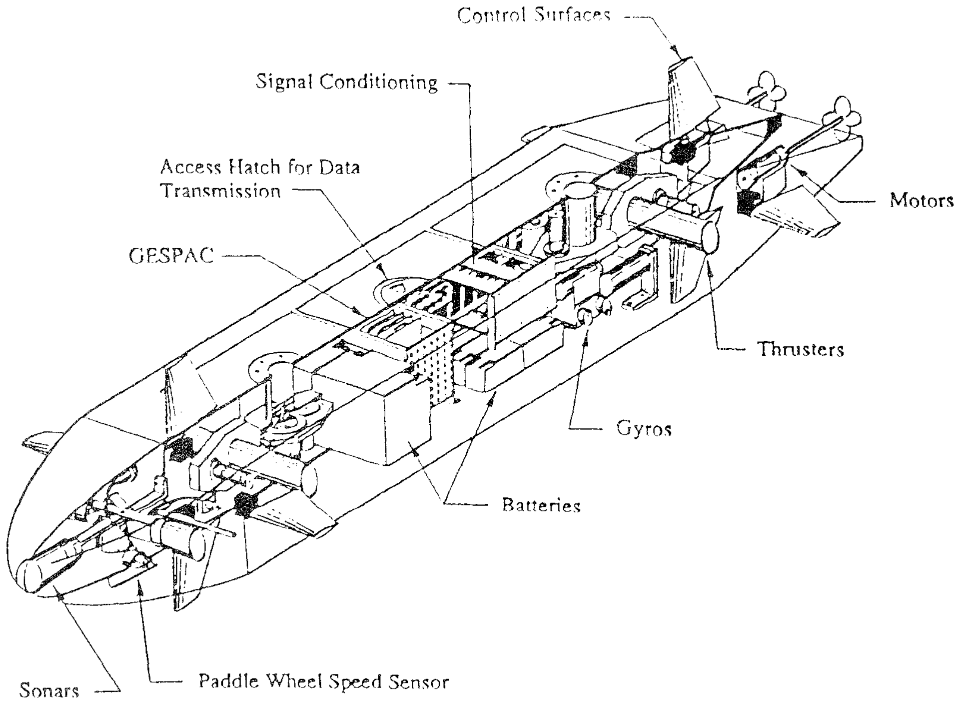
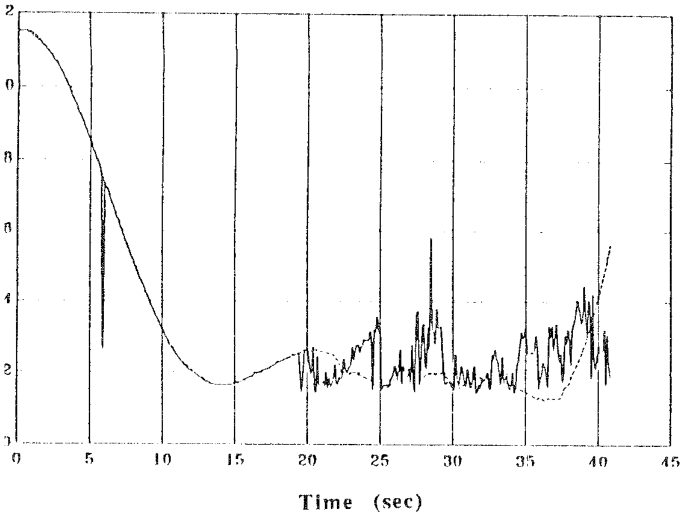


Figure 1. Outline of the NPS AUV II Vehicle



igure 3. Range and Filtered Range Signals From
T1000 Indicating Breakup From Over Ensonification

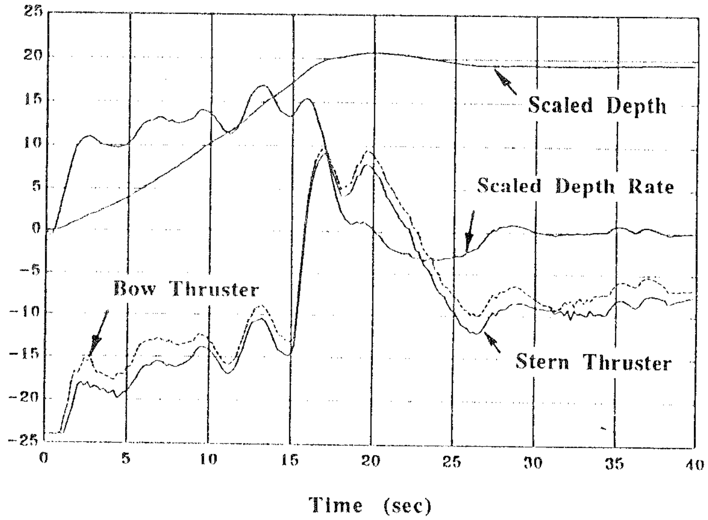


Figure 5. Submerge_and_Pitch_Control Response

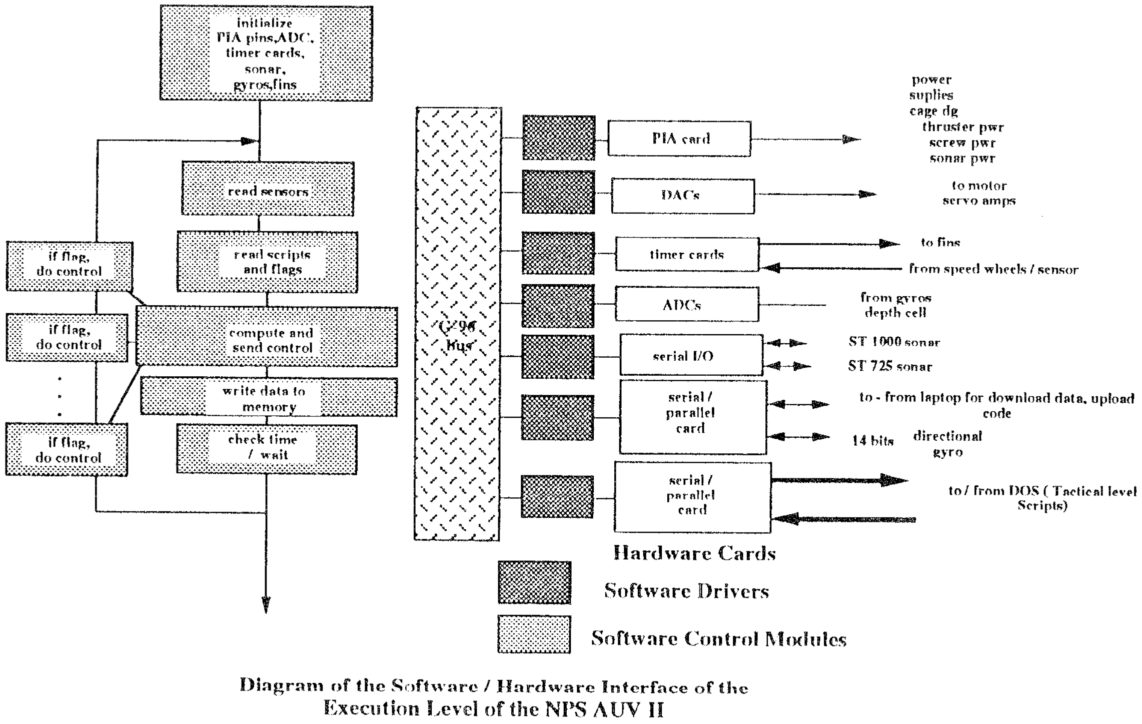


Figure 4. The Structure of the Execution Level Software

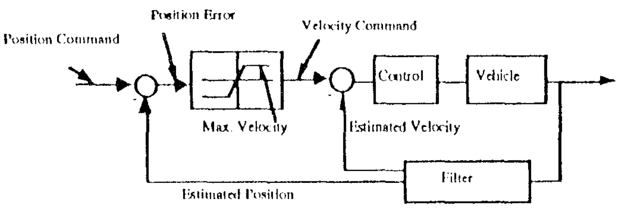


Figure 6. Rate Limited Control of Position Allows for Large Inputs

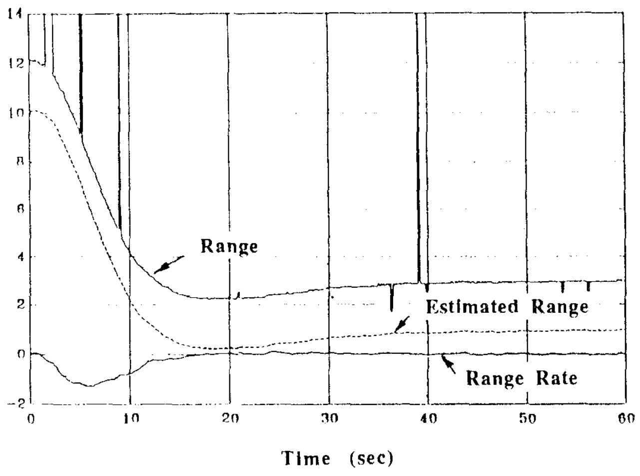


Figure 8. Longitudinal_Position_Control (Low Sonar Gain). Range, Estimated Range (Offset 2 feet), and Range Rate

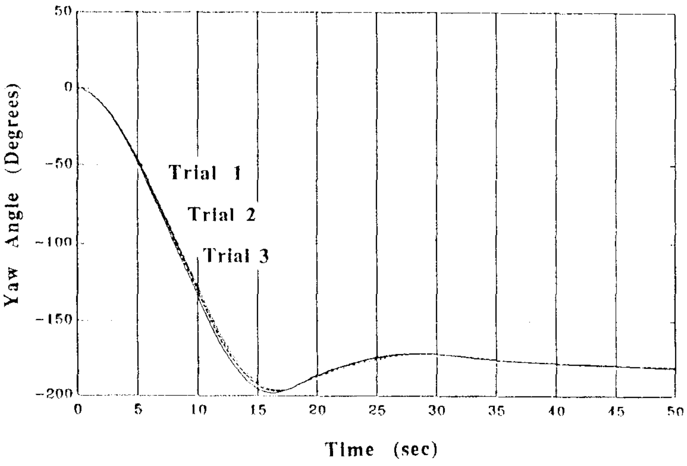


Figure 7. Heading_Control Repeatability from Three Separate Responses for 180 Degree Turn

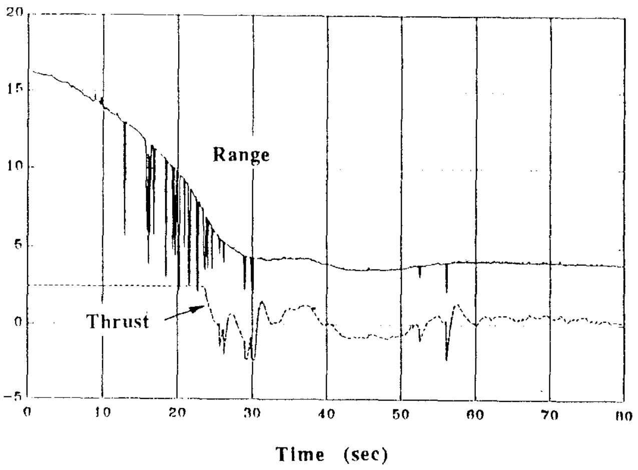


Figure 9. Lateral_Position_Control (Low Sonar Gain). Range and Scaled Thrust Level