# M B A R I

# Autonomous Underwater Vehicle Control with a 3 Actuator False Center Mechanism

**Cheri Everlove, Santa Clara University**
*Mentor: Bill Kirkwood*
*Summer 2004*

**Keywords: AUV, False Center, Steering Mechanism**

## ABSTRACT

This paper describes the need for an alternate steering mechanism as well as the design, testing and some possible improvements for the proposed mechanism. A Three Actuator False Center Control mechanism is the proposed solution to add redundancy, and attempt to reduce the overall steering mechanism size. The successful bench test is described. Additional configurations are then suggested for correction of the out-of-plane issues.

## INTRODUCTION

MBARI's Dorado Class Autonomous Underwater Vehicles (AUV's) are both propelled by and steered by a single thruster mounted at the rear of the vehicle. There are no fins to give rudder or elevator control. Turning the vehicle is accomplished by moving the articulated tailcone, which consists of the propeller, shroud, motor, and gimbal mechanism with two linear actuators. MBARI's gimbal consists of an outer ring that rotates about the vertical axis (providing rudder control or yaw), and an inner ring that rotates about the horizontal axis (providing elevator control or pitch).

The original concept was created by Bill Kirkwood in 1999 and built by the MBARI /Dorado AUV team. MBARI owns the patent and licenses it to Bluefin Robotics. There have been years of successful missions, including over 1700 kilometers logged on MBARI's CTD (temperature, salinity, depth) configured vehicle alone and several successful missions underneath the Arctic ice. These questions need to be

considered: What would happen if one actuator fails? Can we add flexibility to actuator positioning? Is smaller packaging possible?

In the current design if one of the two actuators fails and the AUV is in open water, the vehicle will abort its mission and should eventually float back to the surface. However if that AUV is deep under the arctic ice, recovering the vehicle poses a problem, because it would abort its mission and be far away from any access to recover the vehicle. The current design is also constrained by the physics of the gimbal mechanism requiring that the two actuators be mounted one on the outer gimbal for elevator control and one on the inner gimbal for rudder control. These attachment points are fixed locations and cannot be changed if the actuator is to work. This inflexibility of mounting angles greatly reduces the ability to place additional sensors in the tail section of the AUV. Furthermore the current gimbal mechanism takes up a lot of room in the rear section of the AUV. By replacing the gimbal and reducing the size of the mechanism, and using the mechanical advantage of three motors and three lever arms, we hope to not only gain on power usage but to carve out additional room for scientific sensors.

The primary goal of this project is to develop a tabletop model of a three-actuator mechanism including the linkages, propeller mount and the control box. The secondary goal of the project is to implement a failure tolerant system and the tertiary goal is to design a mechanism that allows for out-of-plane motion created by the new false center.

**MATERIALS AND METHODS**

Initial conceptual design was done with the help of Bill Kirkwood. We identified the key requirements for the new tailcone and then came up with ways of accomplishing them. The requirements included the ability to carry the thruster and transfer the load into the hull, be able to meet or exceed the response of the current mechanism, use the same or less power, reduce the size, weight and electrical noise, and be able to achieve 20-25° of elevator and rudder control.

The initial mechanical concept was to mount three hobby-servo actuators to a flat platform, then connect them to a spherical shape via ball-end joints and connecting rods to allow for full rotational movement of the mechanism. These connecting rods were initially placed 120° apart with future flexibility allowing different angles of separation to

be achieved.  The sphere was then to rest in a cradle, also attached to the platform, which would help to reduce friction and constrain movement.

The next basic conceptual problem was to calculate how much to rotate the servos to achieve the desired rotation of the sphere.  Because the servo motors moved in 2-space while the sphere moved in 3-space, this problem was very difficult to solve.  The solution involved creating a rotational matrix that could take an attachment point, and with desired yaw, pitch and roll angles, transform the point to its new location in 3-space. Then translate that to how much to move the servos to get the connecting rods to the new desired location.  Additional consideration needed to be taken of the false center that is created when one of the actuators fails. When one motor fails, the sphere will rotate about the attachment point of the failed motor, and no longer rotate about the center of the sphere.

The measurement of the yaw, pitch and roll angles was the next necessary piece of information. The sphere needed to be hollowed out to allow for future mounting of the thruster, and current mounting of a feedback sensor. The position information was obtained by mounting a 3-axis clinometer in the center of the sphere to provide knowledge of where the sphere really was. The sensor, from Sparton Electronics, was a serial device capable of providing three-dimensional tilt-compensated bearing with +/- 180° range and X-Y tilt with > +/-80° range.
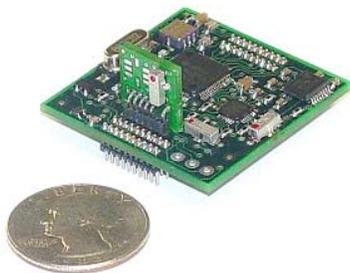


Figure 1: Sparton SP3000D compass.

Finally the control box needed to be designed. The mechanism was to be controlled with a lapbox.  This would have a joystick for command generation, a LCD display for showing information, three separate kill switches for future simulation of servo failure, and contain a microcontroller to receive information from the sensor and deliver position information to the servos. The Microcontroller chosen was a BX-24 from NetMedia Inc.  It has a BASIC true multitasking operating system with hardware interrupts, 16 I/O lines, and onboard TTL serial RS-232 support.
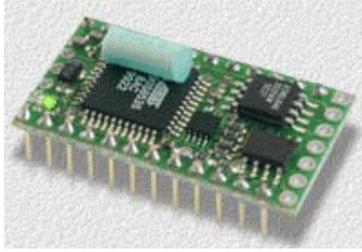
Figure 2: BX-24 Microcontroller from NetMedia.

The joystick chosen was a J40 model from ETI Systems, which had a 60° range of motion. This model consists of two 10KΩ Potentiometers, that act as voltage dividers, so that a 0 to 5V input yields 0 to 5V output for each axis. This 0-5V needed to be translated into arc degrees of movement, to provide commands for yaw and pitch control
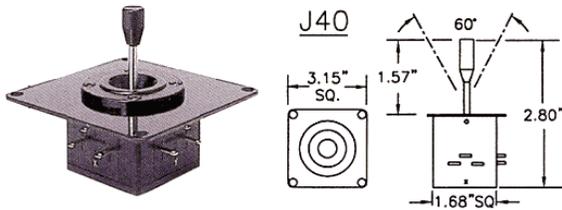


Figure 3: ETI Systems J40 Joystick.

The next phase of the design was the detailed design phase. Solid Works 2003 was used for the majority of the mechanical design. Most parts had several revisions, with one part, the cradle, having 4 revisions. See Appendix A for shop drawings. The last of these revisions was to insert a channel that would accommodate 1/8" Torlon™ bearings. These needed to be set just deep enough to provide a reduction of friction without reducing the ability of the cradle to constrain the movement of the sphere. This was difficult because the bearings were loose in the channel, and would pop-out when the sphere would try to pitch. Attempts were made at finding a suitable lubrication for the bearing, so that they would slide and roll, but in the end, we just used fewer bearing in the groove, so that they weren't rubbing against each other.

After suitable geometry and materials had been found for all of the parts, machining started in the model shop and in the machine shop. The sphere and cradle, which needed large amounts of material removed and needed to fit together perfectly, were made by the machine shop. The rest of the parts were either purchased or fabricated

in the model shop. The mechanism was then assembled, and the lapbox was wired up. See Appendix B for circuit diagram.

Finally, the microcontroller needed to be programmed. See Appendix C for complete program listing. One significant issue involved the sharing of the built-in timer on the BX-24 by the serial port and the pulse command for servo motor control. Servo motors are controlled by varying the length of the pulse from 1.0 to 2.0 ms. A 1.0 ms pulse positions the servo to the far left, and a 2.0 ms pulse positions the servo to the far right so timing is critical to correct positioning. Timing is also critical for RS232 communication. Therefore it was necessary to program the BX-24 to share the timer without interrupting each other. This was possible due to the ability of the microcontroller to have separate tasks, running simultaneously. Additionally, the TTL RS-232 signal from the BX-24 needed to be conditioned to be compatible with the Standard RS-232 that the sensor operated on. An off-board MAX233 chip was used successfully.

**RESULTS**

Only the primary goal was achieved. I was able to successfully program the microcontroller to: effect servo motor positioning, read in sensor position on a serial ComPort, and display actual vs. desired angles on the serial LCD display. The three switches that were originally designed to simulate failure of each of the servo motors were instead used to control the roll, pitch or yaw angles displayed on the LCD. I was unable to fully implement the rotational matrix to determine how much to rotate the servos. Although I can make the servos move, I don't have them moving to the right location. Therefore the error between desired and actual angles is great. This should be fairly easy to add to the existing code, however I just ran out of time. I learned how to use Solid Works to make drawings that can be made into real parts. I was successful at troubleshooting the timing issues involved in both motor control and serial communications. I learned about rotational and translational movement and have more to learn. I also enjoyed watching what other engineers were working on, and getting exposure to science driven engineering.
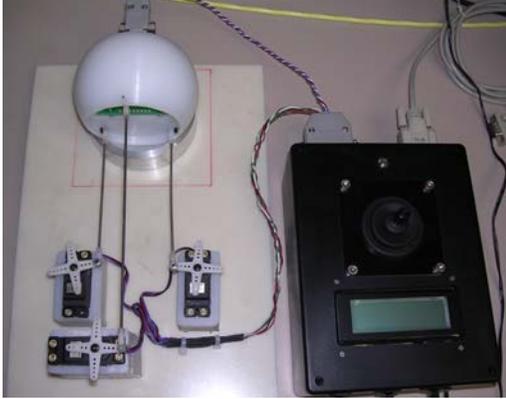
Figure 4: Assembled proof of concept.

## DISCUSSION

Although only the primary goal was fully achieved, much thought was given to the other two goals. One possible approach to take in creating a failure tolerant system is to implement an algorithm that moves each motor in turn, a tiny increment towards minimizing the error in pitch, yaw and distance from the true-center of the sphere. A table can be created to map how each movement affects the position and errors. A weight can be assigned to each error. This method should work with all, two or only one motor working. Since each working motor can only move in or out, each motor has only to choose the one that is the best choice. The full range of motion will not be possible if any of the motors fail, but some degree of control should be attainable. The mechanism to allow the out-of-plane motion while eliminating roll is not discussed here, as it is left for further work.

## CONCLUSIONS/RECOMMENDATIONS

The three-actuator mechanism needs additional modifications to enable its implementation as a fault tolerant system. The next stage is the hardest: Replacing the spherical cradle with a mechanism that allow for out-of-plane movement in the event of failure of one of the actuators, and including in that mechanism a way to prevent roll, as the actuators only move in the planes that control pitch and yaw. Additional work could be done by replacing the tilt sensor with linear sensors that can detect actual distance traveled by the connecting rods, and the software needs to implement an algorithm for

minimization of error of actual vs. commanded pitch and yaw. The need for redundancy is clear, and this mechanism holds promise as a solution.

## ACKNOWLEDGEMENTS