HYDRODYNAMIC AND THRUSTER MODEL VALIDATION FOR AUTONOMOUS UNDERWATER VEHICLES

M. Chyba, T. Haberkorn, R.N. Smith, S.K. Choi and Scott Weatherwax
Department of Mathematics
Department of Ocean & Resources Engineering
Autonomous Systems Laboratory, College of Engineering
University of Hawaii
Honolulu, Hawaii 96822
Email: mchyba@math.hawaii.edu, haberkor@math.hawaii.edu, ryan@ore.hawaii.edu, schoi@hawaii.edu, waxer@artifex.org

ABSTRACT

From Pontryagin’s Maximum Principle to the Duke Kahanamoku Aquatic Complex; we develop the theory and generate implementable time efficient trajectories for a test-bed autonomous underwater vehicle (AUV). This paper is the beginning of the journey from theory to implementation. We begin by considering pure motion trajectories and move into a rectangular trajectory which is a concatenation of pure surge and pure sway. These trajectories are tested using our numerical model and demonstrated by our AUV in the pool. In this paper we demonstrate that the above motions are realizable through our method, and we gain confidence in our numerical model. We conclude that using our current techniques, implementation of time efficient trajectories is likely to succeed.

Introduction

In every area of research, merging theory and application is the ultimate goal, and is usually quite difficult. This difficulty arises especially in hydrodynamics and fluid mechanics due to the complexity of the flow and the lack of a complete understanding of all the components involved. Our main research is focused on the implementation of time and/or energy efficient trajectories onto AUV’s. This research involves developing a mathematical theory in optimal control, numerically calculating trajectories and implementing them onto a testbed AUV. The theory finds the extremals of the Hamiltonian system using Pontryagin’s maximum principle. A numerical algorithm, named STPP, parameterizing the switching times of the thrusters has been developed which creates implementable trajectories which are close in time to the optimal trajectories found in the theory, see [1, 3]. Our ultimate goal is to implement these trajectories on our test-bed vehicle ODIN (Omni-Directional Intelligent Navigator); a spherical vehicle with eight independent thrusters for control. Preliminary tests demonstrated the need for a better understanding of the model for the vehicle and for the thrusters. In this paper, we fo-
focus on the implementation of pure motion trajectories onto ODIN. Pure motion trajectories have been studied as de-coupling vector fields in [2]. Such experiments provide critical information about the model and are essential to our project. Discrepencies between the experimental and theoretical trajectories can be related to many things such as actuator imperfections, inaccurate hydrodynamic model, unknown drift velocities and human error. Through a multitude of tests, we are able to see what parameters affect vehicle performance in what ways, and we can make adjustments in both the theory and the experiments in order to successfully and accurately implement efficient trajectories onto AUV’s in the future.

Model
The model we have chosen is driven by the test bed AUV that we use to test our theory. Any model is suitable, but for our purposes, we have chosen one that we can ground truth. We assume that we have a rigid body deeply submerged in a real fluid; we take real to be an ideal fluid with viscosity. It is understood in practice that a viscous fluid is rotational, but we assume this is not the case for theoretical reasons. We assume that the origin of the body fixed frame is the center of gravity (CG) of the body, the body has three planes of symmetry with body axes that correspond to the three principle axes of inertia.

Equations of Motion
We consider the general equations of motion for marine vehicles in six degrees of freedom which stem from Newton’s Second Law and express them as a controlled mechanical system. A detailed derivation can be found in [3]. We let \( \eta = (x, y, z, \phi, \theta, \psi)^T \) be the position and orientation of the vehicle in the earth-fixed reference frame, where \( (\phi, \theta, \psi) \) are the classical Euler angles. The translational \( \nu = (\nu_1, \nu_2, \nu_3)^T \) and rotational \( \Omega = (\Omega_1, \Omega_2, \Omega_3)^T \) velocities are taken with respect to the body-fixed frame. With this notation, we can write the equations of motion as,

\[
M\ddot{\nu} = M\nu \times \Omega - D_v(\nu)\nu + \rho'g\nabla k + F \quad (1)
\]

\[
J\ddot{\Omega} = J\Omega \times \Omega + M\nu \times \nu - D_\Omega(\Omega)\Omega - r_B \times \rho'g\nabla k + \tau \quad (2)
\]

where \( M \) accounts for the mass of the rigid body and the added mass coefficients, \( J \) accounts for the body moments of inertia and the added moment of inertia coefficients and \( F \) and \( \tau \) account for the control of the vehicle. The term \( r_B \times \rho'g\nabla k \) is the restoring force which is a pure torque induced upon listing where \( r_B \) is the vector from \( CG \) to the center of buoyancy (\( CB \)), \( \rho \) is the fluid density, \( g \) is the acceleration due to gravity, \( \nabla \) is the submerged volume of the body and \( k \) is the unit vector pointing in the direction of gravity.

Test-Bed AUV
The test-bed AUV which we use is the Omni-Directional Intelligent Navigator (ODIN) which is owned and operated by the Autonomous Systems Laboratory (ASL) at the University of Hawaii. As seen in Figure 1, ODIN has a spherical hull which is 65cm in diameter. This sphere is constructed from an aluminum alloy to prevent corrosion. Eight thrusters are attached to the sphere via four fabricated mounts, each holding two thrusters. The thrusters are evenly distributed around the sphere with four vertical and four horizontal, see Figure 1. Fully assembled, ODIN weighs 126.55kg and is positively buoyant by 0.7kg. ODIN is capable of moving in 6 DOF from either a remote or autonomous mode. For our experiments, ODIN is tethered, but only to send commands via TCP/IP protocol from a shore based laptop to be run in an autonomous mode. This setup allows for multiple tests to be conducted without removing ODIN from the water to upload mission sorties. ODIN’s internal CPU is a 800 MHz Intel based processor running on a PC104+ form factor with two external I/O boards providing A/D and D/A operations. Major internal components include a pressure sensor, inertial
measurement unit and 24 batteries. ODIN is able to compute real time, yaw, pitch, tilt, and depth and can run autonomously for up to 5 hours. The software is divided into two components. The first component is based on a real time extension to the Windows 2000 operating system, which provides ODIN real time autonomous control. The second component runs on the remote laptop and allows the operator to upload autonomous mission profiles to ODIN on the fly during testing as well as monitor ODIN in real time during such missions.

**Parameters**

The experiments conducted for this research are carried out at the Duke Kahanamoku Swimming Complex at the University of Hawaii. We use ODIN as the test-bed AUV. In the previous section, we stated that ODIN does not have real time sensors to detect horizontal (x – y) position. Instead, experiments are video taped from the 10m diving platform, giving us a near nadir view of ODIN’s movements. Videos are saved and horizontal position is post processed for later analysis. A real time system utilizing sonar was available on ODIN, but was abandoned for two main reasons. First, the sonar created too much noise in the diving well and led to inaccuracies. More significantly, in the implementation of our optimal trajectories, ODIN is often required to achieve large (> 15°) list angles which render the sonars useless for horizontal position. Many solutions were attempted and video led to the most accurate results.

Unique to ODIN’s construction is the control from an eight dimensional thrust to move in six DOF. This construction puts redundancy into the system in case of thruster failure. ODIN is able to operate in an “underactuated” condition if necessary, and research is active in this area as well. Under certain circumstances of underactuation, ODIN is still fully controllable. Our input trajectories to ODIN take the form of the six DOF controls which are converted onboard ODIN to the eight actual thrusters using the following Thrust Control Matrices (TCM’s) (Eqns. 3 and 4).

**TCMhorizontal** =

\[
\begin{bmatrix}
-0.707 & 0.707 & 0.707 & -0.707 \\
0.707 & 0.707 & -0.707 & -0.707 \\
0.48160 & -0.48160 & 0.48160 & -0.48160
\end{bmatrix}
\]  

(3)

**TCMvertical** =

\[
\begin{bmatrix}
-1.0 & -1.0 & -1.0 & -1.0 \\
-0.26989 & -0.26989 & 0.26989 & 0.26989 \\
0.26989 & -0.26989 & -0.26989 & 0.26989
\end{bmatrix}
\]  

(4)

The numerical values of the various parameters used for the model are given in Table 1. These values were derived from experiments performed on ODIN. The added mass and drag terms were estimated from formulas found in [4, 5]. Moments of inertia were calculated using experiments outlined in [6]. We used inclining experiments to locate and place \( C_B \) in the location seen in Table 1, and we assume that \( C_G \) is located at the center of our body-fixed axis. Note that units for inertia coefficients (denoted by \( \ast \)) are \( \text{kg.m}^2 \). Along with the tests to determine the values

<table>
<thead>
<tr>
<th>m</th>
<th>126.55 kg</th>
<th>( \rho g V )</th>
<th>1243 N</th>
<th>( C_B )</th>
<th>(0,0,-7) mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M^\ast )</td>
<td>70 kg</td>
<td>( M^\ast )</td>
<td>70 kg</td>
<td>( M^w )</td>
<td>70 kg</td>
</tr>
<tr>
<td>( I_x )</td>
<td>5.46*</td>
<td>( I_y )</td>
<td>5.29*</td>
<td>( I_z )</td>
<td>5.72*</td>
</tr>
<tr>
<td>( J_f^p )</td>
<td>0*</td>
<td>( J_f^q )</td>
<td>0*</td>
<td>( J_f^r )</td>
<td>0*</td>
</tr>
<tr>
<td>( D_{11}^{11} )</td>
<td>35.789</td>
<td>( D_{12}^{11} )</td>
<td>35.7953</td>
<td>( D_{21}^{11} )</td>
<td>31.1602</td>
</tr>
<tr>
<td>( D_{12}^{12} )</td>
<td>36.836</td>
<td>( D_{22}^{11} )</td>
<td>36.8312</td>
<td>( D_{22}^{22} )</td>
<td>34.8251</td>
</tr>
<tr>
<td>( D_{11}^{12} )</td>
<td>18.9379</td>
<td>( D_{11}^{21} )</td>
<td>18.9379</td>
<td>( D_{21}^{12} )</td>
<td>16.8545</td>
</tr>
<tr>
<td>( D_{12}^{12} )</td>
<td>35.4235</td>
<td>( D_{22}^{22} )</td>
<td>34.4235</td>
<td>( D_{22}^{22} )</td>
<td>31.5264</td>
</tr>
</tbody>
</table>

Table 1. Numerical values of hydrodynamic coefficients.

in Table 1, we also tested the thrusters. Each thruster has a unique voltage input to power output relationship. This relationship is highly nonlinear and is approximated using a piecewise linear function which we refer to as our thruster model. Hence, we see residual phantom thrusts occurring due to the unbalancing or non-synchronization of the eight thrusters. The most detrimental phantom moment we encounter is in yaw, partially due to the lack of a restoring moment along with small drag resistance.

**Experiments**

Below we analyze two experiments of pure motion trajectories implemented on ODIN. The first is a pure heave, and the second a pure surge. In each case, we analyze three separate trajectories in order to help understand the model and dynamics. The first trajectory is the experimental trajectory which is the actual trajectory realized by ODIN in
the pool. This is plotted using the data collected directly from ODIN’s sensors and the video camera. The second trajectory analyzed is the pre-experiment theoretical trajectory, or pre-theoretical trajectory. This is the path that the computer model says that ODIN should follow with the given theory and hydrodynamic model. The third is the trajectory ODIN would follow in our model given the actual voltage applied to the thrusters in the pool. We will refer to this trajectory as the post-experiment theoretical trajectory, or post-theoretical trajectory. The post-theoretical trajectory is of importance because it highlights inaccuracies we may have in our model that are not apparent by comparing just the pre-theoretical and experimental trajectories. In the experimental trajectories, we have smoothed the depth data due to sensor noise.

Pure Heave

For this experiment, we conducted a closed loop dive to approximately 1.5 m and stabilized the vehicle there. This stabilization is an initialization sequence which starts every experiment. We use 1.5m since this is roughly 2.5 diameters of our vehicle and thus we do not have to consider Froude-Krylov forces or wave damping forces since we are deeply submerged. Also at 1.5m the vehicle is still easily accessible and allows for plenty of depth below with which to run experiments. When the vehicle was stabilized (i.e. no obvious visual motions) we then prescribed a 2.3m pure dive in open loop with feedback control only in pitch and roll. The results are shown in Figure 2 where the solid line is the experimental trajectory, the dotted line is the pre-theoretical trajectory and the dashed line is the post-theoretical trajectory. This legend is the same for all subsequent graphs. From Figure 2, we see that we applied a 30 N force along the z-axis for approximately 5 seconds. The experimental trajectory deviates from the two theoretical by undershooting the prescribed depth, and then rising too quickly after completion of the dive. This is caused by an underestimate of the drag coefficient in heave along with inaccuracies in the thruster model. For the thrusters, we may prescribe a 30 N thrust, but we may get a little more or less. Note that the roll and pitch are not steady which is due to the thruster model, but both are within 2° which is not noticeable when viewing the experiment.

Pure Surge

For this experiment, after initial stabilization, we prescribed a 4m pure surge in the negative direction. The trajectory was applied in open loop with only a yaw feedback loop running. This is necessary due to the recirculation current in the pool and the uniqueness of the thrusters. Without this feedback loop, a straight line is nearly impossible.

In the x-direction, we see that all three trajectories are...
very similar and that we end up about 1m short in the experimental trajectory. This is most likely the result of an underestimate of the drag coefficient in the surge (sway by symmetry). The y-direction shows a deviation even though we did not prescribe any at all. This is due to the yaw controller adjustments. For the \( \tau_3 \) component we are applying less than 1 N thrust for a yaw compensation of 10°.

### Concatenation

Now, we are interested in our ability to combine the above pure motions. In order to test this, we prescribe the trajectory of a rectangle with base 4m and height 3m as our goal trajectory. To achieve this motion, we consider two techniques. First, through the use of ODIN we are very familiar with her operation, and know that given an applied thrust of 15N in surge for 15 seconds, ODIN moves about 4m. Likewise for the same thrust applied for 10 seconds, ODIN moves about 3m. Now, remembering that we are in an unbounded real fluid, we create our rectangle trajectory by applying \( F_1 = 13N \) for 15 sec., \( F_2 = 13N \) for 12 sec., \( F_1 = -13N \) for 15 sec., \( F_2 = -13N \) for 12 sec. At the corners and the end of the trajectory, we give 3 seconds of no applied forces to ensure that ODIN has come to a complete stop before beginning the next leg. This thrust strategy is run through our numerical model to produce the pre-theoretical trajectory, which most times is not a perfect rectangle. This is a purely ad hoc method, yet we achieved descent results in many experiments. An example is shown in Figure 4. ODIN clearly completed the rectangle, but made a scaled down version, and ended up about 1m from where she should have. The post-theoretical trajectory matches the pre-theoretical quite well if you account for a yaw offset at the beginning of each leg. Although ODIN follows a straight line, it may not be in the prescribed direction. This aspect is under investigation, but as above, the angular displacement along each leg is on the order of 15° which can be obtained with as little as 1N of \( \tau_3 \) thrust. Another technique to realize the rectangle is to insist that the vehicle comes to rest at each corner to stabilize before attempting the next leg. This type of trajectory takes longer to realize since we apply a deceleration phase into each leg of the rectangle and stabilize ODIN longer at each corner, but it creates a better rectangle than the earlier described technique. Here we actually figure out the exact thrust that our model predicts to move a given distance with the assumption that the initial and final velocity of the vehicle is zero. We will analyze this experiment a bit closer. To begin, we conducted the same initialization as previous, and then prescribed a 5m by 4m rectangular motion seen in Figure 5. First note that each of the three trajectories is a rectangle with sharp corners. Again ODIN creates a scaled down rectangle from that prescribed. Note that at the first two corners there are loops in the trajectory, this further emphasizes the underestimation of the drag in surge and sway since the vehicle actually moved backward during its deceleration phase when we predicted it to have zero velocity at the end of the phase. Also, we see again the post-theoretical trajectories is more or less straight and only differs from the pre-theoretical trajectory by an angular disturbance along each leg. Further investigation will reveal the effect of the drift in the pool, which was first assumed to be negligible.

### CONCLUSIONS

In this paper we analyze the implementation of open loop trajectories onto a test bed AUV. We consider two pure motion trajectories and analyze the implementation and the model used for computation and theoretical modeling. We have found that our angular disturbances are minimal, with the most problem coming from the yaw direction. We have been able to show that our ability to follow pure motion trajectories is excellent even with a feedback controller only working in yaw. This is a direct relation to producing a good model for our vehicle. These experiments lead us to investigate the drag relationship more closely. It seems that
we need to slightly increase the drag coefficient of our vehicle. More importantly is the need for an accurate model for our thrusters. Since ODIN’s design is for controllability rather than stability, and we lack resistance and restoring forces in yaw, small differences in the thrusters causes noticeable motions during experiments. We are currently running tests to determine more accurate estimates on the thruster models, which will also give insight as to where our drag estimates need adjustment. Along with this, we also see a need to implement a consolation for the current in the pool which was first assumed to be negligible. We have a descent Lagrangian understanding of the pool drift, but are assessing methods to provide an Eulerian approach which can be easily implemented into our model. With a few minor adjustments, we will achieve an accurate rectangle with both the post-theoretical and experimental trajectories. After this success, we plan to begin implementation and tracking of time efficient trajectories. These trajectories can be quite complex and involve a superposition of many if not all degrees of freedom in each movement. Thus, it is important to validate all models on simple concatenations to trust the results extracted from later experiments.

REFERENCES


