Sensor-Based Formation Control of Autonomous Robotic Vehicles

Ana Cristina Resendes Maia

Instituto Superior Técnico - Institute for Systems and Robotics (IST-ISR)
Lisbon 1049-001, Portugal

November 2013

Abstract—This thesis objective is to produce controllers of both depth and altitude specially dedicated to navigate the subaquatic vehicle, Medusa, in the vertical plan.

To fulfill this goal, first it was necessary to model the vehicle behaviour when moving vertically and identifying the parameters of this model. This was done resorting to physical models and empiric measurements.

After, two types of controllers were designed and simulated: a simpler one based on a linearised version of the vehicle’s model and a more complex one that compensated all the non-linearity of the vehicle allowing precise following. These controllers were subjected to extensive simulation in computer environment and the first and simpler one was tested in sea water.

Based on the results of both simulations and real tests, the controllers can successfully control the Medusa vertical movement and follow, in the second controller case, more complex references.

Index Terms—Medusa, modelling, identification, depth, altitude, precise following

I. INTRODUCTION

The purpose of this work, developed in the MORPH project environment, [3], is to obtain controllers that can navigate an under-water specific vehicle, Medusa, on the vertical plan. This includes finding a model that accurately describes the vehicle vertical movement and identifying all its parameters.

Since the model is obtained, it serves as a foundation to build the controllers that shall follow references in altitude and depth. These references are the timed values of depth (or altitude) at which the vehicle should be at each time instant.

Water covers an extensive parcel of the planet Earth. And what is under the surface is many times inaccessible to human divers, or at least dangerous to achieve, but may also present important resources to mankind. As, by the ONU Law of the Sea, [6], this underwater area is on the jurisdiction of each country, the research in subaquatic exploration has increased over the years.

Through the use of Autonomous Under-water Vehicles, AUV’s, bigger depths are accessible and during more time at each dive. For instance, robots as the Medusa can be programmed to scan an entire harbour at constant altitude, which would reduce costs in the operation while providing reliable data. However, this kind of procedure requires controllers as the ones explained on this paper.

The altitude controller as well as the two depth controllers were designed and tested in simulation, and have proven to be efficient on the following of adequate references in that environment. This results can be used in practice to guide the vertical movement of the Medusa, as further explained in Section V.

In order to design these controllers some modelling work was necessary. The vehicle model was done based on the findings of Fossen, [9] and the parameter identification on the work of Ridao, [5], which constitutes the state of the art in the subaquatic robots model and identification.

An intermediate step was the attainment of the same work, but relatively to the Thrusters. In this area, the most recent developments were presented by Healey, [10], and Whitcomb, [11], which offer several models based on the physic parameters of the system to model.

The remainder of the paper is organized as follows: the frame system used and the initial problem formulation - vehicle used and existing forces - are described in Section II. In Section III, the physical model of the vehicle as well as the identification method and results of the model parameters is introduced. Section IV describes the design of each controller, including the final Simulink block diagrams. The results of both simulations and water tests are presented in Section V. Finally, the conclusions are stated in Section VI, which also refers the future work to be done.

II. PROBLEM FORMULATION

In order either to model the vehicle or to control it, first it is necessary to define the frames to use. In this case there are two important sets of frames: the Body Frame, whose origin is in the vehicle centre of mass, and the Inertial Frame, whose origin is at the water’s surface.

The Body Frame has the:
- X-coordinate pointing to the front of the vehicle ($x_B$);
- Y-coordinate to the vehicle’s right ($y_B$);
- Z-coordinate to the bottom of the vehicle ($z_B$).

The Inertial frame, also known as the Universal frame, has the
- X-coordinate pointing towards East ($x_U$);
- Y-coordinate pointing towards North ($y_U$);
- Z-coordinate pointing towards the ocean floor ($z_U$).

This information can be easily visualized in figure 1.

To further reference in the paper, the table I sums up all the denominations of the:
- movements of the Body frame in the Universal one -
- force;
- velocities in the movements directions - velocity;
- position and orientation of the Body frame in the
 Universal frame - position.

<table>
<thead>
<tr>
<th>Movement</th>
<th>Force</th>
<th>Velocity</th>
<th>Position</th>
</tr>
</thead>
<tbody>
<tr>
<td>zdirection</td>
<td>surge</td>
<td>X</td>
<td>u</td>
</tr>
<tr>
<td>ydirection</td>
<td>sway</td>
<td>Y</td>
<td>v</td>
</tr>
<tr>
<td>arroundx</td>
<td>roll</td>
<td>K</td>
<td>p</td>
</tr>
<tr>
<td>arroundy</td>
<td>pitch</td>
<td>M</td>
<td>q</td>
</tr>
<tr>
<td>arroundz</td>
<td>yaw</td>
<td>N</td>
<td>r</td>
</tr>
</tbody>
</table>

TABLE I: Coordinates and orientation of the Robot Frame in the Inertial Frame

As observable in Figure 2, the Medusa robot is an assem-
bling of two acrylic cylinders locked together in an aluminium
frame. Apart from the thrusters, echo-sound and mast, which
are outside the cylinders, and the depth cell, which is in
direct contact with the water though from inside the extremity
aluminium caps, all the remaining components are protected
in vacuum inside the tubes. This includes the PC in the upper
cylinder and the battery sets on the bottom one.

The communications between the vehicle and the console
are done through wi-fi. However, when the vehicle is sub-
merged, this means that an antenna needs to be attached to an
external float and connected to the vehicle through a cable.

Using the notation from Table I and adapting the circum-
stances to the Medusa case, a model of the robot can be
computed as well as controllers to navigate it.

III. MODEL AND PARAMETER IDENTIFICATION

A fundamental physics law states that the sum of all forces
applied to a body are equal to the total mass of that body times
its acceleration. Concerning the particular hydrodynamics of
the vertical movement, the forces applied to a body are quite
similar to what was accounted for in [1]:
- Weight (W) - downwards gravitic force;
- Buoyancy (B) - upwards force from the fluid;
- Thrust (T) - force applied by the thrusters;
- Damping (D(\dot{ij}))- sum of the drag, lift and friction
 between water and vehicle surface;

- Coriolis (C) - force generated by the uniform rotation
of a frame.

However, the Coriolis Force is negligible, once that, for the
study case, the rotation of the frames are minimal. Moreover,
also the damping force can be reduced to its vertical linear
and quadratic drag terms (\(-Z_w \ddot{z} \) and \(-Z_{w|w} |\dot{z}|\ddot{z})\), considering
that this is the predominant effect.

Another important observation is that the total mass of the
vehicle, in hydrodynamics, is the sum of its mass (m)
with an added mass (Ma), resultant from the deflection of a certain
volume of fluid with the body movement in it.

An assumption accounted for, that was afterwards con-
firmed, is that the pitch is nearly null. This means that the
velocity in heave \(\dot{w}\) is equal to the first derivative of the depth
in order to time \(\dot{z}\).

Assembling these statements in one unique equation that
describes the vertical movement of the vehicle, it results on
Equation 1.

\[ W + B + T + (\dot{-Z_w \ddot{z}} - Z_{w|w} |\dot{z}|\ddot{z}) = (m + Ma) \ddot{z} \]  
(1)

Dividing all the previous equation by the total mass of the
vehicle and assembling the Weight and Buoyancy as a resultant
external force (W+B), it results on Expression 2 which can be
simplified as shown in Equation 3.

\[ \alpha \dot{z} + \beta |\dot{z}| + \gamma T + \delta = \ddot{z} \]  
(3)

The equation 3 was then introduced in a Least Squares
algorithm, [2], similarly to what was done in [5], in order to
identify the model parameters: \(\alpha, \beta, \gamma\) and \(\delta\). These parameters
were assembled in a vector and the matricial expression 4 was
implemented with data from real in-water trial, at EMEPC pool
in February 15th, 2013.
\[ y = H\theta \]
\[
\begin{bmatrix}
\dot{z}_1 \\
\dot{z}_2 \\
\vdots \\
\dot{z}_N
\end{bmatrix} =
\begin{bmatrix}
\dot{z}_1 & \dot{z}_1 & T_1 & 1 \\
\dot{z}_2 & \dot{z}_2 & T_2 & 1 \\
\vdots & \vdots & \vdots & \vdots \\
\dot{z}_N & \dot{z}_N & T_N & 1
\end{bmatrix}
\begin{bmatrix}
\alpha \\
\beta \\
\gamma \\
\theta
\end{bmatrix}
\]  
(4)

As the trial only produce the depth values and the rotation velocity of the thrusters, some further computations were needed. For instance, the depth, sampled at 5 Hz, was derivated once to obtain the velocity, \( \dot{z} \), and twice for the acceleration, \( \ddot{z} \). The thrust was computed inputting the rotation velocity, \( \omega \), in the Equations 5.

These equations were obtained, fitting a quadratic curve to groups of collected data for the functioning points of the thrusters when the thrusters were pushing the vehicle up, for negative commands, or down, for positive commands. All the data was collected in Bollard Pool (the vehicle without surge, sway or rotational movement) and assumed to be similar to the one verified when the vehicle was moving.

\[
T_{up} = -8.9241 \times 10^{-3} \omega^2 - 2.2773 \times 10^{-3} \omega |N|, \omega < 0
\]
\[
T_{down} = 8.4661 \times 10^{-3} \omega^2 + 18.9623 \times 10^{-3} \omega |N|, \omega > 0
\]  
(5)

These curves can be seen in image 3, being the blue curve associated with the vehicle going upwards and the red one representing the downwards movement.

![Fig. 3: Command-to-Thrust transfer function - Vehicle moving upwards is blue and downwards red](image)

The results obtained from this procedure are shown in Table II, for the vehicle nominal mass of 30 kg.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Physical Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
<td>-0.1300</td>
<td>( Z_{\omega} )</td>
<td>-6.4369</td>
</tr>
<tr>
<td>( \beta )</td>
<td>-2.4338</td>
<td>( Z_{\omega</td>
<td>\dot{\omega}} )</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>0.0202</td>
<td>( M_\ddot{z} )</td>
<td>-19.4311 [kg]</td>
</tr>
<tr>
<td>( \delta )</td>
<td>-1.1173</td>
<td>( W + B )</td>
<td>-1.0971 [N]</td>
</tr>
</tbody>
</table>

**TABLE II: Results from the Least Square algorithm on the Parameter Identification**

The delta factor only refers to one specific trial, as this value changes with the type of water (salted or fresh), the water temperature and the number and kind of sensors attached to the vehicle. However, this parameter will be neglected on the controller design, as the integral part of the controller learns and compensates its value.

### IV. Controller Design

Three controllers were developed in this project scope: a simple depth controller, based on the vehicle linearised model, Subsection IV-A; a more complex depth controller that compensates the model non-linearities, allowing precise following of complex references, Subsection IV-B; and an altitude controller, based on an adapted version of this last depth controller to altitude references, Subsection IV-C.

To implement this controller, some general blocks were required. These blocks performed general tasks necessary for each one of the controllers.

The Kalman Filter, [8], is used in every implementation, as an observer, in order to remove the noise and outliers added to the measurements by the sensors and estimate the vertical velocity. It uses the Medusa’s dynamics in state space, in form of matrices A and B. For the precise following case, the matrix L is also used to compensate the non-linearities, expression 6. This filter output is a weighted average between the estimation of the vehicle’s position and the measurements from the sensors.

Its states are the vertical position of the robot, its velocity in heave and the delta factor.

\[
\dot{x} = Ax + Bu + L
\]
\[
\begin{bmatrix}
\dot{z} \\
\dot{\delta}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
z \\
\dot{z}
\end{bmatrix} +
\begin{bmatrix}
0 \\
\gamma
\end{bmatrix}
\begin{bmatrix}
T + 0 \\
\beta \dot{z}
\end{bmatrix}
\]  
(6)

So, as mentioned, the simpler depth controller uses only the matrices A and B, whereas the precise following uses also the non-linearities. This filter still needs to be discretized in order to be used, as the data provided by the robot is not continuous in time. This discretization results on the equation 7, bellow.

\[
\dot{x}_{k+1} = Ax_k + Bu_k + L
\]
\[
\begin{bmatrix}
\dot{z}_{k+1} \\
\dot{\delta}_{k+1}
\end{bmatrix} =
\begin{bmatrix}
1 & Ts & 0 \\
0 & 1 + \alpha Ts & Ts
\end{bmatrix}
\begin{bmatrix}
z_k \\
\dot{z}_k
\end{bmatrix} +
\begin{bmatrix}
0 \\
\gamma Ts
\end{bmatrix}
\begin{bmatrix}
0 \\
\beta \dot{z}_k |\dot{z}_k| Ts
\end{bmatrix}
\]  
(7)

When adapting this filter to altitude control, some signal changes are needed, due to the frame changes further explained in subsectionIV-C. The drag factors remain, as its signal is always symmetric to the movement, and only the delta and thrust signals change. This happens because upwards commutes from being the negative direction to be the positive one. The new equations system is reproduced in expression 8.

\[
\dot{x}_{k+1} = Ax_k + Bu_k + L
\]
\[
\begin{bmatrix}
\dot{z}_{k+1} \\
\dot{\delta}_{k+1}
\end{bmatrix} =
\begin{bmatrix}
1 & Ts & 0 \\
0 & 1 + \alpha Ts & -Ts
\end{bmatrix}
\begin{bmatrix}
z_k \\
\dot{z}_k
\end{bmatrix} +
\begin{bmatrix}
0 \\
\gamma Ts
\end{bmatrix}
\begin{bmatrix}
0 \\
\beta \dot{z}_k |\dot{z}_k| Ts
\end{bmatrix}
\]  
(8)
Another essential block is the reference filter, which smooths the references given in depth and computes its first and second derivatives. The smoothing of the depth reference is specially important when it is originally a step, as no vehicle can follow that kind of reference and change its position instantaneously. The results of this filter are presented on image 4.

The remaining signals are essential to the controller computation as further explained in each subsection below.

**A. Simple Depth Controller**

This controller is a PID (proportional integral and derivative controller) based on a linearised version of the Medusa model described previously, and whose architecture is shown in image 6. This new version of the model discards its non linearities, more precisely the delta coefficient and the quadratic term of the model, though the delta term is compensated through the integral part of the controller.

The linearised vehicle model in state space, necessary to compute the gains uses three states and an input, the thrust, as shown in expression 10

\[
\begin{align*}
\dot{x} &= A\dot{x} + Bu \\
\dot{z} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & \alpha \end{bmatrix} \begin{bmatrix} f_z \\ z \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \gamma \end{bmatrix} T
\end{align*}
\]

The gains of this controller were computed through LQR, [7], with the error matrices being a normalization of the errors (by the maximum error admitted for each term) times a weight to calibrate the controller. The objective of this algorithm is to minimize both the state variations, stabilizing the vehicle behaviour, and the actuation, allowing less consume of energy.

The maximum error allowed was 2.5 meters in distance, 0.5 meters per second in velocity, 2.5 x 5 meters second in the integral term and 1% in the command. After dividing the errors by this normalization, the coefficients are then multiplied by the weights which are tuned to allow more rapid and/or stable responses. This tuning is done through simulation in ideal conditions - without noise or disturbances, and using a step as reference.

The weights and resultant gains that were found to optimize the controlling activity are presented in expression 11.

\[
\begin{align*}
&\begin{array}{ccc}
& w_i &= 10 & \rightarrow & \bar{k}_i &= 3.3645 \\
& w_p &= 4 & \rightarrow & \bar{k}_p &= 17.1174 \\
& w_d &= 0.6 & \rightarrow & \bar{k}_d &= 48.2326
\end{array}
\end{align*}
\]

**B. Precise Following Depth Controller**

When facing the results from the previous depth controller, it was found that the discarding of the quadratic term had undesirable and strong effects, related with the stability of the controller. Moreover, in complex references with several steps, the previous controller is unable to follow them.

To compensate these effects, a new controller was designed, based on all the vehicle model and the non-linear control, which allows precise following of the references. The first step of this design was to define the error, equation 12.

\[
\begin{align*}
& \hat{e} = z - r \\
& \dot{\hat{e}} = \hat{z} - \dot{r} \\
& \ddot{\hat{e}} = \ddot{z} - \dddot{r}
\end{align*}
\]

After defining the error, the term \( \dddot{z} \) was substituted by the entire vehicle model. The second derivative of the error was replaced by its corresponding expression from the controller gains in the feedback loop, equation system 13.
\[\ddot{e} = \alpha \dot{z} + \beta |\dot{z}| + \gamma u + \delta - \ddot{r}\]
\[\ddot{e} = -K_1 e - K_2 \dot{e} - K_3 \int e\]  
(13)

This equation system was then solved in order to the actuation variable, the thrust, and the delta factor. As done in the first controller, the delta factor was neglected, once it is already compensated by the integral component of the controller. The resulting expression 14 was then implemented in the Simulink Block Diagram illustrated in image 7.

\[T = \frac{1}{\gamma} (K_1 (r-z) + K_2 (\dot{r} - \dot{z}) + K_3 \int (r-z) - \alpha \dot{z} - \beta |\dot{z}| + \ddot{r})\]  
(14)

As it can be seen, the controller now includes the terms from the non linearity and also the second derivative of the reference. Both these components enable the vehicle to have a smoother and stable movement and to follow the references with less error, even when dealing with complex references.

The gains, as done previously, were computed with the LQR algorithm, but with some differences. The drag terms were not seen as states, but instead as entries of the system. This caused the system to be redefined in terms of state space as in expression 15.

\[\dot{x} = A\dot{x} + Bu\]
\[
\begin{bmatrix}
z \\
\dot{z} \\
\ddot{z} \\
\int z
\end{bmatrix} =
\begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
\alpha \dot{z} \\
\beta |\dot{z}| \\
\gamma T
\end{bmatrix}
\]  
(15)

Also the maximum errors allowed changed: the maximum distance error changed to 1 meter, the maximum velocity error 0.5 meters per second and the maximum integral error 0.5 meters for 5 seconds.

After extensive tuning, again in ideal conditions, the resulting weights and gains used are stated bellow, expression 16. The gains to the entries of the systems are all 10.

\[w_i = 0.08 \rightarrow k_i = 0.0327\]
\[w_p = 0.3 \rightarrow k_p = 0.3015\]
\[w_d = 20 \rightarrow k_d = 1.2387\]  
(16)

![Fig. 5: Complete simulation assembled.](image)

![Fig. 6: Simple Depth Controller Schematic](image)
C. Precise Following Altitude Controller

As a result of the depth precise following controller good performance, the same algorithm was applied to control altitude. This adaptation required a change of frame and, consequently, a new fit of both vehicle model (reflected on the Model block and the Kalman filter block) and the controller.

The new frame adopted for this controller design is shown in image 8. The new referential has its origin at the ocean bottom and the Z-axis pointing towards the surface.

![Image 8: Frame for Altitude Control Design](image)

From the schematic in image 8, the conclusions mathematically expressed in expression 17 can be withdrawn, with some assumptions. The main assumption is that the ocean floor is approximately plan, not having any kind of topography or that it changes very slowly when the vehicle moves in surge. In either case the first and second derivatives of \( f(x, y) \) are null all the time.

These expressions can then be replaced on the original vehicle model, creating a new description for the robot movement in this new frame, equation 18.

\[
\begin{align*}
  f &= z + h \leftrightarrow h = f - z \\
  0 &= \dot{z} + \dot{h} \leftrightarrow \dot{h} = -\dot{z} \\
  0 &= \ddot{z} + \ddot{h} \leftrightarrow \ddot{h} = -\ddot{z} \\
  \ddot{h} &= -(\alpha \dot{z} + \beta \dot{z} | \dot{z} + \gamma T + \delta) = \alpha \dot{h} + \beta \dot{h} | \ddot{h} - \gamma T - \delta
\end{align*}
\] (17)

Similarly to what was done in the precise following depth control, and establishing the error as shown in equation 19, the equalities from expression 20 surge.

\[
\begin{align*}
  e &= h - r \\
  \dot{e} &= \alpha \dot{z} + \beta \dot{z} | \dot{z} + \gamma u + \delta - \ddot{r} \\
  \ddot{e} &= -K_1 \dot{e} - K_2 \ddot{e} - K_3 \int \dot{e}
\end{align*}
\] (20)

When the two equations are equalled and solved in order to the actuation \( T \), the controlling expression to this new model surges as equation 21.

\[
T = u = -\frac{1}{\gamma} (\ddot{r} - \alpha \ddot{h} + \beta | \ddot{h} + k_p (r - h) + k_d (\dddot{r} - \ddot{h}) + k_i \int (r - \int h))
\] (21)

This expression is then translated to a block diagram in Simulink as done in the Precise Following Depth Controller. The resulting schematic is very similar to the one shown in figure 7.

The gains of this controller are the same as the precise following for depth control once they were obtained by the same algorithms, using the same weights and similar matrices. The only change in this field was, similarly to what happened in the Kalman filter case, was the inversion of the signal of the \( \gamma \) which is done directly in the input and not in the matrix, resulting in the expression 22.

\[
\begin{align*}
  \ddot{x} &= A \dot{x} + B u \\
  \begin{bmatrix}
  \dot{z} \\
  \dot{\dot{z}} \\
  \dot{\ddot{z}}
  \end{bmatrix} &=
  \begin{bmatrix}
    \dddot{z} \\
    0 \\
    0
  \end{bmatrix}
  +
  \begin{bmatrix}
    0 & 1 & 0 \\
    0 & 0 & 1 \\
    0 & 0 & 0
  \end{bmatrix}
  \begin{bmatrix}
    \alpha \dot{z} \\
    \beta \dot{z} | \dot{z} \\
    -\gamma T
  \end{bmatrix}
\end{align*}
\] (22)

V. SIMULATIONS AND TEST RESULTS

A. Simulations in Matlab

All the controllers were subjected to extensive simulation in order to detect incorrect behaviours that may happen in reality.

![Image 7: Implementation of the control function 14](image)
These trials include references as steps or combinations of steps changing in quick succession, performing each trial with and without noise and with or without disturbances. These disturbances assumed step form, simulating a force that suddenly starts and maintains at constant value, or pulse form, as short duration forces such as pushes. Also a constant bias on the buoyancy factor was simulated.

The simpler depth controller revealed to be slow to converge in case of disturbances, but eventually follows the reference if it is a step. This happens due to the filters being tuned to a long settling time, which provides more stable results.

On the other hand, if the depth reference is more complex than a succession of steps, the controller is not quick enough to keep up, and eventually diverges, as shown in picture 9.

The precise following however, is able to cope with this kind of references as seen in image 10. Despite the fact that the oscillations when in steady state are bigger that in the previous controller. Relevant oscillations in depth (more than 10 cm of amplitude) are not registered.

The gains presented on Section IV were tuned so that a fluctuation of the model parameters was allowed, not compromising the normal behaviour of the controller. The maximum fluctuation considered acceptable was more or less 50% of each parameter.
In simulation, the worst case scenario happens when all the parameters are at 50% of what was estimated. Even for this huge discrepancy between reality and estimation, the controller was able to react and behave, even though the oscillations in steady state increased to their double. An example of this behaviour is presented in image 13.

**Fig. 12:** Result of simulations of the altitude controller with the ocean floor with topology
- Dark Blue - Altitude Reference [m];
- Green - Simulated Altitude [m];
- Red - Thrusters Commands x0.1 [% rpm];
- Blue - Vertical Velocity [m.s\(^{-1}\)]

In simulation, the worst case scenario happens when all the parameters are at 50% of what was estimated. Even for this huge discrepancy between reality and estimation, the controller was able to react and behave, even though the oscillations in steady state increased to their double. An example of this behaviour is presented in image 13.

**Fig. 13:** Result of simulations of the precise following depth control with 50% mismatch between real parameters of the robot model and the simulated ones.
- Dark Blue - Depth Reference [m];
- Green - Simulated Depth [m];
- Red - Thrusters Commands x0.1 [% rpm];
- Blue - Vertical Velocity [m.s\(^{-1}\)]

B. Real Trial

Due to vehicle malfunctions, only the simpler depth controller could be tested in sea water. Some of the results from this trial are reproduced on figure 14, which consisted in two dives with a pause in between so that the robot may resurface. The vehicle when submerged, should maintain itself at 1 meter of depth, with or without horizontal movement. This trial was executed at July 15th, at EXPO dock, in Lisbon.

The small oscillations, seconds after the robot stabilises at the 1 meter depth, are the consequence of a pitch caused by the initial acceleration of the horizontal movement. This behaviour is, however, quickly compensated, and the vehicle maintains its referenced depth.

**Fig. 14:** Testes performed at EXPO dock with a Medusa Diver, July 15th
- Dark Blue - Thrusters Commands x0.1 [% rpm];
- Red - Depth Readings [m]

The overshoot of 20 cm is a direct consequence of the non-existence of compensation of the non-linearities of the vehicle model, disregarded on the design of this controller.

Although the precise following controllers were unable to be tested in real conditions, the similarities between the behaviour of the simulations and reality of the controller tested lead to the conclusion that these controllers should perform as expected.

**VI. Conclusions**

As seen from the results presented on the previous Section, the simpler controller works well with simple references as steps, but does not respond adequately when subjected to quickly changed references. To that kind of more complex references, it is necessary the precise following controllers, which also provide more quick reactions, specially in case of disturbances.

Although only the simpler controller was tested in water, due to vehicle malfunctions and time constraints. Its similarity with the simulations suggest that the same pattern will be observed on the remaining two controllers.

It was also observed that the chosen gains for the controllers allow a fluctuation of the model parameters until 50%, which gives a certain degree of relaxation when doing minor updates to the vehicle.

Another point to highlight is that, for the study case, the functioning point in steady state is at 10% of the total capacity of the thrusters. This means that, as in this area the slope of the function Thrust-to-Command is very high, so that small changes in the thrust correspond to big fluctuations of command. This is the main reason why there are oscillations on the command when the vehicle maintains itself at constant depth. However, due to the small amplitude of this oscillation, it does not represent a risk to the thrusters live span, reason why it was considered acceptable.

There is still, work to be done. For instance, updating sensors (a sensor that directly provides the velocity would be helpful) or researching new algorithms that automatically calibrate the model parameters when detected inconsistency of the controller. Also algorithms in case of the altitude controller that deal with the terrain geography and other ways...
of communication between the submerged vehicle and the operating console. It would also be advisable to test the precise following designed controller in sea water, both for depth and altitude.

ACKNOWLEDGEMENTS

I could not have written this thesis and accomplished this results without the help of many. Not wanting to minimize the contribution of everyone, some names needed to be highlighted due to their essential role in this process.

This thesis would not have happened without Professor António Pascoal, whose vast expertise in the area guided me through difficult situations, always directing me in the right direction. To the entire project team, specially to João Botelho, Jorge Ribeiro, Luís Sebastião, Manuel Rufino, Miguel Ribeiro and Filipe Morais, by their help and attentive suggestions during the work development as well as the dissertation writing.

REFERENCES