Force Control of Electro-Hydraulic Actuators In An Underwater Fish-Like Vehicle

by

Jonah T. Peskin

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degrees of Bachelor of Science in Electrical Science and Engineering and Master of Engineering in Electrical Engineering and Computer Science at the MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1999

© Jonah Peskin, 1999. All rights reserved.

The author hereby grants to MIT permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part, and to grant others the right to do so.

Author: ........................................... Department of Electrical Engineering and Computer Science May 24, 1999

Certified by: ........................................... Dr. Jamie Anderson Group Leader, Vehicle Systems, Charles Stark Draper Laboratory Thesis Supervisor

Certified by: ........................................... Prof. Jean-Jacques Slotine Professor, Massachusetts Institute of Technology Thesis Supervisor

Accepted by: ........................................... Prof. Arthur C. Smith Chairman, Department Committee on Graduate Theses
Abstract

The Vorticity Control Unmanned Undersea Vehicle (VCUUV) is a robot that mimics fish-like swimming motions. This is accomplished with a semi-flexible tail structure that is actuated by four, independent hydraulic cylinders. Control research on the VCUUV has thus far involved position feedback to precisely place the vehicle's tail. It would be desirable, for several reasons, to control the force applied by each cylinder.

With force control, a spring-like compliant tail could be simulated with programmable spring constants. This would allow investigation as to whether fish-like motion can be achieved with simple mechanical springs. Additionally, compliance of the robotic tail would allow for safer operation since the tail would exhibit some "give" when in contact with its environment.

The VCUUV is equipped with force sensors that can measure the force exerted by each cylinder. Various techniques of linear and non-linear control were applied to the problem of hydraulic force control. The system plant and control systems are described, and the performance of the system is analyzed. This research describes the first known application of nonlinear sliding control to a fish-like, hydraulic robot with multiple degrees of freedom.
Acknowledgments

I would first like to thank my parents. In everything that I have pursued, they have given me all of the support I could have wished for. Their guidance and understanding have been an invaluable part of my education.

I also thank Karen Andrews, whose emotional support has made my thesis, and everything in the past several years, better in all ways.

The person most directly responsible for my pursuit of electrical engineering would be Dr. David Cyganski. His teachings and enthusiasm inspired my interest in engineering.

For my opportunity to work on such an exciting thesis project, I must thank project leader Dr. Jamie Anderson, who has also been a great person to work with.

I also thank Dr. Jean-Jaques Slotine for his valuable guidance and patience as a wonderful thesis advisor.

Over my past two years on the VCUUV project, there were many coworkers who have made work seem more like fun. For the memories, I thank Mark Little, Peter Kerrebrock, Chris Gadda, Paul Marquardt, Tom Trapp, David Korka, Alan DiPietro, Rusty Sammon, Long Phan, Will McFarland, Mike Sherback, Mohan Gurunathan, Jamie Cho, Mike Socha, Ryan Norris, and John Hoctor. Most of all, Bill Kaliardos made the lab a fun place to be whenever he was around. I also have Bill to thank for convincing me to finish my thesis in a timely manner. This thesis would not have been possible without the countless hours of work of those mentioned above.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Research and Development #99-927.

Publication of this thesis does not constitute approval by Draper Labs of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Permission is hereby granted by the Author to the Massachusetts Institute of Technology to reproduce any or all of this thesis.

\(\text{(author's signature)}\)
4 Nonlinear Control

4.1 System Model ........................................... 43
4.2 Sliding Controller Design ................................. 46
4.3 Parameter Bounds ....................................... 49
4.4 Controller Performance in Simulation .................. 52
4.5 Sliding Control In Practice ............................. 54

5 Conclusions .............................................. 58

5.1 Summary ................................................ 58
5.2 Future Work ............................................ 59

A Glossary of Terms ........................................... 62

B Matlab Script Files ......................................... 64
List of Figures

1-1  Vorticity Control Unmanned Undersea Vehicle (VCUUV) . . . . . . . . . . . . 13
1-2  Vehicle tail—CAD model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
1-3  Hydraulics Schematic . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 18

2-1  Servovalve Orifices . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 21
2-2  Hydraulic Cylinder Force Diagram . . . . . . . . . . . . . . . . . . . . . . . . 23
2-3  Inertial Load on Piston (not to scale) . . . . . . . . . . . . . . . . . . . . . . . 26
2-4  VCUUV Tail Mass Ascribed to Each Link . . . . . . . . . . . . . . . . . . . . . 28
2-5  Bode Plot of Open Loop Plant For Link #3 . . . . . . . . . . . . . . . . . . . . 30

3-1  Experimental Bode Plot of Open Loop System, Link #3 . . . . . . . . . . . . 33
3-2  Block Diagram of Plant and Compensator With Feedback . . . . . . . . . . . 34
3-3  Open Loop Bode Plot of Compensated System, Link #3 . . . . . . . . . . . . 35
3-4  Sinusoidal Tracking at 0.25 Hz, Link #3 . . . . . . . . . . . . . . . . . . . . . 36
3-5  Sinusoidal Tracking at 0.5 Hz, Link #3 . . . . . . . . . . . . . . . . . . . . . . 37
3-6  Sinusoidal Tracking at 1.0 Hz, Link #3 . . . . . . . . . . . . . . . . . . . . . 37
3-7  Sinusoidal Tracking at 1.5 Hz, Link #3 . . . . . . . . . . . . . . . . . . . . . 39
3-8  Sinusoidal Tracking of Link #3, Link #2 at 0.5 Hz . . . . . . . . . . . . . . 39
3-9  Sinusoidal Tracking of Link #3, Link #2 at 1.0 Hz . . . . . . . . . . . . . . 41
# List of Tables

2.1 Servovalve flow constants as per manufacturer’s specifications 22
2.2 Tail Structure Mechanical Constants 27
2.3 Parameters of the VCUUV Tail Free-Flood Water Ascribed to Each Link 28
2.4 Tail Structure’s Moment of Inertia As Seen by Each Hydraulic Link 29
2.5 System Parameters for Link #3 30
Chapter 1

Introduction

1.1 Background

It has long been known that aquatic animals exhibit highly efficient swimming propulsion [8, 6]. The incredible efficiency and maneuverability of fish has inspired much research in hydro-propulsion. More recent work has focused on recreating fish-like propulsion through robotics. The MIT RoboTuna project by Triantafyllou and Barrett [14, 3] concluded with promising results; significant drag reduction was observed in a robotic tuna fish.

To verify the results of Barrett, and to learn more about fish-like propulsion, the Charles Stark Draper Laboratory initiated a similar study in which a robotic tuna was to be designed, built, and analyzed for propulsive efficiency. RoboTuna was purely a scientific study in drag reduction that constrained the fish's motion by dragging it through a testing tow-tank. In contrast, Draper Lab's Vorticity Control Unmanned Undersea Vehicle (VCUUV) was intended to investigate both scientific and practical aspects of fish-like motion. Thus the VCUUV was designed to be a fully autonomous, free-swimming vehicle. Now in its second year of operation, the VCUUV's mechanical, electrical, and software design is effectively complete. A cutaway model of the vehicle
is shown in Figure 1-1.

![Figure 1-1: Vorticity Control Unmanned Undersea Vehicle (VCUUV)](image)

1.2 Previous Work and Thesis Motivation

Presently, the VCUUV swims by moving each tail link in a predefined, traveling sinusoid as described by Barrett [3]. The vehicle achieves this motion with four independent linkages as shown in Figure 1-2. The angle of each link is directly related to the linear extension of a hydraulic piston, thus precise position control of each piston is required. This has been accomplished by Gadda [5], and fish-like motion in both air and water has been demonstrated successfully. Gadda’s control system involved modeling each link as independent, linear systems. This allowed for application of traditional linear feedback control techniques for single-input, single-
output (SISO) systems. The system performance was good and allowed the vehicle to achieve almost every desired swimming motion.

![Vehicle tail CAD model](image)

Figure 1-2: Vehicle tail—CAD model

It is not fully fair to treat each link as an independent system. Clearly the motion of one link will cause various forces to be exerted on other links. Because of this, a more advanced, multivariable system model and controller was developed by Trapp [13]. This controller was a theoretical controller that demonstrated excellent performance in simulation, but due to time constraints has not yet been tested on the actual vehicle.

Positional feedback control is now fairly well understood on the VCUUV, however, there are drawbacks to purely positional control. One is that the robot is not sensitive to the forces that it exerts on its environment. Since the vehicle often operates in the presence of humans, it is easy to imagine the robot exerting excessive forces on a person and causing serious injury. It is equally possible for the robot to damage itself
by exerting excessive forces on its environment.

One way to solve this problem would be to install mechanical springs at each pivot point in the tail. This approach would effectively soften the motion of the tail, but has several drawbacks. First, this would require major mechanical redesign and rebuilding. Second, any change in the spring constant of a link would likely involve disassembling the mechanics and installing a new spring.

A better alternative would be to simulate a virtual spring using force feedback control. On the VCUUV, each hydraulic cylinder has a force sensor in series that measures the force exerted by the cylinder. If this sensor is used for feedback, this force can be controlled. One can then create a virtual spring by setting the force exerted by the cylinder to be proportional to the cylinder’s displacement from a reference point. By changing this reference point, there still exists effective position control. One can then program the spring constant of each virtual spring to control the compliance of each link.

It may also be that spring-like links are conducive to efficient fish-like motion. Since real mechanical springs store and restore energy efficiently, their use might improve the VCUUV’s swimming efficiency by demanding less hydraulic effort. Using virtual springs will allow exploration of various theoretical springs to find those that yield the best theoretical efficiency.

Force control using electric motors is a topic that has been thoroughly researched [15]. Because hydraulic pistons are nonlinear, their force control has been more elusive. Linear controllers have been shown to be fairly effective and robust in hydraulic force control [7]. However, high-performance hydraulic control remains an active topic in control research [9, 2, 12]. There is presently no research on force control of MIMO,
underwater hydraulic systems such as the VCUUV.

This thesis describes the first work towards implementing force control on the VCUUV. It also describes the first known use of nonlinear sliding control for force control of a hydraulic, fish-like robot with multiple degrees of freedom.

1.3 System Overview

The VCUUV is a scale replica of a yellow-fin tuna fish and is approximately 2.4 meters in length. The pertinent mechanical and electrical features of the vehicle are described here. For much of this discussion, refer to Figure 1-1.

1.3.1 Pressure Hull

The pressure hull is the sealed, forward section of the vehicle. This compartment contains the batteries, the hydraulic power unit (HPU), and all of the electronics to control the vehicle. The VCUUV contains five 12-volt, lead acid batteries which power the entire system. Much of this power is converted, using a pump, to hydraulic pressure by the HPU which provides the energy needed to move the tail link assembly.

1.3.2 Tail Structure

Hydraulic fluid flows to the tail through hoses that exit the rear of the pressure hull. The link assembly is mechanically fixed, by a pivoting joint, to the back of the pressure hull. Two hydraulic fluid hoses, a supply hose and a return hose, go to each hydraulic piston, and each piston is attached to the link assembly as shown in Figure 1-2.

The link assembly is surrounded by a semi-flexible exostructure which is mechanically fixed to the rear of the pressure hull. This structure is comprised of two “spines” above and below the tail linkage. The spines are made of fairly thin, laminate mate-
rial, and so they are free to flex as the tail assembly moves. For structural support, these spines are connected by concentric “ribs” along the length of the tail. At the end of the last tail link, there is a rigid caudal fin.

The assembly and exostructure are free-flooded; all unused space is occupied by water while swimming. This water, however, is confined to the tail of the fish by a neoprene “skin” that stretches over the tail exostructure. This skin is an effective barrier to water flow, thus the water inside the tail effectively becomes added tail mass.

1.3.3 Hydraulics

The hydraulics system is much of the plant in this control system, and as such it is described in detail in Chapter 2, System Model. An overview is given here, along with a schematic diagram in Figure 1-3.

The system is powered by the hydraulic power unit (HPU) which is a differential pressure source. This pressure differential is supplied, in parallel, to each of the four hydraulic pistons on the VCUUV. Between each piston and the HPU lies a single-stage hydraulic servovalve. Each valve operates independently to regulate fluid flow between the HPU and a hydraulic piston.

Each servovalve has a control input that sets the area of two orifices inside the valve. This is done by moving a spool inside the servovalve. By setting the size of these internal orifices, one can effectively control the rate of fluid flow through the servovalve.

The system outputs are the linear position of the hydraulic piston, and the force exerted by the hydraulic cylinder. Position is measured using a differential variable reluctance transducer (DVRT) mounted on the cylinder. Both compression and ex-
tension forces are measured with a strain gauge mounted in series with the hydraulic cylinder.

1.3.4 Controller Hardware

High level decisions on the VCUUV, such as when to turn left versus swim straight, are controlled by an Intel 486 100MHz processor inside the vehicle. A separate DSP processor, the Texas Instruments TMS320C44 is responsible for the feedback control of each hydraulic piston. This processor samples all sensors and updates control outputs at at a rate of 400 Hertz. All input and output data are logged at a rate of approximately 25 Hertz.

For a detailed description of the VCUUV's electronic subsystems and overall software architecture, refer to J. Cho [4]
Chapter 2

System Model

The goal of the system model is to describe the entire system as a linear, single input-single output system. This requires making several tenuous assumptions about the independence and linearity of each hydraulic link, and hoping that feedback picks up the slack in these modeling oversights. This is a common approach in linear feedback systems, and it was successful for Gadda [5] when performing hydraulic position control, thus it is followed here.

The system model first involved the hydraulics to determine the fluid forces exerted on the cylinder. Second, the operating environment was examined to determine what external forces were acting on the cylinder.

2.1 Hydraulic Power Unit

The hydraulic power unit provides a pressure differential on the order of 1000-1800 psi. The HPU has many mechanical components, and even its own controller to regulate the pressure differential. For the purposes of this research, however, the HPU is treated as a programmable, constant source of pressure. In truth, this source varies around the “ideal pressure” by around ± 50 psi., and is controlled by simple
“on-off” control. Experimentally, it has been observed that a pressure drop of 100 psi takes several seconds to occur, while a pumping action of under a second may cause a brief transient in the pressure differential.

While the pressure is dropping, there is a reasonable interval under which to study the performance of any controller. Precise control of the HPU is beyond the scope of this thesis. While it would certainly be a critical part of a high performance hydraulic system, bang-bang control of the HPU is sufficient to study the performance of a force controller.

2.2 Servovalves

Each servovalve operates in parallel off of the differential pressure source. This makes each link hydraulically independent. The spool servovalve operates by regulating the area of two equally sized orifices. These orifices are on the order of at most 2 \( mm^2 \), which is much smaller than any other conduit in the system. As a result, there is a large hydraulic pressure drop across each orifice. The location of these orifices in the servovalve can be seen in Figure 1-3. The other significant source of pressure loss in the system is the hydraulic piston head. These main sources of pressure loss are shown together on the flow diagram in Figure 2-1.

The standard expression for turbulent fluid flow through an orifice, as described by Merritt [10] is

\[
q = K_i \delta \sqrt{\Delta P} \tag{2.1}
\]

Where \( \Delta P(psi) \) is the pressure differential across the orifice, and \( q(\frac{m^3}{s}) \) is the flow
rate through the orifice. $K_t$ is a constant that is fixed by the physical properties of the orifice such as its shape and surface area when fully open. Determining $K_t$ by analysis is unnecessary because it can be determined more accurately empirically. The quantity $-1 < i_b < 1$ is the control system input. It is the fraction of the orifice that is open.

The equation for flow through each valve is shown in Figure 2-1. The quantity $p_u$ is the pressure across the hydraulic piston. This pressure depends largely on the external load which will be examined in Section 2.4.

Because the two orifices are identical in shape and size at all times, they share the same $K'_t$ and $i_b$. Furthermore, by symmetry of the system and by Kirchhoff's laws, the pressure differential across each orifice is $(P_s - p_u)/2$. Thus the flow equation for both orifices is

$$q = K'_t i_b \sqrt{(P_s - p_u)/2}$$

(2.2)

It is slightly simpler to rewrite this equation with a new $K_t$ that absorbs the 2 from the radical:
\[ q = K_t i_b \sqrt{(P_s - p_u)} \]  

(2.3)

\[ K_t = K'_t / \sqrt{2} \]  

(2.4)

This essentially describes the behavior of the servovalve as a single orifice with flow constant \( K_t \). The manufacturer’s specifications indicate this flow constant for each servovalve as shown in Table 2.1. Tail links are numbered starting with the link closest to the pressure hull, link #1.

<table>
<thead>
<tr>
<th>Servovalve</th>
<th>( K_t \left( \frac{m^3}{s \sqrt{psi}} \right) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>( 3.60 \times 10^{-6} )</td>
</tr>
<tr>
<td>2</td>
<td>( 1.80 \times 10^{-6} )</td>
</tr>
<tr>
<td>3</td>
<td>( 8.00 \times 10^{-7} )</td>
</tr>
<tr>
<td>4</td>
<td>( 3.60 \times 10^{-7} )</td>
</tr>
</tbody>
</table>

Table 2.1: Servovalve flow constants as per manufacturer’s specifications

The above equations for servovalve flow rate are non-linear, and do not immediately lend themselves well to a linear model which is the present goal. A linear form can be found for a specific operating region of \([i_b, p_u] \) based on expected loads. This linearization, as done in Merritt [10] and Laval et al. [7] is summarized as follows:

\[ q = K_c i_b + K_d p_u \]  

(2.5)

Where \( K_c \) and \( K_d \) are constants defined as
\[ K_c = K_t \sqrt{P_s - p_u} \]  
(2.6)

\[ K_d = -\frac{K_t p_0}{2\sqrt{P_s - p_u}} \]  
(2.7)

### 2.3 Cylinder Model

The force that we are trying to control, \( F_c \), is the force applied by the hydraulic piston shaft on its environment. Equivalently, we can examine the load applied by the environment on the piston shaft \((-F_c)\). To do so, we shall perform a summation of all of the forces acting on the piston. A diagram of the piston, cylinder, and all variables involved is shown in Figure 2-2.

![Figure 2-2: Hydraulic Cylinder Force Diagram](image)

- \( p_s \): Pressure across the piston head (psi)
- \( q \): Fluid flow rate through system (m³/s)
- \( A \): Surface area of piston head without shaft (m²)
- \( A' \): Surface area of piston head with shaft (m²)
- \( F \): Force exerted by piston on its environment (N)
- \( F_{mem} \): Force measured by load cell (N)
- \( F_f \): Frictional force exerted on piston (N)
- \( x \): Linear extension of piston (m)
The first major assumption is that one end of the hydraulic cylinder is fixed. When the tail is not swimming, this is a fair assumption due the large mass of the vehicle’s pressure hull. When swimming, however, the links nearest the caudal fin are definitely not fixed. Since we have chosen to limit the design to a SISO model, however, we are forced to make this assumption. This assumption also worked for Gadda [5] with position control and so it is used here.

We also assume there is no leakage of hydraulic fluid across the piston head. We neglect the inertia of the hydraulic fluid and its friction against conduit walls. Furthermore, we shall assume that the surface area of each side of the piston head is equal, i.e. \( A_+ = A_- = A \). This is an assumption that is necessary to maintain a linear model, and was performed by Gadda with no noticeable ill effect.

Making these assumptions, we have the following sum-of-forces equation for the hydraulic piston:

\[
\frac{m_p}{d^2x} = A p_u - F_{fr} - F_c
\]  

(2.8)

We presently neglect coulomb friction \( F_{fr} \), as it is much smaller than the load force \(-F_c\). We also neglect the mass of the piston, as its inertia is extremely small compared to load forces. We then have the equation

\[
F_c = A p_u
\]  

(2.9)

The final equation, that follows from continuity of fluid flow in and out of the cylinder is
\[ q = A \frac{dx}{dt} + \frac{V_t}{4\beta_e} \frac{dp_u}{dt} \] (2.10)

This equation accounts for hydraulic fluid compressability as described in Merritt [10]. \( V_t \) is the total, uncompressed volume of fluid entrained in the pistons and servovalves. \( \beta_e \) is the \textit{effective bulk modulus} of the hydraulic fluid and gas entrained in the system. As it is difficult to know the extraneous gas in the system, this parameter is difficult to estimate.

### 2.4 Load Model

To determine the load acting on the hydraulic piston, we must examine the tail; the source of load. Firstly, we assume that all tail operation will be underwater. The water inside the tail adds significant mass that must be considered. The tail also exhibits some compliance from its flexible spine exostructure. Compared to the large mass of the tail in water, however, this compliance force is negligible.

A very detailed analysis of the hydrodynamic forces involved in swimming has been performed by Trapp [13]. However, results from the RoboTuna project [3] suggest that inertial forces are dominant when swimming, and it is adequate to model the tail as a simple mass. This mass, however, is not only the mass of the tail and the water confined to the tail. We must also include added mass to account for the water surrounding the tail that is moved by the swimming tail structure. This added mass can be approximated as an additional volume of water that is equal to the section of tail that is in motion.

Each link can be seen as a force acting on a crank arm that acts on a pivoting axis.
as seen in Figure 2-3. The moment of inertia can then be determined for the rest of the tail. This moment is calculated assuming that the rest of the tail is straight, an assumption that is rarely true, but is a necessary and reasonable approximation.

![Figure 2-3: Inertial Load on Piston (not to scale)](image)

Performing a simple sum of torques on the axis of rotation, we have the following equation:

\[ F_c d_n = 2I_n \ddot{\theta} \]  

(2.11)

Note that the moment of inertia is doubled to account for the added mass of water external to the tail. The above notation assumes that we are examining hydraulic link \( n \) where links are numbered as shown in Figure 2-4. \( I_n \) then represents the moment of inertia of the straight links aft of link \( n \).

The range of \( \theta \) over which we intend to operate is small enough that we can make the linear approximation

\[ \theta \approx K_a x \]  

(2.12)
Where $K_a$ is a constant that is unique to each link. Combining Equations 2.11 and 2.12, we have a final expression:

$$
F_c = \frac{2I_n K_a}{d} \ddot{x}
$$

(2.13)

This equation shows that we can essentially model the entire hydraulic load as a mass that is attached to the tip of the hydraulic piston. We will call this mass the effective mass:

$$
m_{eff} = \frac{2I_n K_a}{d_n}
$$

(2.14)

and we are left with

$$
F_c = m_{eff} \ddot{x}
$$

(2.15)

Since $d$ and $K_a$ are constants that are well known for each link (see Table 2.2), all that is left is a determination of $I_n$.

<table>
<thead>
<tr>
<th></th>
<th>Link 1</th>
<th>Link 2</th>
<th>Link 3</th>
<th>Link 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_a$ ($\circ / m$)</td>
<td>1008.4</td>
<td>1083.8</td>
<td>1934.2</td>
<td>2452.2</td>
</tr>
<tr>
<td>$d$ ($m$)</td>
<td>0.0584</td>
<td>0.0544</td>
<td>0.0302</td>
<td>0.0241</td>
</tr>
</tbody>
</table>

Table 2.2: Tail Structure Mechanical Constants

The inertias of each separate link section have been calculated by Trapp [13] using a CAD model of the tail assuming water inside each section. These inertias were determined about the centroid of each link and are reproduced in Table 2.3.
Figure 2-4: VCUUV Tail Mass Ascribed to Each Link

<table>
<thead>
<tr>
<th>Link</th>
<th>Mass, m (lbm)</th>
<th>Inertia, ( I ), ( \text{lbm} \cdot \text{in}^2 )</th>
<th>Length, L (in)</th>
<th>Centroid, ( L_c ) (in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link 1</td>
<td>58.6</td>
<td>787</td>
<td>11.45</td>
<td>5.01</td>
</tr>
<tr>
<td>Link 2</td>
<td>18.2</td>
<td>185</td>
<td>11.20</td>
<td>4.18</td>
</tr>
<tr>
<td>Link 3</td>
<td>4.32</td>
<td>44.0</td>
<td>10.33</td>
<td>4.47</td>
</tr>
<tr>
<td>Link 4</td>
<td>1.69</td>
<td>8.00</td>
<td>15.50</td>
<td>4.72</td>
</tr>
</tbody>
</table>

Table 2.3: Parameters of the VCUUV Tail Free-Flood Water Ascribed to Each Link
Our $I_n$, however, is not the inertia of one section, but the total inertia of all links aft of link $n$. Using the parallel axis theorem, it is a simple matter to determine the total inertia of the tail about any given axis of rotation given the centroidal moments of inertia from Table 2.3. This combined moment of inertia is shown for each link in Table 2.4.

<table>
<thead>
<tr>
<th>Link</th>
<th>Inertia, $(I_c \text{ kg \cdot m}^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link 1</td>
<td>3.67</td>
</tr>
<tr>
<td>Link 2</td>
<td>0.814</td>
</tr>
<tr>
<td>Link 3</td>
<td>0.153</td>
</tr>
<tr>
<td>Link 4</td>
<td>0.0134</td>
</tr>
</tbody>
</table>

Table 2.4: Tail Structure's Moment of Inertia As Seen by Each Hydraulic Link

### 2.5 System Transfer Function

By combining the equations 2.5, 2.9, 2.10, and 2.15 we can arrive at a Laplace domain transfer function for $F_c$, given $i_b$:

$$\frac{F_c(s)}{i_b} = \frac{K_c s}{\frac{V_t}{4\beta_k A} s^2 + \frac{K_d s}{A} + \frac{A}{m_{eff}}}$$

(2.16)

The open loop Bode plot of this transfer function is shown in Figure 2-5. This plot corresponds to Link 3 as this was the link that was chosen for experimentation. The parameters of the model were determined from manufacturer’s specifications and measurement of tail mechanics. They are shown in Table 2.5.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Link 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_c$, ($\frac{m^3}{s}$)</td>
<td>$1.11 \times 10^{-4}$</td>
</tr>
<tr>
<td>$K_d$, ($\frac{m^3 Pa}{s}$)</td>
<td>$-9.17 \times 10^{-13}$</td>
</tr>
<tr>
<td>$A$, ($m^2$)</td>
<td>$1.11 \times 10^{-4}$</td>
</tr>
<tr>
<td>$m_{eff}$, ($kg$)</td>
<td>342</td>
</tr>
<tr>
<td>$V_t$, ($m^3$)</td>
<td>0.02</td>
</tr>
<tr>
<td>$\beta_c$, ($Pa$)</td>
<td>$8.89 \times 10^8$</td>
</tr>
</tbody>
</table>

Table 2.5: System Parameters for Link #3

System Transfer Function Bode Plot

Figure 2-5: Bode Plot of Open Loop Plant For Link #3
Chapter 3

Linear Controller Design

Since we have determined a reasonable linear approximation for the plant as a single-input single-output system, we can now apply linear control techniques to control the system. The goal of the controller is to achieve tracking within a bandwidth of 1.5 Hz. This is the highest frequency at which the vehicle is designed to operate. Tracking error on the order of $< 10\%$ from 0 to 1.5 Hz is desired.

To arrive at a controller, we could apply a loop-shaping procedure to the open loop Bode plot found in Chapter 2. However, there is the possibility that this theoretical model neglects significant dynamics such as friction, or the inertia of the hydraulic fluid. Instead of trying to determine these dynamics analytically, we can take a more direct approach since we have access to the entire system. It is possible to create an empirical Bode plot of the system's input-output characteristics.

3.1 System Characterization

A set of experiments were performed wherein a sinusoidal input was programmed as the input to the control system $i_b$. The time-domain input and output, $F_c$, were recorded at 25 Hz. The ratio of the amplitude of the output sinusoid over the input
sinusoid gives the magnitude of the transfer function, and the phase lag between the two signals gives the phase of the transfer function.

This was done with sinusoids of various frequencies, ranging from 1.0 Hz to 10.0 Hz, and the results are shown, in Bode plot form, in Figure 3-1. Very low frequencies were difficult to determine experimentally due to the fact that any low frequency input must be very small to ensure that a hydraulic link does not reach its mechanical stops. Such small inputs tend to operate in a region where non-linear characteristics like servovalve dead-zone, or cylinder stiction come into play. As a result, data from this region is ineffective for linear analysis. Also, the model derived in Chapter 2 suggests that, at low frequencies, the system acts as a derivative. This was observed experimentally, as very low frequency inputs produced force outputs too small to be adequately measured.

This experimental Bode plot reveals some unmodeled high frequency dynamics. At lower frequencies, however, the mathematical model from Chapter 2 predicts the integrating behavior of the system.

If we place the system into a basic feedback configuration as shown in Figure 3-2, we can proceed with a loop-shaping design by choosing an appropriate $G_c$.

With simple P-gain control, this open-loop Bode plot clearly does not guarantee magnitude crossover before a phase of -180 is achieved. Indeed, if P control is used on this system with any significant gain, an unstable system results. Thus to ensure that we crossover before high order dynamics give a phase shift of 180 degrees, we implement several compensators to improve the shape of the system’s loop gain.

Because the system has non-linearities, the open-loop Bode plot does not resemble closely the appearance of a traditional linear system. Consequently, the loop shaping
The compensators are a derivative, two lag, and a gain compensator of the form:

\[ G_c = \frac{0.03162s}{s^2 + 3.557s + 3.162} \] (3.1)

This gives the open loop Bode plot shown in Figure 3-3.

Figure 3-1: Experimental Bode Plot of Open Loop System, Link #3

The procedure used was largely trial and error. The first goal was to squelch the "gain peak" at higher frequencies with lag compensators. This helps ensure that the system stays crossed-over at higher frequencies. Also, gain compensation was used to encourage crossover at the point of least phase lag, around 20 rps. Additionally, a derivative compensator was used to reduce phase lag and thus promote stability.
This is a much more desirable open loop system that provides some phase margin at crossover. The exact frequency of crossover is slightly ambiguous, given the experimental data. However, the desired bandwidth of 1.5 Hz, or 9.4 rad/sec, is well below crossover. The amount of phase margin is equally uncertain, however, the data shown in Figure 3-3 suggest that it is at least 60°.

3.2 Controller Performance

The goal of this force controller is to simulate a virtual spring. As such, performance analysis involved setting the force controller’s desired force input, $F_{ref}$, as a function of hydraulic piston extension:

$$F_{ref} = -K_s(x - x_0) \quad (3.2)$$

Where $K_s$ is a virtual spring constant, $x$ is the linear position of the hydraulic cylinder, and $x_0$ is the spring reference position. In this manner, a compliant position controller is created. The reference position becomes the position that the piston tends to under no load, and the spring constant $K_s$ adjusts the compliance of the
virtual spring.

In order to test the controller's ability to track $F_{ref}$ over various frequencies, we can set the spring reference $x_0$ to a sinusoid of various frequencies which results in an $F_{ref}$ of varying frequency.

A more likely mode of operation for the VCUUV is a constant reference position mode. This would simulate a totally passive, spring-like link, while other links move in sinusoids, towards a swimming motion. However, much of the plant model in Chapter 2 assumes that all other links are stationary and straight. As such, we will first examine the controller performance under ideal conditions with no other moving links. After this, we will then examine performance under more realistic conditions.
in order to see how well the controller performs when its approximations are violated.

In the following experiments, $K_s$ was fixed at 20,000 N/m. This corresponds to a spring constant of around 20 Newtons per degree of tail link movement. This is a reasonable spring constant that is on the order of those that we would like to simulate while swimming. The spring reference was set to a sinusoid of amplitude 4.14 mm ($\pm 8^\circ$) at 0.25, 0.5, 1.0, and 1.5 Hertz. Control performance for these four inputs is shown in Figures 3-4-3-7.

![Desired Trajectory vs Actual Trajectory](image)

Figure 3-4: Sinusoidal Tracking at 0.25 Hz, Link #3

The first striking feature of tracking performance is the apparent superposition of a resonance of about 4 Hz on all force trajectories. This is unfortunate because it severely disrupts force tracking. This resonance is an unmodeled higher order dynamic that some attribute to standing pressure waves in the hydraulic lines [12]. This phenomenon is not presently well understood.
Figure 3-5: Sinusoidal Tracking at 0.5 Hz, Link #3

Figure 3-6: Sinusoidal Tracking at 1.0 Hz, Link #3
Despite this resonance, reasonable tracking is achieved over the desired bandwidth, with no significant attenuation or phase lag. It is important to notice that the position trajectory of the link is not significantly disturbed by the resonance in hydraulic force. In Figures 3-4-3-7, we can observe the smoothness of the position trajectory by noting that the desired force trajectory is just the position trajectory scaled by the spring constant. This smooth position trajectory is valuable when trying to maintain a hydrodynamic swimming motion.

A true test of control performance would involve swimming at full speed while force control is active in one or more links. Because testing was confined to an in-lab testing tank in which limited motion is possible, performance testing consisted of activating force control on link #3 and moving one of its two neighboring links in a sinusoidal motion. This provides a measure of how sensitive controller performance is to the motion of other links.

Tracking data was recorded while link #2 moved sinusoidally with an amplitude of ±8°. Link #3 remained under force control with the same parameters as in the previous experiment, but with a stationary reference position. The data for these experiments show that tracking is still reasonable at low frequencies near 0.5 Hz (see Figure 3-8). However, performance quickly degrades with frequencies near 1 Hz (see Figure 3-9). This suggests that the controller's performance during complex, multi-link swimming motions will likely be unsatisfactory. To achieve satisfactory performance, it may be necessary to develop a multivariable controller that takes into account the interdependence of the tail links.

Control performance was again tested with link #3 acting as a stationary virtual spring with the same parameters as in previous experiments. This time, link #4 was
Figure 3-7: Sinusoidal Tracking at 1.5 Hz, Link #3

Figure 3-8: Sinusoidal Tracking of Link #3, Link #2 at 0.5 Hz
moved in a sinusoidal manner over a ±30° range. The results show that, even at
the low frequency of 0.5 Hz, tracking is very poor (see Figure 3-10). Performance is
similarly poor at 1 Hz (see Figure 3-11).

This experiment again shows that a SISO force controller is very sensitive to the
motion of other linkages, and it's performance while swimming at speed will likely be
inadequate.
Figure 3-9: Sinusoidal Tracking of Link #3, Link #2 at 1.0 Hz

Figure 3-10: Sinusoidal Tracking of Link #3, Link #4 at 0.5 Hz
Figure 3-11: Sinusoidal Tracking of Link #3, Link #4 at 1.0 Hz
Chapter 4

Nonlinear Control

Because the equations for fluid flow are highly nonlinear, one might suspect that a nonlinear controller may be more suitable for hydraulic force control. To investigate this, a SISO nonlinear sliding mode controller was designed for the VCUUV. Many of the approximations of the linearized system were maintained; the system is still considered a single hydraulic cylinder that is fixed at one end and acts on an inertial load. Using the same assumptions as were used in Chapter 3, the system diagram is reproduced here for reference (see Figure 4-1). In this model, we have also added a frictional term to account for the viscous friction in water, and coulomb friction in the hydraulic cylinder.

4.1 System Model

The critical difference in the nonlinear system model, is that we no longer linearize the orifice flow equation which is, in full form:

\[ q = K_i i_b \sqrt{P_a - \text{sgn}(i_b)p_u} \]  

(4.1)
One assumption that we will make is that the load pressure, \( p_u \), is always a damping or inertial load. In other words, when our input \( i_b \) is positive, it is implied that the load pressure \( p_u \) is positive. Likewise, if \( i_b \) is negative, then \( p_u \) is also negative. This same assumption is used when finding a linearized form, and is considered reasonable [10]. This gives the following simplified form:

\[
q = K_e i_b \sqrt{P_0 - |p_u|} \tag{4.2}
\]

We can combine this equation with the equation for fluid compressibility (2.10) as well as the following simple sum of forces:
\[ m_{eff} \ddot{x} = p_u A - b \dot{x} \]  

(4.3)

The result is the following third-order nonlinear system:

\[ \begin{align*}
    \dot{x}_1 &= x_2 \\
    \dot{x}_2 &= \frac{A_p}{m} x_3 - \frac{b}{m} x_2 \\
    \dot{x}_3 &= -\frac{A_p}{\beta'} x_2 + \frac{K_t}{\beta'} u \sqrt{P_s - |x_3|}
\end{align*} \]

(4.4)  
(4.5)  
(4.6)

Where \( \beta' = V_t/4 \beta_c \), the control input \( u = i_b \), and the state vector is defined as follows:

\[ \mathbf{x} = \begin{pmatrix} x \\ \dot{x} \\ p_u \end{pmatrix} \]

We are interested in controlling \( F_c \), which is just the pressure across the piston, \( p_u \), scaled by \( A_p \). Thus if we control \( p_u \) (i.e. \( x_3 \)), it is the same as controlling \( F_c \).

The system’s nonlinearity, combined with the difficulty of precisely determining various hydraulic system parameters led to the choice of a nonlinear sliding-mode controller for robust system control.
4.2 Sliding Controller Design

The initial goal is to arrive at a description of the system that is suitable for sliding mode control. This can be done if we note that the system’s equations describe a second order linear system coupled with a first order nonlinear system. Then, the third state equation alone can be written as

$$\dot{x}_3 = f(x_2) + g(x_3)u$$  \hfill (4.7)

This then becomes our entire system with a relative order of one. This approach was used successfully by Alleyne[1] when performing nonlinear control of an active hydraulic car suspension.

Since we are only interested in controlling $x_3$, we essentially internalize the states $x_1$ and $x_2$. In doing so, we lose explicit control over them, however it is argued here that this is not a problem. As long as we control $x_3$ sufficiently, we will set it as a virtual spring:

$$x_3 = -\frac{K_s}{m}x_1$$  \hfill (4.8)

Which, when substituted into equations 4.4, gives the following system:

$$m\ddot{x} + \frac{b}{m}\dot{x} + \frac{K_s}{m}x = 0$$  \hfill (4.9)

This is a simple mass-spring-damper system that is globally asymptotically stable.
to \( x = 0, \dot{x} = 0 \). Such a system is also generally well behaved in that its states are likely to remain in the operating region of the robot.

Now we can define our sliding surface for this first order system as

\[
s = x_3 - x_{3d}
\]  

(4.10)

We can also rewrite the system as

\[
m(x_3) \dot{x}_3 = h(x, x_3) + u
\]  

(4.11)

Where

\[
m(x_3) = \frac{\beta'}{K_t \sqrt{P_s - |x_3|}}
\]  

(4.12)

\[
h(x_3) = \frac{-A_p x_2}{K_t \sqrt{P_s - |x_3|}}
\]  

(4.13)

To determine the best estimate for control, we set \( \dot{s} = 0 \) and solve for \( u \):

\[
m(x_3) \dot{s} = m(x_3) \dot{x}_3 - m(x_3) \dot{x}_{3d}
\]  

(4.14)

\[
0 = h(x_2, x_3) + u - m(x_3) \dot{x}_{3d}
\]  

(4.15)

\[
\ddot{u} = -\ddot{h}(x_2, x_3) + \dot{m}(x_3) \dot{x}_{3d}
\]  

(4.16)
By adding a switching component,

\[ u = \dot{u} - k \text{sgn}(s) \quad (4.17) \]

we can satisfy the sliding condition as follows:

\[ \frac{1}{2} \frac{d}{dt} s^2 \leq -\eta |s| \quad (4.18) \]

\[ m(x_3) s \dot{s} \leq -m(x_3) \eta |s| \quad (4.19) \]

\[ s[h(x_2, x_3) + u - m(x_3) \dot{x}_3] \leq -m(x_3) \eta |s| \quad (4.20) \]

By substituting for \( u \) (Equation 4.17), we have

\[ s[-\tilde{h}(x_2, x_3) + \tilde{m}(x_3) \dot{x}_3 - k \text{sgn}(s)] \leq -m(x_3) \eta |s| \quad (4.21) \]

\[ k |s| \geq m(x_3) \eta |s| + s[-\tilde{h}(x_2, x_3) + \tilde{m}(x_3) \dot{x}_3] \quad (4.22) \]

\[ k |s| \geq m(x_3) \eta |s| + \text{sgn}(s)[-\tilde{h}(x_2, x_3) + \tilde{m}(x_3) \dot{x}_3] \quad (4.23) \]
\[ k \geq m(x_3) \eta + sgn(s)[-\ddot{h}(x_2, x_3) + \dot{m}(x_3) \dot{x}_3d] \] (4.24)

Thus the following definition of \( k \) ensures that the sliding condition is always satisfied:

\[ k \geq m_{max}(x_3) \eta + H(x_2, x_3) + M(x_3) \dot{x}_3d \] (4.25)

Where

\[ H(x_2, x_3) = |\ddot{h}(x_2, x_3) - h(x_2, x_3)| \] (4.26)

\[ M(x_3) = |\dot{m}(x_3) - m(x_3)| \] (4.27)

And \( m_{max} \) is the largest possible value of \( m(x_3) \).

4.3 Parameter Bounds

We can determine \( H(x_2, x_3) \) and \( M(x_3) \) by varying each system parameter that appears in their expressions. First, we need bounds on each of these parameters. Then, by choosing the extremes of these bounds, and selecting them so as to produce the largest possible value of \( M(x_3) \) for any given state, we can determine the most we can overestimate \( M \). Likewise, by choosing a set of parameters that would minimize \( M \) at any given state, we can find the most we can underestimate \( M \). By choosing the larger of these two values, we then have the magnitude by which our estimate of
$M$ could be in error. A similar approach holds for $H$, and $m_{\text{max}}$.

In determining $M$, $H$, and $m_{\text{max}}$, a simplifying technique was performed wherein $x_3$ was set to its largest expected value. This was done because if $x_3$ is not confined to well below supply pressure $P_s$, then $H$, $M$, and $m_{\text{max}}$ become extremely large. We observe that $H$, $M$ and $m_{\text{max}}$ increase monotonically with $x_3$. Thus if we always use the largest expected $x_3$ when evaluating $M$, $H$, and $m_{\text{max}}$ then our value of $k$ is always larger than necessary to satisfy the sliding condition.

The maximum $x_3$ was restricted to $4 \times 10^6 Pa$. This corresponds to a maximum load force of about 450 N on link #3 which is the largest expected load that should be experienced by link #3.

For example, to determine $M(x_3)$, we can first plot the function $m_{\text{upper}}(x_3) - m_{\text{est}}(x_3)$, where $m_{\text{upper}}$ is $m(x_3)$ evaluated with each system constant chosen, within its bounds, to maximize $m$. This function is plotted in Figure 4-2. In a similar fashion, we plot $m_{\text{lower}}(x_3) - m_{\text{est}}(x_3)$ (Figure 4-3).

By observing these plots, it can been seen that the overestimate and underestimates of $m(x_3)$ increase with $x_3$, thus the largest deviations of $m$ will occur at the maximum estimated load pressure, $4 \times 10^6 Pa$. At this load pressure, $m_{\text{upper}} = 7.31 \times 10^{-5}$ and $m_{\text{lower}} = -4.01 \times 10^{-7}$. The largest magnitude of error in $m(x_3)$ is clearly $7.31 \times 10^{-5}$.

This exact same approach was used to bound $H(x_2, x_3)$. However, in this case, since the bounds are a function of $x_2$ as well, a slight variation is made. We can easily factor $x_2$ out of $H$, leaving $x_2 \cdot H'(x_3)$. After bounding $H'$ in the same manner as described above, we have a bound on $H$ because $x_2$ is a state that is measurable.

A similar approach is then taken to find $m_{\text{max}}$. Refer to the matlab files in Appendix B for the precise values of each system parameter, and for each parameter's
Figure 4-2: Upper bound on $m(x_3)$

Figure 4-3: Lower bound on $m(x_3)$
minimal or maximal bounds. These script files demonstrate the approach described above.

4.4 Controller Performance in Simulation

The system and the controller derived above have been simulated in Matlab. In the following simulation, \( \eta \) was set to 0.1, and the virtual spring constant \( K_s = 3000 \text{N/m} \) was used.

The initial state was set as follows:

\[
\mathbf{x} = \begin{pmatrix}
0 \text{ m} \\
0.2 \text{ m/s} \\
0 \text{ Pa}
\end{pmatrix}
\]

The output load pressure is shown in Figure 4-4. We can see that it follows a very spring-like trajectory. Furthermore, the tracking error is shown in Figure 4-5. This figure shows that the error is on the order of \( 10^3 \text{ Pa} \). This is small when compared with the size of the output which is on the order of \( 10^6 \text{ Pa} \). Thus tracking error is reduced to an excellent 0.1% error.

The control input \( u \) is shown in Figure 4-6, and we can see the tell-tale chattering of sliding control in both this input and in the tracking error.

Simulated sliding mode control has demonstrated potential for excellent force control performance. It was noted, during simulation, that the bounds on each system parameter were best if confined to within around 15% of their actual values. If not, the uncertainty in the estimate of system dynamics would cause \( k \) to be too large, and the control input would saturate. This would be unacceptable because it is effectively
Figure 4-4: Sliding control, Plant output (load pressure $p_u$)

Figure 4-5: Sliding control, Tracking Error
the same as having a $k$ that is smaller than is necessary to satisfy the sliding condition. This result suggests that a reasonable approximation of the system parameters is still necessary for satisfactory control.

4.5 Sliding Control In Practice

A sliding controller similar to the design above was implemented on the VCUUV. An interpolating boundary layer was imposed on the sliding surface in order to minimize chattering by acting as a filter of unmodeled high-frequency dynamics:

$$u = \hat{u} - \bar{k}_{sat}(\frac{\dot{y}}{\alpha}) \tag{4.28}$$

Where $\bar{k}$ is defined as
\[ \tilde{k} = k - k_d + \lambda \phi m_{\text{min}} \]  \hspace{1cm} (4.29)

With \( k_d \) equal to \( k \) evaluated along the desired state trajectory. The boundary layer is updated according to:

\[
\dot{\phi} = \begin{cases} 
\frac{k_d - \lambda \phi m_{\text{min}}}{m_{\text{max}}} & \text{if } k_d \leq \lambda \phi m_{\text{min}} \\
\frac{k_d - \lambda \phi m_{\text{min}}}{m_{\text{min}}} & \text{if } k_d > \lambda \phi m_{\text{min}}
\end{cases} \hspace{1cm} (4.30)
\]

These equations ensure that unmodeled dynamics are filtered, and the sliding condition is maintained\[11].

With \( \lambda = 31.4 \) (5Hz) as a cutoff frequency, this controller was simulated and was found to eliminate chattering on the control input (see Figure 4-7). At the same time, a low tracking error on the order of 0.5% was maintained (see Figure 4-8).

![Figure 4-7: Smoothed sliding control, Plant input (u)](image-url)
Unfortunately, it was not possible to achieve an effective sliding controller on the VCUUV. The reason for this may be because our estimate of system parameters such as effective bulk modulus, $\beta_e$ were not easy to determine. As such, the bounds on the these parameter estimates were very loosely constrained. Simulation results showed that fairly tight bounds on parameter estimates are necessary to avoid saturation of the control input. If the the input is saturated, then the sliding condition is not being met, thus tracking and stability cannot be guaranteed.

As was predicted by simulation, the actual controller exhibited an oscillatory behavior in which the input to the system switched between saturation of its two extremes. This results in a vibrating system that does not appear to track the desired trajectory at all.

Control was further inhibited by the fact that no direct measurement of the piston velocity was available. Instead, it was estimated by taking the derivative of position.
This added uncertainty to the controller and made the system sensitive to noise in the position sensor.
Chapter 5

Conclusions

5.1 Summary

This work has shown that successful force-control of a single DOF hydraulic actuator can be achieved using linear, loop-shaping techniques. Using an empirically derived Bode plot of the system plant, tracking performance of near 10% error was achieved, with a slight resonant disturbance.

In a multi-link system such as the VCUUV, however, the motion of other links was found to strongly influence tracking performance. Thus a multiple-input, multiple-output linear controller may be more effective in achieving good tracking during full swimming.

A nonlinear, SISO sliding controller was simulated and achieved 0.1% tracking error when neighboring hydraulic links were assumed static. This same controller was then implemented on the VCUUV, however simulation results showed that parameter estimates within 15% of their true values would be necessary to achieve stable control. This was verified on the actual system, in which a stable controller was not achieved due to inexact system parameter estimates.
5.2 Future Work

The excellent tracking performance of sliding control under simulation suggests that further research in this direction is warranted. However, to successfully perform hydraulic sliding control, good estimates of system parameters will be necessary.

Alternatively, it may be effective to integrate adaptation in various system parameters. The effective bulk modulus would be an ideal candidate for adaptation, as the amount of gas entrained in the system is both hard to measure and likely to vary slowly with time. To further reduce uncertainty in system dynamics, it may be prudent to add a sensor that directly measures piston velocity, instead of estimating it from the derivative of position.

Judging from the strong inter-link dependence demonstrated by the linear control system, a MIMO nonlinear control system would be most likely to achieve good force tracking while swimming.

With further work on nonlinear force control, it will likely be possible to accurately simulate virtual springs on the VCUUV while swimming in the field.
Bibliography


Appendix A

Glossary of Terms

caudal fin The large, rigid, vertical fin that attaches to the end of the tail’s link assembly.

differential variable reluctance transducer (DVRT) A linear position sensor that is mechanically linked to each hydraulic cylinder to determine the extension of each cylinder.

effective bulk modulus The bulk modulus of the combined liquid and gas in the hydraulic system, $\beta_e$

hydraulic power unit (HPU) The DC-motor and hydraulic pump assembly that is responsible for maintaining a hydraulic pressure differential that is used to drive the link assembly.

link assembly The four aluminum links that are the foundation of the semi-flexible tail. These links are driven by hydraulic pistons.

pressure hull The air-tight, front section of the VCUUV.

RoboTuna A M.I.T. robotic tuna project led by M. Triantafyllou and researched by David Barrett.

spool The controllable component of a hydraulic servovalve whose position determines the size of flow orifices, and thus flow-rate through the servovalve.

spring constant The coefficient $k$ in the first order model of a mechanical spring, in which $\text{Force} = -k \cdot x$
theoretical efficiency A measure of the energy required to achieve swimming motion that ignores the energy losses in the hydraulic systems. This is determined by measuring force applied over linear distance, i.e. $Work = Force \cdot x$

VCUUV Vorticity Control Unmanned Undersea Vehicle. A robotic tuna fish project carried out at the Charles Stark Draper Laboratory.

virtual spring A active mechanical system that is designed to mimic the forces of a simple mechanical spring, often using force-feedback techniques.
Appendix B

Matlab Script Files
Here are all the system constants

% Mechanical Constants

d = 0.0302; % Crank arm leverage distance (m)
Itot = 0.153; % Effective inertia (kg * m^2)
Ka = 193*pi/180; % theta = Ka (X) linear distance to angle constant
Meff = 2*(Itot*Ka/d); % Effective mass of water/tail (kg)

% Hydraulic Constants

A = 1.11E-4; % Average surface area of the piston (m^2)
Be = 6.89E8; % Effective bulk modulus (Pa) estimated at 100,000psi
Vt = 0.02; % Total confined fluid volume in cylinderinose (m^3)
Bp = Vt / (4*Be); % Beta prime (as in my notes)
Kt = 8.00E-7; % Flow constant. originally 8.00E-7 m^3/s*sqrt(psi)
Kt = Kt * 1.2E-2; % m^3/s*sqrt(Pa) (necessary conversion to SI)

Ps = 8.27E6; % Supply pressure (N/m) = 1200psi
Ks = 3000; % Virtual spring constant (N/m) (3000 is good)

% estimates of system params
Kt_E = Kt*0.9;
A_E = A;
Ps_E = Ps;
Bp_E = Bp;
LAM = 2*pi*(5); % sliding controller cutoff
% Plot the function M(x_3), both upper and lower bounds
Be_min = 6.6E8;
Be = 6.89E8;
Be_max = 7.2E8;

Vt_min = 0.015;
Vt = 0.02;
Vt_max = 0.025;

Bp_min = Vt_min/(4*Be_max);
Bp = Vt/(4*Be);
Bp_max = Vt_max/(4*Be_min);

Kt_min = 9.1E-9;
Kt = 9.6E-9;
Kt_max = 10.1E-9;

Ps_min = 8.0E6;
Ps = 8.27E6;
Ps_max = 8.6E6;

x3 = [0:100:Ps_min];
est = Bp ./ (Kt*sqrt( Ps_min+1 - abs(x3) ) );
tooplus = Bp_max ./ ( Kt_min*sqrt( Ps_min+1 - abs(x3) ) );

figure(1);
plot(x3, tooplus);
title('max estimation');

figure(2);
plot(x3, tooplus - est);
title('upper parameter deviation');
xlabel('x_3 (Pa)');
ylabel('m (upper) - m (estimate)');
axis([0 5e6 0 1.5e-7]);

toominus = Bp_min ./ (Kt_max*sqrt( Ps_max - abs(x3) ) );

figure(3);
plot(x3, toominus - est);
xlabel('x_3 (Pa)');
ylabel('m (lower) - m (estimate)');
axis([0 5e6 -1.5e-7 0]);

% Plot the function H'(x_3), both upper and lower deviation.
Ap_min = 1.0E-4;
Ap = 1.11E-4;
Ap_max = 1.22E-4;

Kt_min = 9.1E-9;
Kt = 9.6E-9;
Kt_max = 10.1E-9;

Ps_min = 8.0E6;
Ps = 8.27E6;
Ps_max = 8.6E6;

x3 = [0:100:Ps_min];
est = -Ap ./ (Kt*sqrt( Ps - abs(x3) ) );
tooplus = -Ap_max ./ (Kt_max*sqrt( Ps_max - abs(x3) ) );

figure(1);
plot(x3, tooplus);
title('max estimation');

figure(2);
plot(x3, tooplus - est);
title('upper parameter deviation');
toominus = -Ap_max ./ ( Kt_min*sqrt( Ps_min+1 - abs(x3) ) );

figure(3);
plot(x3, toominus - est);
title('lower parameter deviation');
function x_dot = slide2( t, x )

defcons % Define all system mechanical constants

% Here's where we want to be ( F = Ks*x1 => x3_d dot = (Ks/A)*x2
x3_d = -(Ks/A)*x(1);
x3_d dot = -(Ks/A)*x(2);

% Let's define the control inputs tau based on the present state
s = x(3) - x3_d; % First define s
% Now the estimate of control input
h_hat = -A*x(2) / ( Kt*sqrt( Ps - abs(x(3)) ) );
m_hat = Bp / ( Kt*sqrt( Ps - abs(x(3)) ) );
u_hat = -h_hat + m_hat*x3_d dot;
% Now add in some switching

m_max = 5.42E-7;
m_err = 1.42E-7;
h_err = 1.11;
ETA = 0.1;
v1=m_max*ETA;
v2=abs(m_err*x3_d);
v3=abs( h_err*x(2) );
K = v1 + v2 + v3; % define our switching input
u = u_hat -K*sign(s);

% Now the state equations of the system
x_dot(1) = x(2);
x_dot(2) = (A/Meff)*x(3) - (1000/Meff)*x(2); % The drag term is extraneous
x_dot(3) = - (A/Bp)*x(2) + (Kt/Bp)*u*sqrt( Ps - abs( x(3) ) );

x_dot = x dot;
% Plot the results of controller in slide2.m
% (non-smoothing sliding control)

% Set ode options
defcons % declare all system constants

% Initial state conditions
x0 = [0 0.2 0];
% Solve the system
TIC
[t, x] = ode23('slide2', [0 4], x0);
TOC

% Position
figure(1);
plot( t, x(:,1) );
title('x1');

% Velocity
figure(2);
plot( t, x(:,2) );
title('x2');

% Tracking error
figure(3);
plot( t, (x(:,3) - (Ks/A)*x(:,1)) );
xlabel('Time (sec)');
ylabel('p_{u} (Pa)');
axis([0 4 -8000 8000]);

% Now let's postprocess plot, to find things like u.
x3_d = -(Ks/A)*x(:,1);
x3_d_dot = -(Ks/A)*x(:,2);

% Let's define the control inputs tau based on the present state
s = x(:,3) - x3_d; % First define s
% Now the estimate of control input
h_hat = -A*x(:,2); % Kt*sqrt( Ps - abs(x(:,3)) )
m_hat = Bp ./ (Kt*sqrt( Ps - abs(x(:,3))));
u_hat = -h_hat + m_hat.*x3_d_dot;

% Now add in some switching
m_max = 5.42E-7;
m_err = 1.42E-7;
h_err = 1.11;
ETA = 0.1;
v1=m_max*ETA;
v2=abs(m_err.*x3_d);
v3=abs(h_err.*x(:,2));
K = v1 + v2 + v3; % define our switching input magnitude
u = u_hat -K.*sign(s);

% Control input, u
figure(4);
plot(t, u);
xlabel('Time (sec)');
ylabel('Normalized input u');

figure(5);
plot(t, x(:,3));
title('Controlled Pressure (p_u)');
xlabel('Time (sec)');
ylabel('p_{u} (Pa)');
This is an ODE file (for use with ode23/45)

function x_dot = slide3( t, x )

defcons % Define all system mechanical constants

Here's where we want to be ( F = Ks*x1 => x3_d_dot = (Ks/A)x2
x3_d = -(Ks/A_E)*x(1);
x3_d_dot = -((Ks/A_E))*x(2);

Let's define the control inputs tau based on the present state
s = x(3) - x3_d; % First define s

Now the estimate of control input
h_hat = -A_E*x(2) / ( Kt_E*sqrt( Ps_E - abs(x(3))) );
m_hat = Bp_E / ( Kt_E*sqrt( Ps_E - abs(x(3)) ) );
u_hat = -h_hat + m_hat*x3_d_dot;
% Now add in some switching
m_max = 5.42E-7;
m_err = 1.42E-7;
h_err = 1.11;
ETA = 0.1;
v1=m_max*ETA;
v2=abs(m_err*x3_d);
v3=abs( h_err*x(2) );
K = v1 + v2 + v3; % define our switching input
K_bar = LAM*x(4)*3.8e-7;
u = u_hat - K_bar*sqrt(s/x(4));

Now the state equations of the system
x_dot(1) = x(2);
x_dot(2) = (A/Meff)*x(3) - (1000/Meff)*x(2);
x_dot(3) = -(A/Bp)*x(2) + (Kt/Bp)*u*sqrt( Ps - abs(x(3))) ;

Here's phi
if (K <= 12*x(4)*3.8e-7)
  x_dot(4) = (K - LAM*x(4)*3.8e-7) / 5e-7;
else
  x_dot(4) = (K - LAM*x(4)*3.8e-7) / 3.8e-7;
end

x_dot = x_dot';
% Plot the results of controller in slide3.m

% Set ode options
defcons % declare all system constants

kd0 = 5e-7*0.1 + 1.42e-7 + Ks*0.2/A;
phio = kd0/LAM/3.8e-7;

% Initial state conditions
x0 = [ 0 0.2 0 phio];
% Solve the system
TIC
[t, x] = ode23('slide3', [0 4], x0);
TOC

% Position
figure(1);
plot( t, x(:,1) );
title('x1');

% Velocity
figure(2);
plot( t, x(:,2) );
title('x2');

% Tracking Error
figure(3);
plot( t, ( x(:,3) - (-Ks/A)*x(:,1) ) );
xlabel('Time (sec)');
ylabel('p_u (Pa)');
axis([0 4 -8000 8000]);

% Now let's postprocess plot, to find things like u.
x3_d = -(-Ks/A)*x(:,1);
x3_d_dot = -(-Ks/A)*x(:,2);

% Let's define the control inputs tau based on the present state
s = x(:,3) - x3_d; % First define s
% Now the estimate of control input
h_hat = -A_E*x(:,2) ./ ( Kt_E*sqrt( Ps_E - abs(x(:,3)) ) );
m_hat = Bp_E ./ ( Kt_E*sqrt( Ps_E - abs(x(:,3)) ) );
u_hat = -h_hat + m_hat.*x3_d_dot;

% Now add in some switching
m_max = 5.42E-7;
m_err = 1.42E-7;
h_err = 1.11;
ETA = 0.1;
v1 = m_max*ETA;
v2 = abs(m_err.*x3_d);
v3 = abs(h_err.*x3_d);
K = v1 + v2 + v3; % define our switching input magnitude
K_bar = LAM*x(:,4)*3.8e-7;
u = u_hat - K_bar.*sat(s./x(:,4));

figure(4);
plot(t, u);
xlabel('Time (sec)');
ylabel('Normalized input u');
figure(5);
plot(t, x(:,3));
title('Controlled Pressure (p_u)');
xlabel('Time (sec)');
ylabel('p_u (Pa)');