GEOMETRIC PATH PLANNING FOR A LEGO AUV

by

Michael Richard Charles Andonian

Submitted in Partial Fulfillment
of the Requirements for the Degree of
Master of Arts in Mathematics
Plan B

UNIVERSITY OF HAWI‘I AT MANOA
April 13, 2012

Thesis Committee:

Monique Chyba
Associate Professor of Mathematics
Advisor

George Wilkens
Professor of Mathematics
Advisor

Graduate Committee:

J.B. Nation
Professor of Mathematics
Graduate Chair

Monique Chyba
Associate Professor of Mathematics

George Wilkens
Professor of Mathematics

Robert Little
Professor of Mathematics

Pavel Guerzhoy
Professor of Mathematics

Abstract

For the last thirty years or so, differential geometry and control theory have merged and grown together to produce extraordinary results. When applied to mechanical systems, one sees a system waiting to be exploited for its inherent geometric properties. In this paper, we present the equations of motion for a submerged rigid body from a geometric point of view and use tools from differential geometry to provide solutions to the motion planning problem for an autonomous underwater vehicle. Specifically, the geometry allows us to deduce permissable motions for a vehicle that is underactuated purely from the available degrees of freedom. The geometric equations of motion are then used to path plan for a cost-effective Lego vehicle through simulations and actual implementation as providing a proof of concept.

Acknowledgements

First and foremost, my thanks goes out to Dr. Monique Chyba. I have worked with her for the last two years and can honestly say it has been nothing but exciting, productive, and rewarding. She has been incredibly patient through the years, of which I appreciate immensely. I can never thank her enough, but thank you again for JPL and giving me the opportunity to earn a Masters in Mathematics. To Dr. George Wilkens, thank you for always having your door open. Countless times I would walk by, knock, and ask, "Can I ask you a question?" and even if you were busy or tight on time, your reply was always, "Of course!" Our conversations helped me grow into a better mathematician.

Thank you to Dr. Ryan Smith for the idea of building a Lego AUV which lead to this Masters project. Thank you to his student Brane Pesic, whom I've only collaborated through emails, but without his initial model and guidance for building, Dr. Chyba and I would have no Lego AUV to begin with. Thank you to Justin Toyofuku and Robert Young for helping me build the Lego AUV, concoct ideas, and getting in the pool so I didn't have to.

Thank you to Susan, Shirley, Troy, and the rest of the Mathematics Department for all your support over the years; I've enjoyed every moment as a student in this department. I also give thanks to SUPER-M for the wonderful experience and support. Finally, a special thank you to all my friends and family for their love and support; this Masters is an accomplishment I dedicate to you. There are so many people to thank, but I suppose this is how it feels to win an Academy Awards feel.

Contents

1 Introduction

As humankind moves forward in an age dominated by digital information and advancements in technology, more projects involve the use of robotics to accomplish tasks. Computers far exceed the human mind's ability to process and analyze data where as mechanical systems are more durable than the human body. For these reasons, it is perfectly sensible to utilize robots to execute tasks that prove difficult for humans. Autonomous underwater vehicles are just one such example of how robotics have extended humankinds ability to learn and understand the planet we reside on.

Autonomous underwater Vehicles (AUVs) are robust submersibles capable of transversing vast oceans and narrow passages. Their primary purpose is to provide bathymetric maps of the ocean floor [6] and river surveys [6], but they also investigate hazardous environments in which a human would be endangered. Example of how AUVs are used include sampling areas near hydrothermal vents where extremophiles exist [7], [8] and navigating through fast-flowing environments and shallow waters [9], [10]. NASA has also funded AUV projects such as the DEep Phreatic THermal eXplorer (DEPTHX) and Environmentally Non-Disturbing Under-ice Robotic ANtarctiC Explorer (ENDURANCE). Both of these AUVs employ an array of robust sensors to navigate through their environments in preparation for an unmanned mission to Europa [11],[12].

But there is a very important question involving AUVs, that is how to navigate from one desired location to another. This question is referred to as the motion planning problem and is the study of this paper. Specifically, the motion planning problem for AUVs is to produce a trajectory or path from a given initial configuration to a desired final configuration, all within the confines of the system. Limitations such as thrust bounds and, as we will see, the number of degrees of freedom make this problem complex and non-trivial. By defining equations which describe the dynamical evolution of our system, that is an AUV transversing through a fluid, the motion planning problem falls into control theory. Over the last thirty years or so, differential geometry has tremendous influence on the field of control theory, [20], [21],[15]. And so, we exploit techniques from differential geometry to produce feasible trajectories and solve the motion planning problem. By adjusting the mathematical framework to that of differential geometry, we can account for loses in the number of degrees of freedom. As we will see, the inherent geometry allows us to explicitly produce feasible motions for a given number of degrees of freedom. To investigate this problem, a "test-bed" AUV was constructed using Lego products.

The Lego NXT provides a unique tool for educational purposes. As with most Lego products, the "Lego engineer" can build a plethora of structures and mechanical objects using Lego NXT pieces, limited only by their imagination. But the Lego NXT goes beyond building structures and into the realm of logic and programming. The NXT brick is capable of being programmed and executing a variety of commands as well as acquiring and processing information about its environment. In fact, a variety of groups have looked into using the Lego NXT to provide a foundation of undergraduate engineering courses (see [1], [2], [3]). Others have used the NXT to perform signal processing experiments, [4], as well as investigating the NXT's full capabilities [5]. For the purposes of this Masters project, we investigate a Lego autonomous underwater vehicle (LAUV) and apply geometric concepts to produce paths within the limitations of the system. We begin with a discussion of the equations of motion for a submerged rigid body in a viscous fluid and then apply concepts from geometric control theory to this system. Using techniques from differential geometry, we will re-derive the equations of motion geometrically and exploit inherent geometric properties to provide plausible trajectories for this Lego AUV. This will be followed by simulations as a proof of concept and actual implementation onto the LAUV itself.

2 Geometric Equations of Motion

2.1 Kinetic Equations of Motion

In order to derive the equations of motion for a rigid body, we first need some reference frames. We'll start with an Earth-fixed frame which, for all intents and purposes, we consider to be inertial, denoted $\mathcal{O}_I$, and a body-fixed frame, denoted $\mathcal{O}_B$, fixed to the center of gravity of the rigid body, C_g. Both of these frames are right-handed orthogonal systems. Moreover, both frames have local coordinate systems defined by the axes of their respective reference frame. The inertial frame coordinate system is given by $\Sigma_I = \{\mathcal{O}_I, (s_1, s_2, s_3)\}$ where $s_3 = s_1 \times s_2$ and the body-fixed frame coordinate system is given by $\Sigma_B = \{\mathcal{O}_B, (b_1, b_2, b_3)\}$, where $b_3 = b_1 \times b_2$. Specifically, b_1 will correspond to the longitudinal axis, positive to fore, b_2 corresponds to the transversal axis positive to starboard, and b_3 is the normal axis positive to keel. We denote a vector from $\mathcal{O}_I$ to $\mathcal{O}_B$ by $\mathbf{r} = (r_1, r_2, r_3)$. Figure 1 shows a depiction of our two reference frames. Table 1 shows the notation we will use for the remainder of this paper. This notation differs greatly from the notation typically used in hydrodynamics and defined by the Society of Naval Architects and Marine Engineers (SNAME). However, using the notation in Table 1 will prove to make the read less cumbersome.

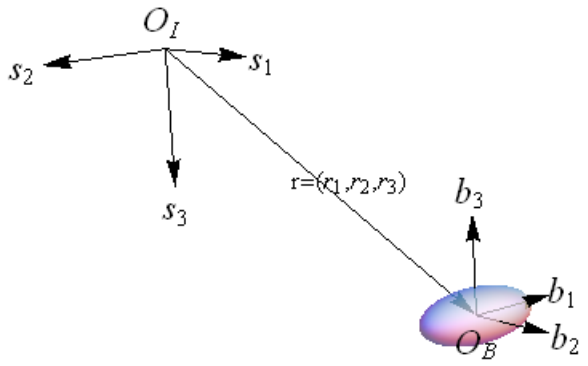

Figure 2.1: Earth-fixed and body-fixed coordinate reference frames

DOF	External Forces & Moments	Linear & Angular Velocities	Positions & Euler Angles
Surge	φ_1	ν_1	r_1
Sway	φ_2	ν_2	r_2
Heave	φ_3	ν_3	r_3
Roll	τ_1	Ω_1	ϕ
Pitch	τ_2	Ω_2	θ
Heave	τ_3	Ω_3	ψ

Table 1: Notation for Various Parameters of Underwater Vehicles (Body-Fixed Frame)

Next, we note that the position and angular orientation of the rigid body with respect to $\mathcal{O}_I$ is given by $\eta = (r_1, r_2, r_3, \phi, \theta, \psi)$. It is clear that the angular orientation of the body-fixed frame with respect to the Earth-fixed inertial frame can be expressed as a rotation matrix, R. Thus, the pairing $(\mathbf{r}, R) \in \mathbb{R}^3 \times SO(3)$ yields the position and orientation of the rigid body with respect to the inertial frame of reference. Note that η also describes the motion of the rigid body with respect

to the inertial frame. In the body-fixed frame of reference, linear velocity is $\nu = (\nu_1, \nu_2, \nu_3)$, angular velocity is $\mathbf{\Omega} = (\Omega_1, \Omega_2, \Omega_3)$, and the external forces and moments are given by $\sigma = (\varphi_1, \varphi_2, \varphi_3, \tau_1, \tau_2, \tau_3)$.

Now, we compute the general rotation matrix for our rigid body. Let $R_i(\alpha)$ denote the rotation matrix which expresses a vector a rotation angle α about the i axis. We choose to adopt the nonstandard notation of letting $s\cdot = \sin(\cdot)$ and $c\cdot = \cos(\cdot)$. This gives

$$R_{s_1}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\phi & s\phi \\ 0 & -s\phi & c\phi \end{bmatrix} R_{s_2}(\theta) = \begin{bmatrix} c\theta & 0 & -s\theta \\ 0 & 1 & 0 \\ s\theta & 0 & c\theta \end{bmatrix} R_{s_3}(\psi) = \begin{bmatrix} c\psi & s\psi & 0 \\ -s\psi & c\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

It is clear to see that each of the matrices above are in $SO(3)$, and since $SO(3)$ is a group, any product of these matrices is also in $SO(3)$. By adopting the xyz- convention for Euler angles, we let $R = R_{s_3}(\psi) R_{s_2}(\theta) R_{s_1}(\phi)$. More explicitly

$$R = \begin{bmatrix} c\theta c\psi & c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi - c\phi s\theta c\psi \\ -c\theta s\psi & c\phi c\psi - s\phi s\theta s\psi & s\phi c\psi + c\phi s\theta s\psi \\ s\theta & -s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{2.1}$$

.

Clearly, the construction of R allows us to deduce that $R\nu$ provides the linear velocity in the inertial reference frame. Finally, the body-fixed angular velocity is given as,

$$\Omega = R_{s_1}(-\phi) R_{s_2}(-\theta) \begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} + R_{s_1}(-\phi) \begin{bmatrix} 0 \\ \dot{\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ 0 \\ 0 \end{bmatrix} = R_\alpha^{-1} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

where $R_\alpha^{-1} = \begin{bmatrix} 1 & 0 & s\theta \\ 0 & c\phi & -c\theta s\phi \\ 0 & s\phi & c\theta c\phi \end{bmatrix}$. This yields the kinematic equations between the body-fixed frame and the inertial frame

$$\dot{\eta} = \begin{bmatrix} R & 0_{3\times 3} \\ 0_{3\times 3} & R_\alpha \end{bmatrix} \begin{bmatrix} \nu \\ \Omega \end{bmatrix}.$$

We now continue on to define the dynamics, both in terms of the coordinates and geometrically.

2.2 Submerged Rigid Body Dynamics

In general, equations of motion describe the evolution of the dynamics of a system in question typically as a function of time. For a rigid-body, the equations of motion most famously correspond the the Newton-Euler equations describing the forces ($F = m\mathbf{a}$) and moments ($\mathcal{F} = J\alpha$) acting on the rigid body. Continuing to define the dynamics of a submerged rigid body, we present the equations of motion of a submerged rigid body with respect to a coordinate system. A detailed derivation of the following equations of motion can be found in Fossen, [13]. The general equations of motion for a submerged rigid body are given as

$$M\dot{v} + C(v)v + D(v)\nu + g(\eta) = \sigma$$

where $M = M_{vehicle} + M_{added} \in \mathbb{R}^{6\times 6}$ is the mass matrix, $C(v) \in \mathbb{R}^{6\times 6}$ is the Coriolis matrix accounting for pseudo-forces, $D(v) \in \mathbb{R}^{6\times 6}$ is the drag matrix accounting for hydrodynamic drag, $g(\eta) \in \mathbb{R}^6$ accounts for potential forces, $v = (\nu, \Omega)^T$, and σ denotes the external forces and moments as before. The added mass matrix can be simplified as long as the rigid body in question has any

number of planes of symmetry. Thus, the added mass matrix is diagonal and the equations of motion in body-fixed coordinate system is given by

$$\begin{pmatrix} mI_3 + M_f & 0_{3\times3} \\ 0_{3\times3} & J_b + J_f \end{pmatrix} \begin{pmatrix} \dot{\nu} \\ \dot{\Omega} \end{pmatrix} - D(v)\, v - Cor_B(v)\, v + g(\eta) = \sigma(t) \tag{2.2}$$

where m is the vehicle's mass, $M_f = diag\left\{M_f^{\nu_1}, M_f^{\nu_2}, M_f^{\nu_3}\right\}$ is the added mass for the translational forces, $J_b = diag\{I_{b_1}, I_{b_2}, I_{b_3}\}$ are the moments of inertia, and $J_f = diag\left\{J_f^{\Omega_1}, J_f^{\Omega_2}, J_f^{\Omega_3}\right\}$ is the added inertia for the rotational moments. These equations describe the motion of a submerged rigid body in a viscous fluid with six degrees-of-freedom. In the case of the LAUV, we assume the vehicle has three planes of symmetry, although it does not. Expanding some of these terms will show that these are in fact the Newton-Euler equations.

Although this coordinate represention is common and easily understandable, there exists geometric equations describing the same dynamics while intrinsic to the system itself. The geometric equations that will be presented also follow the Newton-Euler formulation. One plus to the geometric equations of motion is that they are coordinate invariant, as they arise naturally from the geometry of the system itself. Moreover, as we'll see, these geometric equations offer us insight into the likely scenario of an AUV losing any number of degrees of freedom due to various circumstance. Specifically, in the event our vehicle is or becomes under-actuated, we'll be able to quickly discern trajectories that are feasible.

The geometric equations of motion we are about to present come from [17]. We follow the same formulation, and thus exclude some details. The reader is referenced to [14] and [15] for background on differential geometry as a whole and its application to mechanical systems, respectively. Recall that $\mathbf{r} \in \mathbb{R}^3$ represents the position and $R \in SO(3)$ represents the orientation of a rigid body, with repsect to the inertial frame of reference, which we pair together as $(\mathbf{r}, R) \in \mathbb{R}^3 \times SO(3)$. Clearly there is a natural bijection from $\mathbb{R}^3 \times SO(3)$ to $SE(3)$ given by

$$(\mathbf{r}, R) \mapsto \begin{pmatrix} R & \mathbf{r} \\ 0_{1\times3} & 1 \end{pmatrix}.$$

Let $Q = SE(3)$, and note that Q is a differentiable manifold. The set $\{X_1, \ldots, X_6\}$ will denote the basis of Q whereas the $\{\pi_1, \ldots, \pi_6\}$ will be the basis for the dual space, Q^*. Q is also the configuration space for our submerged rigid body, that is every possible configuration for the submerged rigid body lies on Q. Following the formulation in [17], we can define a Riemannian metric by

$$\mathbb{G} = \begin{pmatrix} M & 0 \\ 0 & J \end{pmatrix}$$

where $M = mI_3 + M_f$ and $J = J_b + J_f$. Uniquely, we can then associate the Levi-Civita connection ∇. This provides us the appropriate notion of geometric acceleration and is given by

$$\nabla_{\gamma'}\gamma' = \begin{pmatrix} \dot{\nu} + M^{-1}(\Omega \times M\nu) \\ \dot{\Omega} + J^{-1}(\Omega \times J\Omega + \nu \times M\nu) \end{pmatrix} \tag{2.3}$$

where $\gamma : I \to Q$ is a curve on Q. With this, we define the equations of motions of a rigid body geometrically. However, this does not include the drag and potential forces an AUV would experience submerged in a fluid. These forces can be modelled geometrically with a few assumptions. To model our forces, we assume the forces are linear (two forces acting on the system is the sum of

the effects and scaling a force scales its effect on the system) and the geometric force F depends on $(\mathbf{r}, R) \in Q$, $\left(\dot{\mathbf{r}}, \dot{R}\right) \in T_{(\mathbf{r},R)}Q$, and time t. Thus, we define the map $F : \mathbb{R} \times TQ \to T^*Q$ to be our external forces, F having the same properties that we just outlined. With this definition, external forces are given geometrically by $\mathbb{G}^\sharp\left(F\left(\gamma'\left(t\right)\right)\right) \in TQ$. Explicitly, the drag forces are given as

$$\mathbb{G}^\sharp\left(F_{drag}\left(\gamma'\left(t\right)\right)\right) = \sum_{i=1}^{3} \frac{F_i\left(\gamma'\left(t\right)\right)}{\mathbb{G}_{ii}} \nu_i^2 X_i + \sum_{i=4}^{6} \frac{F_i\left(\gamma'\left(t\right)\right)}{\mathbb{G}_{ii}} \Omega_{i-3}^2 X_i \tag{2.4}$$

and the potential forces are given as

$$\mathbb{G}^\sharp\left(P\left(\gamma'\left(t\right)\right)\right) = \begin{bmatrix} -\frac{1}{m_1}\left(W-B\right)s\theta \\ \frac{1}{m_2}\left(W-B\right)c\theta s\phi \\ \frac{1}{m_3}\left(W-B\right)c\theta c\phi \\ \frac{1}{j_1}\left(\left(y_G W - y_B B\right)c\theta c\phi - \left(z_G W - z_B B\right)c\theta s\phi\right) \\ -\frac{1}{j_2}\left(\left(z_G W - z_B B\right)s\theta - \left(x_G W - x_B B\right)c\theta c\phi\right) \\ \frac{1}{j_3}\left(\left(x_G W - x_B B\right)c\theta s\phi + \left(y_G W - y_B B\right)s\theta\right) \end{bmatrix} \tag{2.5}$$

where $W = mg$ is the rigid body's weight, $B = \rho\mathcal{V}g$ is the buoyant force, $m_i = m + M_f^{\nu_i}$, $j_i = J_b^{\Omega_i} + J_f^{\Omega_i}$ $(i = 1, 2, 3)$, and $\mathbf{r}_G = (x_G, y_G, z_G)$ $(\mathbf{r}_B = (x_B, y_B, z_B))$ is the location of the center of gravity (center of buoyancy) relative to $\mathcal{O}_B$. Now, we define a forced affine connection control system:

Definition A C^∞- forced affine connection control system is the 6-tuple $(Q, \nabla, \mathcal{D}, Y, \mathcal{Y} = \{Y_1, \ldots, Y_m\}, U)$ where

(i) Q is a manifold,

(ii) ∇ is an affine connection on Q,

(iii) $\mathcal{D}$ is a regular linear velocity constraint such that ∇ restricts to $\mathcal{D}$,

(iv) Y is a vector force,

(v) $\mathcal{Y}$ is a set of input vector fields, typically taken to be the external control of the system.

(vi) $U \in \mathbb{R}^m$ is the control set in which the input controls take their values,

with all items assumed to be C^∞.

Applying this definition directly to our situation with the submerged rigid body, $Q = SE\left(3\right)$, ∇ is the Levi-Civita connection on Q associated with $\mathbb{G}$, $\mathcal{D} \subseteq TQ$, $\mathcal{Y} \subseteq \left\{\mathbb{I}_1^{-1}, \mathbb{I}_2^{-1}, \mathbb{I}_3^{-1}, \mathbb{I}_4^{-1}, \mathbb{I}_5^{-1}, \mathbb{I}_6^{-1}\right\}$, $Y = \left\{\mathbb{G}^\sharp\left(F_{drag}\left(\gamma'\left(t\right)\right)\right), \mathbb{G}^\sharp\left(P\left(\gamma'\left(t\right)\right)\right)\right\}$, and $U = \sigma \in \mathbb{R}^6$. Briefly,
$\mathcal{Y} \subseteq \left\{\mathbb{I}_1^{-1}, \mathbb{I}_2^{-1}, \mathbb{I}_3^{-1}, \mathbb{I}_4^{-1}, \mathbb{I}_5^{-1}, \mathbb{I}_6^{-1}\right\}$ is the supposed to be the set of input vector fields. In the situation with the submerged rigid body, this corresponds to the inverse of the metric, $\mathbb{G}^\sharp$. Recall that $\{\pi_1, \ldots, \pi_6\}$ is the basis for Q^*. Explicitly, $\mathbb{I}_j^{-1} = \mathbb{G}^\sharp\pi_j = \mathbb{G}^{jj}X_j$. Thus, our input vector fields correspond directly to pure motions. This allows us to define the geometric equations of motion for a submerged rigid body in a viscous fluid, analogous to the coordinate equations we saw previously: For a path $\gamma : I \mapsto Q$,

$$\nabla_{\gamma'}\gamma' = \mathbb{G}^\sharp\left(P\left(\gamma\left(t\right)\right)\right) + \mathbb{G}^\sharp\left(F_{drag}\left(\gamma'\left(t\right)\right)\right) + \sum_{i=1}^{6} \mathbb{I}_i^{-1}\left(\gamma\left(t\right)\right)\sigma_i\left(t\right). \tag{2.6}$$

By modelling the drag forces as a $(1,2)$-type tensor, we can modify the Levi-Civita connection above to absorb these forces. Again, the details of this can be found in [17]. We modify the connection as follows:

$$\tilde{\nabla}_{X_i} X_j = \begin{cases} -\frac{D_i}{\mathbb{G}_{ii}} X_i & i = j \\ \nabla_{X_i} X_j & i \neq j \end{cases} \tag{2.7}$$

since $\nabla_{X_i} X_i = 0$ for all i. Thus, we obtain this proposition:

Proposition 1. *Let $Q = SE(3)$, $\widetilde{\nabla}$ be the modified Levi-Civita connection on Q associated with the Riemannian metric $\mathbb{G}$, and let the set of input control vector fields be given by $\mathcal{I} = \left\{\mathbb{I}_1^{-1}, \mathbb{I}_2^{-1}, \mathbb{I}_3^{-1}, \mathbb{I}_4^{-1}, \mathbb{I}_5^{-1}, \mathbb{I}_6^{-1}\right\}$. Let $\mathbb{G}^\sharp\left(P\left(\gamma\left(t\right)\right)\right)$ represent the potential forces arising from gravity and buoyancy. The equations of motion of a rigid body submerged in a viscous fluid subjected to dissipative and potential forces are given by the forced affine connection control system:*

$$\widetilde{\nabla}_{\gamma'}\gamma' = \mathbb{G}^\sharp\left(P\left(\gamma\left(t\right)\right)\right) + \sum_{i=1}^{6} \mathbb{I}_i^{-1}\left(\gamma\left(t\right)\right)\sigma_i\left(t\right). \tag{2.8}$$

This proposition provides the equations of motion relevant to the LAUV. Next, we define kinematic reductions and decoupling vector fields, two concepts which are paramount to our study here. These ideas will allow us to determine levels of controllability for an under-actuated AUV, the primary focus of the application of this theory.

3 Kinematic Reductions and Decoupling Vector Fields

When a vehicle is fully-actuated, it is capable of moving in six degrees of freedom (DOF), that is three translation motions along three principal axes, and three rotations about said axes. In this scenario, the motion planning problem, i.e. whether or not a vehicle can reach a final position and orientation from an initial one, is trivial. However, AUVs are designed to transverse environments that are particularly hazardous. They must navigate through tight tunnels, fast-flowing currents, pass by unstable landscapes, and many other unpredictable situations. Due to these potential dangers, it is highly likely that the vehicle itself may experience some sort damage. This could be an electrical malfunction or external damage to the motors. In any case, the failure of a motor may result in a loss of any number of degrees of freedom. In addition, it is fairly common that the manner in which AUVs are constructed results in the AUV being naturally under-actuated. We define an under-actuated vehicle to be one not capable of performing a motion in all six DOF. It is in this under-actuated scenario that the vehicle must be capable of assessing what trajectories it is capable of accomplishing, be the trajectories complex or simple. This is precisely what kinematic reductions can offer us and, as we'll see, what they can be used to accomplish.

In [15] the authors develop and define kinematic reductions for controlled mechanical systems. However, this presentation does not account for dissipative or potential forces. We remedy this by adjusting the controls at a later time as seen in [16]. Before we introduce the kinematic reduction, we need the following definition. Let us denote our affine connection control system by $\sum_{dym} = \left(Q, \nabla, \mathcal{D}, \mathcal{Y}, \mathbb{R}^6\right)$ (notice that this differs from the *forced* affine connection control system defined previously as there are no force vectors) and a **driftless system** by $\sum_{kin} = \left(Q, \mathcal{X}, \mathbb{R}^{\tilde{m}}\right)$, where $\mathcal{X} = \{X_1, \ldots, X_{\tilde{m}}\}$, $X_i \in \Gamma^\infty\left(TQ\right)$ $(\tilde{m} \leq 6)$. This leads to the definition of a controlled trajectory for our driftless system:

Definition A **controlled trajectory** of our driftless system

$$\gamma'(t) = \sum_{i=1}^{\tilde{m}} \sigma_i(t)\, \mathbb{I}_i^{-1}(\gamma(t)), \tag{3.1}$$

is the pair (γ, σ) where

(i) $\gamma : I \to Q$ and $\sigma : I \to U$ for some $I \subset \mathbb{R}$,

(ii) σ is a kinematic control (abs. continuous)

(iii) (γ, σ) satisfies [3.1].

Similarly, a **controlled trajectory** for our affine connection control system

$$\widetilde{\nabla}_{\gamma'}\gamma' = \sum_{i=1}^{m} \tilde{\mathbb{I}}_i^{-1}(\gamma(t))\, \sigma_i(t) \tag{3.2}$$

where we ask that $\gamma'(t_0)$ satisfies velocity constraints and $\gamma'(t)$ is differentiable.

We now present the formal definition of a kinematic reduction:

Kinematic Reduction Let $\sum_{dyn} = \left(\mathcal{Q}, \nabla, \mathcal{Y}, \mathbb{R}^6\right)$ be a C^∞ affine connection control system with $\mathtt{Y}$ having locally constant rank. A driftless system $\sum_{kin} = \left(\mathcal{Q}, \mathcal{X}, \mathbb{R}^{\tilde{m}}\right)(\tilde{m} \leq 6)$ is a kinematic reduction of $\sum_{dyn}$ if

i $\mathtt{X}$ is a locally constant rank subbundle of $T\mathcal{Q}$ and if,

ii for every controlled trajectory (γ, u_{kin}) for $\sum_{kin}$, there existsu_{dyn} such that (γ, u_{dyn}) is a controlled trajectory of $\sum_{dyn}$.

Here, $\mathtt{Y}$ ($\mathtt{X}$) denotes the C^∞- distribution of $\mathcal{Y}$ ($\mathcal{Y}$) The rank of the kinematic reduction $\sum_{kin}$ at a point $q \in \mathcal{Q}$ is the rank of $\mathtt{X}$ at q.

By definition, the kinematic reduction of [3.2] is of the form

$$\gamma'(t) = \sum_{i=1}^{m} \tilde{\mathbb{I}}_i^{-1}(\gamma(t))\, \sigma_i(t) \tag{3.3}$$

where $Z_i \in \Gamma^\infty(TQ)$, $\sigma^{kin}(t)$ is the associated kinematic control, and $r \leq m$ is the rank of the reduction. Moreover, there exists dynamic controls $\sigma_i(t)$, such that the solution $\left(\gamma, \sigma_i^{kin}\right)$ of [3.3] also provides that (γ, σ_i) is the solution of [3.2]. Theorem 8.18 of [15] provides the characterization of kinematic reductions. Here, we make use of kinematic reductions of rank one, called **decoupling vector fields** (DVFs). A corollary of Theorem 8.18 characterizes decoupling vectors fields for us:
Corollary 1. *A vector field $X \in \Gamma^\infty(TQ)$ is a decoupling vector field for the affine connection control system $\sum_{dym} \iff X, \nabla_X X \in \Gamma^\infty(\mathtt{Y})$.*

Kinematic motions are integral curves of DVFs. For this reason, it is crucial for us to characterize all the DVFs for a given actuation scenario. The details of this characterization can be found in [17] and [18],we will only provide the outline and results here. Given a set of input vector fields $\tilde{\mathcal{I}}^{-1} = \left\{\tilde{\mathbb{I}}_1^{-1}, \ldots, \tilde{\mathbb{I}}_m^{-1}\right\}$, a vector field V is decoupling iff $V \in Span\left(\tilde{\mathcal{I}}^{-1}\right)$ and $\tilde{\nabla}_V V \in Span\left(\tilde{\mathcal{I}}^{-1}\right)$, that is

$$V = \sum_{i=1}^{m} h_i \tilde{\mathbb{I}}_i^{-1} \text{ and } \tilde{\nabla}_V V = \sum_{i=1}^{m}\sum_{j=1}^{m} h_i h_j \tilde{\nabla}_{\sum_{i=1}^{m} h_i \tilde{\mathbb{I}}_i^{-1}} \sum_{j=1}^{m} h_j \tilde{\mathbb{I}}_j^{-1} = \sum_{i=1}^{m}\sum_{j=1}^{m} h_i h_j \tilde{\nabla}_{\tilde{\mathbb{I}}_i^{-1}} \tilde{\mathbb{I}}_j^{-1} \equiv 0 \bmod \tilde{\mathcal{I}}^{-1}$$

for $h_i \in \mathbb{R}$. Recall that $\{X_1, \ldots, X_6\} \in Q$ is the standard basis, and in general to compute $\nabla_{\mathbb{I}_i^{-1}} \mathbb{I}_j^{-1}$ in terms of the standard basis, the equation below is used

$$\begin{aligned}\mathbb{G}\left(\nabla_{\mathbb{I}_i^{-1}} \mathbb{I}_j^{-1}, X_k\right) =& \frac{1}{2}[\mathcal{L}_{\mathbb{I}_i^{-1}}\left(\mathbb{G}\left(\mathbb{I}_j^{-1}, X_k\right)\right) + \mathcal{L}_{\mathbb{I}_j^{-1}}\left(\mathbb{G}\left(X_k, \mathbb{I}_i^{-1}\right)\right) - \mathcal{L}_{X_k}\left(\mathbb{G}\left(\mathbb{I}_i^{-1}, \mathbb{I}_j^{-1}\right)\right) \\ &+ \mathbb{G}\left(\left[\mathbb{I}_i^{-1}, \mathbb{I}_j^{-1}\right], X_k\right) - \mathbb{G}\left(\left[\mathbb{I}_i^{-1}, X_k\right], \mathbb{I}_j^{-1}\right) - \mathbb{G}\left(\left[\mathbb{I}_j^{-1}, X_k\right], \mathbb{I}_i^{-1}\right)]\end{aligned}$$

where $\mathbb{G}(Y_i, Y_j) = Y_i^T \mathbb{G} Y_j$ is the Riemannian inner product representing the kinetic energy metric on $SE(3)$ and $\mathcal{L}_*(\mathbb{G}(*,*)) = 0$ for any left-invariant frame field on $SE(3)$. This leads us to the covariant derivatives for the affine connection control system $\tilde{\nabla}$, computed in [17]. We provide them here in Table [2]:

$(1,1)$	$\frac{D_1}{m_1} X_1$	$(2,1)$	$-\frac{m_1 - m_2}{2j_3} X_6$
$(1,2)$	$-\frac{m_1 - m_2}{2j_3} X_6$	$(2,2)$	$\frac{D_2}{m_2} X_2$
$(1,3)$	$-\frac{m_3 - m_1}{2j_2} X_5$	$(2,3)$	$\frac{m_3 - m_2}{2j_1} X_4$
$(1,4)$	0	$(2,4)$	$-\frac{m_3 - m_2}{2m_3} X_3$
$(1,5)$	$\frac{m_3 - m_1}{2m_3} X_3$	$(2,5)$	0
$(1,6)$	$\frac{m_1 - m_2}{2m_2} X_2$	$(2,6)$	$\frac{m_1 - m_2}{2m_1} X_1$
$(3,1)$	$-\frac{m_3 - m_1}{2j_2} X_5$	$(4,1)$	0
$(3,2)$	$\frac{m_3 - m_2}{2j_1} X_4$	$(4,2)$	$\frac{m_2 + m_3}{2m_3} X_3$
$(3,3)$	$\frac{D_3}{m_3} X_3$	$(4,3)$	$-\frac{m_2 + m_3}{2m_2} X_2$
$(3,4)$	$-\frac{m_3 - m_2}{2m_2} X_2$	$(4,4)$	$\frac{D_4}{j_1} X_4$
$(3,5)$	$\frac{m_3 - m_2}{2m_1} X_1$	$(4,5)$	$\frac{j_3 + j_2 - j_1}{2j_3} X_6$
$(3,6)$	0	$(4,6)$	$-\frac{j_3 + j_2 - j_1}{2j_2} X_5$
$(5,1)$	$-\frac{m_1 + m_3}{2m_3} X_3$	$(6,1)$	$\frac{m_1 + m_2}{2m_2} X_2$
$(5,2)$	0	$(6,2)$	$-\frac{m_1 + m_2}{2m_1} X_1$
$(5,3)$	$\frac{m_1 + m_3}{2m_1} X_1$	$(6,3)$	0
$(5,4)$	$-\frac{j_3 - j_2 + j_1}{2j_3} X_6$	$(6,4)$	$-\frac{j_3 - j_2 - j_1}{2j_2} X_5$
$(5,5)$	$\frac{D_5}{j_2} X_5$	$(6,5)$	$\frac{j_3 - j_2 - j_1}{2j_1} X_4$
$(5,6)$	$\frac{j_3 - j_2 + j_1}{2j_1} X_4$	$(6,6)$	$\frac{D_6}{j_3} X_6$

Table 2: Covariant derivatives in the standard basis X_i notation for the modified affine connection $\tilde{\nabla}$. Here (i,j) stands for $\tilde{\nabla}_{\mathbb{I}_i^{-1}} \mathbb{I}_j^{-1}$

Table 2 now allows us to compute decoupling vector fields V given an under-actuated scenario. We'll apply this knowledge in the subsequent section to the LAUV.

3.1 Control Strategy

We now present the basic idea behind the strategy used to compute the controls to perform kinematic motions. Again, a kinematic motion is an integral curve of a DVF. Table 2 shows that any DVF is a linear combination of pure motions. Thus, to execute more complex motions, we must concatenate kinematic motions. To ensure that the kinematic motions are feasible, we require the vehicle to be

at rest (no velocity) between motions. To do this, we reparameterize the path $\gamma : I \to Q$ with a C^2-reparameterization $\tau_j : [0, t_j] \to I$ such that $\tau_j'(0) = 0$ and $\tau_j'(t_j) = 0$. As such, the reparameterized trajectory $\gamma \circ \tau([0, t_j])$ follows the same path as $\gamma(I)$, but begins and ends at rest. Since γ is an integral curve for V, we have the $\gamma'(t) = V(\gamma(t))$. Using the product rule for covariant derivatives and the chain rule for differentiation,

$$\begin{aligned}\nabla_{(\gamma\circ\tau)'(t)}(\gamma\circ\tau)'(t) &= \nabla_{\tau'(t)\cdot\gamma'(\tau(t))}\tau'(t)\cdot\gamma'(\tau(t)) \\ &= (\tau'(t))^2 \nabla_{\gamma'(\tau(t))}\gamma'(\tau(t)) + \tau''(t)\cdot\gamma'(\tau(t)) \\ &= (\tau'(t))^2 \nabla_V V(\gamma\circ\tau(t)) + \tau''(t)\cdot V(\gamma\circ\tau(t))\end{aligned}$$

The associated control vector σ^{kin} is then derived from

$$\sum_{i=1}^{m} \sigma_i^{kin}(t)\,\tilde{\mathbb{I}}_i^{-1}(\gamma\circ\tau(t)) = (\tau'(t))^2 \tilde{\nabla}_V V(\gamma\circ\tau(t)) + \tau''(t)\cdot V(\gamma\circ\tau(t)) \tag{3.4}$$

and Proposition 13.5 of [15] guarantees then that $\left(\gamma\circ\tau, \sigma^{kin}\right)$ is a controlled trajectory for the affine connection control system equation [3.2]. Now that we have the controls for the driftless system, we add in the controls for the potential forces with the additional term $\overline{\sigma}$ such that

$$\mathbb{G}^{\sharp}(P(\gamma(t))) + \sum_{i=1}^{m} \overline{\sigma}_i(t)\,\tilde{\mathbb{I}}_i^{-1}(\gamma(t)) = 0. \tag{3.5}$$

Thus, the control $\sigma = \sigma^{kin} + \overline{\sigma}$ provides the controls for the complete under-actuated system

$$\tilde{\nabla}_{(\gamma\circ\tau)'(t)}(\gamma\circ\tau)'(t) = \mathbb{G}^{\sharp}(P((\gamma\circ\tau(t)))) + \sum_{i=1}^{m} \tilde{\mathbb{I}}_i^{-1}((\gamma\circ\tau(t)))\,\sigma_i(t) \tag{3.6}$$

meaning $(\gamma\circ\tau, \sigma)$ is a controlled trajectory. However, it is worth noting that equation [3.5] cannot always be satisfied. In general, the potential forces will always be acting on the vehicle, however, the controls to compensate this potential may be non-existant. For this reason, kinematic trajectories must be designed in a manner such that $\overline{\sigma}$ can be calculated in order for equation [3.5] to be satisfied. In other words, the reparameterization not only allows us to concatenate kinematic motions, but it also helps one meet constraints on the thrusters. The following section presents an implementation of this theory onto the LAUV.

4 Implementation on Lego AUV

The Lego AUV (LAUV) was built in order to have a cost-effective test-bed vehicle to implement this theory. Moreover, it offers an excellent outreach education tool for robotics, physics, and programming. In Appendix A, one can find a general description of the construction of the LAUV. Contained in this appendix is a short description of the RobotC program and graphical user interface developed for remote controlling this underwater vehicle. Here, we want to execute the previously developed geometric path planning theory onto the LAUV and discuss the results.

4.1 Degrees of Freedom and Underactuation

The LAUV has four vertical thrusters and two horizontal thrusters. Each motor is controlled individually and allows us to have control over five degrees of freedom. Specifically, the two horizontal thrusters contribute to surge and heave, while the four vertical one contribute to heave, pitch, and roll. Therefore, our set of input control vectors is given by $\tilde{\mathcal{I}}^{-1} = \left\{\mathbb{I}_1^{-1}, \mathbb{I}_3^{-1}, \mathbb{I}_4^{-1}, \mathbb{I}_5^{-1}, \mathbb{I}_6^{-1}\right\}$. Since our vehicle is under-actuated, we wish to find controls which satisfy equation [3.6] for a given DVF. In the previous section, we saw the full list of covariant derivatives in terms of the standard basis on Q. The characterization for DVFs states explicitly that a vector field is only decoupling if it is in the span of the input control vectors. In [18], the DVFs in terms of linear combinations of the input control vectors for any given under-actuated scenario is detailed by Proposition 7. We make use of one portion of the proposition provided and apply it to the LAUV.

Proposition 2. *For the Lego AUV, which is a five input system consisting of two translations and three rotations, under the influence of viscous drag, we assume $C_B = C_g$. The following vector fields characterize all the decoupling vector fields in terms of the five degrees of freedom that can be input into the system.*

Two Translations, Three Rotations: $\tilde{\mathcal{I}}^{-1} = \left\{\mathbb{I}_1^{-1}, \mathbb{I}_3^{-1}, \mathbb{I}_4^{-1}, \mathbb{I}_5^{-1}, \mathbb{I}_6^{-1}\right\}$. For a kinematically unique system, the decoupling vector fields are

(a) $V = h_a\mathbb{I}_a^{-1} + h_{a+3}\mathbb{I}_{a+3}^{-1} + h_5\mathbb{I}_5^{-1}$ *where* $a \in \{1,3\}$,

(b) $V = h_a\mathbb{I}_a^{-1} + h_b\mathbb{I}_b^{-1} + h_5\mathbb{I}_5^{-1}$ *where* $\{a,b\} = \{1,3\}$ *or* $\{a,b\} = \{4,6\}$.

Proof. Strictly by calculating the covariant derivative for $\nabla_V V$ and using Table 2 ,the geometric acceleration for any $\gamma(t) \in \mathcal{Q}$ is given by

$$\nabla_{\gamma'(t)}\gamma'(t) = \mathbb{I}_1^{-1}\sigma_1(t) + \mathbb{I}_3^{-1}\sigma_3(t) + \mathbb{I}_4^{-1}\sigma_4(t) + \mathbb{I}_5^{-1}\sigma_5(t) + \mathbb{I}_6^{-1}\sigma_6(t).$$

As such, for any general vector field $V = h_1\mathbb{I}_1^{-1} + h_3\mathbb{I}_3^{-1} + \sum_{i=4}^{6} h_i\mathbb{I}_i^{-1}$

$$\begin{aligned}
\nabla_V V &= \nabla_{h_1\mathbb{I}_1^{-1} + h_3\mathbb{I}_3^{-1} + \sum_{i=4}^{6} h_i\mathbb{I}_i^{-1}} h_1\mathbb{I}_1^{-1} + h_3\mathbb{I}_3^{-1} + \sum_{i=4}^{6} h_i\mathbb{I}_i^{-1} \\
&= \frac{1}{2}(h_1h_3\left(\frac{m_1 - m_3}{j_2}\right)X_5 + h_1h_5\left(\frac{-m_1}{m_3}\right)X_3 + h_1h_6\left(\frac{m_1}{m_2}\right)X_2 \\
&+ h_3h_4\left(\frac{-m_3}{m_2}\right)X_2 + h_3h_5\left(\frac{m_3}{m_1}\right)X_1 + h_4h_5\left(\frac{j_2 - j_1}{j_3}\right)X_6 \\
&+ h_4h_6\left(\frac{j_1 - j_3}{j_2}\right)X_5 + h_5h_6\left(\frac{j_3 - j_2}{j_1}\right)X_4)\left(\mathrm{mod}\ \mathcal{I}^{-1}\right) \\
&= \frac{1}{2}\left(h_1h_6\left(\frac{m_1}{m_2}\right)X_2 + h_3h_4\left(\frac{-m_3}{m_2}\right)X_2\right)\left(\mathrm{mod}\ \mathcal{I}^{-1}\right)
\end{aligned}$$

So,h_1 or h_6 and h_3 or h_4 have to be zero.The result follows

□

Table 3 provides the full list of decoupling vector fields possible implementable for the input control vector fields $\tilde{\mathcal{I}}^{-1} = \left\{\mathbb{I}_1^{-1}, \mathbb{I}_3^{-1}, \mathbb{I}_4^{-1}, \mathbb{I}_5^{-1}, \mathbb{I}_6^{-1}\right\}$.

$V = h_1\mathbb{I}_1^{-1} + h_4\mathbb{I}_4^{-1} + h_5\mathbb{I}_5^{-1}$	$V = h_3\mathbb{I}_3^{-1} + h_6\mathbb{I}_6^{-1} + h_5\mathbb{I}_5^{-1}$
$V = h_1\mathbb{I}_1^{-1} + h_3\mathbb{I}_3^{-1} + h_5\mathbb{I}_5^{-1}$	$V = h_4\mathbb{I}_4^{-1} + h_5\mathbb{I}_5^{-1} + h_6\mathbb{I}_6^{-1}$

Table 3: Decoupling Vector Fields for the LAUV

It is clear that other decoupling vector fields are realizable by letting $h = 0$. With these DVFs, we can construct trajectories that are practical and can be realized.

4.2 Simulations

In this section, we provide simulations of the LAUV along several different paths, each of varying complexity. Table 4 lists all the pertinent parameters of the vehicle. Added mass values were calculated using slender body theory, see Fossen [13], which calculates the added mass by using 2-D cross section in a particular plane and integrating over the length in the normal direction. The theoretical added masses were computed for each component of the LAUV (hull, RCX motors, XL motors, motor multiplexer, and batteries) and summed up. The same is true of the added inertias. The translational drag values were experimentally determined whereas the rotational drags are estimates. Note that the LAUV is positively buoyant; we will have to account for this in our controls. Figure 4.1 provides a visualization of the LAUV in the simulation environment as well as the thruster configuration.

mass = 2.2680 kg	
$m_1 = 4.0021$ kg	$j_1 = 0.0153$ kg m^2
$m_2 = 4.3360$ kg	$j_2 = 0.2213$ kg m^2
$m_3 = 7.4976$ kg	$j_3 = 0.0178$ kg m^2
dimensions - (0.19, 0.13, 0.08) meters	
$W = \text{mass}g = 22.241$ N	$B = \rho g\mathcal{V} = 24.987$ N
$C_B = \{0, 0, -0.001\}$meters	$C_G = \{0, 0, 0\}$
$D_1 = 7.569$ kg/m^2	$D_2 = 7.717$ kg/m^2
$D_3 = 11.764$ kg/m^2	$D_4 = 4$ kg/m^2
$D_5 = 4$ kg/m^2	$D_6 = 8$ kg/m^2

Table 4: Main Dynamic Parameters for the Lego AUV

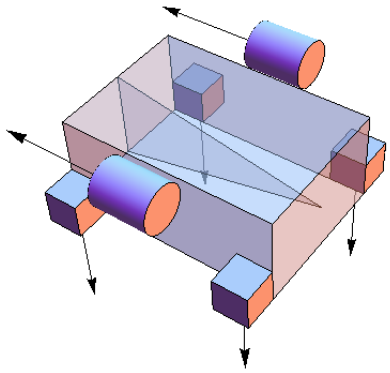

Figure 4.1: Simulated visualization of LAUV. Inner triangle "points" to fore (positive b_1 axis.)

Now, in order to get the controls to the LAUV, we need to apply a linear transformation to the control vector so that we know how to allocate the thrust to the individual motors. This thruster allocation matrix for the LAUV is given by

$$T = \begin{pmatrix} 0 & 0 & 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 0 & 0 \\ -0.0805 & -0.0805 & 0.0805 & 0.0805 & 0. & 0. \\ 0.0795 & -0.0795 & -0.0795 & 0.0795 & -0.08 & -0.08 \\ 0. & 0. & 0. & 0. & 0.095 & -0.095 \end{pmatrix}.$$

Clearly this is singular, so to convert from the geometric controls to the force necessary from the thrusters, we take a pseudo inverse and apply this new matrix to our geometric controls. Using this information and the geometric concepts, we simulate a few missions of varying complexity.

4.2.1 Mission 1

We begin with a simple surge motion. For this mission, we wish to surge a distance of 2 meters at a depth of 1 meter, which we wish to maintain. Our proposition above tells us that indeed the vector field $V = h_1 \mathbb{I}_1^{-1}$ is decoupling. With $h_1 = m_1$, we have that $V = X_1$, the integral curves of which produce the surge trajectory we want. In fact, there is only one integral curve to follow, the straight line from $\eta_i = (0, 0, 1, 0, 0, 0)$ to $\eta_f = (2, 0, 1, 0, 0, 0)$. To ensure the Lego motors can produce the necessary thrust to realize this motion, we will reparameterize $\gamma(t)$ to have an average velocity of $\frac{1}{30}$ m/s. If we let $\tau(t) = \frac{90t^2 - t^3}{1800}$, the correct reparameterization of our path is produced. Using Table 2, we can compute the dynamic control from **??**

$$\sigma_1(t)\,\mathbb{I}_1^{-1}(\gamma \circ \tau(t)) = (\tau'(t))^2\,\tilde{\nabla}_{X_1} X_1(\gamma \circ \tau(t)) + \tau''(t) \cdot X_1(\gamma \circ \tau(t)) = \frac{D_1}{m_1}\left(\frac{180t - 3t^2}{1800}\right)^2 + \frac{180 - 6t}{1800}$$

However, recall that the geometric controls do not account for the potential forces. From Table 4, we see that LAUV is positively buoyant by 2.746 N. As such, we must account for this in our controls. The controls for the with trajectory are then

$$\sigma(t) = \begin{pmatrix} m_1\sigma_1(t) \\ 0 \\ 2.746 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The figure below shows the necessary controls after the thruster allocation matrix is applied.

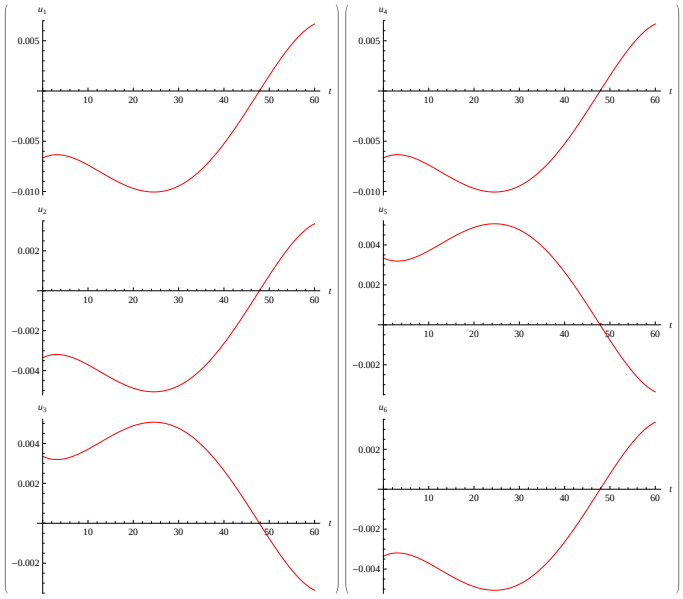

Figure 4.2: Thruster commands for a pure surge. The top two plots show the necessary thrust for the two horizontal XL motors, while the bottom four are for the vertical RCX motors.

By solving the kinetic system and the dynamic system numerically, we obtain the positions and velocities for this kinematic motion seen in the figure below.

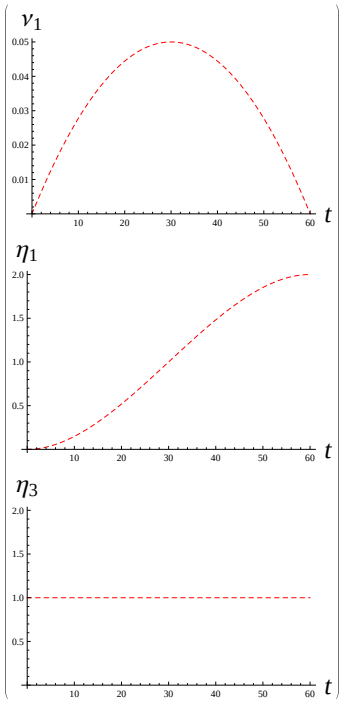

Figure 4.3: Velocity (ν_1) and Position (r_1) Plot in Inertial Frame

It is logical to ask why the vertical facing motors are providing any thrust. The answer is simply that the two horizontal XL motors are not in the same plane as the center of gravity. As such, any thrust from these two motors induces a pitch/roll that the vertical motors must account for to keep the LAUV from pitching. In the end, we can succesfully complete our mission.

4.2.2 Mission 2

The second mission is more simple than the first and involves a pure heave. In this scenario, the LAUV begins 1 meter below the surface and we wish to surface. Remember, the positive "z axis" (which we denoted s_3) is in the direction of gravity. And so, we expect to see the inertial frame position of the LAUV's depth deform from 1 to 0. First, let us compute our controls. But again, the control is simple as we need only to consider the integral curves of X_3. We do this by choosing the decoupling vector field $V = m_3 \mathbb{I}_3^{-1}$. Thus, the only integral curve to follow is from $\eta_i = (0,0,1,0,0,0)$ to $\eta_f = (0,0,0,0,0,0)$, another straight line for the trajectory $\gamma(t) : [0,1] \to Q$. We reparameterize with $\tau(t) = \frac{90t^2 - t^3}{1800}$ in order to have this motion last for 60 seconds. The resulting dynamic controls are

$$\sigma_3(t)\,\mathbb{I}_3^{-1}(\gamma \circ \tau(t)) = (\tau'(t))^2\,\tilde{\nabla}_{X_3} X_3(\gamma \circ \tau(t)) + \tau''(t) \cdot X_3(\gamma \circ \tau(t)) = \frac{D_3}{m_3}\left(\frac{180t - 3t^2}{1800}\right)^2 + \frac{180 - 6t}{1800}.$$

Yet again, we must account for the fact that our vehicle is positively buoyant. Ergo, the true controls are:

$$\sigma(t) = \begin{pmatrix} 0 \\ 0 \\ m_3\sigma_3(t) + 2.746 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Figure 4.4 shows these controls as applied to the Lego motors. Figure 4.5 plots the inertial velocity and position of the LAUV as it progresses through this trajectory.

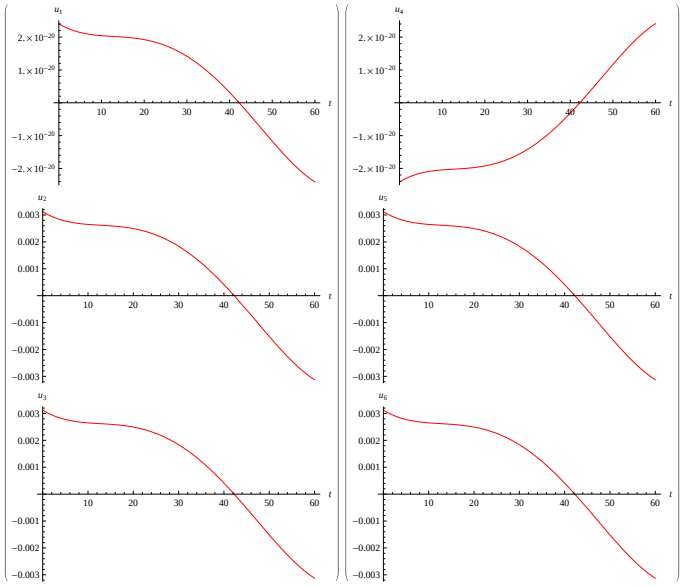

Figure 4.4: Thruster Commands for a Pure Heave 1m Deep. The left column and first plot of the right show the necessary thrust for the four are for the vertical RCX motors. The bottom right-hand two are for the XL motors.

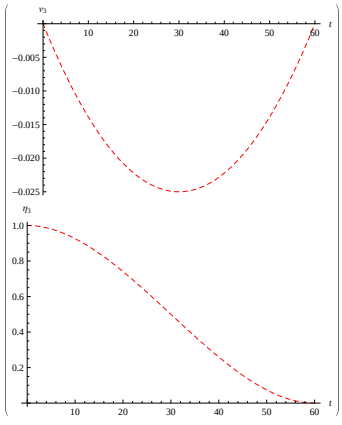

Figure 4.5: Velocity (ν_3) and Position (r_3) Plots in Inertial Frame

In figure 4.4, we see that the only active motors are the vertical ones, as it should be. Finally, in figure 4.5, we see that the LAUV succesfully surfaces at zero velocity.

4.2.3 Mission 3

For the third and final mission of this paper, we wish to execute a more complex motion. This motion will demonstrate not only the significance of the reparameterization, but highlights specifically our ability to concatenate motions successfully. This motion will consist of four trajectories, and thus four different DVFs. In this mission, we wish to surface yet again however in this instance we wish to do so in a nontrivial manner. AUVs exploring underwater labrynths or changing, hostile environments typically to not have the luxury of a trivial surface and it is highly plausible they must navigate under "overpasses" (or over ridges). This mission illustrates a nice trajectory in such cases. For the first motion, we LAUV will "arc" up towards the surface as if it were overcoming a ridge. Second, the LAUV will reorient itself 120 degrees. Third, the LAUV will then "arc" up again, but this time in a fashion as if it were to avoid a an over hanging rock cliff or "ceiling" if you will. Finally, we simply allow the vehicle to orient itself in a manner that there is no pitching.

We begin our simulation again with an initial configuration, $\eta_i^a = \begin{pmatrix} 0 & 0 & 1.2 & 0 & -\frac{\pi}{4} & 0 \end{pmatrix}$. Here, we see that the LAUV begins 1.2 meters deep and pitched initially at a 45 degree angle. Note that this 45 degree angle is in fact towards the surface as the $+z$ (s_3) direction is in the direction of gravity. If we wished (as step four implies) we could calculate the necessary controls to maintain this pitching angle, and so it is not unreasonable for us to begin with such an initial configuration. We wish to end at a final configuration

4.3 Experimental Implementation on Lego AUV

At this time, there have not been any experiments comparing these theoretical simulations to actual implementations on the LAUV. However, it is intended that these missions will be carried out to show these mathematical methods are practical.

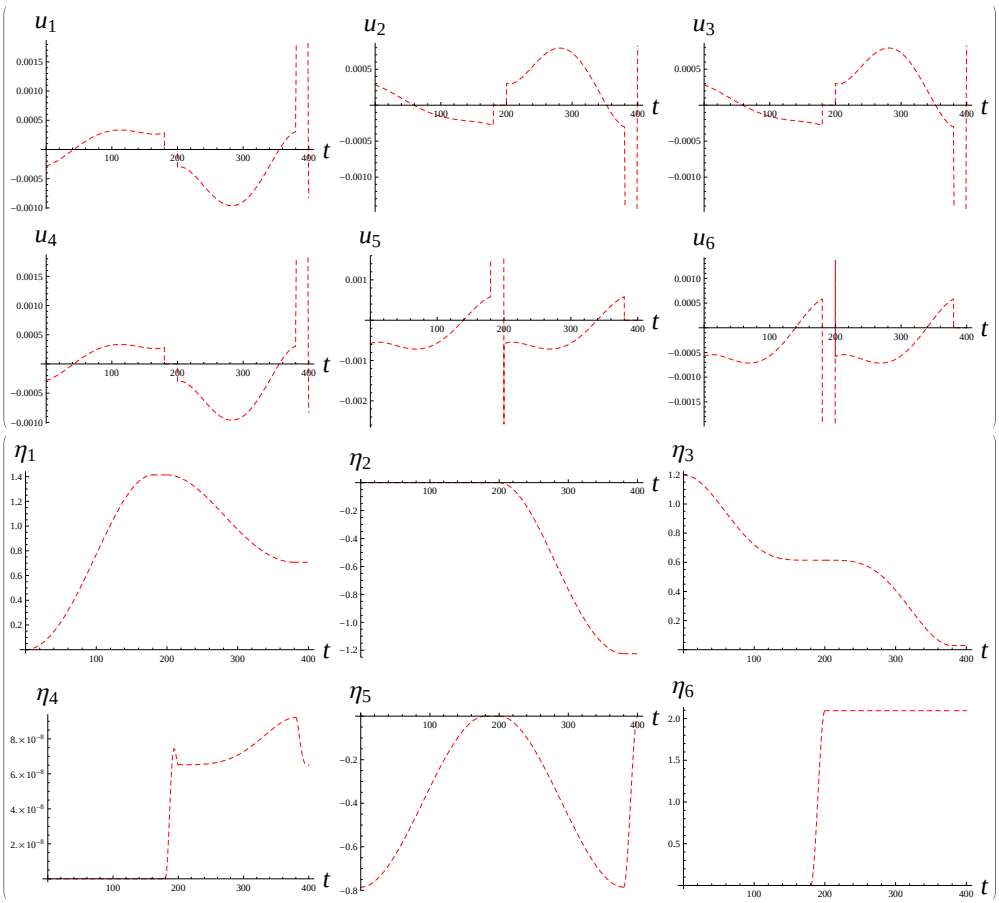

Figure 4.6: Thruster Commands (u_i), Positions (η_1, η_2, η_3), and Orientations (η_4, η_5, η_6) for Concatenated Motions. Horizontal Motors: u_1 & u_4; Front Vertical Motors: u_5 & u_6; Rear Vertical Motors: u_2 & u_3

5 Appendix A Lego AUV Construction

As we've seen, the Lego NXT Mindstorms kit can be used for much more than a simple children's toy, the capabilities limited only to imagination of the Lego engineer. For this Masters thesis, it was desirable to have a cost effective AUV test-bed vehicle to validate the theoretical aspect of this paper. The Lego Mindstorms provided the perfect platform to build an AUV "cheaply" compared to much more advanced and sophisticated AUVs, while providing a tremendous outreach education tool. Moreover, there would be no cost to operate this AUV, again providing an excellent tool to corroborate the geometric theory to be discussed in subsequent section. A complete list of the necessary pats to build the LAUV can be found in the following table:

Item	Quantity
Pelican Case 1050	1
RCX Motor Multiplexer (MTRMX-Nx)	1
RCX Motor	4
LEGO Power Functions XL Motor	2
HiTechnic NXT Gyro Sensor	1
HiTechnic NXT Accelerometer/ Tilt Sensor	2
XBee Explorer USB	1
NXTBee Naked	1
XBee Pro 900 MHz	2
BULGIN - PX0410/08S/6065 - SOCKET, FREE, 8WAY	1
BULGIN - PX0410/04S/4550 - SOCKET, FREE, 4WAY	1
BULGIN - PX0412/08P - PLUG, CHASSIS MOUNT, 8WAY	1
BULGIN - PX0412/04P - PLUG, CHASSIS MOUNT, 4WAY	1
BULGIN - SA3348/1. - PIN, SOLDER, 22-26AWG PK10	2
BULGIN - SA3347/1. - SOLDER SOCKET, 22-26AWG, PK10	2
LEGO MINDSTORMS Education NXT Base Set	1
Sealed Lead Acid Battery 6V 1.2-1.3 Ah - Rechargeable	2
40 mm Propeller	6

Table 5: Parts List for Lego AUV

Building the Lego AUV, LAUV, proved to be a challenge in its own right. Unfortunately, it was not as simple as throwing the NXT into an underwater housing and operating, motors needed to be attached through the housing and compactness was desirable. To this end, the Pelican case provided an adequate solution, having ample space to house the payload consisting of the NXT computer, two sensors, and the XBee radio. The NXT computer communicates with the motors and sensors via 6-position modular connectors, RJ12 cables, which were cut and spliced through waterproof Bulgin sockets and plugs. In order to connect the motors to the NXT computer, holes were drilled into the casing allowing us to epoxy the Bulgin sockets into said holes. First, the plug end of the RJ12 cables which would be jacked into the NXT computer had the wires protruding from the plug soldered to gold sockets which then were placed into the Bulgin sockets. The motors themselves were also soldered to gold pins, these pins then placed into the Bulgin plugs. Now, the motors consist of two separate sets, the two XL RCX motors and the four RCX motors, each having its own socket-plug pairing. The NXT computer has seven separate ports for input/output, three reserved for motors and 4 for sensors. Two of the motor ports were reserved for the two XL RCX motors and one sensor port was connected to an RCX motor multiplexer which basically delegated power to the remaining four RCX motors. With the motor multiplexer housed in a waterproof casing and a frame of Legos built around the Pelican case, the motors were then aligned in a particular configuration. The two

XL motors contribute to the motions surge and yaw, while the four RCX motors contribute to the motions heave, roll, and pitch. The figure below shows one version of the LAUV, as the position of the multiplexer was constantly in flux. For the purposes of this thesis, the image below represents the LAUV which under went all of the experimental procedures. However, as the LAUV is also intended to be an outreach tool, this will most likely not be the final version, as access to the NXT computer itself is troublesome.

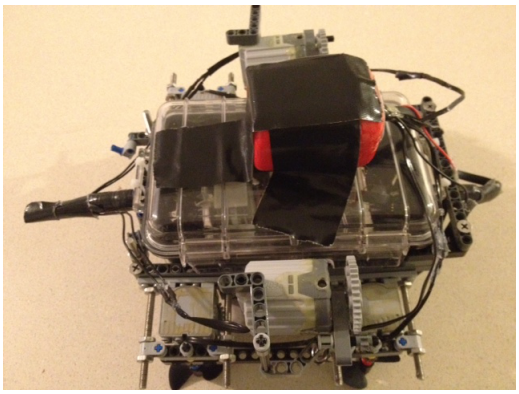

Figure 5.1: Lego AUV partially assembled

5.1 Payload

The payload consists of the NXT computer, two HiTechnic sensors, and a Digi XBee radio. We've discussed the NXT computer to an extent already, so now we'll focus on the radio and sensors. HiTechnic is a third-party component manufacturer which builds Lego certified NXT sensors and other NXT related products. For this project, we chose to use the HiTechnic Accelerometer/Tilt Sensor and Gyro Sensor. The HiTechnic Accelerometer/Tilt sensor measures the acceleration in three axis, $x, y,$ and z. The measurements are given as raw decimals in the range of -200 to $+200$ every hundredth of a second and one can use this to derive a tilt on the sensor. For the LAUV, we are only concerned with the orientation of the vehicle, and this is precisely what the sensor is used to accomplish. In the appendix, we show that the LAUV is not only positively buoyant, but the center of mass and center of buoyancy differ. Therefore, restoring forces act on the vehicle about the x- and y-axes, limiting roll and pitch to the interval $(-90, 90)$. Thus, we can easily map the raw output values to this interval via $x \mapsto 0.45x$. Next, the HiTechnic Gyro sensor measures angular velocity in the vertical plane of the sensor itself. For our purposes, we can easily integrate the angular velocity over a short time interval to obtain an angle. The gyro sensor is positioned inside the casing with the NXT computer in such a manner that it can measure a heading, or the yaw angle, via this integration. With these two sensors, the orientation of the vehicle is known entirely. Finally, we discuss the Digi XBee radio. Specifically, communications between the NXT computer and a PC is accomplished via the Digi XBee Pro 900 which was configured to utilize the NXT's RS-485 line used for high-speed communications. The XBee Pro operates at a frequency band of about 900 MHz with an indoor range up to 100 meters. This is superior to the NXT computer's built-in bluetooth, which only has a range of about 10 meters, and offers us the underwater communication desired.

Now, the LAUV has the capability to be autonomous, that is, navigate freely in its environment with little to no human interaction. We will investigate its autonomy with the geometric path planning to be detailed subsequently. However, the LAUV was originally a remote operated vehicle (ROV), controlled with a gamepad controller via a graphical user interface (GUI) and the XBee

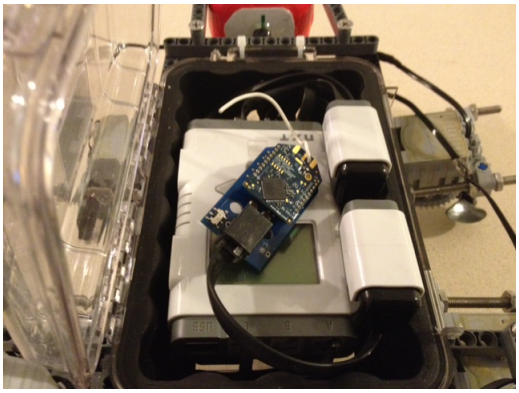

Figure 5.2: Payload of the Lego AUV. Consists of Lego NXT computer, two sensors, and an XBee radio

radios. We will also discuss the GUI in a later section. The primary purpose of controlling remotely was for testing purposes as well as developing a robot that would offer outreach applications.

5.2 RobotC

RobotC is a C-based programming language used to write and debug programs for a variety of robotic platforms. It is particularly useful to program for the Lego NXT. The Lego Mindstorms NXT does come with standard programming software, NXT-G, which features a 'drag and drop' environment for commands/operations to be compiled and downloaded to the NXT computer. Unfortunately, this software does not offer the complexity and freedom this project required. For this reason, the third-party RobotC was used, especially since RobotC even comes with drivers to utilize the HiTechnic sensors. With this C-based programming language, one can write and debug their NXT program and, when ready, compile and download their program to the NXT Intelligent Brick.

With RobotC, there were three main goals that needed to be reached. First, the NXT brick had to be able to communicate through the XBee radios with a computer. Second, controlling the four RCX motors had to be done in RobotC. Moreover, motor control in general needed to be accomplished through commands from the gamepad controller. Finally, the sensors had to initialized, calibrated, and provide the data that would allow us to determine the orientation of the LAUV.

5.3 C Sharp GUI Application

A graphical user interface (GUI) was constructed for this project using Microsoft Visual Basic C#. This environment was chosen solely for its simplicity and the ability to interact with game controllers with ease. Unfortunately, the software is not cross-platform, although one could modify the constructed application in another software development kit such as QT. This GUI provided a user-friendly platform for any individual to interact and control the Lego AUV remotely. One of the more interesting tasks (personally) on this project was the implementation of the gamepad controller into the Visual Basic environment. By no means is this a new idea or accomplishment. In fact, Visual Basic added the assembly reference DirectX precisely for creating applications that used the Microsoft XBOX controller. Now, the gamepad controller used is a typical analog controller with 13 buttons, two analog joysticks, and a directional pad (D-pad). The way the controller interacts with the LAUV is as follows. First, a particular button combination is pressed. The computer then

recognizes this combination as a particular character, which the GUI converts to a string and sends to the LAUV via the XBee radios. In RobotC, certain button combinations were predetermined to correspond to specific maneuvers. The following figure and table show how the mapping of the buttons to commands work. Note that L1, L2, R1, R2, "X", "△", "□", and "◯" are boolean buttons, "Left Ang" and "Right Ang" are left and right analog joysticks, and the Directional-Pad (D-Pad) consists of 8 boolean buttons, which we will refer to by the principal directions on a compass.

Figure 5.3: Image of the Gamepad Controller

Button(s)	Maneuver
L1	Full Thrust Power Forward Left Motor Only
R1	Full Power Forward Thrust Right Motor Only
L2	Full Power Reverse Thrust Left Motor Only
R2	Full Power Reverse Thrust Right Motor Only
D-Pad North	Fore Pitch in the Positive Direction of Grav. Acc.
D-Pad South	Fore Pitch in the Negative Direction of Grav. Acc.
D-Pad East	Starboard Roll
D-Pad West	Port Roll
X	Dive
□	Surface
L1 + R1	Surge in the Direction of the Bow
L2 + R2	Surge in the Direction of the Stern
D-Pad North + L1 + R1	Dive with a Pitch and Surge
D-Pad South + L1 + R1	Surface with a Pitch and Surge
◯+ Any D-Pad Direction	Plays Middle C to G beginning with North Button
△ + Any D-Pad Direction	Plays G#4 to B4 beginning with North Button

Table 6: List of Mappings of Game Pad Buttons to LAUV Maneuvers

Since this portion of the project was mostly intended for outreach education, the addition of buttons that would play sounds (which really works best in a small tank so it can be heard) was included. With additional multiplexers, one could extend this project even further such as adding robotic grips and pressure/depth sensors. The GUI also collects information from the Gyro and Tilt sensors as well. Both of these sensors allow us to determine the full orientation of the vehicle in live-time. The NXT computer has four internal timers, one of which we utilize. Initially, the

RobotC code begins with a heading of 0 (yaw angle is zero). From the onset programs initialization, the Gyro sensor reads the angular velocity data. We make use of the timers in order to calculate the new yaw angle. This is simply the sum of the previous yaw angle and the product of the Gyro's angular velocity and change in time. If the angle exceeds 360 degrees, it is reset to zero. Earlier we mentioned how we determine the roll and pitch angle. All this information is then passed over the radio to the computer, where MATLAB accepts the data and provides a visualization of the orientation of the LAUV.

6 Appendix B - Estimations of Drag Coefficients

All of the experiments performed for this LAUV were very crude in nature, but provided us with a foundation for parameters to be used in our simulations. Specifically, calculating the drag coefficients proved to be the most challenging and time-consuming. As we assumed the LAUV has three planes of symmetry (which it does not), we could assume the hydrodynamic drag matrix is diagonal (as with the added masses). There are many forms of drag for an AUV, but in our case we assumed pressure drag was prevailant. Thus, the calculation for drag coefficients is given by the equation $D = \frac{1}{2} C_D \rho A \left|\nu\right| \nu$, where C_D is the drag coefficient, ρ is the density of water, A is the reference area, and ν is the velocity of the vehicle. Allmendinger (1990) presents that the drag forces can be broken down to the main hull and its appendages. However, since this thesis is primarily intended to show that control of an AUV can be done using geometric methods, we admit there was no sense of urgency to be as accurate as possible. Again, we calculated the drag coefficients crudely, as most novel methods for estimating these parameters were unavailable.

To calculate the translational drag coefficients, a simple machine consisting of two pulleys, a three pound lead weight, a one pound lead weight, a scale, and the LAUV were used. With the LAUV submerged and neutrally bouyant, a string was fastened to the LAUV on the bow, stern, and keel to measure drag forces for surge, sway, and heave, repsectively. On the other end of the string, one of the lead wieghts were fastened and in between were the two pulleys, one fixed in the same plane as the LAUV and the other fixed 0.94 meters out of the water. Finally, the scale was attached to the pulley out of the water. The idea was very simple, with the LAUV and the other components of this apparatus fixed, the scale would read the tension of the string between the lead and the LAUV, which should be precisely the weight of the lead. We would then drop the lead from the 0.94 meter hieght and measure the tension in the rope as the lead fell. This measurement provided us with the force due to drag and was recorded with a camera attached to the scale. Off to the side, we estimated by stopwatch the time between the release of the lead and the time it hit the ground; this provided our measurements for average velocity. From here it was a simple matter of solving the above equation for C_D.

The calculation of the rotational drag coefficients were estimated using experiments in the following fashion. For roll and pitch, the LAUV was fixed at an angle (without thrusters) and then released, allowing it to naturally come to rest. During the LAUV restoring, we measured angular velocity using the gyroscope and a code written in MATLAB which would account for the potential forces. Without the potential forces, the only other forces acting on the vehicle are drag forces and we know exactly how the vehicle behaves. In the simulation environment, we tweaked the drag forces until the simulated righting motion of the LAUV was within a reasonable threshold of what was seen experimentally. The yaw drag coefficient was calculated similarly, but instead of being held at a fixed angle, the thrusters spun the vehicle until there was no acceleration. When the LAUV had no angular acceleration, the thrusters were killed and the vehicle came to a halt only under the influence of the drag force. However, the estimations were still the same using the simulations to determine an estimated coefficient.

References

[1] Kim S.H., Jeon J.W. (2009) Introduction for Freshmen to Embedded Systems Using LEGO Mindstorms, IEEE Transactions on Education, Vol. 52, No. 1.

[2] Behrens A., et. al (2010) MATLAB Meets LEGO Mindstorms—A Freshman Introduction Course Into Practical Engineering , IEEE Transactions on Education, Vol. 53, No.2.

[3] Kim Y. (2011) Control Systems Lab Using a LEGO Mindstorms NXT Motor System, IEEE Transactions on Education, Vol.54, No.3.

[4] Ferri B., et. al (2009) Signal Processing Experiments with the LEGO MINDSTORMS NXT Kit for Use in Signals and Systems Courses, 2009 American Control Conference, St. Louis, MO.

[5] Papadimitriou V., Papadopoulos E. (2007) Putting Low Cost Robotics Components to the Test, IEEE Robotics and Automation Magazine, 99-110.

[6] Chyba, M., Andonian, M., Rader, J. (2011) River Survey Using an Automous Vehicle, Special Edition of Ciencia & Tecnología de Buques/Ship Science and Technology, Vol. 4 (7).

[7] Ferri G., Jakuba M.V., Yoerger D.R. (2008) A novel method for hydrothermal vents prospecting using an autonomous underwater robot. IEEE International Conference on Robotics and Automation, 1055-1060. Pasadena, California.

[8] German C. R., Yoerger D. R., Jakuba M., Shank T. M., Langmuir C. H., Nakamura K.I. (2008) Hydrothermal exploration with the autonomous benthic explorer. Deep Sea Research Part I: Oceanographic Research Papers 55/2, 203-219.

[9] Kruger D., Stolkin R., Blum A., Briganti J. (2007) Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments. IEEE International Conference on Robotics and Automation, 4265 - 4270. Roma, Italy.

[10] Dunbabin M, et. al. (2005) A Hybrid AUV Design for Shallow Water Reef Navigation. IEEE International Conference on Robotics and Automation, 2105 - 2110. Barcelona, Spain.

[11] Fairfield N., Jonak D., Kantor G., Wettergreen D. (2007) Field results of the control, navigation and mapping systems of a hovering AUV, International Symposium on Unmanned Untethered Submersible Technology, Durham, New Hampshire, 2007.

[12] Fairfield N., Kantor G.A., Jonak D., Wettergreen D. (2008) DEPTHX Autonomy Software: Design and Field Results. tech. report CMU-RI-TR-08-09, Robotics Institute, Carnegie Mellon University

[13] Fossen T.I. (1994) Guidance and Control of Ocean Vehicles. John Wiley & Sons 1994.

[14] do Carmo M., (1976) Differential geometry of curves and surfaces. Upper Saddle River, New Jersey: Prentice-Hall, Inc., 1976.

[15] Bullo F., Lewis A. D. (2005) Geometric control of mechanical systems. Springer 2005.

[16] Andonian M., Cazzaro D., Invernizzi L., Chyba M., Grammatico S. (2010) Geometric control for autonomous underwater vehicles: overcoming a thruster failure, IEEE Conference on Decisions and Control, Atlanta, Georgia.

[17] Chyba M., Smith R.N. (2008) A first extension of geometric control theory to underwater vehicles, IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, Killaloe, Ireland.

[18] Smith R. N., Chyba M., Wilkens G. R., Catone C.J. (2009) A geometrical approach to the motion planning problem for a submerged rigid body, International Journal of Control„ vol. 82, no. 9, pp. 1641–1656.

[19] Caiti A., Munafo A., Viviani R. (2007) Adaptive on-line planning of environmental sampling missions with a team of cooperating autonomous underwater vehicles. International Journal of Control 80/7, 1151-1168.

[20] Sussmann H., Tang G. (1991). Shortest Paths for the Reeds-Shepp Car: A Worked out Example of the Use of Geometric Techniques in Nonlinear Optimal Control. Rutgers Center for Systems and Control Technical Report 91-10, September 1991.

[21] Sussmann H. (1989) Thirty Years of Optimal Control: Was the Path Unique?. Proceedings of the conference "Thirty years of Optimal Control", Kingston, R.I., 1988