System Design of Undersea Vehicles with Multiple Sources of Uncertainty

Todd W. Benanzer
Wright State University

Follow this and additional works at: https://corescholar.libraries.wright.edu/etd_all
Part of the Engineering Commons

Repository Citation
https://corescholar.libraries.wright.edu/etd_all/837

This Dissertation is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu, library-corescholar@wright.edu.
System Design of Undersea Vehicles with Multiple Sources of Uncertainty

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

by

TODD W BENANZER
B.S., Wright State University, 2004
M.S., Wright State University, 2006

2008
Wright State University
I HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER MY SUPERVISION BY Todd W. Benanzer ENTITLED System Design of Undersea Vehicles with Multiple Sources of Uncertainty BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Doctor of Philosophy.

Ramana V. Grandhi, Ph.D.
Dissertation Director

Ramana V. Grandhi, Ph.D.
Director, Engineering Ph.D. Program

Joseph F. Thomas, Jr., Ph.D.
Dean of School of Graduate Studies

Committee on Dissertation Defense

Haibo Dong, Ph.D.

Ramana V. Grandhi, Ph.D.

Jay H. Kim, Ph.D.

Ravi C. Penmetsa, Ph.D.

Gregory W. Reich, Ph.D.
ABSTRACT

Benanzer, Todd W. Ph.D. College of Engineering and Computer Science, Wright State University, 2008. System Design of Undersea Vehicles with Multiple Sources of Uncertainty.

The work performed investigates the system design and optimization of an undersea vehicle. The system design is driven by the available components, the missions the vehicle is required to perform, and the performance the system configuration yields. The system design consists of three design modules: path planning, component selection and sizing, and structural analysis. The path planning module uses a novel application of the Particle Swarm Optimization algorithm named Path Planning by Additive Freedom. Additionally, the unknown aspects of the mission space through which the path propagates are dealt with using an uncertainty quantification method known as Evidence Theory. Component selection and sizing are performed using the naval design tool SNARC. This program uses a branch and bound technique called the A* algorithm to choose the components that should be used in the system and what size they should be according to the mission profiles provided by the path. The structural analysis is performed using the ABAQUS finite element program to calculate the structural reliability of the system. This module uses the structure sizing data, as well as the hydrodynamic and hydrostatic forces from the mission profile, to calculate the system’s reliability with respect to a buckling failure, the most common structural failure in undersea vehicles.
Contents

1 Introduction to Unmanned Undersea Vehicles .......................... 1
   1.1 History of the Unmanned Undersea Vehicle .......................... 2
      1.1.1 Navy’s Mission ............................................. 2
      1.1.2 Sea Power 21 Initiative ..................................... 3
      1.1.3 Classes of a UUV ........................................... 4
   1.2 Motivation for UUV Development ...................................... 5
      1.2.1 Operational Advantages of a UUV ............................ 6
      1.2.2 UUV Mission Parameters .................................... 7
   1.3 UUV Design Considerations ......................................... 9
   1.4 Scope of the Research ............................................. 10
   1.5 Motivation for System Design ..................................... 12
   1.6 Prior Work in System Design ..................................... 15
   1.7 Document Overview .............................................. 17

2 Evidence Theory ................................................................... 20
   2.1 Frame of Discernment ............................................... 21
   2.2 Basic Belief Assignments ............................................ 24
   2.3 Mapping a Non-Binary-Based Probability Estimate ............... 29
   2.4 Example Problem .................................................... 32
   2.5 Dempster’s Rule of Combining ..................................... 37

3 Particle Swarm Optimization ............................................... 42
   3.1 The Particle Swarm Optimization Algorithm ....................... 44
   3.2 Intertial Updating ................................................... 45
   3.3 Constriction Factor .................................................. 47
   3.4 Niching ............................................................... 48
   3.5 Implementation Walk Through ..................................... 49
   3.6 The Eggcrate Function .............................................. 53

4 Reliability Based Design Optimization .................................. 56
   4.1 Deterministic Optimization ......................................... 56
   4.2 Reliability-Based Design Optimization ............................ 57
      4.2.1 First Order Reliability Method ............................... 61
      4.2.2 Metamodeling .................................................. 63

5 Path Planning ..................................................................... 66
   5.1 PSO and Path Planning .............................................. 67
   5.2 Trajectory Optimization ............................................. 69
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3</td>
<td>Deterministic Path Planning</td>
<td>70</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Path and Velocity Profile Interpolation</td>
<td>71</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Hazard Penalty Calculation</td>
<td>75</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Checkpoint Penalty Calculation</td>
<td>75</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Maneuvering Limits Penalty Calculation</td>
<td>76</td>
</tr>
<tr>
<td>5.4</td>
<td>Evidence Theory Based Path Planning</td>
<td>79</td>
</tr>
<tr>
<td>5.5</td>
<td>Path Planning by Additive Freedom</td>
<td>79</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Iteration Procedure</td>
<td>81</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Comparison to Current Methods</td>
<td>83</td>
</tr>
<tr>
<td>6</td>
<td>SNARC</td>
<td>85</td>
</tr>
<tr>
<td>6.1</td>
<td>System Configuration Design</td>
<td>88</td>
</tr>
<tr>
<td>6.1.1</td>
<td>System Desirability</td>
<td>88</td>
</tr>
<tr>
<td>6.1.2</td>
<td>A* Algorithm</td>
<td>89</td>
</tr>
<tr>
<td>6.2</td>
<td>Mission Analysis</td>
<td>91</td>
</tr>
<tr>
<td>6.3</td>
<td>Power Systems</td>
<td>92</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Battery</td>
<td>92</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Engine</td>
<td>93</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Fuel System</td>
<td>94</td>
</tr>
<tr>
<td>6.3.4</td>
<td>Power Converters</td>
<td>95</td>
</tr>
<tr>
<td>6.4</td>
<td>Structural Systems</td>
<td>96</td>
</tr>
<tr>
<td>6.4.1</td>
<td>Structural Hull</td>
<td>96</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Thrust Device</td>
<td>99</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Fluid Dynamics</td>
<td>105</td>
</tr>
<tr>
<td>7</td>
<td>Structural Analysis of a UUV</td>
<td>106</td>
</tr>
<tr>
<td>7.1</td>
<td>UUV Modeling</td>
<td>106</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Finite Element Analysis</td>
<td>107</td>
</tr>
<tr>
<td>7.2</td>
<td>Simulated Loadings</td>
<td>109</td>
</tr>
<tr>
<td>7.3</td>
<td>Static Analysis</td>
<td>110</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Hydrostatic Loading</td>
<td>112</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Buckling Analysis</td>
<td>113</td>
</tr>
<tr>
<td>7.4</td>
<td>Dynamic Analysis</td>
<td>119</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Vibration Analysis</td>
<td>120</td>
</tr>
<tr>
<td>7.5</td>
<td>Summary</td>
<td>122</td>
</tr>
<tr>
<td>8</td>
<td>Cost Minimization of a UUV</td>
<td>123</td>
</tr>
<tr>
<td>8.0.1</td>
<td>Cost Modeling</td>
<td>125</td>
</tr>
<tr>
<td>8.1</td>
<td>Problem Formulation</td>
<td>128</td>
</tr>
<tr>
<td>8.2</td>
<td>Discussion</td>
<td>131</td>
</tr>
<tr>
<td>9</td>
<td>System Design of an Undersea Vehicle</td>
<td>132</td>
</tr>
<tr>
<td>9.1</td>
<td>System Design Methodology</td>
<td>132</td>
</tr>
<tr>
<td>9.2</td>
<td>System Design Framework</td>
<td>134</td>
</tr>
<tr>
<td>9.2.1</td>
<td>Path Planning Module</td>
<td>135</td>
</tr>
<tr>
<td>9.3</td>
<td>System Design Results</td>
<td>139</td>
</tr>
</tbody>
</table>
10 Summary and Future Directions
   10.1 Research Contributions ............................................... 146
   10.2 Future Research Directions ........................................... 148

Bibliography ................................................................. 149
List of Figures

1.1 Unmanned Undersea Vehicle ....................................... 2
1.2 Class Sizes of a UUV ............................................... 5
1.3 Undersea Vehicle System Design Framework ....................... 11
1.4 Traditional Methodology for System Design ....................... 13
1.5 Proposed Methodology for System Design ....................... 13

2.1 Frame of Discernment with Purely Exclusive Sets ............... 22
2.2 Frame of Discernment with Purely Inclusive Sets ............... 22
2.3 Frame of Discernment with Mixed Sets ....................... 22
2.4 Translation of a Mixed Frame of Discernment to Elementary Form ............... 23
2.5 Regions of Belief of A for an example Basic Belief Assignment ............... 27
2.6 Regions of Plausibility of A for an example Basic Belief Assignment ............... 27
2.7 Regions of Uncertainty of A for an example Basic Belief Assignment ............... 28
2.8 Further Refined Regions of Uncertainty of A for an example Basic Belief Assignment ............... 29
2.9 Basic Belief Assignment for a Simple Applied Load Example ............... 30
2.10 Plausibility and Belief of Reliability for the System vs. System Resistance ............... 31
2.11 Evidence Theory Example: Cantilever Beam ....................... 32
2.12 Basic Belief Assignment for $k$ ....................... 34
2.13 Evidence Theory Beam Example Vertex Sampling Method Results for Proposition $x_1$ ............... 35
2.14 Evidence Theory Beam Example Vertex Sampling Method Results for Proposition $x_2$ ............... 35
2.15 Dempsters Rule of Combination Example: BBA from Expert 1 ............... 38
2.16 Dempsters Rule of Combination Example: BBA from Expert 2 ............... 38
2.17 Dempsters Rule of Combination Example: Final Combined BBA ............... 41

3.1 Comparison of Nonlinearity Factor to Inertial Updating ............... 47
3.2 Plot of the Objective Function for the PSO Example ............... 50
3.3 Plot of the Objective Function for the Eggcrate Function ............... 54
3.4 Convergence History of PSO for the Eggcrate Function ............... 55

4.1 Graphical Depiction of a Reliability Index ............... 61

5.1 Approximating a Square Hazard with Circle Hazard Definitions ............... 70
5.2 Three interpolation methods used to form a path from four data points ............... 71
5.3 Hazard Penalty Function for a location $[0, 0]$ and $R = 1$ ............... 76
5.4 Checkpoint Penalty Function for a location $[0, 0]$ and $R = 1$ ............... 77
5.5 Path Definition Using Spline Interpolation ............... 80
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>SNARC Screen Capture</td>
<td>86</td>
</tr>
<tr>
<td>6.2</td>
<td>Notable Models Available in SNARC</td>
<td>86</td>
</tr>
<tr>
<td>6.3</td>
<td>A* Algorithm Example System</td>
<td>90</td>
</tr>
<tr>
<td>6.4</td>
<td>A* Algorithm Estimation Technique</td>
<td>91</td>
</tr>
<tr>
<td>6.5</td>
<td>A Simple Steam Turbine Engine Cycle</td>
<td>94</td>
</tr>
<tr>
<td>6.6</td>
<td>Side View of Kinematic Model</td>
<td>96</td>
</tr>
<tr>
<td>6.7</td>
<td>Angle of Attack</td>
<td>98</td>
</tr>
<tr>
<td>6.8</td>
<td>Discretization of a Propulsor Blade for Blade Element Theory</td>
<td>100</td>
</tr>
<tr>
<td>6.9</td>
<td>Two Dimensional Section of a Discretized Propulsor Blade</td>
<td>101</td>
</tr>
<tr>
<td>7.1</td>
<td>UUV Finite Element Model</td>
<td>108</td>
</tr>
<tr>
<td>7.2</td>
<td>UUV Shell Stresses When Subjected to Statically Applied Depth Loading</td>
<td>113</td>
</tr>
<tr>
<td>7.3</td>
<td>UUV Shell Depth Loading: First and Second Buckling Modes</td>
<td>115</td>
</tr>
<tr>
<td>7.4</td>
<td>UUV Shell Depth Loading: Third and Fourth Buckling Modes</td>
<td>115</td>
</tr>
<tr>
<td>7.5</td>
<td>Snap-through Buckling: Load vs. Displacement Plot</td>
<td>116</td>
</tr>
<tr>
<td>7.6</td>
<td>Riks Analysis of the UUV Shell</td>
<td>117</td>
</tr>
<tr>
<td>7.7</td>
<td>Undeformed UUV Shell</td>
<td>117</td>
</tr>
<tr>
<td>7.8</td>
<td>UUV Shell Deformation at Point A</td>
<td>118</td>
</tr>
<tr>
<td>7.9</td>
<td>UUV Shell Deformation at Point B</td>
<td>118</td>
</tr>
<tr>
<td>7.10</td>
<td>UUV General Instability Buckling</td>
<td>118</td>
</tr>
<tr>
<td>7.11</td>
<td>UUV Lobar Buckling</td>
<td>119</td>
</tr>
<tr>
<td>7.12</td>
<td>UUV Shell First and Second Vibrational Modes</td>
<td>121</td>
</tr>
<tr>
<td>7.13</td>
<td>UUV Shell Third and Fourth Vibrational Modes</td>
<td>122</td>
</tr>
<tr>
<td>8.1</td>
<td>Material Cost Function</td>
<td>127</td>
</tr>
<tr>
<td>8.2</td>
<td>UUV Finite Element Model</td>
<td>129</td>
</tr>
<tr>
<td>8.3</td>
<td>UUV Design Sections</td>
<td>130</td>
</tr>
<tr>
<td>9.1</td>
<td>Undersea Vehicle System Design Framework</td>
<td>133</td>
</tr>
<tr>
<td>9.2</td>
<td>Optimal Mission Profiles for Missions 1-3</td>
<td>140</td>
</tr>
<tr>
<td>9.3</td>
<td>System Design Iteration History</td>
<td>141</td>
</tr>
</tbody>
</table>
List of Tables

1.1 Summation of Various Recent System Design Research .................. 18
2.1 Evidence Theory Beam Example Input Parameters ....................... 33
2.2 Evidence Theory Beam Example Propositions and Belief Mapping Func-
tion Values ........................................................................ 34
2.3 Evidence Theory Beam Example BEL and PL Values for Each Proposition 36
2.4 Evidence Theory Beam Example BEL and PL Values for the System .. 37
2.5 Dempsters Rule of Combining: Example Results ......................... 40
3.1 PSO Example: Beginning Particle Locations ............................. 50
3.2 PSO Parameters for the Eggcrate Function Example .................... 54
5.1 PSO vs Additive Freedom Method Comparison for Example Missions .. 84
7.1 UUV Shell Properties .......................................................... 107
8.1 Material Properties of the UUV Shell ...................................... 130
8.2 History of Design Variables: UUV Shell .................................. 131
9.1 Mission 1 Definition with Basic Belief Assignment Data (all units in meters) 135
9.2 Mission 2 Definition with Basic Belief Assignment Data (all units in meters) 136
9.3 Mission 3 Definition with Basic Belief Assignment Data (all units in meters) 136
9.4 Available Components for Use in the Vehicle ........................... 137
9.5 Undersea Vehicle Structural Parameters .................................. 138
9.6 Optimal Mission Performance Parameters ............................... 140
9.7 Optimal Mission Component Sizes ....................................... 141
9.8 Buckling Failure Probability of the Optimal System .................... 141
Acknowledgments

I would like to thank my adviser Dr. Ramana V. Grandhi for his help in the performance of this work and preparation of this document, along with my committee members, Jay Kim, Haibo Dong, Ravi Penmetsa, and Greg Reich, who provided valuable input. Thanks to my ULI lab mentor Bill Krol at NUWC-Newport for his advise concerning all things naval. I would like to thank the sponsors of this work: the Dayton Area Graduate Studies Institute fellowship program and the U.S. Navy Office of Naval Research (ONR) Grant N000-14-06-1-0273.

Additionally, I would like to thank my parents for the constant encouragement, advise, and free meals I received along the way. I would also like to thank my best friend Erin who always believed in me, even when I didn’t. A special thanks goes out to my friends for their understanding while I made my way through this process. Finally, thanks to everyone at CDOC, without their help, advise, and criticism I would not be here today.
Introduction to Unmanned Undersea Vehicles

This work performs the systems design and optimization for tactical-sized undersea vehicles. It primarily deals with vehicles that do not carry an explosive payload. Much work has gone into the design of naval torpedoes, and these systems are fairly well defined. This work delves into the new area of Unmanned Undersea Vehicles, as the missions they will complete may be similar to those missions that torpedoes complete, or they may be vastly different. Because of these new missions, this work focuses on Unmanned Undersea Vehicles.

An Unmanned Undersea Vehicle, or UUV, shown in Figure 1.1 is a vehicle that operates with no human on board, and is capable of complete submersion below the ocean’s surface for extended periods of time. UUVs came about as a Naval idea aimed at increasing the capacity and ability of the Navy to meet current and future sea mission demands. These vehicles can range in size from a very small, backpack sized unit up to a unit
weighing several tons. The range of use of UUVs may span most current
manned missions as well as several missions identified solely for unmanned
systems. The Navy is pressing forward with the idea, and the resulting
deigns may well shape the Navy’s future capabilities and role in national
defense.

![Unmanned Undersea Vehicle](image)

Figure 1.1: Unmanned Undersea Vehicle

## 1.1 History of the Unmanned Undersea Vehicle

### 1.1.1 Navy’s Mission

According to the Navy League, “The mission of the U.S. Navy is to main-
tain, train, and equip combat-ready naval forces capable of winning wars,
deterring aggression, and maintaining freedom of the seas [1].” The Navy
currently has 281 total war ships with 55 of those being ship submersibles
(submarines) [2]. With the persistent demands resulting from terrorism
and other wars throughout the world, the demand for a flexible and quick
reacting force is becoming ever important. Even in 1775 when the United
State’s Naval force was initially assembled, the need for such a force was
needed then and is still needed now [1]. Chief of Naval Operations Admi-
rall Vern Clark stated that to fight terrorism and future threats the Navy
must be “on-scene, on-call, and on-demand.”

Despite this need for a strong global sea presence, the number of ships in the Navy is shrinking down to the size it once was during the Great Depression [1]. This means that the existing ships in operation must be designed to be even more flexible and responsive to global threats than in the past. One way the Navy aims to meet both of these demands is by developing a small, highly flexible, and proactive force, under the guidance of its FORCEnet Implementation Plan. The idea of a highly integrated round-the-clock fighting force was initially touched upon in a paper entitled “Forward...From the Sea [3].” This idea was then more formalized with the FORCEnet and Sea Power initiatives [4].

1.1.2 Sea Power 21 Initiative

Sea Power 21 provides the layout for the incremental growth of the Navy’s fighting force to meet the needs of today as well as continue to meet future defense demands in whatever form may be. Admiral Vern Clark, Chief of Naval Operations 2000-2005, stated that the Navy must leverage the great advances in the 21st century in the areas of precision, area of influence, and communication, to form a new and highly-cooperative fighting force [5]. This new fighting force has been identified as Sea Power 21. Sea Power 21 is divided into three main components: Sea Basing, Sea Strike, and Sea Shield [4].

Sea Basing is the cooperative effort of a collection of sea vessels as
well as land and air units that work in a synergistic fashion to provide constant support to one another in a localized area. Sea Strike is the ability to provide substantial offensive force, using a combination of land, air, and sea units to do so. Sea Shield is the initiative that the integrated force will be able to operationally have control over its desired area of responsibility. The UUV will play a critical role in each facet of the Sea Power 21 Initiative, by working closely with other units, thus maximizing the operational potential of naval units.

1.1.3 Classes of a UUV

The Navy has defined several classes of UUVs, which are shown in Figure 1.2 to illustrate relative size. These different classes vary in size, as well as operational abilities [6]. The smallest class is the man-portable class. These UUVs can range in size up to 100 pounds. Due to their small size and weight, these can be deployed by hand, either from the shore line or off of a small rubber inflatable boat. The next size up is the light-weight vehicle class. The UUVs in this category are 12.75 inches in diameter. This allows these UUVs to operate within the existing infrastructure in place for torpedoes of this dimension. Similarly, the heavy-weight vehicle class have 21 inch diameters for the same reason. The UUVs that belong in the large vehicle class could range in diameter up to 72 inches. These types of UUVs, due to their large size, would be able to hold large amounts of fuel which translates into longer operational time, however require an
extensive infrastructure to deploy.

Figure 1.2: Class Sizes of a UUV

1.2 Motivation for UUV Development

UUVs offer many things to the Navy. Their ability to perform many duties that a traditional submarine can do without humans on board is a definite advantage. The computer guidance systems, or brains, of the UUV will be in close contact with other UUVs as well as other military units within the FORCEnet infrastructure. Information sharing will allow better decisions to be made during peacetime as well as in battle, due to a greater knowledge of the surroundings. The UUVs will have the capability to take such repetitious tasks as patrolling and ocean floor mapping from Navy personnel and allow these people to perform more complicated decision-based tasks.
1.2.1 Operational Advantages of a UUV

The Navy’s UUV Master Plan [6] lists several operational advantages that an Unmanned Undersea Vehicle would have over its manned counterparts. The ability of an unmanned system to act without constant human intervention allows for the vehicle to increase the influence that a manned station has on its environment by allowing several units to operate under the control of a single person. Allowing the UUV to complete routine tasks, and even more complex tasks allows for the operator to concentrate on other needs. The autonomous nature also further removes the human from dangers that may occur on a UUV mission. UUVs are also much smaller in size than a manned undersea unit. The lower propulsion demands necessary make for quieter operations. When a UUV surfaces, only a small portion would need to break the water surface in order to communicate. A small antenna would be all that is necessary. Each of these advantages provide the UUV little probability for unintentional detection.

In addition, their small size allows UUVs to be easily deployed. They can be sized such that they could be launched in any location built for standard (21-inch) or lightweight (12.75-inch) torpedoes. UUVs can also be used much more robustly with respect to their environment. Towed arrays are currently used heavily in conjunction with submarines. A towed array is typically attached to a submerged submarine and connected by a cable. These arrays are used to transmit data through the air, or to gather data above sea level. A towed array may have difficulty in particularly
inclement weather, while a UUV would be able to navigate such waters and return to its home base without the worry of a broken cable. UUVs also provide a persistence factor. Due to their autonomous nature they may be set to an idle mode in which they can rest along the ocean floor and await further instructions.

Also, UUVs will be able to make use of the newest power and energy technologies. These new technologies include fuel cells as well as high density battery configurations. Currently, research is being performed in both of these areas. The Navy’s goal is to attain a power supply that can be quickly refueled by refilling a liquid tank or by switching a battery.

1.2.2 UUV Mission Parameters

The stated operational advantages mentioned in Section 1.2.1 allow the UUV to perform several interesting missions. These missions are explained thoroughly in the 2004 UUV Master Plan [6] and outlined briefly here. For intelligence gathering, UUVs would be ideal. These types of missions often involve entering a hostile area and gathering as much information about the surroundings as possible. The environmental hostilities may be due to a near-proximity enemy, a chemical or nuclear exposure risk, or unsafe geographic passage ways for a large submarine.

Additionally UUVs provide excellent support conducting mine countermeasures. Frequently, large, safe, water routes must be created for large surface vessels. Scanning for mines and other obstacles along these paths
can become quite tedious and dangerous. UUVs would be able to be deployed in large numbers, and work as a cooperative unit to quickly identify safe paths as well as identify mined areas.

Anti-submarine warfare is another high-importance mission that the UUV will be able to assist in. UUVs would be able to monitor enemy submarine movement through a ‘Hold at Risk’ [6] mission. A ‘Maritime Shield’ mission would consist of clearing a large area of outside threats, while ‘Protected Passage’ missions require the UUV to act as a security escort to a passing vehicle. UUVs are also capable of an identification mission. During this type of mission, the UUV inspects shorelines or ship hulls for abnormalities. This type of mission would be able to greatly reduce the demand of dive teams who currently take on these inspection tasks.

UUVs will also be used oceanography missions. Currently, oceanographic tasks are performed using submarines and attached towed arrays or surface ships. This becomes a problem when ocean floor topography is needed in areas that are hotly contested, or stealth is required. UUVs would be able to enter these areas while avoiding detection and attain ocean floor data that can be valuable when making tactical maritime decisions.

To a lesser extent, UUVs may be sent out to act as communication relays. The UUV could travel to near surface depths while remaining out of the line-of-sight field of vision. It could then act as a relay for a fully
submersed submarine to allow high bandwidth communications through the airwaves that would be impossible through sea water. UUVs could also be used as a payload delivery system to provide needed supplies to areas where stealth is required.

Another interesting capability for the UUV would be to confuse near-by hostile vessels. In this type of mission, the UUV would deliberately project a large acoustic or electronic signature. This signature would confuse near-by enemies into detecting a full sized submarine or perhaps some sort of other obstacle that they would need to avoid.

Each of these missions would provide numerous advantages and options to the Navy while lowering the risk of human lives. However, producing a UUV that is capable of one of these missions is not nearly enough. UUVs must be designed such that they are capable of carrying out many of the aforementioned missions. This type of design requires that initial designs of UUVs must take into account the interaction between systems for optimal trade-offs rather than simply trying to optimize each subsystem individually.

1.3 UUV Design Considerations

A UUV must be designed with many criteria in mind. A sufficient propulsion system must be employed to provide thrust at an efficient rate. A reliable and long-lasting power source should be included into the system to power the propulsion system as well as all the on board electronics. A
highly robust and complicated guidance and control computer algorithm must be input into the brain of the UUV to allow for proper navigation, as well as object identification and decision making.

Efficient design of each of these components is necessary in order to create a highly flexible UUV. The interactions between these systems must be considered. The proposed research focuses on the interaction between many of the systems that make up the UUV. In this multidisciplinary UUV system-level design, multiple fidelity models are adopted depending on the necessity of the needed details. Typically high fidelity structural models will be integrated with other lower fidelity system models to design the UUV that has optimal mission performance characteristics.

1.4 Scope of the Research

The work performed here is aimed at performing a system-level optimization of a UUV, or any tactical-scale undersea vehicle. The research devises a framework that uses desired missions and available components in order to optimally design a system configuration. The system design and optimization framework consists of three major design modules, as seen in Figure 1.3. The three design modules are the path planning module, the component selection and sizing module, and the structural analysis module.

This framework allows for the incorporation of uncertainty, both in the mission definitions as well as in the hull structure. Furthermore, the incor-
Figure 1.3: Undersea Vehicle System Design Framework

Incorporation of feed-forward and feedback loops allow for information to be passed from module to module in order to ensure that a complete mission profile, component configuration, and structural analysis can be performed in an iterative fashion. The framework developed in this research provides a mission-in configuration-out design system. In addition, the components available for use in the UUV can be updated allowing for the prediction of future capabilities and performance. Valuable mission and structural risk
values are output, too.

These values give designers information about the certainty that the vehicle will complete the supplied missions. This type of mission-in configuration-out design removes the intermediate step of defining vehicle specifications, such as top speed and range. Designing to these intermediate specifications may yield sub-optimal results when creating a complete system configuration. Furthermore, by performing a systems design on low-fidelity modeling, and reserving the high-fidelity structural modeling for model validation and analysis, a verified system is designed that is able to fully explore each possibility while keeping the process computationally efficient.

1.5 Motivation for System Design

Traditionally system design of undersea vehicles, as many vehicles, is performed by first identifying key performance characteristics, such as speed, endurance, and various maneuvering metrics. These characteristics then dictated which components need to be incorporated into the system in order to form a vehicle with satisfactory performance relative to each metric. Once the system components are selected, the system is constructed and used to perform various missions. This methodology is posed in Figure 1.4.

An alternate formulation of the system design process is proposed in this research. Figure 1.5 shows this methodology. In this research, the
The process of determining performance characteristics, selecting components, and evaluating mission performance is arranged as a design loop. This methodology involves an iterative solution methodology that updates the design based on feed-forward and feed-back information. This allows for decisions made about the component selection and sizing to be made intelligently so that they best affect mission performance. These trade-off considerations are omitted in the traditional system design methodology.

System design involves many different disciplines in order to fully cap-
ture the physics of the problem. Many researchers have proposed alternative ways of performing system-level design and optimization. Various methods include multi-fidelity analysis in which each of the system components and subcomponents are modeled using varying fidelities ranging from closed-form equations to highly iterative processes in order to model the inner workings of the component or subcomponent.

Performing design and optimization on the system level also makes the problem complex. Typically, one or more inner design loops are used to design the separate components, and an outer design loop deals with the overall system performance. Because of this, the computational time necessary to analyze and design entire systems is very large. This is where making efficient inner-loop design optimizations can drastically reduce the required computational time and make full-scale system design and optimization a useful solution methodology.

Despite the large computational effort, the benefits of performing design and optimization on the system-level allow for more robust and efficient system designs. Performing designs on the system level gives great insights into how various choices on the component and subcomponent level affect the overall system performance. In highly coupled systems where a component may drastically affect all other aspects of the performance of the vehicle, system level design is vital. In these types of systems, changes in one area of the system affect the performance of many other components. This coupling of sub-systems and components must be analyzed, and due
to these couplings an iterative solution is necessary.

1.6 Prior Work in System Design

Kamada et al. [7, 8] performed a system level design of a supercavitating torpedo. In this work they aimed to design a supercavitating torpedo so that it could best perform a fixed array of maneuvers including fixed radius turns and simple dive and surface maneuvers. While accurately modeling the control scheme and cavitator size necessary to perform such maneuvers, they failed to address the components necessary to power the vehicle. Additionally, the structural response was also omitted.

Patel et al. [9, 10] also focused on the design of undersea vehicles. Their work focused on the design of a traditional torpedo. This work involved the design of a torpedo meant to intercept a moving target. They included hydrodynamic approximations into their design in order to illustrate a control system using proportional integral control. This work however, also failed to account for the selection of component systems and the structural response of the torpedo hull.

Much work has been done in the aerospace field with regards to system level design. This is sometimes also referred to in literature as conceptual design or trajectory optimization. Baker et al. [11–13] created a highly detailed system design framework called the Integrated Hypersonic Aeromechanics Tool (IHAT) for designing high speed missile systems in their collaboration with the Navy Surface Warfare Center. This is one of
the most complete system design frameworks available in the literature for the design of tactical-scale vehicles. The IHAT program is based on the trajectory optimization tool Programs To Optimize Spacecraft And Aircraft Trajectories (POST) [14]. This program performs path planning over an open design space with no hazards or checkpoints as proposed in this research.

The POST program is written especially for space vehicles to plan orbital paths, and atmospheric entry and exit paths. IHAT also incorporates aeroelastic structural behavior with the NASTRAN [15] finite element program. However, Baker et al. do not incorporate component selection as performed in this work. Each specified component is sized by the IHAT tool, but the inclusion or exclusion of any individual component must be performed by the user prior to input into the IHAT software. Furthermore, the IHAT system only performs deterministic calculations.

Mor and Livne [16] perform a similar system design for a reentry vehicle. Their problem formulation does not address component selection or sizing however. Similarly, the trajectory optimization, or path planning, performed in their work is driven primarily by limits on aerodynamic loads, and therefore is fairly unimodal, allowing for convergence with the use of gradient methods rather than the multi-modal design space of path planning performed in this work.

Foo et al. [17, 18] present a path planning technique for use with unmanned aerial vehicles that uses the particle swarm optimization algo-
rithm. While they include hazards into the path planning algorithm, the aerodynamics required from the system are not considered. Additionally, no mention is made of the component design of the vehicle, as their work only looks at path planning, and not the vehicle’s design as well. No uncertainty in models and processes is addressed in their work.

Recently, in the more general system design field Mavris et al. [19–21] have performed the design of a anti-aircraft gun while incorporating uncertainty. This work involves the joint design of a gun and its ammunition with respect to system cost and effectiveness (kill rate). The path of the projectile is a function of its firing velocity and its aerodynamic drag, as the system is not powered. The projectile is assumed to have some control of its flight through a bang-bang controller which fine tune it’s trajectory toward the end target. This work incorporates the uncertainty involved with the firing rate as well as using experimental data to form a response surface for the damage potential with regards to projectile size and relative velocity of the projectile. An abridged summary of relevant system design research is shown in Table 1.1.

1.7 Document Overview

This document begins by explaining the concepts of Evidence Theory, an uncertainty quantification technique, in Chapter 2. Once these concepts are explained and several examples of its applications are performed, the optimization methodology of Particle Swarm Optimization is presented
<table>
<thead>
<tr>
<th>Author(s)</th>
<th>Path Planning</th>
<th>Vehicle Dynamics</th>
<th>Vehicle Control</th>
<th>Component Design</th>
<th>Structural Consideration</th>
<th>System Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benanzer</td>
<td>Highly Nonlinear w/ Uncertainty</td>
<td>Hydrodynamic Approximations</td>
<td>Ideal Controller Assumed</td>
<td>Full Component Selection and Sizing</td>
<td>Finite Element Analysis w/ Uncertainties</td>
<td>Undersea Vehicle w/ Uncertainty(Torpedo/UUV)</td>
</tr>
<tr>
<td>Kamada et al.</td>
<td>Simple Turn and Dive Maneuvers</td>
<td>6 DOF Calculation</td>
<td>Calculated Using Optimal Control Theory</td>
<td>Not Included</td>
<td>Not Included</td>
<td>Undersea Vehicle (Supercav Torpedo)</td>
</tr>
<tr>
<td>Foo et al.</td>
<td>Highly Nonlinear</td>
<td>Not Included</td>
<td>Ideal Controller Assumed</td>
<td>Not Included</td>
<td>Not Included</td>
<td>Air Vehicle (UAV)</td>
</tr>
<tr>
<td>Mor and Livne</td>
<td>Unimodal open Design Space</td>
<td>Aeroelasticity Calculations</td>
<td>Ideal Controller Assumed</td>
<td>Not Included</td>
<td>Finite Element Analysis</td>
<td>Air Vehicle (Reentry Vehicle)</td>
</tr>
<tr>
<td>Baker et al.</td>
<td>Unimodal Open Design Space</td>
<td>Aeroelasticity Calculations</td>
<td>Ideal Controller Assumed</td>
<td>Component Sizing</td>
<td>Finite Element Analysis</td>
<td>Air Vehicle (Missile)</td>
</tr>
<tr>
<td>Patel et al.</td>
<td>Intercept Trajectory</td>
<td>Hydrodynamic Approximations</td>
<td>Proportional Integral Control</td>
<td>Not Included</td>
<td>Not Included</td>
<td>Undersea Vehicle (Torpedo)</td>
</tr>
</tbody>
</table>
in Chapter 3. Another form of optimization, Reliability Based Design Optimization is presented in Chapter 4. This methodology uses a gradient-based search to find optimal designs while incorporating uncertainty into the system.

Chapter 5 presents the path planning techniques used in this work, which uses the Particle Swarm Optimization algorithm and Evidence Theory uncertainty quantification method already presented. The component selection and sizing program, SNARC, is described in detail in Chapter 6. Next, the structural analysis techniques used in this work are provided in Chapter 7. A cost minimization is presented in Chapter 8. This Chapter combines the techniques from Chapters 4 and 7 to minimize production costs of an undersea vehicle while incorporating structural uncertainty.

An undersea vehicle system design is preformed in Chapter 9. This combines the methodologies and techniques described in Chapters 2 through 7 to perform a system level design and optimization while incorporating uncertainty. Finally, Chapter 10 provides some summary remarks about the work.
Evidence Theory

In this research, Evidence Theory is used to quantify the location and size of potential hazards. These hazards are used in the path planning phase of the system design and optimization process.

Evidence Theory came out of the work originally developed by Dempster [22]. Later, Shafer, in his book [23], updated Dempster’s theories and developed the modern Evidence Theory, sometimes referred to as Dempster-Schafer Theory. Evidence theory has recently been found to be valuable in uncertainty analysis when applied to engineering problems where little information exists. It has been a valuable tool in the quantification of epistemic, or subjective uncertainty.

In modern engineering problems, uncertainty is increasingly considered to minimize design risk in deployment. Two types of uncertainty exist in nearly all problems. As coined by Helton [24] uncertainties can be divided into two groups: Aleatory and Epistemic. Aleatory, or stochastic, uncertainty exist because of predictable variations in the analysis of the
problem. Often, aleatory uncertainty involves variables where a large volume of information exists, and fairly accurate predictions of behavior can be made due to the wealth of data.

However, epistemic uncertainty exists because of a lack of prior knowledge. The variables most likely to exhibit some form of epistemic uncertainty are variables in which there is little or no historical data to draw upon. Often times, the known data consists of extrapolated estimations, a small number of experiments, or expert opinion. Both epistemic and aleatory statistics play a large role in engineering analyses. Often, designers must combine the two types of uncertainty in order to fully capture the random nature of a problem and best formulate a design that deals with all of the uncertain factors affecting performance.

Evidence theory acts in a way that combines several frames of discernment into basic belief assignments (BBAs). These basic belief assignments can then be arranged in such a way that they formulate the analysis set in which the belief and plausibility of a system can be calculated. The belief of a system stands as the lower bound on the probability of the system, while the plausibility acts as the upper bound. As more and more data becomes available, these bounds converge on the true probability.

2.1 Frame of Discernment

Frames of discernment are the basic definitions of sets that comprise the foundation of evidence theory. These frames depict intervals on which
the set of numbers along a number line may fall. The value of a variable may take any value along the \( X \) number line shown in Figures 2.1-2.4. The rules that frames of discernment are built upon are quite flexible when compared to traditional statistics. Figure 2.1 shows a frame of discernment of purely exclusive sets. In this type of frame of discernment each point along the line belongs only to one set. This definition poses the frame of discernment in its most basic, or elementary form. Figure 2.2 shows a frame of discernment in which the each set is a subset of a larger set, with the exclusion of the largest defined set. This represents a purely inclusive frame of discernment. Figure 2.3 shows a frame of discernment definition in which the sets overlap in a non-structured fashion. This definition is called a mixed frame of discernment.

As mentioned, only the purely exclusive frame of discernment defini-
tion, shown in Figure 2.1, is in its most basic form. The purely inclusive and mixed definitions need to be further refined to purely exclusive forms so that they may be in their elementary form as well. For illustrative purposes, the mixed frame of discernment example will be redefined into its elementary form. Each set in this definition must be divided until each point on the number line is delegated to no more than one set. For the example in Figure 2.3, this requires set $x_1$ be divided into two subsets, $x_{(1,1)}$, the subset of $x_1$ that does not coincide with any other sets, and $x_{(1,2)}$, the subset of $x_1$ that coincides with $x_2$. The set $x_2$ must be divided into three subsets, each comprising a specific portion of the set. Subset $x_{(2,2)}$ comprises the portion of set $x_2$ that is not included in any other sets. The subset of the set $x_2$ that comprises the interval also included in set $x_1$ has already been defined by the subset definition $x_{(1,2)}$. Using the developed pattern and naming notation, the third subset of the set $x_2$ is $x_{(2,3)}$. Similarly, the set $x_3$ can be divided into the subsets $x_{(3,3)}$ and the previously defined subset $x_{(2,3)}$. The graphical depiction of this set definition is shown in Figure 2.4.

Figure 2.4: Translation of a Mixed Frame of Discernment to Elementary Form

Using the notation that the set of all subsets defined in the frame of discernment can be represented as $\Omega$, we can form various propositions about the set. Considering the elementary frame of discernment shown in
Figure 2.1, a set of particular importance is the power set of $\Omega$, as shown in Equation 2.1.

\[
2^\Omega = \{\emptyset, \{x_1\}, \{x_2\}, \{x_3\}, \{x_1, x_2\}, \{x_1, x_3\}, \{x_2, x_3\}, \{X\}\}
\] (2.1)

It should be observed that the power set of $\Omega$ is noted as $2^\Omega$ because the number of propositions within the power set is equal to $2^n$, where $n$ represents the number of propositions in the elementary frame of discernment. For instance, in the example shown in Figure 2.4 there are five propositions, or interval sets, in the elementary frame of discernment. This power set includes thirty-two unique propositions, or $2^5$.

Each proposition within the set $\Omega$ is shown in Equation 2.1. Of these propositions, if the proposition of $\{x_1, x_3\}$ is selected, then the inference is that either the value falls within the subset $x_1$ or $x_2$. It is important to note that in order for the frame of discernment set definition to be true, the value must fall into one or the other, and not both. Additionally, no further inference is made as to which subset the value is more likely to fall, only that the value falls into either subset.

2.2 Basic Belief Assignments

Frames of discernment form the building blocks for the basic belief assignment, (BBA). BBAs are the organized way in which the degree of propositional belief is expressed. A mapping function, $m$ is used to de-
clare values of belief on propositions. The mapping function specifies the amount of belief present in the specified proposition and must satisfy the three axioms stated in Equations 2.2-2.4. It is of note that these definitions are not as strict in definition as those that apply to traditional probability theory. This allows for greater flexibility for their implementation in engineering problems.

\[ m(\omega) \geq 0 \forall \omega \in 2^{\Omega} \quad (2.2) \]

\[ m(\emptyset) = 0 \quad (2.3) \]

\[ \sum_{\omega \in 2^{\Omega}} m(\omega) = 1 \quad (2.4) \]

where \( \omega \) is a proposition.

There are several important deductions that can be made using the above listed axioms. These deductions deal with the way that evidence theory and traditional probability theory differ. The most prominent of which is that a belief assigned to a proposition does not infer any belief about any other propositions. Assume the following belief of a proposition:

\[ \text{Belief} = m_1 \text{ for } \{x_1\} \in \Omega. \quad (2.5) \]

In probability theory, assuming that belief and probability are equivalent in this instance, we could deduce that any subset of \( \Omega \) containing the
subset $x_1$ must have a probability of greater than or equal to $m_1$. More simply:

$$\text{for any } x_i \subset x_1 \; p(x_i) \geq p(x_1) \quad (2.6)$$

However, in evidence theory, no further deductions can be made about the belief of any of the other subsets of $\Omega$ other than the sum of the belief of each of the remaining sets in the power series $2^\Omega$ must not be greater than one. Thus the only deduction that can be made is the following:

$$\sum_{\omega \in 2^\Omega - x_1} m(\omega) \leq 1 \quad (2.7)$$

From the basic belief assignments, calculations of an event’s belief and plausibility can be made. Given an occurrence $A$, the belief and plausibility that event $A$ will occur can be calculated through the use of the following two equations:

$$\text{BEL}(A) = \sum_{x \subset A} m(x) \quad (2.8)$$

$$\text{PL}(A) = \sum_{x \cap A \neq \emptyset} m(x) \quad (2.9)$$

In Equations 2.8 and 2.9, $x$ represents the entire set of propositions in which the system’s basic belief assignment is based upon. Often the difference between the two values, belief and plausibility, are considered to be the uncertainty in the systems outcome. A graphical depiction of the
concepts of belief and plausibility is included to aid in the understanding of these topics.

Figure 2.5 shows the regions of belief, highlighted in red, for the occurrence of an event $A$, outlined with a dashed line. These regions satisfy Equation 2.8 and are highlighted in red. Notice that the regions $x_1$ and $x_2$ represent their respective propositions that are numerical sets where each point within the set satisfies the event $A$.

Figure 2.5: Regions of Belief of $A$ for an example Basic Belief Assignment

Figure 2.6 shows the region of the plausibility, highlighted in red, for the occurrence of an event $A$. Each of the regions, $x_1$, $x_2$, $x_3$, and $x_4$ represent the proposition sets that satisfy Equation 2.9 for calculating plausibility.

Figure 2.6: Regions of Plausibility of $A$ for an example Basic Belief Assignment

As previously discussed, the uncertainty in the probability of the system performance can be represented by the difference between the values of
belief and plausibility. This region of uncertainty is expressed in Figure 2.7. The regions of uncertainty, highlighted in red, contain both areas that satisfy event $A$ as well as areas where event $A$ is not satisfied.

![Figure 2.7: Regions of Uncertainty of A for an example Basic Belief Assignment](image)

One way to reduce the amount of uncertainty is to acquire additional information about the design space. In Figure 2.8, it is assumed that additional information has been included that divided each postulate into four equal regions. If these regions are then reanalyzed and the region of uncertainty is attained, it can be seen that in this case, the uncertainty has been reduced to half of its prior amount. The new uncertainty region, shown in red, is smaller because of the additional information. Thus if more information was included into the basic belief assignment, further and further refinements could be made. It is of particular interest that should the postulate regions become infinitesimally small, the belief and plausibility values would converge to the true probability of the system, provided accurate BBAs were assigned.
2.3 Mapping a Non-Binary-Based Probability Estimate

Due to the need for a continuous measure of the probability of a system so that optimization algorithms can better meet constraints and further refine objective values, several techniques have been developed. The belief and plausibility values from evidence theory are discontinuous when plotted across the design space. As each proposition experiences or does not experience an event due to the adjustment of the input variables, the belief and plausibility of a proposition (thus forming a basic belief assignment) may jump between zero and one. This creates discontinuities as well as regions with zero slope.

Consider the system with a load resistance $R$, and the applied load $S$. For this system, assume that input parameters can be adjusted to increase or decrease the system’s load resistance and that the applied load is an unknown variable. In this example, the applied load uncertainty will be addressed using evidence theory. The basic belief assignment for the
uncertain applied load is shown in Figure 2.9.

Figure 2.9: Basic Belief Assignment for a Simple Applied Load Example

With this information, the system can be analyzed using the limit state in Equation 2.10:

\[ R - S = 0 \] (2.10)

Figure 2.10 shows the system plausibility and belief relative to changing the system’s resistance, \( R \). Here, the failure region is

\[ R - S < 0 \] (2.11)

while the safe region can be defined as

\[ R - S > 0 \] (2.12)

Notice the discontinuities in the function values of the belief and plausibility in Figure 2.10. Also notice the large regions where the gradients of belief and plausibility are zero relative to the system resistance. This type of response variable makes it very difficult for hill climbing or gradient-
Bae et al. [25] developed a continuous intervening variable plausibility decision definition. In this method, the probability estimate was based on multiplying the area of each proposition that satisfies the given event, $A$, by the mapping function, $m(A)$. This method requires many function evaluations in order to properly estimate the area within each postulate that satisfies event $A$. Alyanak [26] used a linear approximation method to more efficiently calculate the event area. This calculation looked for corners of the proposition hypercube space that had endpoints where one was within the event space and one was outside the event space. Using the magnitude for which each point was in or out of the event space, the
approximation locates a point along the edge that represents the boundary of the event space. By performing this approximation along each edge, the algorithm is able to make an estimation of the area of the event space within the proposition.

### 2.4 Example Problem

Consider the cantilever beam shown in Figure 2.11. For this example the uncertainty is in the stiffness of the tip spring $k$. Through the stiffness value $k$, it could represent the boundary condition at the tip from a free end (low $k$ values) to a fixed support (high $k$ values).

![Figure 2.11: Evidence Theory Example: Cantilever Beam](image)

For this example, the limit state is a displacement limit. For the beam shown, the tip displacement, $y$ is calculated to be:

$$ y = \frac{3PL^4}{24EI + 8L^3k} \quad (2.13) $$

where $P$ is the uniform load per length and $L$ is the beam length.

Table 2.1 shows the values of the input parameters used in this example.
Here, a structural failure is considered to be a tip displacement of greater than 1.83 cm. The limit state then becomes the following:

$$g(x) = y - 1.83$$  \hspace{1cm} (2.14)

or

$$g(x) = \frac{3PL^4}{24EI + 8L^3k} - 1.83$$  \hspace{1cm} (2.15)

where $g(x) > 0$ represents a failure.

There exists particular information that forms the Basic Belief Assignment for the stiffness of the tip spring. This BBA is shown in Figure 2.12. There are ten propositions defined in this Basic Belief Assignment. In this example, since the information is assumed to come from one source, Dempster’s rule of combination is not needed.

The propositions along with their belief mapping function values for the spring stiffness are shown in Table 2.2. This data matches the data illustrated in Figure 2.12.

The first proposition to be analyzed is proposition $x_1$ and its corresponding belief $m_1$. For this example, the vertex sampling method is used in order to determine the existence or non-existence of a violation of the
limit state function, $g$. Recall that a positive value of $g$ represents a failure. The vertex sampling method is an efficient sampling technique as long as the limit state function is monotonically increasing or decreasing with respect to the proposition space. For this example it is safe to use the vertex sampling method. For a more non-linear problem, an advanced sampling technique should be used. Figure 2.13 shows the limit state evaluation values at each vertex of proposition $x_1$.

Since both vertices are in the failure region, it is assumed that the entire region within the proposition is an area of a failure event. Recall

<table>
<thead>
<tr>
<th>Proposition</th>
<th>Belief Mapping Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = [30, 50] kN/m$</td>
<td>$m_1 = 0.0188$</td>
</tr>
<tr>
<td>$x_2 = [30, 70] kN/m$</td>
<td>$m_2 = 0.0352$</td>
</tr>
<tr>
<td>$x_3 = [30, 90] kN/m$</td>
<td>$m_3 = 0.0196$</td>
</tr>
<tr>
<td>$x_4 = [30, 130] kN/m$</td>
<td>$m_4 = 0.0065$</td>
</tr>
<tr>
<td>$x_5 = [50, 70] kN/m$</td>
<td>$m_5 = 0.2749$</td>
</tr>
<tr>
<td>$x_6 = [50, 90] kN/m$</td>
<td>$m_6 = 0.1529$</td>
</tr>
<tr>
<td>$x_7 = [50, 130] kN/m$</td>
<td>$m_7 = 0.0523$</td>
</tr>
<tr>
<td>$x_8 = [70, 90] kN/m$</td>
<td>$m_8 = 0.2871$</td>
</tr>
<tr>
<td>$x_9 = [70, 130] kN/m$</td>
<td>$m_9 = 0.0981$</td>
</tr>
<tr>
<td>$x_{10} = [90, 130] kN/m$</td>
<td>$m_{10} = 0.0546$</td>
</tr>
</tbody>
</table>
the equations for Belief and Plausibility:

\[
\text{BEL}(A) = \sum_{x \subset A} m(x) \quad (2.16)
\]

\[
\text{PL}(A) = \sum_{x \cap A \neq \emptyset} m(x) \quad (2.17)
\]

Using Equations 2.16 and 2.17, the \text{BEL} and \text{PL} can be calculated for a failure in proposition \(x_1\) as both being equal to 1. This means that for the proposition set \(x_1\) the failure probability is 1.

The next proposition analyzed is proposition \(x_2\) and its corresponding belief \(m_2\). Here, Figure 2.14 shows the limit state evaluation values at each vertex of proposition \(x_2\).

For this proposition, the lower vertex is a failure event, while the upper vertex yields a safe design. Because of the mixed nature of the design
region across this proposition the $BEL = 0$ and $PL = 1$ for failure within this proposition. This process can be continued for each of the eight remaining propositions presented for the stiffness of the spring. The $BEL$ and $PL$ values for each proposition are presented in Table 2.3.

Table 2.3: Evidence Theory Beam Example $BEL$ and $PL$ Values for Each Proposition

<table>
<thead>
<tr>
<th>Proposition $m(x)$</th>
<th>$BEL$</th>
<th>$PL$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1 = [30, 50] \text{kN/m}$</td>
<td>$m_1 = 0.0188$</td>
<td>0</td>
</tr>
<tr>
<td>$x_2 = [30, 70] \text{kN/m}$</td>
<td>$m_2 = 0.0352$</td>
<td>0</td>
</tr>
<tr>
<td>$x_3 = [30, 90] \text{kN/m}$</td>
<td>$m_3 = 0.0196$</td>
<td>0</td>
</tr>
<tr>
<td>$x_4 = [30, 130] \text{kN/m}$</td>
<td>$m_4 = 0.0065$</td>
<td>0</td>
</tr>
<tr>
<td>$x_5 = [50, 70] \text{kN/m}$</td>
<td>$m_5 = 0.2749$</td>
<td>0</td>
</tr>
<tr>
<td>$x_6 = [50, 90] \text{kN/m}$</td>
<td>$m_6 = 0.1529$</td>
<td>0</td>
</tr>
<tr>
<td>$x_7 = [50, 130] \text{kN/m}$</td>
<td>$m_7 = 0.0523$</td>
<td>0</td>
</tr>
<tr>
<td>$x_8 = [70, 90] \text{kN/m}$</td>
<td>$m_8 = 0.2871$</td>
<td>1</td>
</tr>
<tr>
<td>$x_9 = [70, 130] \text{kN/m}$</td>
<td>$m_9 = 0.0981$</td>
<td>1</td>
</tr>
<tr>
<td>$x_{10} = [90, 130] \text{kN/m}$</td>
<td>$m_{10} = 0.0546$</td>
<td>1</td>
</tr>
</tbody>
</table>

The plausibility and belief for failure for the beam’s displacement failure can be reached by using the information calculated in Table 2.3. By multiplying the $BEL$ and $PL$ values for each proposition by their corresponding $m$ value, the propositions effect on the overall plausibility and belief can be calculated. This is shown in Table 2.4 as the fifth and sixth columns. Each cell in these two columns represents the contribution of the proposition to the overall Belief and Plausibility.

The fifth and sixth columns of Table 2.4 can then be summed to find the plausibility and belief values for the failure criteria. The beam’s displacement belief of failure is 0.4398 while the plausibility of failure is 0.9812. From this is can be concluded that the true probability of failure is between 0.4398 and 0.9812. Recall the BBA illustrated in Table 2.2. Because of several of the broad set definitions, those that span across most of the
Table 2.4: Evidence Theory Beam Example BEL and PL Values for the System

<table>
<thead>
<tr>
<th>Proposition</th>
<th>m(x)</th>
<th>BEL</th>
<th>PL</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1 = [30, 50] kN/m)</td>
<td>(m_1 = 0.0188)</td>
<td>0</td>
<td>0 * 0.0188</td>
</tr>
<tr>
<td>(x_2 = [30, 70] kN/m)</td>
<td>(m_2 = 0.0352)</td>
<td>0</td>
<td>1 * 0.0352</td>
</tr>
<tr>
<td>(x_3 = [30, 90] kN/m)</td>
<td>(m_3 = 0.0196)</td>
<td>0</td>
<td>1 * 0.0196</td>
</tr>
<tr>
<td>(x_4 = [30, 130] kN/m)</td>
<td>(m_4 = 0.0065)</td>
<td>0</td>
<td>1 * 0.0065</td>
</tr>
<tr>
<td>(x_5 = [50, 70] kN/m)</td>
<td>(m_5 = 0.2749)</td>
<td>0</td>
<td>1 * 0.2749</td>
</tr>
<tr>
<td>(x_6 = [50, 90] kN/m)</td>
<td>(m_6 = 0.1529)</td>
<td>0</td>
<td>1 * 0.1529</td>
</tr>
<tr>
<td>(x_7 = [50, 130] kN/m)</td>
<td>(m_7 = 0.0523)</td>
<td>0</td>
<td>1 * 0.0523</td>
</tr>
<tr>
<td>(x_8 = [70, 90] kN/m)</td>
<td>(m_8 = 0.2871)</td>
<td>1</td>
<td>1 * 0.2871</td>
</tr>
<tr>
<td>(x_9 = [70, 130] kN/m)</td>
<td>(m_9 = 0.0981)</td>
<td>1</td>
<td>1 * 0.0981</td>
</tr>
<tr>
<td>(x_{10} = [90, 130] kN/m)</td>
<td>(m_{10} = 0.0546)</td>
<td>1</td>
<td>1 * 0.0546</td>
</tr>
</tbody>
</table>

design space, the uncertainty region between the belief and plausibility of this problem is quite large. If there were a way to reduce the mapping function values, or the ranges of these wide-spanning sets, the uncertainty of the outcome could be reduced.

### 2.5 Dempster’s Rule of Combining

Dempster’s rule of combining, stated in Equations 2.18 and 2.19 is one method widely used to combine multiple belief mapping functions. It is most useful when combining two or more different belief mapping function of the same variable. Dempster’s rule of combining, while useful, can be problematic if dealing with two mapping functions that have little agreement, as it disregards all areas of disagreement. It should be noted that Yager and Inagaki also present alternative forms of combining evidence from multiple sources, however, a further discussion of these methods is explained by Bae [27] and it omitted here.
\[ m(A) = \frac{1}{1 - K} \sum_{B \cap C = A} m_1(B)m_2(C) \quad (2.18) \]

where

\[ K = \sum_{B \cap C = \emptyset} m_1(B)m_2(C) \quad (2.19) \]

For a given problem formulation, two basic belief assignments are derived from two independent sources. These basic belief assignments both represent information over the same variable and are shown in Figures 2.15 and 2.16.

For this example, the value of \( K \) from Equation 2.19 will be determined first. This value is calculated using propositions from each expert that
do not intersect. Upon inspection it is seen that \( m_{1,1} \) and \( m_{2,1} \) do not intersect. This is the only pair of non-intersecting propositions. The value of \( K \) then becomes

\[
K = \sum_{B \cap C = \emptyset} m_1(B)m_2(C) = (m_{1,1})(m_{2,1}) = (0.4)(0.1) = 0.04 \quad (2.20)
\]

Now, the intersecting propositions must be found in order to create new proposition sets based on the two existing BBAs. The first intersecting pair to be analyzed is the propositions \( m_{1,1} \) and \( m_{2,2} \). For the intersection to be calculated, the Equation 2.18 is needed:

\[
m(A) = \frac{1}{1-K} \sum_{B \cap C = A} m_1(B)m_2(C) = \frac{1}{1-K} (m_{1,1})(m_{2,2}) \quad (2.21)
\]

The intersecting region \( A \) must also be found. For the intersecting propositions \( m_{1,1} \) and \( m_{2,2} \), this region is the interval \([0.2, 0.4]\). Substituting the value of \( K \) calculated in Equation 2.20,

\[
m(A) = \frac{1}{1-0.04} (m_{1,1})(m_{2,2}) = \frac{1}{1-0.04} (0.4)(0.3) = 0.125 \quad (2.22)
\]

Similarly, the second intersecting pair, \( m_{1,1} \) and \( m_{2,3} \), can be analyzed. The formulation for \( K \) does not change, but the rest of Equation 2.18 must be recalculated.
\[
m(A) = \frac{1}{1 - K} \sum_{B \cap C = A} m_1(B)m_2(C) = \frac{1}{1 - K} (m_{1,1})(m_{2,3})
\]  \hspace{1cm} (2.23)

Substituting in the known values, the combined mapping function is shown to be

\[
m(A) = \frac{1}{1 - 0.04} (m_{1,1})(m_{2,3}) = \frac{1}{1 - 0.04} (0.4)(0.6) = 0.25 \hspace{1cm} (2.24)
\]

This process is repeated for each pair of intersecting propositions. The results are shown in Table 2.5.

<table>
<thead>
<tr>
<th>Expert 1 Index</th>
<th>Expert 1 C</th>
<th>Expert 1 m</th>
<th>Expert 2 Index</th>
<th>Expert 2 C</th>
<th>Expert 2 m</th>
<th>Combined Index</th>
<th>Combined A</th>
<th>Combined m</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[0.0, 0.4]</td>
<td>0.4</td>
<td></td>
<td>[0.4, 0.6]</td>
<td>0.1</td>
<td>XXX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>m_{1,1}</td>
<td>[0.0, 0.4]</td>
<td>0.4</td>
<td>m_{2,1}</td>
<td>[0.2, 0.8]</td>
<td>0.3</td>
<td>[0.2, 0.4]</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td>m_{1,1}</td>
<td>[0.0, 0.4]</td>
<td>0.4</td>
<td>m_{2,2}</td>
<td>[0.0, 1.0]</td>
<td>0.6</td>
<td>[0.0, 0.4]</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>m_{1,2}</td>
<td>[0.2, 0.8]</td>
<td>0.4</td>
<td>m_{2,3}</td>
<td>[0.4, 0.6]</td>
<td>0.1</td>
<td>[0.4, 0.6]</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>m_{1,2}</td>
<td>[0.2, 0.8]</td>
<td>0.4</td>
<td>m_{2,4}</td>
<td>[0.2, 0.8]</td>
<td>0.3</td>
<td>[0.2, 0.8]</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td>m_{1,2}</td>
<td>[0.2, 0.8]</td>
<td>0.4</td>
<td>m_{2,5}</td>
<td>[0.0, 1.0]</td>
<td>0.6</td>
<td>[0.2, 0.8]</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>m_{1,3}</td>
<td>[0.4, 1.0]</td>
<td>0.2</td>
<td>m_{2,6}</td>
<td>[0.4, 0.6]</td>
<td>0.1</td>
<td>[0.4, 0.6]</td>
<td>0.02</td>
<td></td>
</tr>
<tr>
<td>m_{1,3}</td>
<td>[0.4, 1.0]</td>
<td>0.2</td>
<td>m_{2,7}</td>
<td>[0.2, 0.8]</td>
<td>0.3</td>
<td>[0.4, 0.8]</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>m_{1,3}</td>
<td>[0.4, 1.0]</td>
<td>0.2</td>
<td>m_{2,8}</td>
<td>[0.0, 1.0]</td>
<td>0.6</td>
<td>[0.4, 1.0]</td>
<td>0.13</td>
<td></td>
</tr>
</tbody>
</table>

Several of the combined propositions have exactly the same interval. Only in this case are the belief mapping values able to be combined. The simplified BBA is shown in Figure 2.17.
Figure 2.17: Dempsters Rule of Combination Example: Final Combined BBA
Particle Swarm Optimization

Particle Swarm Optimization is used in this work in order to solve the difficult problem of path planning, which will be explained further in Chapter 5. Kennedy and Eberhart [28] discovered Particle Swarm Optimization in 1995. Their method, which was based on swarm theory, incorporated randomness and swarm cooperation in order to optimize highly non-linear systems. Gradient-based methods, such as steepest descent, conjugate gradient methods and Sequential Quadratic Programming are increasingly efficient methods of finding minima of convex functions. However, these methods typically cannot distinguish local minima from the global minima. These methods will find the nearest local minima relative to their starting point. For most structural systems, if a valid starting point is known, these methods will be able to improve upon the current design; however, they are highly dependent upon the starting point of the optimization algorithm.

Particle Swarm Optimization (PSO) is classified as a heuristic opt-
mization method. Other heuristic optimization methods include genetic algorithms [29], simulated annealing [30], and ant colony optimization [31]. Most heuristic methods act in a similar way. Typically, they mirror phenomena found in nature. Particle Swarm Optimization is often described to mirror the way that a swarm of birds can find a bird feeder. At the beginning of the process, the birds are spread out over a large area. As time passes, more and more birds find their way to the bird house by basing their movements on each bird’s own knowledge and knowledge of the birds around them. Genetic algorithms mimic the way genes and traits are passed down from generation to generation resulting in a survival of the fittest atmosphere. Simulated annealing mimics the process in which atoms come to rest during the cooling process, while ant colony optimization simulates the way that ants are able to find the quickest means to bring food to the hive.

Of all of the stated methods, genetic algorithms (GA) and Particle Swarm Optimization (PSO) are the most widely used. While each method has its own strengths, PSO was chosen for this research over other heuristic search methods due to its rapidly increasing knowledge base, as shown by its recent popularity, and its statistically shown efficiency advantage over GA [32].
3.1 The Particle Swarm Optimization Algorithm

The Particle Swarm Optimization algorithm simulates the movement of particles through a design space at incrementing time steps. At each point in time, every particle has a position $x^i_k$ and a velocity $v^i_k$, where $i$ represents the index of the particle and $k$ is the time step index. During each time step both the velocity and position of the particles are updated according to Equations 3.1 and 3.2.

$$v^i_{k+1} = w_k v^i_k + c_1 r_1 \frac{p^i_k - x^i_k}{\Delta t} + c_2 r_2 \frac{p^g_k - x^i_k}{\Delta t}$$  \hspace{1cm} (3.1)

$$x^i_{k+1} = x^i_k + v^i_{k+1} \Delta t$$  \hspace{1cm} (3.2)

In Equation 3.1, $w$ represents an inertial factor, $c_1$ is the individuality or self-confidence factor, $c_2$ is the sociability or social confidence factor, $r_1$ and $r_2$ are uniform random variables, $p^i$ is particle $i$'s previous best location, $p^g_k$ is the best neighbor location during step $k$, and $\Delta t$ is the time step size. This equation calculates the velocity of each particle that is used to update the particle’s position according to in Equation 3.2. Typically the time step size is one in order to reduce complexity.

These equations illustrate how the PSO algorithm simulates particle movements over time. The individual particle movements are based on each particle’s previous velocity, each particle’s previous best location, and the best particle location in its neighborhood. One issue experienced by all
heuristic methods is how to best turn the dials, or adjustable factors, of the optimization algorithm. The factors that can be adjusted in PSO include the particle inertia, $w_k$, the individuality factor, $c_1$, and the sociability factor, $c_2$. While Kennedy and Eberhart [28] initially used the conditions:

\begin{align*}
w &= 0.0 \\
c_1 &= 2.0 \\
c_2 &= 2.0
\end{align*} \hspace{1cm} (3.3)

the literature [32] shows that over a wide array of benchmark problems, the following values may be better suited for more efficient convergence:

\begin{align*}
w &= 0.5 \\
c_1 &= 1.5 \\
c_2 &= 1.5
\end{align*} \hspace{1cm} (3.4)

The values for $w$, $c_1$, and $c_2$ shown in Equation 3.5 were used in this work, due to the increased efficiency shown by Hassan et al.

### 3.2 Intertial Updating

Additionally, work has gone into updating the inertia value, $w$, as the optimization algorithm progresses. Typically, the value for inertia is quite
large during the initial iterations. This causes the particles to perform a more global search due to the increased weight of the particle’s past velocity with respect to the individuality and sociability factors. Venter and Sobieszczanski-Sobieski [33] propose that the inertial factor be updated each iteration as in Equation 3.5:

$$w_{k+1} = w_k f_w$$  \hspace{1cm} (3.5)

In this work, Venter and Sobieszczanski-Sobieski found the best value for $f_w$ to be 0.975 across a wide array of benchmark problems. Chatterjee and Siarry [34] proposed an alternate method of inertial updating. They proposed a nonlinear updating equation to achieve efficient convergence behavior. Their updating function, shown in Equation 3.6, is dependent upon a nonlinearity factor $n$.

$$w_{k+1} = \left( \frac{(k_{max} - k)^n}{k^n} \right) (w_{initial} - w_{final}) + w_{final}$$  \hspace{1cm} (3.6)

In this equation, $k$ is the number of the current iteration, $k_{max}$ is the maximum number of iterations, $w_{initial}$ is the initial inertial value, and $w_{final}$ is the final inertial value. Figure 3.1 shows how the inertial updating functions acts with varying degrees of nonlinearity. As the value of $n$ increases, the time in which the PSO algorithm converts to a more localized search is increased. Chatterjee and Siarry [34] suggest that setting $n = 1.2$ is the best choice for most of the benchmark problems they investigated. Because of these studies, for the PSO methodology implemented here, the
updating function shown in Equation 3.5 was used to adjust the inertial weight throughout the design process.

![Figure 3.1: Comparison of Nonlinearity Factor to Inertial Updating](image)

### 3.3 Constriction Factor

Another way of improving the PSO algorithm by increasing convergence efficiency was developed by Clerc and Kennedy [35]. They called their method the constriction method. The constriction factor acts to limit the maximum velocity for any particle. It does this as a function of the individuality and sociability factors, $c_1$ and $c_2$ respectively. The constriction factor is shown here in Equations 3.7-3.9:

$$v_{k+1}^i = K \left( v_k^i + c_1 r_1 \frac{p^i - x_k^i}{\Delta t} + c_2 r_2 \frac{p_k^g - x_k^i}{\Delta t} \right) \quad (3.7)$$

where
\[ K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|} \]  \hspace{1cm} (3.8)

and

\[ \varphi = c_1 + c_2 > 4 \]  \hspace{1cm} (3.9)

Schutte and Groenwold [36] suggest that setting \( c_1 = 2.8 \) and \( c_2 = 1.3 \) yield optimal convergence results when applying the constriction factor method. In lieu of using the constriction factor method, an inertial updating function was chosen to aid in convergence in the work performed in this document.

### 3.4 Niching

In addition to increasing the chance of finding the global optima, or at least a better optima than could be achieved with a gradient-based search, global search methods such as PSO have been used to identify many of the local optima. Because PSO is a global search method, it lends itself well to finding many of the local optima with some slight adjustments in its implementation. There are many reasons why one would prefer to attain a list of local optima rather than outputting only the best position found. Many times when dealing with multi-objective optimization problems, it is unclear as to which design should be chosen. Often several of the results may be superior to others, even if they have similar objective function
values.

One popular way that PSO has been used to identify many local optima is through niching. NichePSO was developed by Parsopoulos and Vrahatis [37], although currently there are several methods of obtaining niching behavior. The idea behind niching is to create several subpopulations of particles. These subpopulations will isolate themselves from other populations based on some criteria, typically though some function combining each particle’s location as well as its objective function value, or fitness, in relation to its neighbors. Niching provides several design solutions where further analyses can be used, such as a pareto analysis or a conjoint analysis [38].

3.5 Implementation Walk Through

In order to more fully understand the physics of the Particle Swarm Optimization algorithm, an example walk-through is presented where one particle’s movements are calculated. For the example problem, the objective function to be minimized is shown in Equation 3.10:

\[ y = x_1^2 + x_2^3 + 2 \times |x_2| \]  

Equation 3.10

A plot of this function is shown in Figure 3.2. For this example the population size is four. These particles will be chosen at random, and their values are shown in Table 3.1. Additionally, particle one has been assigned a previous best value, \( p^1 \) which was at its location in the previous
iteration, making its velocity \( v = [0.059, -0.022]^T \). Also, the weighting factors \( w = 0.5 \), \( c_1 = 1.5 \), and \( c_2 = 1.5 \) are considered here.

![Plot of the Objective Function for the PSO Example](image)

Figure 3.2: Plot of the Objective Function for the PSO Example

<table>
<thead>
<tr>
<th>( p )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.643</td>
<td>0.844</td>
<td>2.703</td>
</tr>
<tr>
<td>2</td>
<td>-0.111</td>
<td>0.476</td>
<td>1.072</td>
</tr>
<tr>
<td>3</td>
<td>0.231</td>
<td>-0.648</td>
<td>1.077</td>
</tr>
<tr>
<td>4</td>
<td>0.584</td>
<td>-0.189</td>
<td>0.712</td>
</tr>
<tr>
<td>( p^1 )</td>
<td>0.521</td>
<td>0.866</td>
<td>2.644</td>
</tr>
</tbody>
</table>

Table 3.1: PSO Example: Beginning Particle Locations

From this information, the velocity for the current iteration as well as the position for the next iteration of particle one. Consider again the equation for calculating velocity within the Particle Swarm Optimization algorithm:

\[
v_{k+1}^i = wv_k^i + c_1 r_1 \frac{p_i^i - x_k^i}{\Delta t} + c_2 r_2 \frac{p_g^k - x_k^i}{\Delta t}
\]

(3.11)

Now consider particle 1, iteration 1:
\[ v_{k+1}^1 = wv_k^1 + c_1 r_1 \frac{p_1^1 - x_k^1}{\Delta t} + c_2 r_2 \frac{p_g^q - x_k^1}{\Delta t} \] (3.12)

Generating two uniform random values for the first particle, \( r_1 \) and \( r_2 \) become 0.392 and 0.485, respectively. The equation then becomes:

\[ v_{k+1}^1 = wv_k^1 + c_1 0.392 \frac{p_1^1 - x_k^1}{\Delta t} + c_2 0.485 \frac{p_g^q - x_k^1}{\Delta t} \] (3.13)

Recall the previous velocity of the first particle, as well as the weighting factors and a time step size of one and the equation is now:

\[ v_{k+1}^1 = 0.5 \times [0.059, -0.022]^T + 1.5 \times 0.392 \frac{p_1^1 - x_k^1}{1} + 1.5 \times 0.485 \frac{p_g^q - x_k^1}{1} \] (3.14)

The value for \( p_1^1 \) is the location of the previous best value of particle one. This, as shown in Table 3.1, is \([0.521, 0.866]^T\). \( p_g^q \) however is the location of the neighbor particle with the best, or fittest value. In this example the fittest value is the lowest value because this is a minimization problem. For a general problem, the calculation of \( p_g^q \) depends upon the number of neighbors specified to each particle. The neighbors of each particle are found by starting with the closest particle to the particle of interest using the particle locations at the current iteration. The next closest particle is then added to the neighbor set and the process continues until the specified number of neighbors is attained. This set of neighbor particles is then searched for the particle with the best value or fitness.
For this example, the number of neighbors is three, making each particle able to interact with each other. This makes particle four the fittest neighbor to particle one, making $p_k^0 = [0.584, -0.189]^T$. These values for $p^1$ and $p_k^0$ along with $x^1_k$, the particle’s current location, can now be substituted into Equation 3.14.

$$v_{k+1}^1 = 0.5 * [0.059, -0.022]^T + 1.5 * 0.392 \frac{[0.521, 0.866]^T - [0.643, 0.844]^T}{1} + 1.5 * 0.485 \frac{[0.584, -0.189]^T - [0.643, 0.844]^T}{1} \tag{3.15}$$

Simplifying this yields:

$$v_{k+1}^1 = [0.030, -0.011]^T + [-0.0717, 0.0129]^T + [-0.0429, -0.7515]^T \tag{3.16}$$

$$v_{k+1}^1 = [-0.0846, -0.7496]^T \tag{3.17}$$

Now, the new location of the particle can be calculated using the velocity from Equation 3.17.

$$x_{k+1}^1 = x_k^1 + v_{k+1}^1 \Delta t \tag{3.18}$$

Substituting the velocity, $v_{k+1}^1$, the current location, $x_k^1$, and the time step size, the new location can be calculated.
\[ x_{k+1}^1 = [0.643, 0.844]^T + [-0.0846, -0.7496]^T \times 1 \quad (3.19) \]

\[ x_{k+1}^1 = [0.5584, 0.0944]^T \quad (3.20) \]

Evaluating the objective function value at the new location of the particle is 0.502, a large improvement over the particle’s previous value of 2.703. This process is then repeated for each particle during a single iteration. This process then repeats itself for several iterations. Often, once a string of iterations fails to yield a significant improvement to the objective function, then the process ceases.

### 3.6 The Eggcrate Function

The Eggcrate Function is a function often used to test various heuristic optimization algorithms. Here, it is chosen because of its high nonlinearity and many local optima. The Eggcrate Function is shown in Equation 3.21:

\[ f(x) = x_1^2 + x_2^2 + 25 \left( \sin^2 x_1 + \sin^2 x_2 \right) \quad (3.21) \]

For this benchmark problem, the input variables \(x_1\) and \(x_2\) are both constrained to be between \(-2\pi\) and \(2\pi\). The global minimum of the Eggcrate function is at \(x = [0, 0]^T\) where \(f(x = [0, 0]^T) = 0\). A plot of this function is shown in Figure 3.3. For this example, 20 particles were used during each iteration, and their initial positions were spread over the design space.
Table 3.2: PSO Parameters for the Eggcrate Function Example

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inertia</td>
<td>$w$</td>
</tr>
<tr>
<td>Linear Inertial Updating Value</td>
<td>$f_w$</td>
</tr>
<tr>
<td>Individuality Factor</td>
<td>$c_1$</td>
</tr>
<tr>
<td>Sociability Factor</td>
<td>$c_2$</td>
</tr>
<tr>
<td>Number of Particles</td>
<td>$n_{part}$</td>
</tr>
<tr>
<td>Number of Neighbors</td>
<td>$n_{nbrs}$</td>
</tr>
</tbody>
</table>

using Latin Hypercube Sampling.

The parameters for the PSO algorithm for the Eggcrate Function are shown in Table 3.2. For this implementation of PSO, the linear inertial updating was performed using Equation 3.5.

Figure 3.4 shows the particle history of the PSO solution to the Eggcrate Function. Notice how during the initial stages the particles evaluate many
locations throughout the design space. As the algorithm progresses, the particles begin to converge upon the global optima at $x = [0, 0]^T$. The solution was found after 293 iterations.

Figure 3.4: Convergence History of PSO for the Eggcrate Function
4

Reliability Based Design

Optimization

4.1 Deterministic Optimization

Deterministic design optimization focuses on the results of the system with respect to given input values. The analysis will yield several design outputs, such as the objective function value, constraint function values, and gradients, or sensitivities, at the input value. The deterministic optimization problem is usually formulated as shown in Equations 4.1-4.3.

Minimize

\[ f(x) \] (4.1)

subject to

\[ g_i(x) \leq 0 \] (4.2)

\[ h_j(x) = 0 \] (4.3)

In this case, \( x \) is the vector of design variables, \( g_i(x) \) are the inequality
constraints, and $h_j(x)$ are the equality constraints. As previously stated, the result of this optimization is a single design point that satisfies the constraint conditions.

4.2 Reliability-Based Design Optimization

Reliability-Based Design Optimization (RBDO) has been a useful evolution of deterministic design optimization. Deterministic design optimization, as explained in Section 4.1, selects a single design point with specific boundary conditions for evaluation. The results of this evaluation determine a new design point. This process repeats itself until the optimization algorithm can no longer find a better solution than the current design point.

One unfortunate downfall of deterministic optimization is that in most engineering problems, the design variables, boundary conditions, and other parameters rather than remaining at the exact specified value or state form some statistical distribution around the mean value. RBDO accounts for these statistical distributions in the analysis through stochastic finite elements [39, 40], FORM [41], SORM [42], or one of many other reliability-based methods [43]. The uncertainty addressed using the methods in this chapter address a slightly different form of uncertainty than the uncertainties addressed by Evidence Theory, explained in chapter 2. Evidence Theory is useful in addressing issues where little information exists about the state of a design variable. Traditional RBDO techniques, like those
described here, address uncertainties due to variations for which much information exists.

Take a given system dependent upon two variables, \( R \) and \( S \). Here, \( S \) represents the load, while \( R \) represents the systems load capacity. For this simple case the limit state becomes:

\[
R - S \leq 0
\] (4.4)

Now we will consider both \( R \) and \( S \) to be random in nature. Thus the probability density function (PDF) and cumulative density (CDF) functions for \( R \) become \( f_R \) and \( F_R \), respectively. Similarly for \( S \), \( f_S \) and \( F_S \) are created as its PDF and CDF, respectively. The failure state is considered when Equation 4.4 is violated. This means that the probability of failure, \( p_f \), is the probability when \( S > R \). This probability can be calculated as

\[
p_f = \int_0^\infty \int_0^S f_R( r ) f_S( s ) dr ds
\] (4.5)

By definition

\[
F_R( s ) = \int_0^s f_R( r ) dr
\] (4.6)

By combining Equations 4.5 and 4.6, the following definition of the probability of failure can be attained:
\[ p_f = \int_{0}^{\infty} F_R(r) f_S(s) \, ds \quad (4.7) \]

Reliability-Based Design Optimization, or RBDO, takes a slightly different approach to designing a system. RBDO takes into account both the value of the input variables and the associated statistical distributions. This implies that the mean value of an input variable, its type of distribution, and its distribution parameters are each taken into account when determining the characteristics of the system output. A typical RBDO problem has a form shown in Equations 4.8 and 4.9:

Minimize

\[ \text{cost}(x) \quad (4.8) \]

subject to

\[ P(G_j(X, z) \leq 0) - P_{fj} \leq 0 \quad (4.9) \]

where \( G_j(X, z) \) is the limit state of failure criteria \( j \) and \( P_{fj} \) is the probability of failure of that limit state.

The probability of failure can be calculated using several different methods. Madsen and Egeland [44] divide these methods into four levels. Level one methods use a deterministic result to form a reliability estimate. These methods often employ a safety factor to find an allowable level of a response value. Level two methods use two values of a response to determine the reliability of a system. Often these two values are the mean response and the response variance. Reliability methods that fall into level three require
an understanding of the joint distributions of inputs and responses. These are needed to calculate a probability of failure. The highest level denoted by Madsen and Egeland, level four, involves considering non-typical metrics in formulating a design including cost of production, maintenance, and repair.

Monte Carlo simulations and the more efficient Latin Hypercube Sampling [45] method are two widely used sampling methods. These two methods are used to generate numerous design points throughout an area of interest. When these methods are used to perform RBDO, each design point is evaluated and a decision is made as to whether the design point yields a failure. Based on the number of failures, a reliability estimate can be made. Sampling methods, although typically computationally demanding, work well when the input variables may be highly correlated, and are often easily applied to most engineering problems.

Another type of RBDO methods rely on an optimization routine to form a reliability estimate. The method now commonly referred to as the Hasofer-Lind algorithm [46] is one method that calculates probability of failure through the use of a normalized design space and linear response approximations. A graphical depiction of the reliability index, $\beta$ is shown in figure 4.1. The Hasofer-Lind algorithm attempts to locate the nearest failure point to the current design point, called the most probable point of failure. Optimization routines work well when the input variables are independent of one another and are computationally more efficient than
sampling methods. However, these methods sometimes struggle to deal with correlation due to the statistical assumptions made in their formulation.

More recent evolutions in reliability assessment, such as Polynomial Chaos Expansion [47] and the Stochastic Finite Element Method [48] have shown value when applied to structural problems by Choi et al. [43]. These methods consist of a combination of sampling methods as well as optimization routines in order to leverage the strengths of both methods.

4.2.1 First Order Reliability Method

A first order reliability method (FORM) calculates a reliability index $\beta$ that is based on the distance of the most probable failure point from the origin in the normalized design space. The FORM reliability estimate is
created by solving the optimization problem depicted in Equations 4.10 and 4.11:

Minimize

\[
\beta = \sqrt{\hat{x}^t \hat{x}} \tag{4.10}
\]

subject to

\[
g(x) = 0 \tag{4.11}
\]

Here, in Equation 4.10, \( \hat{x} \) represents the value of \( x \), the input vector, in the normal space. This conversion is shown here in Equation 4.12 where \( \mu_i \) and \( \sigma_i \) represent the mean and standard deviation of the \( i^{th} \) design variable, respectively:

\[
\hat{x}_i = \frac{x_i - \mu_i}{\sigma_i} \tag{4.12}
\]

Once the random variables have been converted to the normal space, the optimization routine can be run, which will locate the most probable point of failure (MPP). In this case, the limit state is the response where \( g(x) = 0 \).

The performance measure approach (PMA) [49] is a relatively new advancement in Reliability-Based Design. This method poses the optimization problem statement in a slightly different manner. PMA poses the optimization problem as

Minimize

\[
g(x) \tag{4.13}
\]
subject to

\[ \beta = \sqrt{x^T \dot{x}} = \beta_t \]  \hspace{1cm} (4.14)

where \( \beta_t \) is the target reliability index. Notice how the objective and constraint functions are reversed when compared to the reliability index method formulation in Equations 4.10 and 4.11. Although this formulation typically converges faster, due to the more easily satisfiable constraint function, it requires the target reliability function to be updated through an updating function [50].

4.2.2 Metamodeling

In order to reduce the required computational time inherent in any form of optimization, metamodels are often used in the place of full physical models. The optimization problem is often initialized with a full computational analysis. However, for each ensuing iteration, the full computational analysis may only need to be performed if the desired point of evaluation does not have two previous evaluations within the desired move limits. For example, the move limits may be set to a maximum change of any input variable of 10%. These move limits ensure that an function approximation is not used in an unexplored region of the design space.

TANA2, Two-point Adaptive Nonlinear Approximations, shown in Equation 4.15, can be used to create the metamodels. TANA2 takes into account the nonlinearity of the function. The approximation accounts for the second-order terms of the expansion. Here the TANA2 approximation
\( \tilde{f}(x) \) is evaluated at point \( x \) and based around function evaluations and gradients at two design points, \( X_1 \) and \( X_2 \):

\[
\tilde{g}(X) = g(X_1) + \sum_{i=1}^{n} \left( \frac{\partial g}{\partial x_i} \right)_{X_1} \frac{x_{i,1}^{1-p_i}}{p_i} (x_i^{p_i} - x_{i,1}^{p_i}) + \frac{\epsilon_1}{2} \sum_{i=1}^{n} (x_i^{p_i} - x_{i,1}^{p_i})^2
\]

(4.15)

The values of \( p_i \) and \( \epsilon_1 \) are solved so that the approximation function value and derivatives equal the true function value and derivatives at both \( X_1 \) and \( X_2 \). More clearly, \( \tilde{g}(X_1) = g(X_1), \tilde{g}(X_2) = g(X_2), \left( \frac{\partial \tilde{g}}{\partial x_i} \right)_{X_1} = \left( \frac{\partial g}{\partial x_i} \right)_{X_1}, \) and \( \left( \frac{\partial \tilde{g}}{\partial x_i} \right)_{X_2} = \left( \frac{\partial g}{\partial x_i} \right)_{X_2} \) for \( i \) from 1 to \( n \) design variables. Once the desired number of approximations, \( N \), are made about \( X \) are computed using each previously evaluated point, they are combined into a multipoint approximation (MPA) using Equation 4.16. This represents the response surface of the performance measures:

\[
\tilde{f}(x) = \sum_{i=1}^{N} w_i(x) \tilde{f}_i(x)
\]

(4.16)

The weighting factor, \( w_i \), is calculated using Equation 4.17, which is shown here:

\[
w_i(x) = \frac{\phi_i(x)}{\sum_{i=1}^{N} \phi_i(x)}
\]

(4.17)

The weighting function, \( \phi_i \), is calculated based on the distance between the point of interest and the point at which the approximation is based, as shown in Equation 4.18. This distance is denoted as \( L_i \):
\[ \phi_i = \frac{1}{L_i^2} \] (4.18)

This multipoint approximation method can be used either by performing a design of experiments or by continually updating the approximation as needed. One way to form a multipoint approximation function is by performing a design of experiments sampling on the full-physics model. Once these full-physics evaluations are performed, the approximation can be formed around these points, and all future evaluations are calculated using the approximations.

Another way to use multipoint approximations is to update the approximations continuously. In this case, the full-physics evaluations are performed any time there are less than the desired amount of previous evaluations within the specified move limits. This ensures that approximations are not based on evaluations that are too far away from the desired evaluation point.
Path Planning

Many authors have addressed path planning of autonomous vehicles. Much of this research aims to closely match the actual path of the vehicle to the desired path. For this task, many algorithms have been developed to minimize path following error. This is typically measured as a sum of the distance of the actual path from the desired path, while dealing with mass and inertial properties of the vehicle. Although Naeem et al. [51] make several interesting points about the self-locating capabilities of undersea vehicles, this research assumes that the UUV has precise knowledge of its location and target at all times. Furthermore, it is not the interest of this work to define a control algorithm that can properly follow the specified path. Many authors, including Do et al. [52], address the issue of path following control methods. That is not to say that controls are not considered in this work. The dynamic forces on the vehicle are monitored throughout the path, however, as long as maximum design forces are not exceeded, it is assumed that the on-board controller will provide adequate
forces for the vehicle to maintain its predefined course.

There are two major types of path planning. One form of path planning involves planning where a finite number of pathways that are available. This type of path planning would be similar to the methods that various mapping softwares that produce driving directions. Typically, the A* algorithm or similar algorithms are used to solve these finite problems. The other form of path planning, the type that this research involves itself with, is the problem of infinite path planning. This type of path planning is best suited to solve problems such as aircraft and seacraft where there are no defined paths they must follow. This should not be confused with aircraft routing problems where an airplane must follow a general path from one airport to another. This type of infinite path planning may find the best way to cross a forest with no restriction on following the existing trails. It may be easier to get through the forest by following the trails for much of the journey, but there may be instances where it is advantageous to depart from the existing trails, and in infinite path planning, this is allowable. Cameron [53] provides additional descriptions on the various forms of path planning.

5.1 PSO and Path Planning

An efficient way of posing infinite path planning as an optimization problem is the potential field approach addressed by Hwang and Ahuja [54]. Here, they applied a penalty to the objective function in areas in and sur-
rounding an object to be avoided, or a hazard. In their paper, they suggest that a penalty be applied that is roughly the inverse of the distance away from the hazard.

A popular area for path planning is research concerning unmanned aerial vehicles. Undersea vehicles as well as aerial vehicles have similar spaces to explore. Foo et al. [18] presents the use of particle swarm optimization in order to find ideal pathways. Here, they illustrated the advantages of using a simplistic optimization algorithm for path planning. Since typically the design space is nearly unlimited for path planning, particle swarm optimization (PSO) explores a large area and is able to offer an increased likelihood of finding many possible pathways. This research also includes object avoidance.

In the work by Foo et al., the forces on the system are not taken into account when designing a path. The paths used in this work are also greatly simplified. The paths created are point-to-point piecewise linear paths that further do not account for turning forces. Similarly, Hu et al. [55] imposes a Genetic Algorithm in order to solve the parallel problem of commercial airline path planning. Their research however also fails to properly account for any dynamic forces on the structure due to the chosen path. Again, the path consists of a point-to-point path. In both of these works, by Foo et al. and Hu et al. speed is completely omitted.

Wang et al. [56] aims to address the inefficiency of the particle swarm method. Here, the research attempts to use constrained and semi-infinite
constrained optimization to develop paths for exploring an area while avoiding collision. These methods need far less function evaluations than PSO, but are vulnerable to local optima. Due to the high nonlinearity that path planning objectives and constraints can create, many local optima may exist. Because of the high nonlinearity of the path planning problem as well as the presented literature [32], PSO is chosen for path planning in this research.

5.2 Trajectory Optimization

Much information can also be found with regards to trajectory optimization. Much current work has gone into using trajectory optimization techniques in order to create paths for space vehicles [16, 57, 58]. This work is typically coupled highly with the thermal and aerodynamic forces that are created upon exit from and reentry to Earth’s atmosphere. Unlike the previously mentioned work that excluded the dynamic forces, in these studies, the paths are primarily driven by the vehicle loading and are typically indifferent to the time in which a path takes. However, the design space for the path in trajectory optimization are usually limited to various arcs, and gradient based optimization algorithms are sufficient in finding optimal solutions. Trajectory optimization softwares, such as POST (Program to Optimize Simulated Trajectories) [14], deal with similar design spaces. POST optimizes elliptical orbit paths as well as Earth exit and reentry paths.
5.3 Deterministic Path Planning

The path planning in this research considers the two-dimensional movement of the undersea vehicle, while also taking into account its velocity. The paths are defined by specifying both a starting point, an ending point, and the operating depth below sea level. Other entities that may be defined are the areas that the undersea vehicle must avoid, or hazards, and the areas that the undersea vehicle must pass through, or waypoints. Both the hazards and waypoints are described by specifying a center point and a radius. For simplicity, all hazards and waypoints are defined using circles. However, most shapes can be attained by combining circles of different radii if required, as seen in Figure 5.1.

![Approximating a Square Hazard with Circle Hazard Definitions](image)
5.3.1 Path and Velocity Profile Interpolation

The path is defined by connecting any number of design points consecutively. Many ways of connecting these design points exist. Figure 5.2 shows a comparison of a linear, cubic spline, and cubic Hermite interpolation methods.

In order to interpolate in both the $x$ and $y$ directions, the locations of each interpolation point along the $x$ any $y$ axes was created as a function of an intervening variable $t$. The equation for linear interpolation between the data points $x_1 = f(t_1)$ and $x_2 = f(t_2)$ is shown here:
\[ x = x_1 + \frac{(x - x_1)(t_2 - t_1)}{x_2 - x_1} \] (5.1)

The cubic spline interpolation method works slightly differently. While a linear interpolation ensures that the intersection of two adjoining interpolation functions be continuous, a cubic spline interpolation enforces the first and second derivatives be continuous as well. Assume the interpolation function \( S_1 \) interpolates a value for the x-coordinate between the design points \( x_1 \) and \( x_2 \). Similarly, assume the interpolation function \( S_2 \) interpolates a value for the x-coordinate between the design points \( x_2 \) and \( x_3 \). A linear interpolation function ensures that \( S_1(x_2) = S_1(x_2) \). A cubic spline interpolation ensures that not only does \( S_1(x_2) = S_2(x_2) \), but also that \( \frac{dS_1}{dt}(x_2) = \frac{dS_2}{dt}(x_2) \) and \( \frac{d^2S_1}{dt^2}(x_2) = \frac{d^2S_2}{dt^2}(x_2) \). This means that the the first and second derivatives are continuous. In order to ensure these properties a system of equations must be solved. One interesting issue when performing a spline is the issue of what the values of \( \frac{dS}{dt} \) and \( \frac{d^2S}{dt^2} \) should be at the end points. If information about the endpoints is known, then the values can be set to the known first and second derivative value. However, if no derivative information is known about the endpoints then typically a natural spline is considered. in this case both derivatives at both endpoints are considered to be zero. This calculation of a natural cubic spline is used in the path planning included in this research.
The piecewise function for a cubic spline is shown here:

\[ x = ((a_i(t - t_i) + b_i)(t - t_i) + c_i)(t - t_i) + x_i \]  
(5.2)

where

\[ a_i = \frac{b_{i+1} - b_i}{3h_i} \]  
(5.3)

\[ c_i = \frac{x_{i+1} - x_i}{h_i} - \frac{h_i(b_{i+a} + 2b_i)}{3} \]  
(5.4)

and

\[ h_i = t_{i+1} - t_i \]  
(5.5)

The \( N - 1 \) values of \( b_i \) can then be satisfied by solving the following equation:

\[ \frac{h_i b_i}{3} + \frac{2(h_i + h_{i+1})b_{i+1}}{3} + \frac{h_{i+1}b_{i+2}}{h_{i+1}} = \frac{x_{i+2} - x_{i+1}}{h_{i+1}} - \frac{x_{i+1} - x_i}{h_i} \]  
(5.6)

The Hermite polynomial interpolation method is similar to the cubic spline interpolation method except that the slope at each data point is required.

As previously mentioned, several prior path planning researchers have used linear interpolation because of it’s ease of implementation. However, by using a linear interpolation, the body forces on the vehicle during mis-
sion simulation are difficult to calculate because of the discontinuities in the first and second derivatives. While both the cubic spline and Hermite polynomial interpolation methods have continuous first and second derivatives, the slopes of the path at each of the design points are not specified. Due to this, the cubic spline interpolation method is used for creating a path between the design points.

Additionally the velocity is modeled similarly. Like the path calculation, a cubic spline is fit to the velocity design points. The difference is that both the starting and finishing velocities are design points, and each design point only has one input value. The points are spaced evenly along the length of the path, not along the time passage of the path. This ensures that changing the value of the one velocity design variable does not affect the location of the remaining design variables. This minimizes the coupling between velocity design variables.

From the path profile, a radius of curvature can be calculated. This value can be used on its own in order to limit a minimum turning radius for an undersea vehicle, or it can be coupled with the velocity profile to calculate an estimate on the required turning forces of the vehicle as it performs a turning maneuver. The radius of curvature of the path can be calculated by fitting a circle through three consecutive points. Using the radius of the circle is found, the required centripetal force can be calculated by combining the turning radius with the velocity at that point. Respectively, letting \( r_i \) and \( v_i \) represent the radius of curvature and velocity
of the path at the point \( i \), and \( m \) be the mass of the vehicle, the centripetal force, \( F_{c}^{i} \), can be calculated:

\[
F_{c}^{i} = \frac{mv_{i}^{2}}{r_{i}}
\]  

(5.7)

5.3.2 Hazard Penalty Calculation

The path’s inclination to avoid hazards is enforced through a penalty function. The penalty function enforced is proportional to the distance of a path point to the nearest edge of the hazard that it has violated. This function is shown in Equation 5.8:

\[
P_{H} = \sum_{p_{i} \in H} R_{H} - (D_{p_{i}(H)})
\]  

(5.8)

where \( P_{H} \) is the value of the penalty, \( p_{i} \) is the location of the path point \( i \), \( H \) is the hazard area, \( R_{H} \) is the radius of the hazard, and \( D_{p_{i}(H)} \) is the distance of the path point \( p_{i} \) to the center of the hazard.

5.3.3 Checkpoint Penalty Calculation

The checkpoints also enforce a penalty function in order to ensure that the path passes through each checkpoint. The checkpoint penalty is calculated by finding the nearest path point to the center of the checkpoint. If the distance between the path point and the center of the checkpoint is less than that of the radius of the checkpoint, then the penalty is zero. Otherwise, the penalty is proportional to the distance of the nearest path point
minus the radius of the checkpoint. This penalty is shown in Equation 5.9:

\[ P_C = \delta(p_n \in C) (D_{p_n} - R_C) \]  

(5.9)

where \( P_C \) is the value of the checkpoint penalty, \( p_n \) is the location of the path point nearest to the checkpoint area, \( C \). \( R_C \) is the radius of the hazard, and \( D_{p_n} \) is the distance of the path point \( p_n \) to the center of the checkpoint.

### 5.3.4 Maneuvering Limits Penalty Calculation

The path planning module enforces two maneuvering limitations on the vehicle. The first is a simple lower bound on the turn radius. The lower bound is enforced without considering the vehicle’s velocity. This lower...
limit is due to the limits on the control surface deflections. Again a penalty function is used to enforce this limit. The penalty is enforced as a linear function of the difference between the lower limit on the actual turning radius at each point.

The turning radius at each point is calculated by fitting a circle through the current path point, $p_i$, and the points before and after the current point, $p_{i-1}$ and $p_{i+1}$. Let the coordinates of $p_i$ be $[x_i, y_i]$. Similarly the points $p_{i-1}$ and $p_{i+1}$ have the coordinates $[x_{i-1}, y_{i-1}]$ and $[x_{i+1}, y_{i+1}]$. Given this notation, the equation of a circle is:

$$x^2 + y^2 + Dx + Ey + F = 0$$

Through substitution, the following linear system can be created:
\[
\begin{bmatrix}
  x_{i-1} & y_{i-1} & 1 \\
  x_i & y_i & 1 \\
  x_{i+1} & y_{i+1} & 1
\end{bmatrix}
\begin{bmatrix}
  D \\
  E \\
  F
\end{bmatrix}
= -\begin{bmatrix}
  x_{i-1}^2 + y_{i-1}^2 \\
  x_i^2 + y_i^2 \\
  x_{i+1}^2 + y_{i+1}^2
\end{bmatrix}
\] (5.11)

Equation 5.11 can then be solved for \(D\), \(E\), and \(F\). Completing the squares of Equation 5.10, yields the following:

\[
\left( x + \frac{D}{2} \right)^2 + \left( y + \frac{E}{2} \right)^2 = \frac{D^2 - 4F + E^2}{4}
\] (5.12)

which makes the radius:

\[
r_i = \sqrt{\frac{D^2 - 4F + E^2}{4}}
\] (5.13)

Using the velocity at each path point and combining these values with the approximated turn radii, the centripetal force on the vehicle at each path point can also be calculated. The centripetal force exerted perpendicular to the direction the vehicle is moving in is calculated

\[
F_C = \frac{mv_i^2}{r_i}
\] (5.14)

where \(m\) is the mass of the vehicle, \(v_i\) is the velocity of the vehicle at path point \(i\), and \(r_i\) is the turning radius calculated in Equation 5.13. The maximum centripetal force is limited by an upper bound by a linear hydrodynamics approximation of the control force produced by a standard control fin.
5.4 Evidence Theory Based Path Planning

Evidence Theory, as explained in Chapter 2, is incorporated into the path planning to address the uncertain locations of hazards in the mission definitions. Evidence Theory is incorporated by adjusting the hazard penalty function shown in Equation 5.8. This value, $P_H$, is multiplied by the plausibility decision value, discussed in Section 2.3. This allows for a continuous penalty function value across the design space, which aids in optimization convergence.

5.5 Path Planning by Additive Freedom

The path planning is performed using the Particle Swarm Optimization algorithm (PSO). Here, in this work the path planning is performed assuming that complete knowledge is known of the design space, and no hazards or checkpoints are initiated after the start of a mission. This is to prevent the mission profile and subsequent system design from being affected by a suboptimal artificial intelligence controller. In essence, this work focuses on what the optimal path would be for ideal artificial intelligence controller. Additionally, for better graphical interpretation, the work is limited to two-dimensions. These two dimensions could be looked at as a top-down view, where the vehicle has a fixed depth, or from the side where the vehicle’s motion is fixed to a vertical plane.

The path planning is done by defining one or more points through which
the vehicle must pass through. This point(s), along with the starting and ending points, is used to fit a spline which defines the path. A similar method is used to define the velocity profile. The starting and ending velocities, as well as any number of intermediate velocities, are considered to be design points. Then at any given point along the path, the velocity is found by a cubic spline interpolating between the defined velocity design points. The procedure for the path calculation using a cubic spline interpolation is shown in Figure 5.5.

![Figure 5.5: Path Definition Using Spline Interpolation](image)

Initially, a straightforward application of PSO was used to find the optimal path, however, the convergence was very slow for complex mission design spaces where a large number of design variables was needed to identify a sufficient path. If the number of design variables was reduced, the convergence time was reduced, however the path was not given adequate freedom to pass through each way point without avoiding the
mission hazard areas.

In order to overcome this difficulty, path planning by additive freedom (PPAF) was developed in this work. The PPAF method allows the PSO algorithm to converge quickly, while still maintaining adequate freedom for the path to maneuver around hazards and through way points efficiently. The additive freedom method begins with a reduced number of design variables to specify the path as well as the velocity profile. These design variables are given large move limits so that they may fully explore the design space. Once a convergence criterion is reached for this stage of the additive freedom method, an additional design point is added to the path profile design problem and to the velocity profile design problem. The design point move limits are reduced and the optimization is performed again, based on the solution from the previous stage. This process is repeated until an overall convergence criterion is met. The convergence criteria is met when an additional design point is added and the optimizer fails to find a solution better than the previous stage with fewer design variables.

5.5.1 Iteration Procedure

The initial iteration of the path planning by additive freedom method involves first defining how many spline points will be used to determine the initial design problem. It was found that the most efficient initial number of spline points for the path was equivalent to the number of
checkpoints in the mission plus one. For missions with many checkpoints, this may require additional computational time, however for missions with only a few checkpoints and many hazards, this formulation is very efficient. This allowed a base level of flexibility to define a rough path through the mission space.

Furthermore, the initial freedom given to each design point to move about the mission space was initially set to the entire design space. That is, they are allowed to move 50% of the range of the design space in any direction. Initially, the center points for the spline points are placed in the middle of the design space, so that they may span the entire design space. The particle swarm optimization algorithm is then used to converge upon a solution for this initial step of the additive freedom method.

Once a solution has been reached for this initial case, the optimal path is passed on to the next step of the additive freedom method. This path is used to define the central values of the design points for the new current step of the additive freedom method. The number of spline points is increased by one. These new spline points are then distributed along the current optimal path equally across the length of the spline. These points then act as center points for the new move limits. The move limits on each spline point are then reduced by 25%. Recall in the previous step that the move limits were 50% of the range in any direction. This means that during the second step of the additive freedom method, the move limits would be reduced to 37.5% or 50% * 75% in any direction. The particle
swarm optimization algorithm is then used to converge on a new optimal solution for the current step of the additive freedom method.

This process is then repeated until the addition of more spline points fails to yield a better result than a previous step. In order to use the additive freedom method to provide both path and velocity profiles, the iteration steps are alternated to allow a simultaneous convergence of both the path and mission profiles. More clearly, the initial step of the additive freedom method is used to define an optimal path profile. Using this designed path profile, an optimal velocity profile is defined using an initial step of the additive freedom method. Using this velocity profile, the second step of the additive freedom method is used define an updated path profile. Again, this path profile is used in combination with the second step of the additive freedom method for the velocity profile. This process then continues until the convergence criterion is met.

5.5.2 Comparison to Current Methods

In order to examine the effectiveness of this method a comparison test was run fro three sample missions. For this comparison test, an identical particle swarm algorithm was used where the number of particles for each optimization run was $2N^2$, where $N$ is the number of design variables, and the number of neighbors for each particle was the nearest 5%. For this example, the number of design variables for the PSO method for problem was 18, while for the PPAF method the last iteration had 18
design variables, starting from 9 variables. This method was tested for three mission profiles of differing complexity and the results are shown in Table 5.1.

Table 5.1: PSO vs Additive Freedom Method Comparison for Example Missions

<table>
<thead>
<tr>
<th>Mission 1 (9 Hazards)</th>
<th>Method</th>
<th>Mission Time (s)</th>
<th>Calculation Time (s)</th>
<th>Full Function Evaluations</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSO</td>
<td>390</td>
<td>6801</td>
<td>4258008</td>
<td></td>
</tr>
<tr>
<td>PPAF</td>
<td>287</td>
<td>292</td>
<td>67810</td>
<td></td>
</tr>
<tr>
<td>Mission 2 (7 Hazards)</td>
<td>PSO</td>
<td>436</td>
<td>3318</td>
<td>490536</td>
</tr>
<tr>
<td>PPAF</td>
<td>341</td>
<td>210</td>
<td>56476</td>
<td></td>
</tr>
<tr>
<td>Mission 3 (2 Hazards)</td>
<td>PSO</td>
<td>699</td>
<td>508</td>
<td>82295</td>
</tr>
<tr>
<td>PPAF</td>
<td>692</td>
<td>424</td>
<td>85088</td>
<td></td>
</tr>
</tbody>
</table>

The path planning by additive freedom method shows itself to be a much more efficient approach for path planning than that of simply applying the PSO method. These three examples show that using additive freedom greatly reduces the required computational effort needed for complex design spaces. When the mission is more simple, as in example mission 3, the additive freedom method simply matches the efficiency of the standard PSO approach. In missions 1 and 2, where there were 9 and 7 hazards, respectively, the reduction in calculation time was 96% for mission 1 and 94% for mission 2.
6

SNARC

The Synthesis of Naval ARchitectural Concepts (SNARC) program is a software currently in development at the Naval Undersea Warfare Center in Newport, Rhode Island [59]. This program is capable of calculating an underwater vehicle assembly configuration that meets certain design and performance criteria. SNARC’s primary focus is on component selection and sizing. The different sizes of components can be seen in the screen capture displayed in Figure 6.1. SNARC has built-in models for payloads, hull designs, propulsors, motors, sonar systems, and batteries.

Each of these models are calculated using various empirical equations, simplified physics calculations and several naval component design programs. Often, the user is able to choose the desired fidelity or type of model to be included in the design. For instance, the engine model can be modeled using a simple thermal engine model, a thermal engine model with depth-dependent efficiency, or a detailed Self-Contained Chemical Energy Propulsion System (SCEPS) model. SNARC has models that account for
power conversion, structural response, fluid dynamics, thermodynamics, and acoustics. Because of the wide array of included models, this is truly a multidisciplinary component configuration design program. The most notable available models are shown in Figure 6.2.

SNARC is a weapon synthesis environment built to quickly automate tradeoffs between component selection and sizing. This allows expert-in-
the-loop type design while minimizing the mundane task of repetitive hand calculations. The framework that SNARC is built upon is the Weapon Synthesis Environment package developed by Kusmik [60]. This framework was developed to manage interactions of subcomponents and components at the system level. Also, this framework allowed for components to be altered to allow for variable-fidelity modeling as well as for the input of new component models as the available undersea weapons technology evolves. The majority of the SNARC program is written in C++ to allow for speed of computation as well as the fact that it is in a language that is familiar to most. Additionally, this allows for various MATLAB script files to be incorporated more easily.

As previously stated, the key aspect of SNARC is to allow designers to more easily weigh design decisions, as well as to get immediate feedback as to how component availability affect the undersea vehicle’s ability to perform specified missions. These types of trade studies allow designers to look closely at complex competing variables. It also allows the designer to adapt a forward looking design mentality that can help identify what the key limiters to performance for these undersea vehicles.

These limiters can then be analyzed and studies can be done that can accurately quantify what type of benefits may be reaped if a technological breakthrough could be made increasing the ability and efficiency of a component. These types of studies allow military decision makers to identify key limiting technologies that need to be researched.
6.1 System Configuration Design

SNARC, as mentioned in the previous section, performs a component selection and sizing process in order to identify possible component configurations that work together to form an undersea vehicle system. The inputs to SNARC require the user to select the components that are available for use in the undersea vehicle as well as the desired mission that the system must complete. These inputs are sent to the SNARC executable file through means of textual input. While the details of the text input format are not included, there is no loss in the understanding of the usability of SNARC.

6.1.1 System Desirability

SNARC has several methods that it uses to measure models against each other. One way that SNARC uses to select the best model among competing designs is total system volume. The user can also specify that the primary representation of the desirability of a system’s design is the vehicle’s weight. The number of systems as well as the system noise can be used as measures of system desirability. For any design case, upper limits may be placed on any or all of the specified attributes. Additionally, indifference limits may be placed on any of the system attributes as well. Since SNARC may find many similar configurations that all complete the specified mission, the results may produce numerous nearly identical configurations. Indifference limits ensure that in order to be considered
a different design in SNARC, the attribute be different by a specified amount. For instance, the user may specify that the system volume must differ by at least ten kilograms than any current configuration in order to be considered a new configuration. In order to identify optimal component configurations SNARC uses the A* algorithm.

6.1.2 A* Algorithm

The A* (A Star) Algorithm was discovered by Hart et al. [61] in 1968. The A* Algorithm was developed as a search algorithm based on branch and cut methods. This method uses predefined nodes to determine the best path to get from the initial node to then ending node. Within SNARC, the A* Algorithm is used to determine which components should be included in the vehicle’s configuration. Once an initial set of components are selected, an iterative process is used to size each component to provide adequate system performance.

The A* Algorithm first defines an initial set of nodes. At this point that the algorithm attempts to reduce the number of valid branches. It reduces the number of branches at each step by creating an estimate of the cost of following the path initiated by that individual branch. The estimation of the cost is an important aspect of the A* Algorithm. This estimate must be an admissible function. An admissible function is a function where the estimate never overestimates the cost of getting from a given node to the final node. If this is true, then at each step, only the branch or branches
having the lowest cost need to be further explored. An example of a simple system path is shown in Figure 6.3.

In Figure 6.3, $g_i(x)$ represent a true cost calculation. For this example, twelve system evaluations must be examined in order to explore each combinatory option. The A* algorithm uses estimations to reduce this task of comparing path costs. The estimation technique used by the A* Algorithm is shown in Figure 6.4. The estimation function, $h(x)$, approximates the cost of getting from the current node to the final node. As previously stated, this allows various branched to be cut, thus reducing computational effort. For this example, if two of the initial four branches can be eliminated, the number of systems that need to be evaluated are reduced by half.
6.2 Mission Analysis

SNARC finds system configurations that are able to complete the user specified mission. Each component must be of sufficient size to provide enough energy, force, or other vehicle property so that the velocities, electrical energy, noise requirements, depth, and duration can be attained. The mission specification section within SNARC requires the user to input each portion of a mission as a leg. Each leg is defined by specifying the operating depth, the distance traveled, the velocity, and sonar status. Once these are specified, SNARC ensures that each criteria is met by the selected valid configurations.
6.3 Power Systems

6.3.1 Battery

There are two different base models for the simulated performance of a battery. The base model “Battery” is a simple battery model that performs straightforward algebraic relations for the power output of the battery. The required inputs for this model are the battery efficiency \( e_{\text{f\,bat}} \), stowage factor \( s_{\text{f\,bat}} \) and specific energy \( s_{\text{e\,bat}} \). The battery’s efficiency is the ratio of the energy available for use versus the total calculated energy of the battery. The stowage factor of the battery is the ratio of the battery’s mass to its volume. This may also be referred to as density. Finally, the specific energy is simply the ratio of the battery’s total energy to its mass. The available energy out, \( E_{\text{out}} \), can be related to the volume of the battery, \( vol_{\text{bat}} \) with the following relationship:

\[
\frac{e_{\text{f\,bat}} \times s_{\text{f\,bat}} \times s_{\text{e\,bat}} \times vol_{\text{bat}}}{E_{\text{out}}} = 1 \tag{6.1}
\]

The more complex battery model is based on the FORTRAN battery models developed by Raymond W. Roberts of the Naval Undersea Warfare Center in Newport, RI. This battery model is based on an aluminum-silver oxide battery. This battery is based on the chemical reaction shown here:

\[
2Al + 3AgO + 2OH^- + 3H_2O = 2Al(OH)_4^- + 3Ag, E_{cell} = 2.7V \tag{6.2}
\]
where $E_{cell}$ is the resulting output voltage of the reaction.

### 6.3.2 Engine

SNARC has three different engine models: a simple thermal engine, a thermal engine with depth-dependent performance, and a detailed stored chemical energy propulsion system model. The most basic, the simple thermal engine acts as a converter for changing chemical energy into rotational power. This relationship is based on an equation similar to Equation 6.1. To use this model, the user must input the engine’s efficiency ($e_{f_{eng}}$), stowage factor ($s_{f_{eng}}$) and specific energy ($s_{e_{eng}}$). Using the notation $E_{out}$ for available energy out and $vol_{eng}$ the volume of the engine, the following relationship can be made.

$$
\frac{e_{f_{eng}} * s_{f_{eng}} * s_{e_{eng}} * vol_{eng}}{E_{out}} = 1
$$

(6.3)

The more detailed engine with depth dependent efficiency involves the use of a slightly higher fidelity thermal engine model than the base design. For this model, the engine’s efficiency is not a static number, but a number that varies with depth. The highest level of detail, uses a high fidelity analysis of a stored chemical energy propulsion system (SCEPS). This module models a typical engine used in the Mk 50 torpedo. It is a closed steam turbine cycle. A basic closed steam turbine cycle is shown in Figure 6.5. The self-containment is made possible by using the chemical reaction between sulfur hexafluoride and lithium. The reaction equation is shown
The chemical reaction in Equation 6.4 provides a closed cycle heat generation plant for the engine cycle. This equation is capable of creating 20,000 BTU’s of heat per pound of lithium.

\[ 8Li + SF_6 \rightarrow Li_2S + 6LiF + Heat \quad (6.4) \]

6.3.3 Fuel System

There are three models within SNARC that can model the fuel system. The three model encapsulate both the fuel itself and the tank which holds the fuel. The most simple fuel model represents a basic monopropellant-type fuel which is specified by providing the fuel’s efficiency \( ef_{fuel} \), stowage factor \( sf_{fuel} \) and specific energy \( se_{fuel} \). Once these are provided, the fuel volume, \( vol_{fuel} \), and available energy, \( E_{out} \) can be related using Equation 6.5. This fuel model assumes that the fuel bladder, if applicable, is of negligible weight and volume.
The more detailed fuel model includes a pressure compensated shell around the fuel tank. The fuel’s efficiency, stowage factor, and specific energy must be specified similarly to the more simple fuel model, and Equation 6.5 is also used here to calculate energy and volume.

The most detailed fuel model mimics that of the most simple model, but contains an adjustment to the volume calculations. The fuel system for this model compensates for hydraulic, fuel, and electrical lines that must be run past the fuel tank. The two previously mentioned models do not account for this.

6.3.4 Power Converters

There are many types of power conversion models although each acts very similarly. The available power conversion components include alternators (rotational to AC), DC to DC converters (DC voltage to different DC voltage), generators (rotational power to DC), inverters (DC to AC), rectifiers (AC to DC), and transformers (AC voltage to different AC voltage). In the previous statement, AC stands for alternating current, and DC stands for direct current. Again, each of these models acts in a similar way. For each model, an efficiency, \( e_{f_{pc}} \), stowage factor, \( s_{f_{pc}} \), and specific energy, \( s_{e_{pc}} \), must be specified. They are related to volume \( vol_{pc} \) and available energy out, \( E_{out} \) using the following equation:

\[
\frac{e_{f_{fuel}} \times s_{f_{fuel}} \times s_{e_{fuel}} \times vol_{fuel}}{E_{out}} = 1 \quad (6.5)
\]
\[
\frac{e f_{pc} * s f_{pc} * s e_{pc} * v o l_{pc}}{E_{out}} = 1
\]  

(6.6)

6.4 Structural Systems

6.4.1 Structural Hull

Within SNARC, only one structural hull model is present. The hull model used is based on a curve fit of previous designs, however the data used for the curve fit is unavailable. Calculations of drag, volume and lift are calculated using basic hydrodynamic equations based on hull shape assumptions. Figure 6.6 shows a side view of a kinematic UUV model.

![Figure 6.6: Side View of Kinematic Model](image)

From this figure, the basic equations of motion can be calculated. Here \( F_{gravity} \) represents the force of gravity pulling down on the UUV, which is proportional to the mass. Since the UUV will be submerged during most of its operation, buoyancy forces, depicted as \( F_{buoyancy} \) push it toward the surface. This is due to Archimedes principle. Archimedes stated that the buoyant force acting on an object is equal and opposite to the gravitational forces acting on the amount of fluid displaced by the object. The buoyant
force, rather than acting at the center of mass similar to gravitational forces, acts at the center of volume.

Several other forces are present in this kinematic model. These forces are due to more controllable inputs however. The axial force that results from the spinning propulsor, \( F_{\text{propulsion}} \), is a resultant of the propulsion system. Additionally, the control fins of the UUV are artificially added in by supplying point forces at the center of where the fins would act. Initially only a two-dimensional model will be used. In future work the fins will be incorporated into a three-dimensional problem.

The kinematic model in Figure 6.6 shows the basic forces acting on the body during operation. If the propulsive, drag, and lift forces are assumed to rotate with the UUV, and the buoyancy and gravitational forces are assumed to act regardless of the UUV’s orientation, the equations of motion can be derived as shown.

The UUV is assumed to have a given angle of attack, denoted as \( \theta \) in Figure 6.7. The forces in the horizontal direction can be calculated using simple geometric translations. \( F_X \) and \( F_Y \) represent the global forces in the \( X \) (horizontal) and \( Y \) (vertical) directions while \( F_x \) and \( F_y \) represent the local forces in the \( x \) (axial) and \( y \) (transverse) directions. These forces can be calculated using Equations 6.7–6.10.

\[
F_x = F_{\text{propulsion}} - F_{\text{drag}} \quad (6.7)
\]

\[
F_y = F_{\text{lift}} \quad (6.8)
\]
Figure 6.7: Angle of Attack

\[ F_X = F_x \cos \theta + F_y \sin \theta \]  \hspace{1cm} (6.9)

\[ F_Y = F_{buoyancy} - F_{gravity} + F_x \sin \theta + F_y \cos \theta \]  \hspace{1cm} (6.10)

The moments about the center of gravity can also be calculated. This is important because the moments about the center of gravity control the pitching moment of the UUV. For these calculations, it is assumed that all forces are acting along the center axis of the UUV such that the propulsion and drag forces provide no moment about the center of mass. It should also be noted that the moments also depend on the angle of attack of the UUV. The following equation is used to calculate the applied moment about the center of mass:

\[ M_{CM} = F_{lift} (x_{CL} - x_{CM}) \cos \theta + F_{buoyancy} x_{CV} \cos \theta \]  \hspace{1cm} (6.11)

These equations allow for calculation of necessary pitch angles to follow the prescribed path. These pitch angles are then analyzed with respect to the control fin control force limitations to ensure that the vehicle is capable of such maneuvers. Additionally SNARC will calculate the applied drag
forces that the system must overcome in order to maintain or increase velocity. Because of this lack of a reliable structural analysis of the hull sections, a full finite element model has been created to make up for this lack of sufficient structural analysis.

6.4.2 Thrust Device

SNARC has four models for propulsion. The most basic is a simple power converter. Two additional models for propulsor are included. They include one model which uses blade element momentum theory and another which uses a lifting line propulsor model. The fourth thrust model is a rocket model. The rocket model is for use with the supercavitating torpedo designs, which is currently under development and is not yet operable.

The most basic of the propulsor models is similar to many of the other basic components that deal with energy conversion. Again, the user must specify the propulsor’s efficiency ($e_f^\text{prop}$), stowage factor ($s_f^\text{prop}$) and specific energy ($s_e^\text{prop}$). Once these are provided, the propulsor volume, $v_{ol}^\text{prop}$, and output energy, $E_{out}$ can be related using Equation 6.12.

$$e_f^\text{prop} \cdot s_f^\text{prop} \cdot s_e^\text{prop} \cdot v_{ol}^\text{prop} \cdot E_{out} = 1 \quad (6.12)$$

The more detailed model that uses blade element momentum theory has a more physics-based formulation. [62] Blade element theory is one method to calculate the lifting force of predicting the propulsor performance. The first step in using this method involved discretizing a single propulsor blade
into two-dimensional sections along the length of the blade, as shown in Figure 6.8. Given a rotational velocity of the propulsor, \( \Omega \), the lift and drag of each two-dimensional section can be calculated.

Figure 6.8: Discretization of a Propulsor Blade for Blade Element Theory

Figure 6.9 shows the two-dimensional section taken from Figure 6.8. In Figure 6.9, \( V_0 \) represents the flow velocity along the axis of rotation for the propulsor. \( V_2 \) represents the flow resulting from the angular rotation of the propulsor at the blade section. \( V_1 \) is the sum of the flow velocities \( V_0 \) and \( V_2 \), and represents the flow velocity that the wing section sees, or the fluid’s relative velocity with respect to the rotating wing section. \( \theta \) is the geometric pitch angle of the wing section while \( \alpha \) is the wing angle of attack relative to the fluid flow velocity. From these geometric relationships, the values of the blade section lift, drag, thrust and torque can be calculated.

The angle between the lift and thrust vectors can be calculated as
Using the angle $\phi$, calculated in Equation 6.13, the thrust of the blade section, $T_e$, can be calculated using the following equation:

$$T_e = L_e \cos(\phi) - D_e \sin(\phi) \quad (6.14)$$

where $L_e$ and $D_e$ represent blade section lift and drag, respectively. Similarly, the torque, $R_e$ can be calculated:

$$R_e = r_e \left( D_e \cos(\phi) + L_e \sin(\phi) \right) \quad (6.15)$$

The blade section lift and drag are found using the following two equations, where $C_L$ is the coefficient of lift and $C_D$ is the coefficient of drag for the angle of attack $\alpha$, $c$ is the chord of the element, and $r_e$ is the span of the element.

$$L_e = \frac{1}{2} C_L \rho V_1^2 c r_e \quad (6.16)$$
Substituting Equations 6.16 and 6.17 into Equations 6.14 and 6.15, the thrust and torque can be calculated for each section of the blade.

The equation for thrust becomes:

\[ T_e = \frac{1}{2} (C_L \cos(\phi) - C_D \sin(\phi)) \rho V_1^2 cr_e \]  

(6.18)

Similarly, the equation for torque becomes:

\[ R_e = \frac{1}{2} r (C_D \cos(\phi) + C_L \sin(\phi)) \rho V_1^2 cr_e \]  

(6.19)

The thrust from the entire blade can now be calculated by taking the integral across the entire length of the blade. This is shown in Equation 6.20.

\[ T = \frac{1}{2} \int_R^0 \left[ (C_L(r) \cos(\phi(r)) - C_D(r) \sin(\phi(r))) \rho V_1^2 c(r) \right] dr \]  

(6.20)

The torque from the propulsor can be calculated as shown in Equation 6.21:

\[ R = \frac{1}{2} \int_R^0 (C_D(r) \cos(\phi(r)) + C_L(r) \sin(\phi(r))) \rho V_1^2 c(r) rdr \]  

(6.21)

Simple blade element theory assumes that the velocities of \( V_0 \) and \( V_2 \)
are known. These values are assumed to be the velocity that the vehicle is traveling through the fluid for $V_0$ and the velocity that the propulsor section due to rotation for $V_2$. However, momentum theory takes into account the acceleration of the axial fluid velocity, $V_0$, and the rotational fluid velocity, $V_2$, due to upstream effects of the propulsor.

These upstream effects are often quantified by using the following two equations:

\[ V_0 = (1 + a)V_{inf} \]  
\[ V_2 = (1 - b)\Omega r \]

where $a$ is the axial inflow factor, $b$ is the angular inflow factor and $V_{inf}$ is the free stream velocity.

In order to solve for the axial inflow factor and the angular inflow factor, conservation of momentum must be considered. First, the momentum in the axial direction, $T$, is quantified.

\[ \Delta T = (\text{axial mass flow rate})(\Delta \text{velocity}) \]  
\[ \Delta T = \rho 2\pi r dr V_0 (V_{out} - V_{in}) \]

where $V_{in}$ is the flow velocity into and $V_{out}$ is the flow velocity out of the control volume encompassing the propeller. Additionally we can assume
that the velocity of the flow axial to the propeller is the average of the velocities in and out of the control volume. More clearly:

\[ V_0 = \frac{V_{in} + V_{out}}{2} \]  \hspace{1cm} (6.26)

The conservation of momentum of the flow in the radial direction about the propeller, \( R \) must also be considered. Similarly, the momentum can be quantified as follows:

\[ \Delta R = \text{(radial mass flow rate)(\Delta velocity)} \]  \hspace{1cm} (6.27)

\[ \Delta R = \rho 2\pi r dr V_0 (V_{out} - V_{in}) \]  \hspace{1cm} (6.28)

In this case, the radial flow velocity at the entrance of the flow volume is considered to be zero, simplifying the equation to:

\[ \Delta R = \rho 2\pi r dr V_0 V_{out} \]  \hspace{1cm} (6.29)

Again using a similar relationship as equation 6.26, \( V_{out} \) can be calculated as shown:

\[ V_2 = \frac{V_{in} + V_{out}}{2} \]  \hspace{1cm} (6.30)

Recall that the radial flow velocity at the entrance of the flow volume is considered to be zero.
\[ V_2 = \frac{0 + V_{out}}{2} \]  \hspace{1cm} (6.31)

Substituting 6.23, the rotational flow velocity can be calculated:

\[ V_{out} = 2b\Omega r \]  \hspace{1cm} (6.32)

This system of equations can then be solved to find the thrust for each element, \( T_e \), the torque for each element, \( R_e \), as well as the axial inflow factor, \( a \), and the angular inflow factor, \( b \).

### 6.4.3 Fluid Dynamics

Drag and lift calculations are calculated based on linear hydrodynamic assumptions. These calculations are based on a semi-hemispherical endcap attached to a cylinder of equal diameter. These calculations use a Reynolds number dependent drag coefficient. The drag calculation performed in SNARC also takes into account standard tail appendages, such as stators and enclosures.
Structural Analysis of a UUV

7.1 UUV Modeling

Computer-based structural modeling of a UUV allows for many different designs and materials to be investigated with much less time and financial resources invested than physical trial-and-error tests. Although physical tests must ultimately be performed for validation, simulations based in sound mathematical representations of physics allow for many “what-if” studies, as well as optimization procedures that reduce prototyping time. Simulation, when performed efficiently and accurately, allows the designer to assess most of the critical design criteria before these prototypes are built, saving in production costs and prototype iterations.

For each analysis in this chapter, the same UUV configuration was used. In each case the geometric properties, as well as stiffener configuration, are shown in Table 7.1. The load cases will be further explained when the example is presented.
7.1.1 Finite Element Analysis

The primary structural analysis will be performed within the Finite Element Analysis (FEA) framework. FEA involves creating a mesh of the physical object, thus dividing the object into many different pieces. Each of these pieces is given a certain number of degrees of freedom. The mathematical representation of these degrees of freedom are then compiled into matrices, and the solution, which may be displacements, stresses, or one of many other responses, can then be calculated. This equation can be altered in order to take into account dynamic effects, as well as non-linear responses through various transformations.

The FEM model of the UUV is shown in Figure 7.1. The shell is modeled here using the commercial finite element package ABAQUS [63]. Quadrilateral elements, shown in green, make up the primary shell of the UUV’s structure. A combination of triangular and quadrilateral shell elements are used to model each end cap. Beam elements, shown with red lines, are used to model both the longitudinal and ring stiffeners.
Linear FEA

For a static analysis of a structure, Equation 7.1 shows the governing equation. Here $[K]$ is the stiffness matrix, $\{\Phi\}$ is the displacement vector, and $\{F\}$ is the loading vector resulting from externally applied loading conditions.

$$ [K] \{\Phi\} - \{F\} = \{0\} \quad (7.1) $$

Typically this equation can be solved using various numerical methods based in linear algebra. The displacement values, $\Phi$, then can be transformed into strain and stress values through matrix multiplication.

Nonlinear FEA

Nonlinear finite element analysis involves a slightly different method. Nonlinear analysis can involve a stiffness or external forces that change with displacement. In a fully nonlinear analysis, both stiffness and loading could be functions of displacement. In this case the basic equation is shown here:

$$ [K (\Phi)] \{\Phi\} - \{F (\Phi)\} = \{0\} \quad (7.2) $$
In linear FEA, the assumption is made that the displacements are small enough that the load is the same in the undeformed shape as well as the deformed shape. Additionally linear FEA assumes that once the loading is released from the structure, it will return to its undeformed shape. Nonlinear FEA takes into account the changes in the loading and stiffness.

It does this by solving an equation similar to Equation 7.2 through iterative methods. The finite element program will make an initial guess for the displacements, then calculate both sides of the equation. If these solutions do not match within a specified tolerance, the displacements are updated, and the process repeats until a convergence criteria is met. Nonlinear analysis will be implemented if large deformations of the UUV design may occur. Large deformations may occur during extreme external pressure loading or impact loading. Typically for standard stress analyses resulting from a shallow water operating environment, linear solution methods can be used.

7.2 Simulated Loadings

An Unmanned Undersea Vehicle has many different loading conditions that affect its performance as well as its operational life. These loads come from many sources, such as testing requirements, transportation concerns, as well as operational loadings such as hydrostatic and hydrodynamic loads [64]. These loadings, when taken into account during the design phase, must be applied so that they best represent real-world physics,
while also keeping computational expense of simulation to a reasonable degree. This creates quite a contradiction to the designer. In order to best represent the load conditions, large models with many degrees of freedom are desirable. However, when considering the computational cost involved, fully expanded models with little or no assumptions can be quite costly. Therefore it is the aim of this research to accurately represent true-life physics, while maintaining a reasonable computational cost.

7.3 Static Analysis

An Unmanned Undersea Vehicle is a vehicle that must perform a majority of its mission undetected. In order to keep from plain sight, it must be designed to travel completely submerged, sometimes at large depths. The mission of the UUV will determine the depth that it must dive down to. However, due to the high density of sea water, as the dive depth of the UUV increases, so do the hydrostatic forces. These forces can cause material yielding as well as buckling if the UUV is not structurally capable to handle them. As explained in a later section, the UUV should be designed so that it is neutrally buoyant. This allows the UUV the best maneuvering capabilities. It is also desirable for the structural weight to be low, as this allows for more component mass, including: guidance and control computers, additional payload, and fuel to name a few.

Also, UUVs that are built to only perform a single mission are not desirable as if the mission becomes no longer needed, then the UUV would
become obsolete. In order to avoid this type of scenario, a UUV should be designed so that it can be designed efficiently and still able to perform a wide array of missions. The committee on Autonomous Vehicles [65] cited several dive depths of UUVs currently in limited use. The operating depths ranged from 100 to 300 meters.

These are merely the operating depths. The crush depth, the depth at which the UUV structure fails due to buckling or material yielding, is typically designed to be several times that. The depth buffer is due to the fact that if a UUV were to dive to a depth greater than anticipated due to a logistics error or mechanical failure, it would likely sink to the bottom, where it could be rendered useless. The UUV may even be unrecoverable due to the hostility of the waters in which it was performing a mission. In the worst scenario, the UUV could be captured, and the information could be extracted from its hard drive thus providing an enemy with valuable information. For these reasons, UUVs must be designed to withstand a substantial depth loading.

Material yielding occurs when the forces placed on the structure cause the stresses to exceed the materials yield point. At this stress level, the deformations transition from elastic to plastic deformations. A structure undergoing elastic deformation will return to its unloaded state once the external forces are removed. When plastic deformation occurs, the structure will only partly return to its undeformed state. Buckling is a different analysis that involves finding an external load where an additional
infinitesimally small force will cause a very large displacement. This point is described as an instability point.

### 7.3.1 Hydrostatic Loading

One major source of failure modes during a UUV’s operation is due to the pressure of water when the UUV is at full depth. Because of the density of water, immense pressures can be exerted on the surface of the UUV hull at desired operating depths. Often times the water depth at which the UUV has a mission-ending structural failure is called its crush depth.

Hydrostatic loading exerts a normal force to the wetted portion of the UUV. During linear analysis this force remains the same throughout the analysis, however when nonlinear responses are taken into account, the force applied must be derived stepwise. The load application is applied incrementally, allowing the structure to deform. Once the resulting deformations are calculated, an incrementally increased load is applied and the resulting deformation is calculated once again. In this case, the hydrostatic pressure is applied normally to the deformed structure at the particular load step in question.

Here, the UUV shell was subjected to depth pressure at 200 meters. The maximum von Mises stress was found to be 29.8 MPa. The stress distribution across the structure is shown in Figure 7.2. The stress distribution appears to be nearly uniform across the center section of the UUV. However, the stress values are slightly less at each end due to stiffening
from the end caps and near each stiffener due to the added rigidity they provide.

Figure 7.2: UUV Shell Stresses When Subjected to Statically Applied Depth Loading

7.3.2 Buckling Analysis

In general, buckling occurs due to a structural instability in the system. When a load is applied to a structure, certain displacements result. The relationship between applied force and resulting displacements is commonly referred to as the stiffness. Buckling instabilities cause the structure to experience a large drop in stiffness. A buckling mode occurs when displacements deform the structure in such a way that a small perturbation force causes large displacement values.

Bifurcation Buckling

Bifurcation buckling occurs when a system can occupy two structural configurations under the same loading conditions. This type of behavior gives the structure essentially zero stiffness in the direction of bifurcation. One way to derive the location of the bifurcation point, the configuration where this type of buckling occurs, is by the solution of the following eigenvalue problem:
\[
\left( [K] + \lambda [K_{\sigma}]_{\text{ref}} \right) \{\delta D\} = \{0\} 
\]  
(7.3)

In the above equation \([K]\) is the standard stiffness matrix, while \([K_{\sigma}]_{\text{ref}}\) is the geometric stress stiffness matrix. \(\lambda\) is the vector of eigenvalues that represents the load factors. These load factors, when multiplied by the applied load, give the load that would result in buckling. \(\{\delta D\}\) is the associated eigenvector that gives the buckling mode shapes for the corresponding buckling deformations.

The stress stiffness matrix, \([K_{\sigma}]_{\text{ref}}\), calculates the effect that the loading-induced stresses has on the stability of the structure. The equation for calculating the stress stiffness is shown here:

\[
[K_{\sigma}]_{\text{ref}} = \int G^T S G |J| 
\]
(7.4)

where the integral is taken over the entire element, \(G\) is the derivative of the elemental shape functions, \(S\) is the stress matrix for the prescribed load, and \(J\) is the Jacobian matrix.

The UUV shell described by Table 7.1 was subjected to a linear buckling analysis described above. For this analysis a reference load of 200 meters of depth pressure was applied. The eigenvalue for the first mode shape was 0.75. This means the calculated crush depth for the first buckling mode shape, shown in Figure 7.3, is 150 meters (200*0.75). Due to the symmetry of the configuration, the second mode shape is merely the first mode shape rotated axially 90 degrees. The same can be said of the third
Snap-Through Buckling

Snap-through buckling calculation mandates the use of nonlinear techniques. This type of buckling occurs when a deformation state exists such that greater deformation occurs, even as the load magnitude is reduced. The snap through buckling case must be solved by using force verses displacement plots at seen in Figure 7.5. These force verses displacement plots are formed by incrementally applying loads as described in Section 7.3.1. On this plot, the initial loads cause this hypothetical case to initially deform in a linear fashion.

Once a certain deformation configuration occurs, the force required for additional displacement is actually less than was required previously. This point referred to as the critical state, labeled $C_0$ in Figure 7.5. It is at this point that snap-through buckling occurs. Here the buckling factor is denoted as $\lambda$, and is the force at which the curve reaches its first peak. It
should be noted however that once snap-through buckling has occurred, a new configuration is formed that can also be stable. $C_0$ represents the point where the structure becomes stable again. This type of response can be seen in Figure 7.5 as the right portion of the curve that continues upward.

![Figure 7.5: Snap-through Buckling: Load vs. Displacement Plot](image)

The UUV is subjected to a snap-through buckling analysis here. For this analysis the properties of the UUV shell are again given in Table 7.1. In this analysis, the UUV was subjected to a depth pressure load. This load was applied externally and normal to the surface, causing the UUV to compress. At each increment, the deformations and load factor were calculated using the Riks method. This method creates a displacement...
versus load curve which is used to determine the critical buckling load.

The results of the Riks analysis are shown in Figure 7.6. Figure 7.7 shows the UUV’s undeformed configuration. Also, the point used to create the Riks Analysis plot is marked with a red dot. Figures 7.8 and 7.9 show the deformation plots of points A and B labeled on Figure 7.6. From this study, the crush depth of the UUV appears to be just under 350 meters, which is the highest point on the force versus displacement plot.

Figure 7.6: Riks Analysis of the UUV Shell

Figure 7.7: Undeformed UUV Shell
General Instability Buckling

General instability buckling occurs in UUVs when the ring stiffeners provide less stiffness to depth loading than the skin. The result of general instability buckling is a large buckling mode spanning the length of the UUV. An illustration of a UUV undergoing general instability buckling is shown in Figure 7.10. The buckling mode spans the length of the UUV indenting both the skin and the stiffeners.

Lobar Buckling

Lobar buckling is another form of buckling that a UUV may encounter at depth. When the ring stiffeners provide more stiffness against buckling than the skin provides, lobar buckling is likely to occur. Lobar buckling is a form of buckling where the buckling eigenvector seems to be more
localized in one section, rather than across the length of the UUV as in general instability buckling. Figure 7.11 shows a UUV where lobar buckling has occurred.

![Figure 7.11: UUV Lobar Buckling](image)

### 7.4 Dynamic Analysis

Modeling the test environment involves applying time sensitive loads and analyzing the accelerations, stresses, and displacements throughout the model. Stresses should be less than their yield values such that plastic deformation does not occur. In addition, steady-state loadings, such as engine vibration, should not cause loads to exceed the fatigue limit of the material in use.

Another important characteristic of a dynamic response is the acceleration data. The stress information will evaluate the structural integrity of the shell, however acceleration data will provide insight into the survival of the sensors, battery, as well as other internal equipment. If the accelerations are too great at certain location of the UUV, then the forces generated may significantly damage the internal components.
7.4.1 Vibration Analysis

The vibrational modes of the UUV are extremely important to its operation and performance. A free vibration analysis will determine the natural vibrational modes of the UUV. These modes and frequencies are important because if the UUV is excited at one of these natural frequencies, a tremendous amount of energy can be put into the system. This energy will then cause the UUV to vibrate violently, causing possible damage to an internal component.

Because of these reasons it is important that the natural frequencies are not near excitation frequencies. In addition, various components such as the control systems, sensing systems, and any other on board system that is sensitive to vibration should be taken into account when analyzing the free vibrational modes of the UUV.

In addition to free vibration analysis, a forced vibration analysis can also be a valuable tool. Specific to a UUV, the forced vibration analysis has the capability of tracking how cyclic internal and external forces affect the structures and subcomponents of the UUV. A forced vibration analysis requires a time-dependent loading that is used to excite the structure. The resulting displacements of the structure are also a function of time. This response can contain two parts.

One portion of the forced vibration response is the transient response. The transient response is the structure’s time-dependent displacement response that is a result of the initial impact of the force. This portion
typically dies out over time, and it disregarded during most long-term vibration analyses. The other portion of the total forced vibration response is the steady-state vibration response. The steady-state vibration response is the response that is a result of the cyclic loading applied. These type of vibration responses are important when analyzing the long-term effects of such a load condition.

For the frequency analysis, nonstructural mass was added to the shell. The masses were added to simulate the mass distribution of a undersea vehicle hull loaded with its system’s components. This mass was distributed throughout the length of the UUV such that the neutral buoyancy condition was attained. Neutral buoyancy occurs when the mass of the volume of the water displaced equals the mass of the UUV. This allows for near optimal operating conditions of an undersea vehicle. The first fundamental frequency of the UUV is 133 Hz. The second vibrational frequency occurs at 160 Hz. Similarly to the linear buckling analysis, the vibrational mode shapes are repeated in sets of two. The vibrational mode shapes are shown in Figures 7.12-7.13.

Figure 7.12: UUV Shell First and Second Vibrational Modes
7.5 Summary

For the remainder of this work, the buckling load will be the primary mode of determining safety of the undersea vehicle hull. The vibration frequencies are not further incorporated as the acoustic portion of this work was left for future investigations. The cost minimization of an undersea vehicle work described in Chapter 8 uses a linear buckling analysis to determine the structural hull’s integrity. A linear buckling analysis is also included into the system-level analysis performed in Chapter 9. The buckling analysis described here is used as a high-fidelity analysis to supplement the low-fidelity analysis used within the component selection module. In each case, a Rik’s analysis could have been implemented to determine structural stability. This would have allowed for the identification of non-linear buckling behavior, but due to each hull design having roughly the same thickness throughout, and the lack of apparent localized buckling, or buckling between stiffeners, this type of analysis was not needed.
Cost Minimization of a UUV

Incorporating component cost into optimization is a concept that has been considered in the literature. Giesing and Wakayama [66] mention the benefits of incorporating costs in their analysis of a transport aircraft. They argue that structural weight is often used as the objective function, which can be inefficient at achieving a cost savings. They go on to say that many times a reduction in overall weight can lead to increased machine time, which in turn drives up cost. Johnson [67] performs a cost minimization on a commercial aircraft by considering life cycle cost. Iqbal and Hansen [68] cite a need to shift from a product performance design philosophy to one driven by cost-effectiveness. These examples and many others exist; however, little work has been done that incorporates reliability and design variation with these costs concerns. The methodology developed in this research aims to incorporate uncertainties in costs due to manufacturing variabilities.

In many instances, the statistical distribution is controllable within
certain limits. A manufacturer is often able to alter the production process to produce different material property distributions. Sometimes tighter tolerances are needed for shape manufacturing where several parts are assembled in a machine system, such as a piece of electronic equipment. In most cases, reducing the statistical spread of the material property distributions can be costly. This is due to the increased cost of quality controls and increased production time involved in producing products within a smaller tolerance range. In turn, widening the tolerance of these inputs allows the manufacturer to produce the material or product at a lower cost. This means that for a manufacturer, if the statistical tolerances of the input variables are tightened, the costs of production will increase due to the rise in material or production costs. Likewise, if the statistical tolerances of the input variables are widened, the material costs and the production costs could decrease significantly.

The relationship between product and material costs and design tolerance provides manufacturers an opportunity to reduce production costs. A standard objective of the optimization process is to reduce the weight of a structure, thereby reducing the cost of material. This being accepted as true, if the designer is allowed to relax the engineering tolerances on the material properties, the cost of the product can similarly be reduced. In addition to reducing material costs by relaxing tolerances, the cost of production can also be reduced. However, relaxing these tolerances could cause the part to be rejected, so it must be done with care.
Typically after a traditional RBDO process, entirely new design geometries are developed. In order for the component’s manufacturer to produce the new design, entire production lines may need to be altered, and machines may require retooling. Once a statistical distribution optimization has been performed, very little retooling or redesign needs to be done due to the nature of the adjustments. The part being produced will have the same mean dimensions and be constructed with similar materials. This gives the new design the ability to run on the existing production lines.

This research performs a reliability-based design optimization while using the statistical spread of structural properties as design variables rather than the means as the design variable. In order to accomplish this and prohibit the statistical spread from approaching zero, the costs of adjusting the statistical spread are taken into account during this procedure. The result of this method is a cost-efficient design that meets reliability performance demands.

8.0.1 Cost Modeling

Intuitive Cost Model

In order to perform a cost minimization, a pricing function must be formulated to map certain design criteria to cost. One way to perform this mapping is to relate the standard deviation of the material properties to some cost function. Since very few general cost models for uncertainty exist in the literature, two cost models are proposed for uncorrelated input
variables here.

One such cost function is expressed in Equation 8.1. A plot of this cost function is shown in Figure 8.1. The cost function used here was chosen because of its properties as $\sigma_i$ approaches 0 and as $\sigma_i$ approaches $\infty$. It can be reasonably assumed that as the required design standard deviation approaches zero, the cost required to produce such a design would go to infinity. As the standard deviation expands toward infinity, the process becomes uncontrolled and the cost to produce this design shrinks to some baseline value. The cost of production of such material then approaches some minimum value, or in the case of Equation 8.1, zero:

$$\text{Cost of Tolerance} = \sum_{i=1}^{n} \frac{a_i e^{-\sigma_i}}{\sigma_i}$$  \hspace{1cm} (8.1)

where

- $a_i = i^{th}$ variable cost contribution coefficient
- $\sigma_i = \text{standard deviation of the } i^{th} \text{ variable}$
- $n = \text{number of variables considered in the cost function}$

This behavior is illustrated in Figure 8.1.

**Spotts’ Cost Function**

Another cost function that can be used for mapping geometric or material standard deviation to financial cost was created by Spotts [69] and more recently applied by Jeang [70]. This equation describes the cost as a
function of two constants, \(a\) and \(b\), as well as the tolerance, \(t\):

\[ C_M(t) = \sum_{i=1}^{n} a_i + \frac{b_i}{t_i^2} \]  

(8.2)

Here, the standard deviation is substituted as a means of defining tolerance. An alternative to this method would be to analyze the design at the tolerance extremes. This optimization could then be performed using interval analysis methods. Rao and Berke investigated this further [71]. However, this type of analysis is not the focus of this research.

Under Spotts’ cost model, the constants \(a_i\) and \(b_i\) are dependent upon current material supply and demand, market fluctuations, as well as many other factors, and \(N\) is the number of components included in the analysis. However, a discussion of how to attain these numbers is not within the scope of this paper. Looking more closely at this equation, if \(a_i\) and \(b_i\) are considered to be the same for each increment of \(i\), then they can
be pulled out of the sum. This is a reasonable assumption if each design variable shares similar characteristics. For instance, if each design variable represents the standard deviation of the length of steel shafts with identical diameters, then this assumption would appear to be valid:

\[
C_M(t) = na + b \sum_{i=1}^{n} \frac{1}{t_i^2}
\]  

(8.3)

It has been shown that \(a_i\) and \(b_i\) are simply scaling factors and may be ignored during the minimization of Equation 8.2. Simplifying the process, these constants are set to 0 and 1 respectively and the new cost function is used, shown in Equation 8.4:

\[
C_M(t) = \sum_{i=1}^{n} \frac{1}{t_i^2}
\]

(8.4)

Ideally these cost models could take into account variabilities of different forms of manufacturing processes, however there is a very limited number of models available in the literature. The intent of this work is not to accurately model cost as a dollar amount, but rather model the effort or resources needed, and if anything, provide a motivation for the development of further cost of variability models.

### 8.1 Problem Formulation

The primary structural forces acting on any undersea vehicle during use are hydrostatic and hydrodynamic forces. The hydrostatic force is a uni-
form load that acts normal to the wetted surfaces of the vehicle and is primarily related to depth. Hydrodynamic forces result from movement of the vehicle within the medium. As previously stated, UUVs typically travel at low speeds that are ideal for data collection. The low speeds create small hydrodynamic forces when compared to hydrostatic forces at greater depths. In this case, the UUV shell is designed to perform at a depth of 2000 feet, more than enough to dominate the dynamic forces.

For this investigation a buckling load factor of less than 1.0 is considered a failure. Buckling analysis was performed by solving for the linear bifurcation eigenvalue problem. The smallest resulting eigenvalue is termed the buckling load factor. This is the ratio of the actual buckling load to the applied load. Again, the applied load is a hydrostatic pressure load equivalent to 2000 feet below the water’s surface. The UUV was modeled using the NASTRAN finite element package (Figure 8.2). The outer shell of the UUV was modeled using 1992 plane-strain quadrilateral plate elements and 48 plane-strain triangular plate elements. There were no longitudinal or ring stiffeners included in this model. The hydrostatic pressure was simulated by applying a static uniform pressure load normal to the UUV’s outer surface.

![UUV Finite Element Model](image)

Figure 8.2: UUV Finite Element Model

The mean skin thickness was set to 0.472 in throughout, and the mate-
rial properties of Aluminum 1050 were used and are displayed in Table 8.1 for this design. The required reliability relative to buckling failure is once again 99.8%. This is equivalent to a reliability index $\beta$ of 3.0. The design variables in this case were the coefficients of variations of this thickness mean. The shell was divided into three tube sections of equal length along the body of the UUV, as shown in Figure 8.3. Each of these ring sections was assigned a coefficient of variation, bounded between 0.001 and 0.25. The coefficient of variation is the ratio of the standard deviation to the mean value. In this example, the mean value is a constant. Each design variable was assumed to have a normal distribution independent of each of the other design variables.

![Figure 8.3: UUV Design Sections](image)

Table 8.1: Material Properties of the UUV Shell

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Young’s Modulus</td>
<td>$10 \times 10^6$ psi</td>
</tr>
<tr>
<td>Poisson’s Ratio</td>
<td>0.33</td>
</tr>
<tr>
<td>Density</td>
<td>0.098 $lb/in^3$</td>
</tr>
</tbody>
</table>

For this analysis the First Order Reliability Method is used. The cost function used in this example is the Spotts’ cost model described by Equation 8.2. As shown previously, both $a_i$ and $b_i$ can be factored out as shown in Equations 8.3 and 8.4.
8.2 Discussion

The optimization process yielded the results shown in Table 8.2. The initial design point was chosen to have a uniform coefficient of variation with a $\beta$ equal to 3.0 to allow for a fair comparison of designs. Here it is seen that the front and middle sections are allowed to expand to a large coefficient of variation value, while the rear section coefficient of variation is reduced slightly. This results in an 11.5% cost reduction. These results are valuable information for the designer. Relating these optimal values to one another, the designer can identify the shell sections that require more manufacturing precision, and to what degree the precision should be relaxed or increased.

<table>
<thead>
<tr>
<th>Name</th>
<th>Initial Value</th>
<th>Final Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$COV_1$ (in percent)</td>
<td>5.50</td>
<td>5.311</td>
</tr>
<tr>
<td>$COV_2$ (in percent)</td>
<td>5.50</td>
<td>6.124</td>
</tr>
<tr>
<td>$COV_3$ (in percent)</td>
<td>5.50</td>
<td>6.249</td>
</tr>
<tr>
<td>$\beta$</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>Normalized Cost</td>
<td>1.000</td>
<td>0.885</td>
</tr>
</tbody>
</table>
System Design of an Undersea Vehicle

9.1 System Design Methodology

This work focuses on the design of an unmanned undersea vehicle while considering uncertainty. The system design inputs are the desired missions that will be performed, as well as the components available for inclusion into the system. Once these are supplied, a preliminary mission path is defined. This information is then passed to the component selection module. If a sufficient component configuration can be identified, a structural analysis is performed. The structural analysis incorporates both the component and path data to perform a high fidelity analysis of the outer hull of the vehicle. The flow of information is shown in Figure 9.1.

In this work, uncertainty is included in two stages of the analysis as well as in two differing forms of uncertainty. The uncertainty associated with the missions involves the uncertain location and size of the hazards.
Since this information is likely limited to expert opinion or a small amount of information, Evidence Theory is used to quantify this risk as explained in Section 5.4. Additionally, the structural hull is considered to have uncertain thickness. The structural uncertainty here is assumed to be due to the manufacturing of the three hull sections, and is assumed to be normally distributed about a mean value. A reliability index method, as described in Section 4.2.1, is used to quantify this uncertainty.
Once an optimal path, component configuration and structural hull design are found, the system design framework outputs the mission profile and mission risk (in the form of Evidence Theory variables) for each mission as well as the included components and their sizes. The structural reliability of the vehicle during each of the missions is output as well.

9.2 System Design Framework

In order to perform a coupled system-level design, a framework was developed. This framework consists of three major components, or modules. The three modules are the path planning, component selection, and structural analysis modules. The path planning module uses the particle swarm optimization algorithm defined earlier, PPAF, to identify the best path and velocity profile that the vehicle should take for each mission. The component selection module uses the SNARC program in order to select which components should be used, and properly size them based on the demands from each of the missions. The structural analysis module uses the forces from the path and velocity profiles, and couples these with the hull information generated in SNARC in order to perform an reliability analysis of the structure.

The system framework was developed using MATLAB scripting. All information that is pass from one module to another is done using this software. Additionally, the modular framework developed allows for additional modules to be added or removed from the framework. Furthermore,
the existing modules can be altered to include either higher fidelity models that would increase the overall prediction of the system and the system’s performance, or lower fidelity models that would allow for faster computational times, and the ability to perform many what-if studies that may prove useful.

### 9.2.1 Path Planning Module

The path planning module initiates the system design process. The initial limits on the vehicle are set as a maximum velocity of 50 m/s. This is set as an extreme upper limit, as it is unlikely that the vehicle will be able to travel that fast when the vehicle’s components are selected. Furthermore, each mission profile is minimized with respect to the time it will take to complete the mission. The three mission profiles are defined in Tables 9.1-9.3.

Once an mission profile is found for each mission specification, the optimal mission profiles are passed along to SNARC for component selection and sizing. The mission profile includes the vehicle’s path, operating
Table 9.2: Mission 2 Definition with Basic Belief Assignment Data (all units in meters)

<table>
<thead>
<tr>
<th>Hazard</th>
<th>X-Cord</th>
<th>Y-Cord</th>
<th>Radius</th>
<th>X-Cord Range</th>
<th>Y-Cord Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hazard 1</td>
<td>$m = 0.8, [950 - 1050]$</td>
<td>1000</td>
<td>200</td>
<td>$[0, 1000]$</td>
<td>$[1000, 1000]$</td>
</tr>
<tr>
<td>Hazard 2</td>
<td>1000</td>
<td>800</td>
<td>$m = 0.5, [50 - 300]$</td>
<td>$m = 0.5, [200 - 500]$</td>
<td></td>
</tr>
<tr>
<td>Hazard 3</td>
<td>1000</td>
<td>500</td>
<td>$m = 0.9, [300 - 500]$</td>
<td>$m = 0.1, [200 - 600]$</td>
<td></td>
</tr>
<tr>
<td>Hazard 4</td>
<td>3000</td>
<td>2000</td>
<td>$m = 0.6, [500 - 500]$</td>
<td>$m = 0.4, [400 - 500]$</td>
<td></td>
</tr>
<tr>
<td>Hazard 5</td>
<td>$m = 0.99, [4000 - 4000]$</td>
<td>0</td>
<td>500</td>
<td>$m = 0.1, [4000 - 4500]$</td>
<td></td>
</tr>
<tr>
<td>Hazard 6</td>
<td>6000</td>
<td>1000</td>
<td>$m = 0.9, [800 - 1000]$</td>
<td>$m = 0.1, [1000 - 2000]$</td>
<td></td>
</tr>
<tr>
<td>Hazard 7</td>
<td>$m = 0.5, [7000 - 8000]$</td>
<td>0</td>
<td>600</td>
<td>$m = 0.5, [8500 - 9000]$</td>
<td></td>
</tr>
<tr>
<td>Hazard 8</td>
<td>8000</td>
<td>1000</td>
<td>$m = 0.75, [1000 - 1500]$</td>
<td>$m = 0.25, [2000 - 3000]$</td>
<td></td>
</tr>
<tr>
<td>Hazard 9</td>
<td>9000</td>
<td>1000</td>
<td>$m = 0.7, [750 - 800]$</td>
<td>$m = 0.3, [750 - 900]$</td>
<td></td>
</tr>
<tr>
<td>Checkpoint 1</td>
<td>2000</td>
<td>1000</td>
<td>500</td>
<td>500</td>
<td></td>
</tr>
<tr>
<td>Checkpoint 2</td>
<td>8000</td>
<td>1000</td>
<td>500</td>
<td>500</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.3: Mission 3 Definition with Basic Belief Assignment Data (all units in meters)

<table>
<thead>
<tr>
<th>Hazard</th>
<th>X-Cord</th>
<th>Y-Cord</th>
<th>Radius</th>
<th>X-Cord Range</th>
<th>Y-Cord Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hazard 1</td>
<td>3000</td>
<td>1000</td>
<td>300</td>
<td>$[0, 1000]$</td>
<td>$[1000, 1000]$</td>
</tr>
<tr>
<td>Hazard 2</td>
<td>2750</td>
<td>500</td>
<td>300</td>
<td>$[5000, 1000]$</td>
<td>100</td>
</tr>
<tr>
<td>Hazard 3</td>
<td>2750</td>
<td>1500</td>
<td>300</td>
<td>$[5000, 1000]$</td>
<td>100</td>
</tr>
<tr>
<td>Hazard 4</td>
<td>2400</td>
<td>500</td>
<td>150</td>
<td>$[5000, 1000]$</td>
<td>100</td>
</tr>
<tr>
<td>Hazard 5</td>
<td>2400</td>
<td>1500</td>
<td>150</td>
<td>$[5000, 1000]$</td>
<td>100</td>
</tr>
<tr>
<td>Hazard 6</td>
<td>3000</td>
<td>750</td>
<td>300</td>
<td>$[5000, 1000]$</td>
<td>100</td>
</tr>
<tr>
<td>Hazard 7</td>
<td>3000</td>
<td>1250</td>
<td>300</td>
<td>$[5000, 1000]$</td>
<td>100</td>
</tr>
<tr>
<td>Checkpoint 1</td>
<td>2500</td>
<td>1000</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>
Component Selection Module

The component selection and sizing module uses SNARC to identify a component configuration. This program takes into account the information passed along from the path planning module. Once this information is received by SNARC, it chooses components based on the available component list. It then finds an optimal system for the missions specified. If SNARC is unable to find a feasible component configuration for one or more of the mission profiles, then the path planning optimization criteria is altered and new mission profiles must be calculated by the path planning module. For instance, if SNARC is unable to incorporate a sufficient fuel supply for an abnormally long mission, the path planning optimizer is adjusted so that it will find a new mission profile for the offending mission with a shorter path length. This will likely come at the expense of traveling through hazardous areas. The components available for selection in the design process are shown in Table 9.4.

Table 9.4: Available Components for Use in the Vehicle

<table>
<thead>
<tr>
<th>Component</th>
<th>Model</th>
<th>Stowage Factor</th>
<th>Specific Energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Payload</td>
<td>Simple Payload</td>
<td>$1209 \text{ kg/m}^3$</td>
<td>$-200 \text{ W/unit}$</td>
</tr>
<tr>
<td>Battery</td>
<td>Simple Battery</td>
<td>$2527 \text{ kg/m}^3$</td>
<td>$0.391 \text{ MJ/kg}$</td>
</tr>
<tr>
<td>Fuel</td>
<td>Otto Fuel Tank</td>
<td>$1170 \text{ kg/m}^3$</td>
<td>$2.19 \text{ MJ/kg}$</td>
</tr>
<tr>
<td>Engine</td>
<td>Thermal Engine</td>
<td>$1295 \text{ kg/m}^3$</td>
<td>$-2.197 \text{ kW/kg}$</td>
</tr>
<tr>
<td>Propulsor</td>
<td>Simple Propulsor</td>
<td>$2000 \text{ kg/m}^3$</td>
<td>$-17 \text{ kW/kg}$</td>
</tr>
<tr>
<td>Sonar</td>
<td>Simple Sonar</td>
<td>$2000 \text{ kg/m}^3$</td>
<td>$-0 \text{ W/kg}$</td>
</tr>
</tbody>
</table>

SNARC provides the component configuration that is able to complete all mission profiles and is capable of ranking designs in a number of ways.
including: the number of systems, the size of the vehicle, the mass of the
vehicle, the noise created, and functions of any of the previous value. In
this study the lightest feasible component configuration was used. Once a
component configuration has been reached for all specified missions, then
the vehicle’s size and shape are passed along to the structural analysis
module. Accompanying this information is the hydrodynamic and hydro-
static forces that are a resultant of the mission profile and vehicle shape.

**Structural Analysis Module**

The structural analysis, performed in ABAQUS, analyzes how the struc-
ture interacts with the hydrodynamically and hydrostatically induced loads.
SNARC provides the vehicle’s length and diameter as well as the hydrody-
namic drag forces on the vehicle. The additional structural design variable
are listed in Table 9.5.

A reliability analysis with respect to the buckling load factor is per-
formed. If the buckling load factor is less than 2, then the load case is
considered a failure. The probability of success is held to be above 99%
for each mission profile. If the vehicle is deficiently designed for a specified mission profile and vehicle size, the mission is reanalyzed with a lower limit on the maximum velocity to reduce the hydrodynamic loads.

9.3 System Design Results

The total system design took roughly 6 hours on a PC with a Pentium 4 processor and 2 GB of RAM. The optimal mission profiles are shown in Figure 9.2. Each solution properly balances the amount of risk while minimizing the mission time. Notice that missions one and two have a large amount of uncertainty, denoted by the translucent areas, while mission three has absolute hazard certainty, noted by the opaque areas, and thereby total avoidance of the area. Table 9.6 shows the performance parameters for each of the missions.

Calculating the hazard risk of a mission involves dealing with the likelihood that the mission profile may pass through one or more different hazards. Because of this, the individual hazard risk must be combined to form an overall mission risk. This calculation is performed using De Morgan’s Rule. De Morgans Rule is shown here in Equation 9.1:

\[ A_1 \cup A_2 \cup A_3 = \overline{A_1} \cap \overline{A_2} \cap \overline{A_3} \]  

(9.1)

Now, if \( A_i \) represents the probability of the mission passing through hazard \( i \), then \( \overline{A_1} \cup \overline{A_2} \cup \overline{A_3} \) represents the probability of the mission avoiding each hazard. This probability is equal to the the product of the probabil-
ities of the mission avoiding each hazard, $A_i$.

Figure 9.2: Optimal Mission Profiles for Missions 1-3

Table 9.6: Optimal Mission Performance Parameters

<table>
<thead>
<tr>
<th></th>
<th>Time (s)</th>
<th>Length (m)</th>
<th>BEL(Hazard)</th>
<th>PL(Hazard)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission 1</td>
<td>1209</td>
<td>31352</td>
<td>0.10</td>
<td>0.40</td>
</tr>
<tr>
<td>Mission 2</td>
<td>278</td>
<td>10022</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Mission 3</td>
<td>507</td>
<td>10355</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Table 9.7 shows the component sizes for the optimal system configuration. These values were calculated from the SNARC design tool.

Table 9.8 shows the probability of buckling failure of the structure during each of the mission profiles.

In order to get a more clear understanding of the system design process,
Table 9.7: Optimal Mission Component Sizes

<table>
<thead>
<tr>
<th>Component</th>
<th>Model</th>
<th>Mass (kg)</th>
<th>Volume ($m^3$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sonar</td>
<td>Simple Sonar</td>
<td>163.0</td>
<td>0.1248</td>
</tr>
<tr>
<td>Hull</td>
<td>Hull</td>
<td>959.6</td>
<td>0.7692</td>
</tr>
<tr>
<td>Propulsor</td>
<td>Simple Propulsor</td>
<td>22.4</td>
<td>0.0112</td>
</tr>
<tr>
<td>Engine</td>
<td>Thermal Engine</td>
<td>236.6</td>
<td>0.2087</td>
</tr>
<tr>
<td>Fuel Tank</td>
<td>Otto Fuel Tank</td>
<td>487.2</td>
<td>0.3610</td>
</tr>
</tbody>
</table>

Table 9.8: Buckling Failure Probability of the Optimal System

<table>
<thead>
<tr>
<th>Mission</th>
<th>$\beta$</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mission 1</td>
<td>2.9299</td>
<td>0.1695%</td>
</tr>
<tr>
<td>Mission 2</td>
<td>2.9287</td>
<td>0.1702%</td>
</tr>
<tr>
<td>Mission 3</td>
<td>2.9299</td>
<td>0.1695%</td>
</tr>
</tbody>
</table>

A discussion of the iteration history is included here. Figure 9.3 shows a graphical representation of the iteration history.

The first iteration shows that three new mission profiles were calculated. The mission times for the three missions were initially 645 seconds, 200 seconds, and 593 seconds, respectively. This mission profiles were then passed along to SNARC to identify a component configuration for the three mission profiles. As denoted in Figure 9.3, SNARC was unable to attain a configuration that was able to complete the first and second
mission profiles. This required the system design process to redefine mission profiles for missions one and two. The mission profile for mission three can remained unchanged, as SNARC was able to design a component configuration that completes this mission profile. During the second iteration, the mission times for the three missions were 790 seconds, 250 seconds, and 593 seconds, respectively.

The demands on the components were reduced by reducing the top speed of the vehicle by 20% for each mission for which a component configuration was unavailable. Additionally, the scale factor on the hazard penalty was reduced by a factor of 10. This allows for more risk to be taken during the mission, but also for more mission design freedom needed to attain a feasible component configuration for the mission profile.

These two mission profiles that do not have a feasible component configuration are then updated in the second iteration. Again, however, SNARC is unable to design a component configuration capable of completing all three missions. Again the velocities and hazard penalties of the offending mission are reduced and a new mission profile is calculated. On the third iteration, SNARC was able to identify a component configuration capable of performing each of the three mission profiles. For the third iteration the mission times for the three missions were 977 seconds, 250 seconds, and 593 seconds, respectively, and the overall system mass was 2618 kg. Since a feasible component configuration was able to be attained, even though the maximum mass of the configuration is slightly higher than the upper
limit of 2500 kg, that completes each mission, a high fidelity structural analysis was performed.

The structural analysis estimates the structural reliability of the configuration designed by SNARC during each defined mission. In iteration three, the structural reliability was insufficient for each of the defined missions. The reliability index values of the three missions were 1.84, 1.99, and 2.00 respectively. Each of these values are below the lower boundary of 2.33. This requires each mission to be redefined so that the loading on the structure can be reduced.

In order to reduce the structural loading, the maximum velocity for each offending mission is reduced by 20%. The three mission profiles were redefined and the mission times were 1087 seconds, 278 seconds, and 507 seconds, respectively. The system mass of the component configuration found by SNARC was 2575 kg. Iteration four, however, found one structural reliability deficiency. This deficiency was found in the first mission which had a reliability index value of 2.28. Mission two had a reliability index value of 2.34, as did mission three. Again, the velocity allowable for the first mission was reduced and a new mission profile for the first mission was defined, as well as a new component configuration during the fifth iteration. The mission times for each mission profile now become 1209 seconds, 278 seconds, and 507 seconds, respectively. For this iteration, a feasible component configuration was found. The mass of the system was 2067 kg. This system configuration was found to be sufficiently reliable.
with reliability indicies of 2.930, 2.929, and 2.930 for the three missions, respectfully. Because a valid system was designed and the design’s reliability was validated, the system design process is complete.
Summary and Future Directions

This work performed a system level design of an undersea vehicle while incorporating uncertainty. The system design framework consists of modules. The path planning module finds the optimal path for the given mission space. This module takes into account uncertainty in the location of hazards using evidence theory. The next module is the component selection and sizing module that designs a system configuration using SNARC. Finally the structural analysis module performs a finite element analysis of the buckling reliability with respect to variations of the hull’s skin thickness. These three modules are pulled together through a framework that efficiently updates the intermediate variables in order to design a undersea vehicle with an optimal component configuration and a minimal amount of mission risk.

The system design framework developed in this work integrated models of varying fidelities, performed uncertainty quantification, and optimized an undersea vehicle system for optimal mission performance. This work
efficiently measured the effects of the system configuration, not simply on vague performance metrics such as range or top speed, but to mission performance parameters such as mission success metrics, structural safety estimates, and mission completion time. This framework allows for undersea vehicle designers to analyze directly how changes in available components or the performance of components affects the ability of the vehicle to perform.

10.1 Research Contributions

In this work several contributions to the system design community were developed. The path planning performed in this work incorporated Evidence Theory in order to deal with the uncertainties associated with hazard location and size. This implementation of Evidence Theory allows for expert opinions and limited sample information to be incorporated into how these types of uncertainties and affect a mission’s risk. Additionally, this uncertainty was propagated through the design process where it helped drive the system design, such that the effects could be traced through the component selection and sizing and system design process.

In order to include uncertainty into the path planning process, the current methods of finding optimal mission profiles needed to be made more efficient. Due to the nature of uncertainty methods, often, including uncertainty into an analysis require many more calculations to be made to measure its effects. This created the need for the path planning by
additive freedom method. This method allows for much more efficient identification of acceptable and effective mission profiles. In this work, up to a twenty-fold improvement was found when compared to the traditional particle swarm optimization method. Further refinement of this method would no doubt solidify its standing as being superior to the particle swarm algorithm when applied to path planning.

Furthermore, much scientific contribution was made with the development of an integrated system design framework. As explained in the prior chapters, the system framework developed here incorporated a combination of available tools and methods that prior to this work, had not been able to be combined. This work represents cutting edge system-level design through the numerous aspects considered in the framework. Furthermore, this framework incorporated two forms of uncertainty encountered through the design process. Evidence Theory was integrating into the path planning analysis to deal with epistemic uncertainty, and the Hasofer-Lind algorithm was incorporated into the structural analysis to address the aleatory uncertainty present in system design. By incorporating both forms of uncertainty, the groundwork was laid for future researchers to use this framework to address either type of uncertainty present during the design of a system.

Also, the framework presented one of the few works of coupling a low-fidelity design validated by a high-fidelity analysis. This design and validation arrangement developed here reduces the computational cost by
performing the iterative system design process using low-fidelity models. This allows for many configurations to be analyzed very quickly. Once a candidate system design is identified, then the high-fidelity analysis either validates or invalidates the design. Although the high-fidelity analysis requires more computational effort, it needs to only be performed once a candidate design is reached. It is this removal of the high-fidelity analyses from the design cycle and into the analysis phase that allows for such computational efficiencies seen here to be realized.

10.2 Future Research Directions

This work, as mentioned developed a framework methodology for complimenting a low-fidelity design with a high-fidelity analysis. In this work, the high-fidelity analysis was limited to a structural analysis of the undersea vehicle hull. In the future, the framework will be expanded to include a sonar module. This will provide accurate responses of the sonar’s structural integrity as well as the dynamic calculations of sonar vibrations. These attributes can then be used to determine the actual sonar performance as part of the system design. Furthermore, a high-fidelity acoustic model will be included in the analysis portion of the system design. This will complement SNARC’s low-fidelity approximations and provide a more accurate depiction of the acoustic response due to the coupling of the shell design and the component noise.

The integration of acoustics into the system design framework allows
for a tighter coupling of acoustic performance to mission performance. Once estimates on the sound propagation are available, these sound data can then be used to determine the feasibility of a mission. For instance, a particular hazard may not be able to detect sound levels below a certain value. Additionally, the detection radius of a potential hazard may be mapped as a function of the sound produced by the vehicle at any particular point in time along the mission. All of this allows for a more accurate representation of the mission space, further advancing system design.

In order to make the system framework developed here more accessible to the research community, PHX ModelCenter will be used as the framework software. PHX ModelCenter by Phoenix Integration has a graphical user interface as well as a large support infrastructure which will allow for a more visual representation of the data flow within the system design framework.
Bibliography


Brauer, G., Petersen, F., Cornick, D., Stevenson, R., and Olson, D., Programs To Optimize Spacecraft And Aircraft Trajectories, NASA Center: Langley Research Center, 1997.


[40] Ghanem, R., “Ingredients for a General Purpose Stochastic Finite Elements Imple-

[41] Chiralaksanakul, A. and Mahadevan, S., “First-Order Approximation Methods in

[42] Zhao, Y. and Ono, T., “A General Procedure for First/Second-Order Reliability


Selecting Values of Input Variables in the Analysis of Output from a Computer


[51] Naeem, W., Sutton, R., Ahmad, S. M., and Burns, R. S., “A Review of Guid-
ance Laws Applicable to Unmanned Underwater Vehicles,” Journal of Navigation,

[52] Do, K., Pan, J., and Jiang, Z., “Robust and adaptive path following for under-


