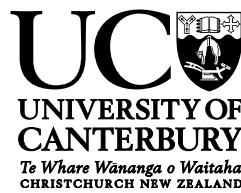


Intersection Between Natural and Artificial Swimmers

A scaling approach to underwater vehicle design.



Michael Coe

Supervisors: Ass. Prof. Stefanie Gutschmidt

Prof. Mark Jermy

Dr. Natalia Kabaliuk

Department of Mechanical Engineering
University of Canterbury

This dissertation is submitted for the degree of
Doctor of Philosophy

I would like to dedicate this thesis to my partner and loving parents . . .

Acknowledgements

I would like to thank my supervisor Associate Professor Stefanie Gutschmidt for taking me on as a student, mentoring, and support, Dr. Natalia Kabaliuk for her knowledge and support, Dr. Tim Giffney for entertaining some of my crazy ideas, and to Professor Mark Jermy for supporting my research. I would also wish to thank the technical staff of the University of Canterbury Mechanical Engineering Department. A special acknowledgement to the IT staff: Paul Southward, Adam Latham, and François Bissey, for their excellent support. I would also like to thank the technical staff at New Zealand eScience Infrastructure for their support. Finally, to all the other postgraduate students that have shared in the highs and lows of this journey.

Abstract

Approximately 72% of the Earth's surface is covered by water, yet only 20% has been mapped [1]. Autonomous Underwater Vehicles (AUVs) are one of the main tools for ocean exploration. The demand for AUVs is expected to increase rapidly in the coming years [2], so there is a need for faster and more energy efficient AUVs. A drawback to using this type of vehicle is the finite amount of energy that is stored onboard in the form of batteries. Science and roboticists have been studying nature for ways to move more efficiently. Phillips *et al.* [3] presents data that contradicts the idea that fish are better swimmers than conventional AUVs when comparing the energetic cost of swimming in the form of the Cost of Transport (COT). The data presented by Phillips *et al.* only applies to AUVs at higher length and naval displacement (mass) scales, so the question arises of whether an AUV built at different displacements and length scales is more efficient than biological animals and if current bio-inspired platforms are better than conventional AUVs.

Besides power requirements, it is also useful to compare the kinematic parameters of natural and artificial swimmers. In this case, kinematic parameters indicate how fast the swimmer travels through the water. Also, they describe how fast the propulsion mechanism must act to reach a certain swimming speed. This research adopts the approach of Gazzola *et al.* [4] where the Reynolds number is associated with a dimensionless number, Swim number (Sw) in this case, that has all the kinematic information. A newly developed number that extends the swim number to conventional AUVs is the Propulsion number (Jw), which demonstrates excellent agreement with the kinematics of conventional AUVs. Despite being functionally similar, Sw and Jw do not have a one-to-one relationship. Sw , Jw , COT represent key performance metrics for an AUV, herein called performance criteria, which can be used to compare existing platforms with each other and estimate the performance of non-existent designs.

The scaling laws are derived by evaluating the performance of 229 biological animals, 163 bioinspire platforms, and 109 conventional AUVs. AUVs and bio-inspired platforms have scarce data compared with biological swimmers. Only 5% of conventional and 38% of bio-inspired AUVs have kinematic data while 30% of conventional and 18% of bio-inspired AUVs have energetic data. The low amount of performance criteria data is due to the nature

of most conventional AUVs as commercial products. Only recently has the COT metric been included in the performance criteria for bio-inspired AUVs. For this reason, the research here formulates everything in terms of allometric scaling laws. This type of formulation is used extensively when referring to biological systems and is defined by an exponential relationship $f(x) = ax^b$, where x is a physical parameter of the fish or vehicle, like length or displacement. Scaling laws have the added benefit of allowing comparisons with limited data, as is the case for AUVs.

The length and displacement scale (physical scale) must be established before estimating the performance criteria. Scale is primarily determined by the payload needed for a particular application. For instance, surveying the water column in deep water will require different scientific tools than taking images of an oyster bed in an estuary. There is no way to identify the size of an AUV until it is designed for that application, since these scientific instruments each have their own volume, length, and weight. A methodology for estimating physical parameters using computer vision is presented to help determine the scale for the vehicle. It allows accurate scaling of physical parameters of biological and bio-inspired swimmers with only a side and top view of the platform. A physical scale can also be determined based on the vehicle's overall volume, which is useful when determining how much payload is needed for a particular application. Further, this can be used in conjunction with 3D modeling software to scale nonexistent platforms.

Following the establishment of a physical scale, which locomotion mode would be most appropriate? Unlike conventional AUVs that use propeller or glider locomotion, bio-inspired platforms use a variety of modes. Kinematics and energy expenditures are different for each of these modes. For bio-inspired vehicles, the focus will be on the body-caudal fin (BCF) locomotion, of which four types exist: anguilliform, carangiform, thunniform, and ostraciiform. There is ample research on anguilliform and carangiform locomotion modes, but little research on thunniform and ostraciiform modes. In order to determine which locomotion mode scales best for a bio-inspired AUV, this research examines the power output and kinematic parameters for all four BCF modes. In order to achieve this, computational fluid dynamics simulations are performed on a 2D swimmer for all four modes. Overset meshes are used in lieu of body-fitted meshes to increase stability and decrease computational time. These simulations were used to scale output power over several decades of Reynolds numbers for each locomotion mode. Carangiform locomotion was found to be the most energy efficient, followed by anguilliform, thunniform, and ostraciiform.

In order to utilize the above scaling laws in designing a novel platform, or comparing an existing one, there must be a unifying framework. The framework for choosing a suitable platform is presented with a case study of two bio-inspired vehicles and a conventional one.

The framework begins by determining how the platform can be physically scaled depending on the payload. Based on the physical scale and derived scaling laws, it then determines performance criteria. It also describes a method for relative cost scaling for each vehicle, which is not covered in the literature. The cost scaling is based on the assumption that all payloads and materials are the same. The case study shows that a conventional AUV performs better on all performance criteria and would cost less to build.

Deputy Vice-Chancellor's Office
Postgraduate Research Office



Co-Authorship Form

This form is to accompany the submission of any thesis that contains research reported in co-authored work that has been published, accepted for publication, or submitted for publication. A copy of this form should be included for each co-authored work that is included in the thesis. Completed forms should be included at the front (after the thesis abstract) of each copy of the thesis submitted for examination and library deposit.

Please indicate the chapter/section/pages of this thesis that are extracted from co-authored work and provide details of the publication or submission from the extract comes:

Chapter 3: Better Estimates of Surface Area

Published in

M. Coe and S. Gutschmidt, "Computer vision estimation of physical parameters and its application to power requirements of natural and artificial swimmers," Designs, vol. 5, no. 4, 2021.

Please detail the nature and extent (%) of contribution by the candidate:

100%

Certification by Co-authors:

If there is more than one co-author then a single co-author can sign on behalf of all

The undersigned certifies that:

- The above statement correctly reflects the nature and extent of the Doctoral candidate's contribution to this co-authored work
- In cases where the candidate was the lead author of the co-authored work he or she wrote the text

Name: *Stefanie Gutschmidt* Signature:  Date: 15/06/2022

Publications

1. M. Coe and S. Gutschmidt, “Computer vision estimation of physical parameters and its application to power requirements of natural and artificial swimmers,” *Designs*, vol. 5, no. 4, 2021.

Table of contents

List of figures	xv
List of tables	xxv
1 Are Fish More Efficient in Locomotion Than Engineered Systems?	1
1.1 Motivation	2
1.2 Objectives	4
1.3 Structure	5
2 Cost of Transport and Scaling Laws	7
2.1 Introduction	7
2.2 Natural Swimmers Versus Engineered Systems	8
2.3 Borrowing from Biologists: The Cost of Transport	14
2.3.1 Another Model for COT in Engineered Systems	20
2.3.2 Cost of Transport in Natural swimmers	22
2.3.3 Cost of Transport in Artificial Swimmers	25
2.4 Influence of Kinematic and Power Scaling	34
2.4.1 Kinematic Scaling	34
2.5 Power Scaling Laws	37
2.6 Hotel Power	40
2.6.1 Hotel Power (BMR) For Biological Animals	41
2.7 Are Biological Fins Better Than Engineered Thrusters?	49
2.7.1 Hybrid Biological Locomotion with Thrusters	50
2.7.2 Where Does Biological Design Excel?	52
2.8 Concluding Remarks	54
3 Better Estimates of Surface Area	55
3.1 Introduction	55
3.2 Results	57

3.2.1	Validation with 3D Scanned Model	57
3.2.2	Validation with Biological Animals	58
3.2.3	Application to Cost of Transport	65
3.3	Concluding Remarks	71
4	Do Computers Dream of Swimming Fish?	73
4.1	Introduction	73
4.2	BCF Motion	74
4.2.1	Anguilliform Motion	76
4.2.2	Carangiform Motion	77
4.2.3	Thunniform Motion	80
4.2.4	Ostraciiform Motion	81
4.3	Overset Grid	83
4.4	Computational Domain	84
4.5	Ika-Flow Implementation	88
4.6	Numerical Setup	92
4.7	Forces, Power, and Efficiencies	95
4.8	Post-processing	98
4.9	Limitations and Comparisons	100
4.9.1	Pimple Iterations	102
4.9.2	Time Delay	104
4.9.3	Grid Independence and Validation	104
4.9.4	The Effect of Max Courant Number	107
4.9.5	Ansys vs. GMSH mesher	108
4.9.6	Turbulence Models	110
4.10	Simulation Results	114
4.10.1	Pressure Coefficient	119
4.10.2	Wake	125
4.10.3	Optimum Linkages	125
4.11	Concluding Remarks	129
5	How Do Things Scale?	131
5.1	Introduction	131
5.2	Energetic Scaling of Natural and Artificial Swimmers	131
5.3	Kinematic Scaling of Bio-inspired Robotic Platforms	133
5.4	Adapting Swim Number to Conventional AUVs	139
5.5	Cost Scaling of AUVs	143

5.6	Concluding Remarks	149
6	An Algorithm to Bring It All Together	151
6.1	Introduction	151
6.2	Application Requirements	151
6.3	Propulsion Modes	152
6.4	Payload	153
6.5	Kinematic Performance and Parameters	156
6.6	Cost of Transport	158
6.7	Cost Estimation	159
6.8	Concluding Remarks	163
7	Conclusions and Recommendations for Future Work	165
7.1	Novelties and Conclusions	165
7.1.1	Comparing Performance of Natural and Artificial Swimmers	165
7.1.2	Computer Vision Estimation of Physical Parameters	166
7.1.3	Simulation of BCF Locomotion Modes	166
7.1.4	Scaling Laws	167
7.1.5	Framework for Design	169
7.2	Recommendations for Future Work	169
	References	171
	Appendix A Data For models	201
A.1	Biological Kinematic Data	201
A.2	Conventional AUV Data	203
A.3	Bio-inspired BCF AUV Data	205
A.4	Bio-inspired MPF AUV Data	208
A.5	Bio-inspired Lift Based AUV Data	211
A.6	Bio-inspired Jet Based AUV Data	211
A.7	Conventional AUV Propeller Number Data	212
A.8	AUV Applications Data	214
	Appendix B The Ika-Fit Method	217
B.0.1	Ika-Fit Method	217
B.0.2	Ellipsoid Approximation	226

Appendix C Implementation of Turbulence Models in OpenFOAM®	231
C.1 General Reynolds Averaged Navier-Stokes (RANS) Formulation	231
C.2 k - ω Shear stress Transport (SST)	233
C.3 k - k_L - ω	234
C.4 Spalart-Allmaras	238
C.5 Turbulent Boundary Conditions	240
C.6 First Layer Thickness	241
Appendix D Ika-Flow Code	245
D.1 Ika-Flow Overset motion code	245
D.2 Ika-Flow patch motion code	266
D.3 Ansys Fluent UDF code	278

List of figures

1.1	AUV growth prior to 2022 from Westwood Global Energy Group [2]. . . .	3
2.1	Sketch of the anatomy of a fish showing the terminology in the text for fins and other features. Adapted from [5].	9
2.2	Locomotion modes of fish Swimming. Top row (a) represents BCF swimmers and bottom two rows (b) represent MPF propulsion. Red and black shaded areas show areas that contribute to thrust generation. Adapted from [6]. . .	9
2.3	Wake profiles for propeller (top) and fish (bottom) during locomotion. . . .	11
2.4	Schematic diagram for the actuators of Autonomous Underwater Vehicles. SMA is Shape Memory Alloy, MFC is Macro Fiber Composite, DEA is Dielectric Elastomer Actuator, and IPMC is Ionic Polymer Metal Composite.	12
2.5	Diagram showing the similarities, efficiency, and energy losses for an AUV and silver eel. Adapted from [3]. The top red box represents the energy generation for both swimmers. The combination of the red and blue boxes represents the hotel power, and the yellow box represents the propulsive power of the two swimmers.	15
2.6	(a) The influence on COT (2.6) of changing the (a) wetted surface area, (b) slenderness ratio, (c) mass, and (d) $1/(\eta_a\eta_p)$ with all other variables being equal. P_B in all cases is kept at a constant 25 Watts. The dashed black line represented the division between the region in which hotel power dominates at low velocities and the region where propulsion power dominates at higher velocities.	18
2.7	COT versus body weight for flying, running, and swimming animals fit with an allometric power law. Original data is from Schmidt-Nielsen [7]. COT_{opt} is the COT at optimum velocity where COT is at a minimum.	22

2.8	(a) Dimensionless GCOT (J/Nm) versus Re number. (b) Energy needed to transport the body weight of a swimmer at the optimum speed over its body length related to body mass. (c) and (d) Energy needed to transport the the body weight of a swimmer at optimum speed related to Reynolds Number and body mass. Data from Videler and Nolet [8].	23
2.9	COT for aquatic animals categorized in trophic levels as defined in Table 2.4. Ellipses cover relative area of each trophic level for this data set. COT_{opt} is the COT at U_{opt} which represents the velocity where COT is lowest.	26
2.10	Specific power of single vehicles versus maximum speed from before 1950. Redrawn from [9].	27
2.11	Comparison of Cost of Transport versus displacement for natural and artificial swimmers. Data collected and collated by Phillips <i>et al.</i> [3]. COT_{opt} is the COT at U_{opt} which represents the velocity where COT is lowest.	29
2.12	COT of biological and artificial swimmers. Biological data is from Videler and Nolet [8]. Conventional Propeller and Glider information is taken from [10–15]. BI Carangiform data is from [16–27]. BI Ostraciiform data is from [28, 29]. BI Thunniform data is from [30–32]. BI Anguiliform data is from [33–35]. BI Sub-Carangiform data is from [36]. MPF data is from [37–39]. Lift based data is from [40, 41]. Best fit allometric trend line for power law fit is shown in legend. COT_{opt} is the COT at U_{opt} which represents the velocity where COT is lowest.	31
2.13	COT of biological and artificial swimmers. Biological data is from Videler and Nolet [8]. Conventional Propeller and Glider information is taken from [10–15]. BI Carangiform data is from [16–27]. BI Ostraciiform data is from [28, 29]. BI Thunniform data is from [30–32]. BI Anguiliform data is from [33–35]. BI Sub-Carangiform data is from [36]. MPF data is from [37–39]. Lift based data is from [40, 41]. Best fit allometric trend line for power law fit is shown in legend. ϵ is the COT at U_{opt} which represents the velocity where COT is lowest.	32
2.14	Extension of the von Kármán-Grabrielli methodology to AUVs [9]. ϵ is the COT at U_{opt} which represents the velocity where COT is lowest.	33
2.15	Swim vs. Reynolds number for a collection of swimming and flying animals. Data for fish taken from Videler [42] and all other data from Gazzola <i>et al.</i> [4]. Figure adapted from Gazzola <i>et al</i> [4].	35

2.16	Swim vs. Reynolds number for 2D and 3D simulations. The 2D and 3D simulations scale with the same exponent in both the laminar and turbulent regimes. Adapted from Gazzola <i>et al</i> [4].	36
2.17	(a) $\langle C_T \rangle$ and (b) COT versus Re for the University of Virginia Tunabot using the propulsion scaling model of Yu <i>et al.</i> [43]. Numbers in (b) represent the different regions of the Cost of Transport model. 1 is the region where hotel power is the dominate energy consumption of the vehicle and 2 is the region where propulsion power is the dominate energy sink.	40
2.18	Data collected in [3] and fit with a power-law curve to obtain the hotel power scaling for biological animals. Displacement in this case is the mass of the swimmer, assuming neutrally buoyant.	42
2.19	(a) Actuator Power and (b) Actuator Torque versus actuator mass for Maxon motors from [44] and Hitec servos from [45].	44
2.20	Actuator power versus torque for Maxon motors from [44] and Hitec servos from [45].	45
2.21	CAD rendering of common servo linkage transmission system from Malec <i>et al.</i> [46, Fig. 3].	46
2.22	(a) hotel power and (b) propulsion power for bio-inspired platforms plotted over the number of actuators. (c) hotel power and (d) propulsion power for bio-inspired platforms plotted over the number of linkages. Linkages and actuators were were taken from pictures of platforms from published works.	47
2.23	Diagram showing that more linkages lead to a more optimized undulation form. Green circle and links show curvature for a 4 link tail. Red circle and links represent the same curvature with 2 links.	48
2.24	Strouhal versus Reynolds number for 3000 swimming animals from Gazzola <i>et al.</i> [4, Fig. 2.b].	50
2.25	(a) Power versus Thrust and (b) Comparative Efficiency versus Thrust showing the power and efficiency of commercially available thrusters compared to a biomimetic fin from Palmisano <i>et al.</i> [47, Figures 1 and 12].	51
2.26	(a) velocity versus undulation speed and (b) average power versus undulation speed showing different cases for an eel-like robot with thrusters from Kelasidi <i>et al.</i> [35, Figures 11.e and 11.f].	52

2.27	(a) Length versus minimum turn radius and (b) length versus maximum turn rate for conventional AUVs, bio-inspired robotic platforms, and biological animals. Turn Radius BI BCF data from [19, 20, 23–27, 31, 34, 48–62]. BI MPF data from [63–66]. BI Lift data from [67]. All diamond data collected by Fish [68].	53
3.1	(a,b) Validation of Ika-Fit (a) surface area and (b) volume algorithm with scanned salmon data scaled using Blender software.	57
3.2	(a) The calculated mass versus published data for (a) Atlantic salmon, (b) Atlantic cod, (c) killer whale, and (d) European silver eel.	59
3.3	Comparison of surface area of all algorithms with the scanned salmon model that has been scaled inside Blender.	60
3.6	Comparison plot between the surface area estimation algorithm, prolate spheroid method [69, 70], partition disc method [70], and O’Shea regression lines [71]. (a) Unscaled Atlantic salmon data, and (b) scaled Atlantic salmon data.	64
3.7	(a) Comparison of surface area and (b) mass for bio-robotic platforms. Ika-Fit NACA and Ika-Fit Poly are this study’s method using the NACA fit described in Section B.0.1, and Ika-Fit Poly is the top contour’s fit with a fourth degree polynomial in a similar way to the side contours.	66
3.8	Comparison of the Ika-Fit and PSM method in calculating COT with published data. (a) Killer whale data from [72] and European silver eel data from [73]. (b) Sockeye salmon data from [74] and Atlantic salmon data from [75].	67
3.9	Comparison of the Ika-Fit and PSM method in calculating COT with published bio-mimetic robot data. RoboSalmon data from [36], Knifebot data from [76], and UVTunabot data from [30]. The dashed black line marks the demarcation of regions 1 and 2 for the UVTunabot.	68
4.1	Comparison of one cycle of the all BCF swimmers.	75
4.2	(a) Diagram showing the envelope and motion of an anguilliform swimmer. (b) Midline motion of an anguilliform swimmer at different times steps for a is set to 0.1, α is set to 1.	77
4.3	Comparison of shape functions for anguilliform and carangiform mode. Amplitude function given in legend.	78

4.4	Visualization showing (a) shape of an airfoil swimming and (b) shape of the midline during anguilliform motion. t/τ represents the time (t) over 1 tail beat (τ).	78
4.5	(a) Diagram showing the envelope and motion of an carangiform swimmer. (b) Midline motion of an carangiform swimmer at different times steps. a is 0.1 and coefficients of (4.10)	79
4.6	Visualization showing (a) shape of an airfoil swimming and (b) shape of the midline during carangiform motion. t/τ represents the time (t) over 1 tail beat (τ).	80
4.7	(a) Diagram showing the envelope and motion of an thunniform swimmer. The pitching foil is shown inset to the right. (b) Midline motion of an thunniform swimmer at different times steps. a is 0.1 and coefficients of (4.10)	81
4.8	Visualization showing (a) shape of an airfoil swimming and (b) shape of the midline during carangiform motion. t/τ represents the time (t) over 1 tail beat (τ).	81
4.9	(a) Diagram showing the envelope for the body and tail. (b) Midline motion of an ostraciiform swimmer at different times steps.	82
4.10	Visualization showing (a) shape of an airfoil swimming and (b) shape of the midline during carangiform motion. t/τ represents the time (t) over 1 tail beat (τ).	82
4.11	Schematic showing overset interpolation from the source grid (donor cells in black) to the destination grid (acceptor cell in red).	84
4.12	Overset mesh showing the types of cells as rendered by Paraview [77]. Blue cells represent the calculated cells, red cells are the holes, and white cells represent the interpolation region. The number assigned to the cells is given by the cell types colormap, 2.0 for holes, 1.0 for interpolated, and 0.0 for calculated. Red cells outside the fish airfoil are cut from the background mesh and are an artifact from the rendering of the two meshes on top of each other.	85
4.13	Schematic of computational fluid domain. Overset region is shown around the "airfoil", 2 refinement regions are used as shaded in red and blue. Note that the coordinate system is set so that $-x$ is in the direction of flow as is customary for fish simulations.	86

4.14	(a) Overview of the mesh with background mesh and two refined regions encompassing the overset mesh and (b) a zoomed view of the overset mesh showing the interpolation region of the overset to the background mesh being near the same size.	87
4.15	Original Solid Body class in UML	88
4.16	Modified fish body class in UML. The new class does not call the transformField class, but instead provides it's own transformation class inside appropriate motion function. This can be seen when comparing the oscillatingRotatingMotion class to the carangiformMotion class.	89
4.17	Two meshes showing the effect of the damping factor, controlled by the ramp entry in the dynamicMeshDict. The left mesh shows a ramp factor of 10.0 and the right mesh shows a ramp factor of 40.0.	91
4.18	PISO algorithm with non-iterative marching	93
4.19	PISO algorithm with iterative marching (PIMPLE). Adapted from Guerrero [78]	94
4.20	Comparison of forces in the x-direction before and after filtering using a Hanning filter.	98
4.21	Q criterion for carangiform swimming at a Strouhal of 40. Simulation flow time is 60 seconds. 3D Circular vortexes are superimposed onto a velocity contour. Rendering is done in ParaView [77].	99
4.22	(a) C_D , (b) C_L , (c) C_D zoomed in, and (d) C_L zoomed in for all four BCF swimming modes for 60 seconds at a strouhal number of 40 using the PISO algorithm (1 nOuterCorrectors).	101
4.23	(a) C_D , (b) C_L , (c) C_D zoomed in, and (d) C_L zoomed in of carangiform swimming for 60 seconds utilizing 1, 5, and 25 pimpleFoam iterations. Simulations are performed at Reynolds number $5 \cdot 10^3$ and at Strouhal number of 40.	103
4.24	(a) C_D , (b) C_L for carangiform swimming for 60 seconds utilizing 1, 5, and 25 pimpleFoam iterations. (c) C_D and (d) C_L for the same simulation with a delay of 5 and 15 seconds. Simulations were performed at Reynolds number $5 \cdot 10^3$ and at a Strouhal number of 40.	105
4.25	(a) C_D and (b) C_L for the same simulation data from Dong and Lu [79] and Yu <i>et al.</i> [43]. Simulations performed at a Reynolds number of $5 \cdot 10^3$ at a Strouhal number of 40.0 with only the carangiform swim mode. Dashed lines represent the time averaged values over the tail beat cycles.	106
4.26	(a) C_D and (b) C_L for max Courant numbers of 0.5, 1.0, and 2.0. Simulations performed at Reynolds numbers $5 \cdot 10^3$ and Strouhal number of 40.	107

4.27	(a) Ansys and (b) GMSH mesh generated using the same parameters in Table 4.1. (c) C_D and (d) C_L zoomed in for both meshes after steady state has been reached.	109
4.28	(a) C_D and (b) C_L for different turbulence models. Simulations performed at Reynolds numbers $5e03$ (left plots) and $4e05$ (right plots) with a Strouhal number of 40.0.	113
4.29	(a) Anguilliform drag / thrust coefficient and (b) zoomed in of (a) to show zero crossing. (c) Carangiform drag / thrust coefficient and (d) zoomed in of (c) to show zero crossing.	116
4.30	(a) Ostraciiform drag / thrust coefficient, (b) zoomed in of (a) to show zero crossing, (c) thunniform drag / thrust coefficient, and (d) zoomed in of (c) to show zero crossing. Note that the $5 \cdot 10^3$ case for the ostraciiform and all cases for thunniform mode never pass the zero line. In these cases, a third degree polynomial was fit and the self propulsion case was taken from that fit.	117
4.31	Scaling of self propelled Strouhal numbers for all four locomotion modes. Note that the scaling is broken into two parts consistent with experimental observations given by Gazolla <i>et al.</i> [4]. The divergent scaling occurs at the transition from laminar to turbulent regimes.	118
4.32	(a) Anguilliform, (b) carangiform, (c) ostraciiform, and (d) thunniform total power.	120
4.33	Total power scaling for each locomotion mode.	121
4.34	(a), (c) Swimmer at first half of cycle time, $\frac{t}{T} = 0.25, 0.5$. Anguilliform, carangiform, ostraciiform, and thunniform modes from top to bottom. (b), (d) upper and lower pressure coefficient, C_p , for each locomotion mode plotted over the chord length. Simulations performed at Reynolds number of $5 \cdot 10^3$ and Strouhal number of 40. Solid lines represent the lower airfoil and dashed lines represent the upper airfoil.	123
4.35	(a), (c) Swimmer at first half of cycle time, $\frac{t}{T} = 0.75, 0.5$. Anguilliform, carangiform, ostraciiform, and thunniform modes from top to bottom. (b), (d) upper and lower pressure coefficient, C_p , for each locomotion mode plotted over the chord length. Simulations performed at Reynolds number of $5 \cdot 10^3$ and Strouhal number of 40. Solid lines represent the lower airfoil and dashed lines represent the upper airfoil.	124
4.36	Velocity magnitude and q-criterion for (a) anguilliform and (b) carangiform locomotion modes. Vortices show the characteristic reverse Kármán street present in biological swimmers.	126

4.37	Velocity magnitude and q-criterion for (a) ostraciiform and (b) thunniform locomotion modes. Vortices show the characteristic reverse Kármán street present in biological swimmers.	127
4.38	Linkages and actuators for bio-inspired BCF vehicles.	128
5.1	COT of biological and artificial swimmers. Biological data is from Videler and Nolet [8]. Conventional Propeller and Glider information is taken from [10–15]. BI Carangiform data is from [16–27]. BI Ostraciiform data is from [28, 29]. BI Thunniform data is from [30–32]. BI Anguiliform data is from [33–35]. BI Sub-Carangiform data is from [36]. MPF data is from [37–39]. Lift based data is from [40, 41]. Best fit allometric trend line for power law fit is shown in legend underneath their corresponding data set.	132
5.2	COT of biological and artificial swimmers. Biological data is from Videler and Nolet [8]. Conventional Propeller and Glider information is taken from [10–15]. BI Carangiform data is from [16–27]. BI Ostraciiform data is from [28, 29]. BI Thunniform data is from [30–32]. BI Anguiliform data is from [33–35]. BI Sub-Carangiform data is from [36]. MPF data is from [37–39]. Lift based data is from [40, 41]. Best fit allometric trend line for power law fit is shown in legend underneath their corresponding data set.	134
5.3	Swim vs. Reynolds number for a collection of swimming animals and bio-inspired robotic platforms. Data for fish taken from Videler [42]. BCF data taken from [16, 17, 19, 27, 28, 30, 36, 53, 57, 80–110]. MPF data taken from [37, 111–127]. Lift data taken from [40, 128]. Best fit allometric trend line for power law fit is shown in legend underneath their corresponding data set.	136
5.4	Optimum COT versus Reynolds number and Reynolds number versus Swim number for selected AUV platforms. Arrows give the correlation between the same platforms on the different plots. Black dashed line is the swim number scaling line from Gazzolla <i>et al.</i> [4].	137
5.5	Swim vs. Reynolds number for a collection of AUVs. Data taken from the following: AMC Explorer [129], MUN Explorer [130], HUGIN 4500 [131], Autosub3 [132], IM and Wayamba [133]. Black dashed line is the scaling relationship derived in Gazzolla <i>et al.</i> [4]. Orange dashed line is the scaling relationship of the torpedo shaped single thruster AUVs and the brown dashed line is the scaling line for the flatfish type AUV, Wayamba. .	140
5.6	Diameter of AUVs versus their propeller diameter. Data taken from the following: AMC Explorer [129], MUN Explorer [130], HUGIN 4500 [131], Autosub3 [132], IM and Wayamba [133].	142

5.7	Schematic of the interior layout of Dorado Class AUV adapted from a combination of schematics given in Sanchez <i>et al.</i> [134] and Henthorn <i>et al.</i> [135]. Real picture given in Henthorn <i>et al.</i> [135].	143
5.8	Maximum depth versus Reynolds Number for numerous conventional AUVs (blue circle), bio-inspired BCF (orange triangle), bio-inspired MPF (red star), and bio-inspired lift based (upside down purple triangle).	145
5.9	Surface area versus (a) displacement and (b) length for all conventional and bio-inspired AUVs.	146
5.10	Motor price versus (a) nominal torque and (b) nominal speed for Maxon motors from Maxon Inc. [44].	148
6.1	Schematic of the overall diagram to design an Autonomous Underwater Vehicle. Each block is comprised of sub block diagrams.	152
6.2	Schematic of Propulsion modes.	153
6.3	Schematic of payload for an AUV.	155
6.4	Surface area scaling for the three designs given in table 6.1. Inset graph shows the surface area of each design at the correct scaling to match the volume of the MBARI AUV.	161
6.5	Length versus Weight for AUVs with power scaling laws given.	162
B.1	Images used in the validation of the Ika-Fit method with salmon modeled in Blender. (a) Salmon with caudal fin, (b) Salmon with no caudal fin, and (c) Salmon top with fins.	218
B.2	The steps taken for image segmentation.	219
B.3	<i>Cont.</i>	220
B.3	Images showing the algorithm on a complex fish specimen. (a) Initial image, (b) S and (c) V components, (d) K-means color quantized image, (e) image after morphological opening and closing, (f) canny edge-detected Image, and (g) contours overlayed over image (e).	221
B.4	(a,c) Side and top view with the overlayed centroid (blue dot), line slicing through the middle of the specimen (blue line), dividing line (red line) to extrema points, and white line representing the contours. (b,d) Plotted contour after dividing the top and bottom of the specimen and zeroing the resulting contours. Image obtained from photographs of salmon purchased from local fishery.	222

- B.5 Contour lines (side and top views) of scanned salmon; the yellow line is the real contour line, solid (blue and red) lines are computed lines for side contour, and the blue dashed line is the computed line for top views. **(a)** King salmon side contour with fit equations shown in the legend. **(b)** King salmon top view contour with NACA fit overlayed. Area output from the area matching algorithm is shown. 223
- B.6 Schematic view of a symmetric airfoil. c is the chord length, t is the maximum thickness as a percentage of chord length, and m is the x-position where maximum thickness occurs. x_L represents the x-coordinate of the Leading Edge and x_T is the coordinates of the Trailing Edge. 224
- B.7 **(a)** Prolate spheroid with elliptical section shown. Spheroid is partitioned into discs shown in red. **(b)** Schematic of elliptical section showing parameters used in surface area approximation: b_T is the top contour, b_B is the bottom contour, a is the thickness of the NACA airfoil fit, and dx is the thickness of the elliptical section. 228

List of tables

2.1	Number of existing AUVs. Data represents a mix of company brochure as well as academic journals. Fish modes are described in Fig. 2.2, with oscillatory referring to Rajiform locomotion but with oscillating fin motions versus undulating fin motions, such as that of a cuttlefish.	10
2.2	Breakdown of the number of actuators per each type of platform. IPMC is Ionic Polymer Metal Composite, MFC is Macro Fiber Composite, SMA is Shape Memory Alloy, DEA is Dielectric Elastomer Actuator, and EAP is Electroactive Polymer.	13
2.3	Table of typical efficiencies summarized in [3] and expanded on with data from a review in [136].	19
2.4	Description of the trophic levels as defined by the Food and Agricultural Organization of the United Nations [137].	25
2.5	Allometric scaling laws for BMR from meta analysis given by Table 1 in Ballestoros <i>et al.</i> [138]. The model used for all data reported is of the form $BMR = aM^k$ where M is the body mass of the animals. N is the number of animals in each data correlation.	43
2.6	Allometric scaling laws for motor and servo data. The model used for all data reported is of the form $y = ax^k$ where y is the independent variable and x is the dependent variable.	45
2.7	Comparison of calculated hotel power and propulsion power of different bio-inspired platforms and their actuator type.	48
2.8	Data showing the power consumption from onboard electronics for the SEMBIO platform from Amory and Maehle [139, Table 2].	49
3.1	RMSD and MAE of the Ika-Fit method of four animal species. Error is given as a percentage of the Ika-Fit estimate versus published data.	59

3.2	RMSD and MAE of compared methods with laser-scanned salmon. Error is given as a percentage of the estimated algorithm value versus the direct value given by the 3D model.	61
3.3	RMSD and MAE of compared methods with Atlantic salmon data from O'Shea et al. [71]. Error is given as a percentage of the estimated algorithm value versus the direct value given by the 3D model.	61
3.4	RMSD and MAE of compared methods with Atlantic cod data from O'Shea et al. [71]. Error is given as a percentage of the estimated algorithm value versus the direct value given by the 3D model.	62
3.5	Relative time difference between methods. Time was measured from the image input to the value output.	64
3.6	Bio-inspired platforms used as validation in this research. Relevant parameters and citations are given.	65
3.7	Pros and Cons of the using the Ika-Fit method.	70
4.1	Summary of the mesh composition and metrics for the overset and background mesh. Metrics were given by OpenFOAM [®] checkMesh utility with allGeometry and allTopology flags turned on.	87
4.2	Summary of dynamicMeshDict entries for all four BCF locomotion modes. All angles are in degrees.	91
4.3	Numerical schemes and their respective term. Schemes are limited to second order except for time scheme. x_j represents the partial derivative where 1 – 3 is a Cartesian coordinate direction (i.e.) $1 = \partial x, 2 = \partial y, 3 = \partial z$	95
4.4	Summary of simulation times reaching a quasi steady-state condition and completion of 60 seconds of flow time using 8 cores.	102
4.5	Summary of simulation times for 1, 5, and 25 pimple iterations (nOuterCorrectors).	103
4.6	Summary of simulation times the same simulation with no delay, a 5 second delay, and a 15 second delay.	104
4.7	Summary of simulations for increasing grid sizes.	106
4.8	Summary of simulations for increasing grid sizes.	108
4.9	Summary of simulation times for two different meshing programs.	108
4.10	Lift C_L and drag C_D for three different turbulence models, XFOIL, and data from McCrosky [140] at Reynolds numbers of 5e03 and 4e05.	112
4.11	Summary of simulation times for three different turbulence models at $Re = [5 \cdot 10^3, 4 \cdot 10^5]$	114

4.12	Strouhal numbers that satisfy the self-propulsion (SPS) condition at tested Reynolds numbers.	115
4.13	Summary of power Scaling for natural and artificial swimmers. All equations are of the form ax^b where x is the dependent variable and Reynolds number in for all data in this table.	119
5.1	Summary of COT Scaling with Reynolds Number for natural and artificial swimmers. All equations are of the form ax^b where x is the dependent variable and in this case is Reynolds number (Re).	133
5.2	Published Images of designs from Fig. 5.4.	138
5.3	Specifics of common AUV materials adapted from [141]. GFRP is Glass Fiber Reinforced Polymer and CFRP is Carbon Fiber Reinforced Polymer. Costs are provided in units of USD. Costs are given for raw material and alloys of metals may have different prices.	147
6.1	Published Images of designs used in cost scaling analysis. Numbers in parenthesis is the scaled value by the most limiting dimension.	156
6.2	Summary of Kinematic Scaling for natural and artificial swimmers. All equations are of the form ax^b where x is the dependent variable.	156
6.3	Summary of the kinematic scaling parameters for the case study designs. . .	158
6.4	Summary of COT Scaling for natural and artificial swimmers. All equations are of the form ax^b where x is the dependent variable. COT is in units of [J].	159
6.5	Summary of the COT scaling parameters for the case study designs.	159
6.6	Scaled parameters for MBARI AUV and three designs given in table 6.1. . .	161
A.1	Table of kinematic parameters for various fish taken from Videler [42]. . . .	201
A.2	Table of kinematic parameters for various fish taken from Videler and Nolet [8].	202
A.3	Table of kinematic and COT data for conventional AUV platforms.	203
A.4	Table of kinematic and COT data for conventional AUV platforms (Continued).	204
A.5	Table of kinematic and COT data for BCF Bio-inspired Robotic Platforms. .	205
A.6	Table of kinematic and COT data for BCF Bio-inspired Robotic Platforms (Continued).	206
A.7	Table of kinematic and COT data for BCF Bio-inspired Robotic Platforms (Continued).	207
A.8	Table of kinematic and COT data for MPF Bio-inspired Robotic Platforms.	208
A.9	Table of kinematic and COT data for MPF Bio-inspired Robotic Platforms (Continued).	209

A.10 Table of kinematic and COT data for MPF Bio-inspired Robotic Platforms (Continued).	210
A.11 Table of kinematic and COT data for Lift-Based Bio-inspired Robotic Platforms.	211
A.12 Table of kinematic and COT data for Jet Bio-inspired Robotic Platforms. . .	211
A.13 Legend for AUV applications. Application classification and Data for con- ventional AUVs adapted from Haroutunian [142].	214
A.14 Applications for conventional AUVs. Data from Haroutunian [142] and AUVAC.org [10].	215
A.15 Applications for BCF AUVs.	215
A.16 Applications for MPF AUVs.	215
A.17 Applications for Lift Based AUVs.	215
B.1 Experimental values to define the coefficient d_1 as a function of m from Abbot [143].	225
B.2 Abbreviation of methods, description of how the methods are implemented, and references.	226
C.1 Default SST Model coefficients implemented in OpenFOAM® [144]. . . .	235
C.2 Default $k-k_L-\omega$ coefficients implemented in OpenFOAM® [144].	239
C.3 Default Spalart-Allmaras Model coefficients implemented in OpenFOAM® [144].	241

Chapter 1

Are Fish More Efficient in Locomotion Than Engineered Systems?

Fish swimming has been of interest to scientists and engineers since Gray's paradox was introduced by Gray [145] in 1936. Gray wanted to determine how much energy a dolphin and porpoise expend during steady swimming. Gray's approach used Blasius' model for flow over a flat plate to calculate the drag over the animal [146]. Muscle power data from rowing oarsmen was generalized to the two swimmers to determine muscle output. It was found that the power expended by the animals to overcome drag would need to be approximately seven times that of the muscle power available [145]. This is known as Gray's paradox in the biological and hydrodynamics communities.

This paradox has since been reconciled by several theories and explanations. A particularly convincing explanation provided by Bale *et al.* [147]. Bale presents two main arguments to the paradox, the first being that the ratio of drag force to muscle power is highly dependent on which model is used. The group investigated this by applying three different drag models to swimming larval zebra fish. They found three different correlations to muscle power output without violating any energy conservation principles. The second argument is that the muscle energy used by the swimmer is primarily utilized for the deformation of the body. This is because the drag power is balanced by thrust power and not by muscle power. Since thrust power is a function of muscle power, this implies that drag power can be greater than muscle power without being paradoxical.

The paradox may have been reconciled, but it sparked a plethora of research into the fields of fish swimming, hydrodynamics, and robotics. The main theme around all this research is building an engineered system that is as maneuverable and meets the efficiency of fish swimming. The first output of this research was a robotic tuna developed by Barrett at the Department of Ocean Engineering, Massachusetts Institute of Technology (MIT)

[148]. Barrett's aim was to investigate an alternative form of propulsion for Autonomous Underwater Vehicles (AUVs) that was based on the kinematics of a tuna. The robot utilized a flexible streamlined body propelled by an oscillating foil. The team at MIT found that the flow past the body can be controlled to reduce drag and increase thrust [148].

Since MIT's first robotic fish, many other researchers have made AUVs based on biological analogs. AUVs in this sense are any vehicle that can swim and control its trajectory without the use of an electronic tether. There are many bio-inspired robots that are made to study the kinematics and hydrodynamics of fish swimming with few having an emphasis of providing for the shortcomings of AUVs, as was Barrett's motivation. This raises the question of why, if the locomotion mode of fish is more efficient, are there not more commercial AUVs based on fish locomotion? Another logical progression from the above question is: are current AUVs less efficient than natural swimmers, and how close have robotic platforms gotten, in terms of total energy output, to biological animals?

A possible answer to this lies in the maturity of conventional AUV design. The first AUV, Self-Propelled Underwater Research Vehicle (SPURV), was developed in 1957 at the Applied Physics Laboratory at the University of Washington. Since then, the technology has been popular for oceanic exploration, commercial, and military applications. There have been continual advancements aimed at increasing the speed and efficiency of these vehicles. Could we have designed a better engineered system than nature has demonstrated to us?

The first and only study to answer these questions was performed by: A.B. Phillips, M. Haroutunian, S.K. Man, A.J. Murphy, S.W. Boyd, J.I.R. Blake, and G. Griffiths [3]. The team collected a large data set on both biological and AUV characteristics and capabilities. They compared them all using the Cost of Transport (COT). This is a metric borrowed from biology that represents the total system energy needed to move a mass a certain distance. The most compelling finding of this research was that when comparing AUVs to biological animals, AUVs appear to be the more efficient swimmers [3, 69, 142].

1.1 Motivation

The Westwood Global Energy Group [2] has forecasted AUV demand to steadily rise over the next years as shown in Fig. 1.1. With this increase, there will be a demand for AUVs that are more efficient in terms of energy consumption and perform missions faster. This research seeks to create a unified framework that can be used to design new AUV platforms or compare current platforms to each other. This framework will also include bio-inspired platforms as well as conventional AUV platforms. There is a lack of information on the energetics (energy

consumption) and kinematics (speed) of both bio-inspired and conventional platforms so scaling laws are developed to give an estimation when no real data is provided.

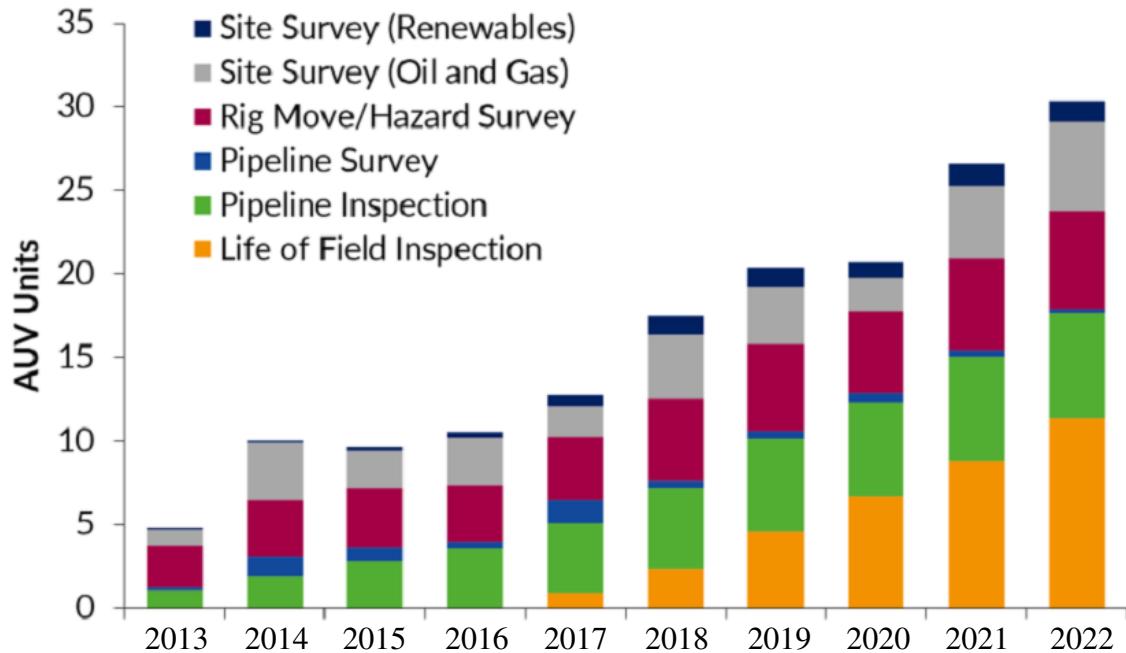


Fig. 1.1 AUV growth prior to 2022 from Westwood Global Energy Group [2].

The research presented by A.B. Phillips [3] at Newcastle University provides the basis for the scaling laws developed in this manuscript. In the authors' research, scaling laws for COT are developed and compared to biological animals. By the authors' own admission, the model used for their comparison is rather simple [3]. The only drag taken into account is that provided by the body form of the swimmer. It is not able to account for mechanisms in fish that, like Barrett's original robot, can reduce drag and increase thrust. This research would like to build upon Phillips's methodology by incorporating data from biological animals and bio-inspired robotic platforms.

The scaling methodology is extended to kinematic scaling, as introduced by Gazzola *et al.* [4]. This group's research shows that you can combine kinematic parameters for biological animals, such as undulation amplitude or tail beat frequency, to the Reynolds Number (Re). Gazzola *et al.*'s research can be further extended to bio-inspired robotic platforms and conventional AUVs in order to obtain scaling laws. These kinematic scaling laws combined with the energetic scaling laws following the research of Phillips *et al.* [3] can be used to compare AUV platforms that are of different types (i.e. conventional and bio-inspired) and can give an estimation of energy consumption and kinematic performance with no specific knowledge of a platform.

Finally, the research in this manuscript seeks to understand what design features contribute to the performance of AUV platforms. For instance, what locomotion mode performs better than others? or what type of body geometry gives the best energetic and kinematic performance? Through the use of the above scaling laws, this research further seeks to determine if performance changes depending on the physical scale (length, displacement, Reynolds Number).

1.2 Objectives

The main objective of this research is to present a unified framework for the design of AUVs with an emphasis on energy consumption, kinematic performance, and cost. To accomplish this unified framework, the scaling law methodology described in [3, 69, 142] is expanded upon with bio-inspired data as it presents a more holistic way to compare biological and engineered systems at different physical scales. Using this methodology further allows for the determination of what locomotion mode works best at different lengths, displacements, or velocities. Furthermore, utilizing these scaling laws permits for the comparison of the energetic and kinematic performance of the platforms using the same type of locomotion. This allows for the understanding of design features that contribute to one platform performing better than another.

Concerning biological and bio-inspired swimmers, there is a lack of data regarding the scaling of output power for biological locomotion modes. The second objective of this research is to fill this gap in knowledge and derive energy consumption laws for different types of BCF fish locomotion. This is accomplished through the use of Computational Fluid Dynamics (CFD) and an overset grid technique. This allows for the determination of an optimal biological locomotion mode in terms of energy consumption. A secondary objective to this is to provide a primer for a new researcher to understand how to perform CFD simulations of fish locomotion with an overset grid.

The last objective of this research is to understand the cost of building an AUV depending on the application. As with the above objectives, the cost of an AUV is formulated into a scaling law. This cost scaling must take into account the scaling of the payload for a specific mission or application. To accomplish this, a reliable and accurate way to scale the physical parameters of different platforms is needed. To accomplish this, a novel estimator of physical parameters (surface area, volume, mass) using computer vision is developed that can scale physical parameters with length. Through the use of this newly developed algorithm, an estimation of the cost of a vehicle dependent on payload size can be made.

Finally, performance can be estimated using the scaling laws for energy consumption and kinematic performance.

1.3 Structure

This thesis is structured in the following way: Chapter 2 gives a review of current literature, jargon associated with biological and natural swimming, a review of the Cost of Transport (CoT) metric, a review of kinematic scaling for biological animals, and the development of scaling models for Hotel Power. Chapter 3 describes how to use computer vision to estimate physical parameters for the physical scaling of any type of platform. Physical scaling is needed to obtain a one-to-one comparison of different platforms but also for scaling platforms depending on the type of payload needed for an application. Chapter 4 gives details on how to simulate fish locomotion using an open-source computational fluid dynamics (CFD) package, OpenFOAM®. This allows for the development of power scaling for different types of body-caudal fin (BCF) locomotion that can be applied to bio-inspired designs. Chapter 5 presents the kinematic and energetic scaling for biological, bio-inspired, and conventional AUVs. This chapter further introduces a new dimensionless number, the Propulsion number, that defines the kinematic scaling of conventional AUVs. Lastly, a first ever model is proposed for the scaling of cost for bio-inspired and conventional AUVs. Chapter 6 describes a developed framework used to design and compare different AUV platforms starting with defining the payload, then estimating the kinematic and energetic payload, and finally estimating the cost. A case study is also provided to illustrate how to apply the developed framework for different platforms with different locomotion styles.

Chapter 2

Cost of Transport and Scaling Laws

2.1 Introduction

The oceans comprise approximately 72% of the surface area of the Earth and only 20% has been mapped [1]. One of the primary tools for ocean exploration is the Autonomous Underwater Vehicle (AUV). A drawback to using this type of vehicle is the finite amount of energy that is stored onboard in the form of batteries. Lately, scientist and roboticists have been looking to nature for more efficient means of locomotion. Phillips *et al.* [3] presents data that contradicts the idea that fish are better swimmers than conventional AUVs when comparing the energetic cost of swimming in the form of the Cost of Transport (COT). The data presented by Phillips *et al.* only applies to AUVs at higher length and naval displacement (mass) scales, so the question arises of whether an AUV built at different displacements and length scales are more efficient than biological animals and if current bio-inspired platforms are better than conventional AUVs.

Besides power requirements, it is also useful to compare kinematic parameters of natural and artificial swimmers. Kinematic parameters in this case give information on how fast the swimmer travels through the water with how fast the propulsion mechanism must actuate to reach that certain swimming speed. This research adopts the approach of Gazzola *et al.* [4] where the Reynolds number is associated with a dimensionless number, Swim number (Sw) in this case, that has all the kinematic information. Sw and COT form two important metrics for the performance of an AUV, herein called performance criteria, that can be used to compare existing platforms to one other or to estimate the performance of non existent designs.

To derive the scaling laws for kinematic and energetic cost performance, data on the design and performance of 229 biological animals, 163 bio-inspired platforms, and 109 conventional AUVs is collected. While there is an abundance of data for biological swimmers,

but the data for conventional AUVs and Bio-inspired platforms is scarce. Only 5% of conventional and 38% of bio-inspired AUVs have kinematic data while 30% of conventional and 18% of bio-inspired AUVs have energetic data. The reason for the low amount of performance criteria data is that most conventional AUVs are commercial and therefore are protected as intellectual property. For bio-inspired AUVs, the performance criteria, and the COT metric in particular, has only been included as recently as 2018. For this reason, the research here formulates everything in terms of allometric scaling laws. This type of formulation is used extensively when referring to biological systems and is defined by an exponential relationship $f(x) = ax^b$, where x is a physical parameter of the fish or vehicle, like length or displacement. Scaling laws have the added benefit of allowing comparisons with limited data, as is the case for AUVs.

The following chapter reviews scaling laws for COT and kinematic performance. Here COT is decomposed into two components for engineered systems, hotel power (P_H) and propulsive power (P_P). It is found that there is a lack of literature addressing scaling of hotel power. A model for the scaling of different components of hotel power is presented and it is found that this type of energy consumption is difficult to model as a whole. The purpose of scaling the different components is to support future research covered in this thesis.

2.2 Natural Swimmers Versus Engineered Systems

Fish motion was reviewed in 1999 by Sfakiotakis *et al.* [5] and the hydrodynamics of fish compared to propeller propulsion was reviewed in 2011 by Govardhan and Arakeri [149]. The following is a brief review of fish motion and associated hydrodynamics in order to understand the nomenclature for this chapter. Fish are categorized into two different swimming modes: Body-Caudal Fin (BCF) and Median-paired Fin (MPF). To understand the reason for this naming, the general anatomy of fish is presented in Fig. 2.1.

Momentum transfer from the fish to the surrounding fluid is either done by the undulation of the entire body (BCF) or by using only fins (MPF) which leads to the classification of the two swimming modes. Each mode is further separated into sub-swimming modes depending on the body or fin motion from undulation to pure oscillation, as shown in Fig. 2.2. Not pictured in Fig. 2.2 is lift-based and jet swimming. Lift-based swimming is the locomotion mode for turtles and sea lions in which their flippers move the fluid past their bodies to create lift. Jet swimming is utilized by jellyfish and cuttlefish in which a part of the body fills with fluid and ejects it out of a nozzle for thrust.

This research focuses on BCF propulsion which includes all the top row of locomotion types in Fig. 2.2. The justification for this focus is that the majority of bio-inspired fish

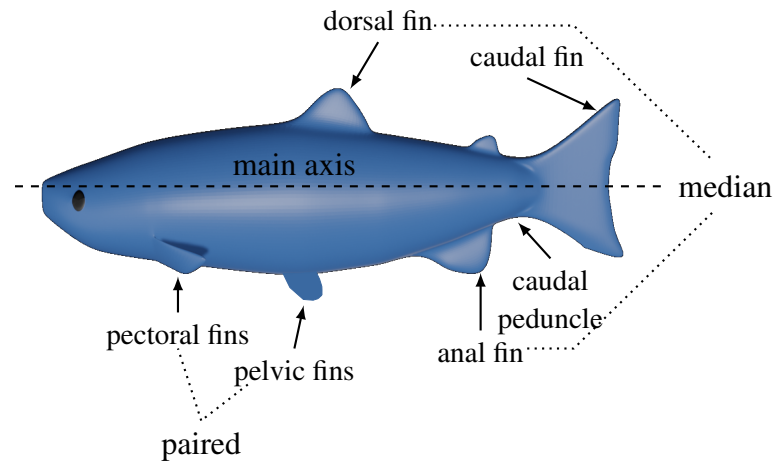


Fig. 2.1 Sketch of the anatomy of a fish showing the terminology in the text for fins and other features. Adapted from [5].

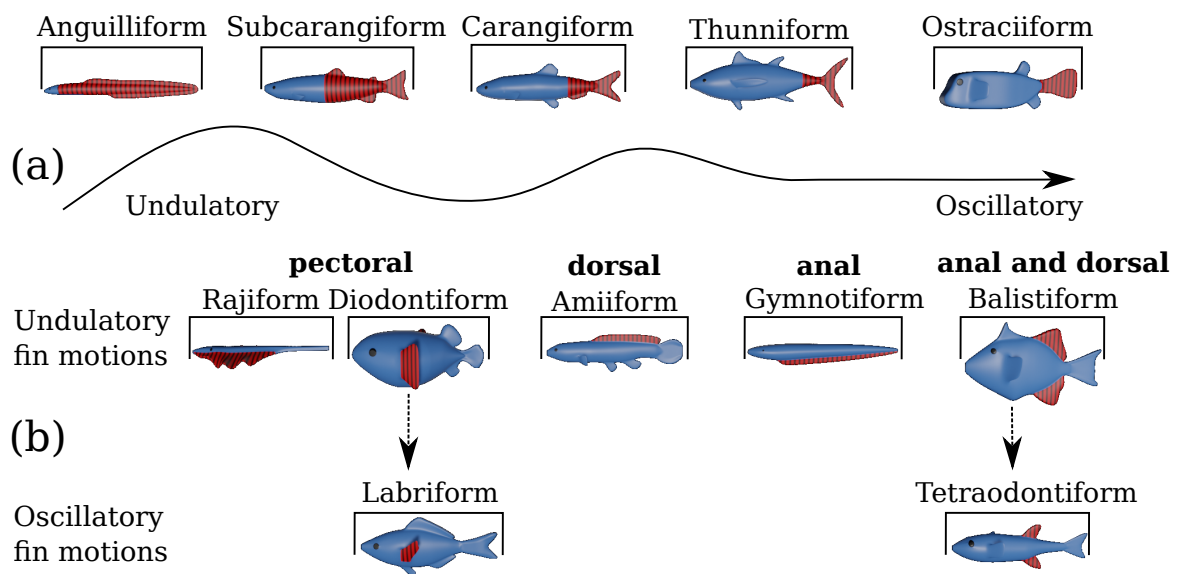


Fig. 2.2 Locomotion modes of fish Swimming. Top row (a) represents BCF swimmers and bottom two rows (b) represent MPF propulsion. Red and black shaded areas show areas that contribute to thrust generation. Adapted from [6].

Table 2.1 Number of existing AUVs. Data represents a mix of company brochure as well as academic journals. Fish modes are described in Fig. 2.2, with oscillatory referring to Rajiform locomotion but with oscillating fin motions versus undulating fin motions, such as that of a cuttlefish.

Mode	Sub Mode	Total Published
Conventional		109
	Propeller	103
	Glider	6
BCF		106
	Anguilliform	13
	Sub-Carangiform	5
	Carangiform	55
	Thunniform	15
	Ostraciiform	18
MPF		41
	Rajiform	25
	Labriform	8
	Oscillatory	4
	Gymnotiform	4
Lift Based		16

take inspiration from BCF swimmers, as shown in Table 2.1. The majority of existing MPF robots are based on Rajiform swimming, also known as Mobuliform, which represents the locomotion mode for skates and rays. This means that a majority of robotic platforms utilize undulatory locomotion instead of a simpler oscillatory mechanism. This is at odds with data that shows the more oscillatory Thunniform motion is the most efficient swim mode for biological swimmers as given by Sfakiotakis *et al.* [5]. A possible explanation is that almost all artificial fish are made for vehicle and controls research. Since undulatory motion requires the use of a more complex actuation mechanism and control scheme, undulatory robotic platforms offer a more attractive platform for controls research.

Compared to conventional propeller propulsion, body undulation propulsion differs in how thrust is produced and how wake vortices are formed in the wake. Figure 2.3 shows the vortices in the wake of both swimmers and how the shear layer forms across their bodies. Fish generate thrust both laterally (perpendicular to the body) and tangentially (in the line of propagation) to the body as they undulate. The vortices in the wake form what is known as a reverse Kármán street characterized by alternating directions of vortices. As opposed to fish motion, propeller thrust is only generated in one direction with the wake shed as helical

vortices equal to the number of blades instead of a reverse Kármán street. The detailed review of both types of propulsion is given by Govardhan and Arakeri [149].

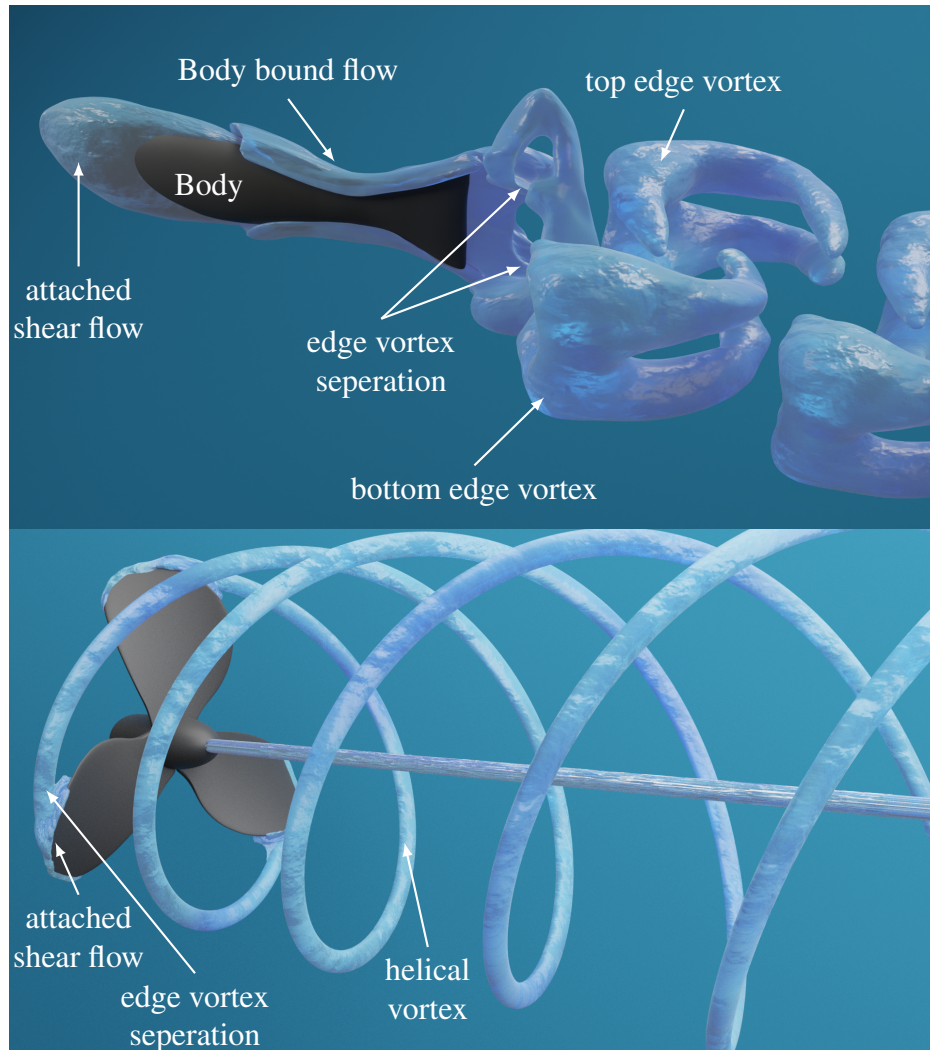


Fig. 2.3 Wake profiles for propeller (top) and fish (bottom) during locomotion.

Table 2.1 shows that conventional AUVs and BCF propulsion are similar in the number of vehicles. For bio-inspired robotic platforms, the actuation schemes to enable full body undulation are quite varied with concepts from link-driven [98], wire-driven [150], or compliant tails [30]. Roboticists have also used alternative "smart" actuators other than motors such as: Ionic polymer metal composites [151], shape-memory alloy [51], or pneumatic artificial muscles [61]. Some designers have even combined technologies by putting a traditional propeller thruster on the end of a fish tail as in the BioSwimmer [61].

In comparison to fish locomotion, conventional AUV locomotion is not as varied and the actuation scheme is less complicated and thus well established. A typical AUV is powered

by one or more conventional propeller thrusters that are powered by a motor. The hull of the AUV is typically a rigid pressure hull in contrast to a bio-inspired robot that requires some amount of the hull to be flexible for undulation. The exception to this rule is found in the glider-type movement mode. This generally uses a ballast to fill with water to become negatively buoyant. It will then perform a sinusoidal descent and ascent maneuver using this buoyancy drive. To help in this, the battery is typically moved from the front to the back or vice versa in order to keep the nose pointing in the direction of forward velocity. A schematic diagram of all the actuation modes is given in Fig. 2.4 and a breakdown of the actuation schemes for various platforms is given in Table 2.2.

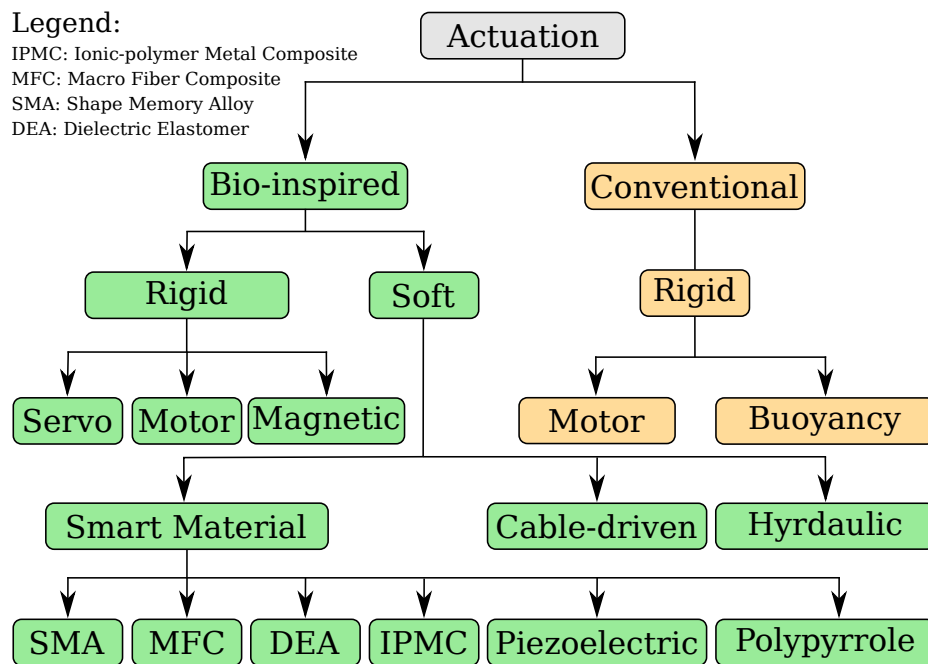


Fig. 2.4 Schematic diagram for the actuators of Autonomous Underwater Vehicles. SMA is Shape Memory Alloy, MFC is Macro Fiber Composite, DEA is Dielectric Elastomer Actuator, and IPMC is Ionic Polymer Metal Composite.

From the table, we can see that the majority of bio-inspired robots utilize a servo or motor as an actuation scheme. Similarly, the dominating actuation scheme for conventional AUVs is a motor.

Table 2.2 Breakdown of the number of actuators per each type of platform. IPMC is Ionic Polymer Metal Composite, MFC is Macro Fiber Composite, SMA is Shape Memory Alloy, DEA is Dielectric Elastomer Actuator, and EAP is Electroactive Polymer.

Mode	Sub Mode	Total Published
Conventional		109
	Propeller (Motor)	103
	Glider (Buoyancy)	6
BCF		106
	Servo	52
	Motor	21
	IPMC	10
	Piezoelectric	7
	Magnetic	5
	MFC	4
	Hydraulic	3
	SMA	3
	Polypropole	2
	DEA	1
	Live Muscle	1
MPF		40
	Servo	18
	Motor	8
	Hydraulic	3
	IPMC	3
	EAP	2
	SMA	2
	DEA	1
	MFC	1
	Piezoelectric	1
Lift Based		16
	Motor	8
	Servo	7
	SMA	1

2.3 Borrowing from Biologists: The Cost of Transport

This research is concerned with the scaling of the energy consumption of natural and engineered swimmers. In 1972, a metric that described the energy consumption of biological animals, the Cost of Transport (COT), was introduced by Schmidt-Nielsen [7]. Since its introduction, this metric has been used widely in scientific articles as the accepted way to describe animal energetics. Initially, the COT utilized the metabolic rate of the animal as a measure of the power output. This section will show how this metric is extended from biological to engineered systems. First, COT for biological animals is defined and normalized in two ways by Schmidt-Nielsen [7] and is reported as unit energy over unit mass over unit length but is often normalized over unit mass as

$$COT = \frac{AMR}{U}, \quad (2.1)$$

$$COT = \frac{AMR}{M \cdot U}, \quad (2.2)$$

where AMR is the active metabolic rate [$\text{cal kg}^{-1} \text{min}^{-1}$] or [$\text{cal kg}^{-1} \text{h}^{-1}$], cal (calorie) is a unit of energy for biological systems, M is mass [kg], and U is the speed of the animal [m/s]. Equation (2.1) is commonly used in biological texts, as (2.2) is the mass-normalized form and is more common for engineering applications. AMR is converted to Watts using a conversion factor as shown in (2.5).

AMR can further be broken down into $AMR = BMR + NMR$ where BMR is the base metabolic rate and NMR is the net metabolic rate. Phillips *et al.* [3] correlates these values into engineered systems by using the silver eel as an example. Silver eels migrate around 5,000 km without eating making them an ideal analogue to an AUV. Figure 2.5 shows a diagrammatic representation of the COT for a silver eel and a typical AUV.

Figure 2.5 shows that for biological and artificial swimmers, there is a base energy cost associated with systems that do not contribute to propulsion, as represented by the red and blue boxes. This is referred to as hotel power for AUVs and BMR for biological swimmers. Following the work done by [3, 152], BMR is replaced with hotel power P_H and NMR is replaced by propulsive power P_p . For engineered systems, P_H represents the power needed for the vehicle to remain stationary but powered on. As an example, an AUV on the surface of the water transmitting its GPS coordinates. P_p is the power needed for the vehicle to overcome fluid drag and generate thrust represented by the yellow box in Fig. 2.5. The model relating biological animals and engineered systems is given by substituting these Power terms into (2.2) giving the relationship

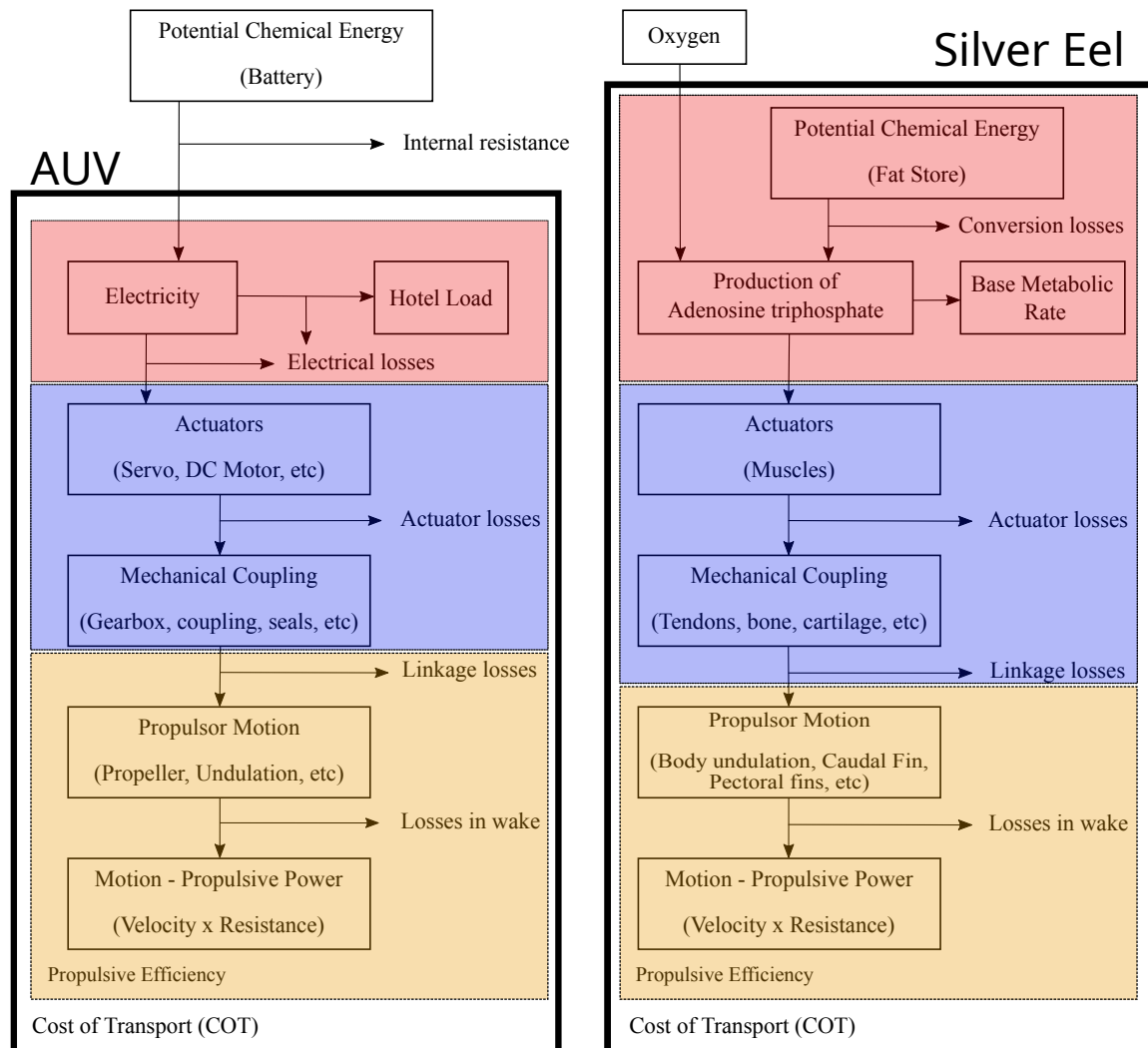


Fig. 2.5 Diagram showing the similarities, efficiency, and energy losses for an AUV and silver eel. Adapted from [3]. The top red box represents the energy generation for both swimmers. The combination of the red and blue boxes represents the hotel power, and the yellow box represents the propulsive power of the two swimmers.

$$COT = \frac{BMR + NMR}{M \cdot U} = \frac{P_H + P_P}{M \cdot U}. \quad (2.3)$$

For (2.3) to be valid, BMR and NMR must be converted to Watts. For animals, obtaining the direct AMR is evasive and difficult to measure; therefore, oxygen uptake \dot{M}_{O_2} is used and measured by a respirometer in either a circular tank, pond [153] or a water tunnel [74, 154]. \dot{M}_{O_2} for animals is generally $[\text{mgO}_2\text{kg}^{-1}\text{min}^{-1}]$, which can be converted to metabolic power ($[P_M] = W$) by assuming that all the oxygen is converted to energy, with the conversion factor from mgO_2 to J given by Elliott and Davison [155]:

$$CF = 14.14 \frac{\text{J}}{\text{mgO}_2}, \quad (2.4)$$

which gives a metabolic power of:

$$P_M = \frac{CF}{60[\text{sec}/\text{min}]} \dot{M}_{O_2} \cdot M, \quad (2.5)$$

where 60 [sec/min] is used by (2.5) to convert minutes to seconds and M is the mass of the animal [kg].

Fluid dynamicists have satisfactorily approached calculating COT separately using drag theory by calculating the amount of thrust that an animal would need to overcome the viscous drag of the fluid [156, 7, 8]. This approach has the advantage of not needing physical laboratory testing of the animal to measure \dot{M}_{O_2} at different swimming velocities. Using this model, the COT for the animal or an engineered system can be expressed as a base metabolic power (P_B), a propulsive power (P_P), a mass (M), and velocity (U) [142, 3, 69]:

$$COT = \frac{P_B + P_P}{M \cdot U} = \frac{P_B + \frac{1}{2\eta_a\eta_p}\rho C_D A_s U^3}{M \cdot U}, \quad (2.6)$$

where P_B is found by extrapolating \dot{M}_{O_2} to $U = 0$ to obtain the power at 0 velocity and converted using the conversion factor, (2.4), and η_a and η_p are the unit-less actuator and propulsive efficiency, respectively. η_a represents the efficiency of the linkages and actuation mechanisms, such as motors, shafts, and couplings. η_p represents the efficiency of the propulsion mode, such as the propeller efficiency for conventional AUVs or the flapping propulsor efficiency for bio-mimetic robots. ρ in (2.6) is the fluid density in kgm^{-3} , C_D is the unit-less drag coefficient, A_s is the wetted surface area in m^2 , and U is the free-stream velocity in ms^{-1} .

In practice, C_D is a measure of towed resistance calculated using the resistance procedure outlined by the International Towing Tank Committee (ITTC) [157]. This recommended procedure fits the drag coefficient to an empirical line using:

$$C_D = (1 + k) C_{FM} = (1 + k) \frac{0.075}{(\log_{10} Re - 2)^2}, \quad (2.7)$$

where $(1 + k)$ is the form factor, given by Hoerner [158], and Re is the Reynolds number. The form factor is based on the slenderness ratio

$$SR = \frac{D}{L}, \quad (2.8)$$

where L is the length of the object and D is the diameter of the cross section. The full form factor is given as [157, 158]:

$$FF = (1 + k) = 1 + 1.5 SR^{1.5} + 7 SR^3. \quad (2.9)$$

From inspection, COT changes significantly with the subject's physical dimensions. Specifically, the wetted surface area is directly proportional to propulsive power and the drag coefficient, (2.7), is proportional to the slenderness ratio in (2.8) [158] since $(1 + k)$ (2.9) is directly proportional to the drag coefficient. If it is assumed that the animal is neutrally buoyant, then the physical dimensions also affect its mass.

To further elucidate the importance of using accurate physical parameters when using the COT model, Fig. 2.6 shows several parameters are varied with all other parameters held constant. In all cases, P_B was kept constant at 25 Watts, while the parameters in P_P were varied in (2.6). Regions in Figure 2.6 are divided by a vertical dashed line with region 1 on the left hand side and region 2 on the right hand side. The demarcation of these regions is marked by the minimum point in the u-shaped curve.

Important to note is that P_B dominates the COT at lower velocities (region 1), while P_P is dominant at higher speeds (region 2), which gives the characteristic U-shaped curve [3]. Another important point in these plots is U_{opt} , which is defined as the velocity in which COT is minimized seen in Fig. 2.6. Varying the wetted surface area will shift the right side of the graph upwards, resulting in a higher COT in region 2. Shifting the slenderness ratio (2.8) effects the form factor (2.9) which leads to an increase in drag (2.7). This causes the COT to increase in region 2, but the effect is much less than that of the surface area. Decreasing the mass in the COT (2.6) will cause a shift in the denominator, which will cause the COT to increase throughout the entire COT curve. Finally, decreasing efficiency will increase the COT in region 2, since it is inversely proportional to propulsion power.

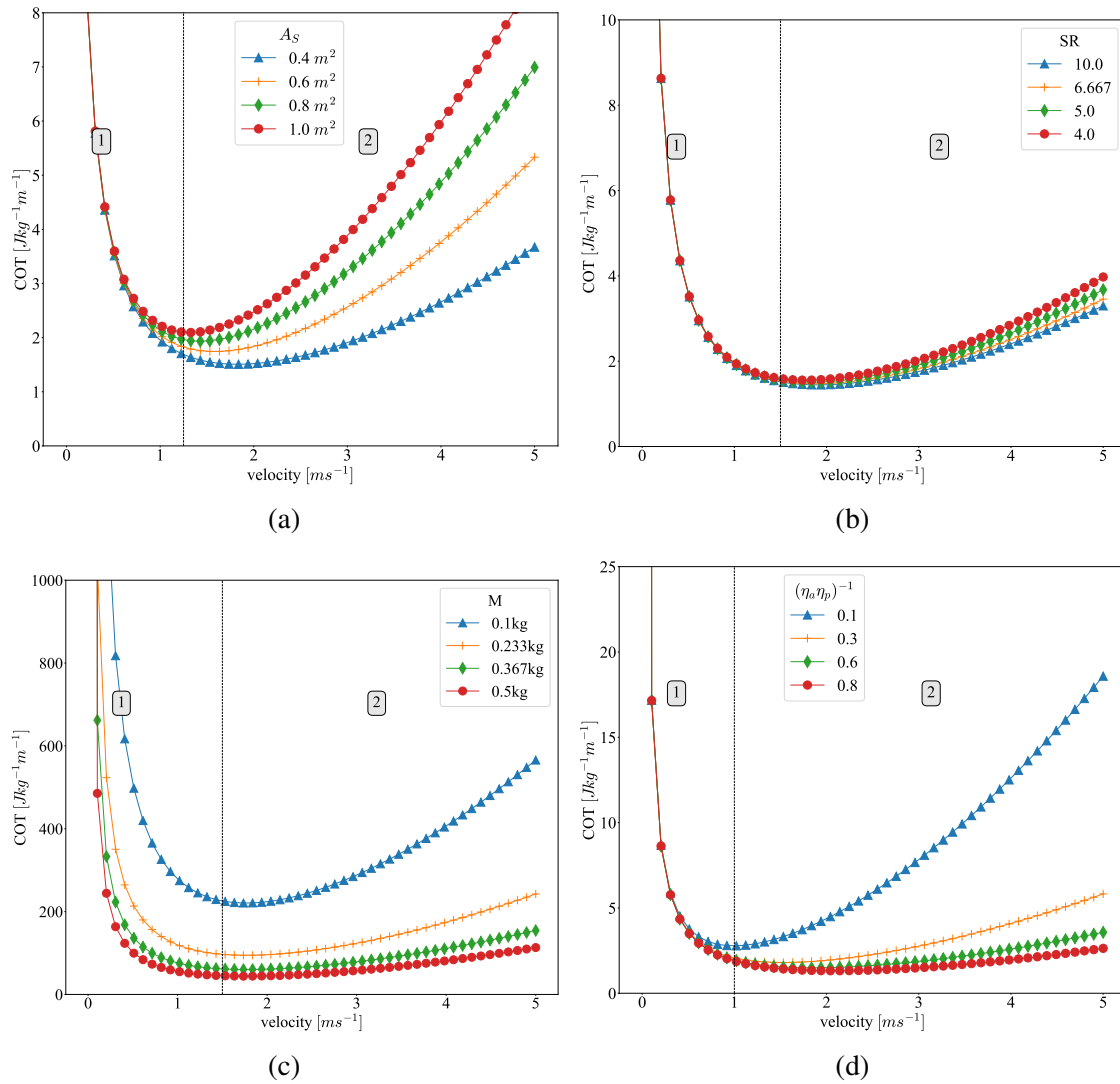


Fig. 2.6 (a) The influence on COT (2.6) of changing the (a) wetted surface area, (b) slenderness ratio, (c) mass, and (d) $1/(\eta_a \eta_p)$ with all other variables being equal. P_B in all cases is kept at a constant 25 Watts. The dashed black line represented the division between the region in which hotel power dominates at low velocities and the region where propulsion power dominates at higher velocities.

Concerning wetted surface area, data on wetted surface area is scarce for both engineered and biological systems alike. Chapter 3 describes a novel method developed for this research that uses computer vision to accurately estimate the surface area of both natural and bio-inspired swimmers. The method developed in Coe and Gutschmidt [159] is more accurate than previous methods while also being simple and easily implemented.

Total efficiency in biological animals is a combination of adenosine triphosphate (ATP) conversion and muscle efficiency [160]. Conventional propeller underwater vehicles and bio-mimetic robots have an actuator efficiency that is a combination of the actuator itself and the linkages linking the actuator to the propulsor. Propulsion efficiency is dependent on the type of propulsion employed; this encompasses body or fin undulation for biological and bio-mimetic models, and either buoyancy or propeller propulsion for conventional AUVs [91, 161–163]. A summary of the typical efficiencies (η_a , η_p) are given by Phillips *et al.* [3] and is expanded upon in Table 2.3.

Table 2.3 Table of typical efficiencies summarized in [3] and expanded on with data from a review in [136].

Actuator Type	Typical Efficiencies	Reference
Direct current motor	0.60–0.90	[44]
Pneumatic cylinders	<0.67	[164]
Dogfish red muscle	<0.51	[160]
Dogfish white muscle	<0.41	[160]
Diesel engine	<0.40	[165]
Bluegill sunfish	0.26–0.37	[166]
Electroactive polymers	<0.38	[136]
Shape memory alloys	<0.10	[167]
Nanoparticle-based	<0.01	[168]
Twisted coil polymer	0.10–10	[169]
Ionic polymer metal composites	<3.0	[170]
Dielectric elastomer	<90	[171]
Conducting polymer	<18	[172]
Piezoelectric	90	[173]
Propulsor Type	Typical Efficiencies	Reference
Buoyancy engine	<0.50	[162]
Propeller	<0.53	[174]
Biological	0.80–0.90	[175]

Utilizing the parameters of Table 2.3 as efficiencies η_a and η_p in equation 2.6 for an appropriate actuator scheme gives a sufficient estimate of the COT needed for both artificial and natural swimmers. The accuracy of this model is discussed in Section 3.2.3, when an accurate way to determine the physical parameters is developed. The comparison of this model with published data shows that the model underestimates COT by 0.2% to 20% for

bio-inspired platforms and up to 80% for the case of a killer whale. One explanation for this is that the model only considers the towed resistance of the vehicle or fish, but not other sources of drag and Barrett's original research [148, 176] discovered that fish have the ability to sense vortices in water and use them to decrease their overall drag. An earlier study by Magnuson [156] shows that form and friction drag, which the COT model (2.2) accounts for, only makes up approximately 53% of the total drag while induced drag accounts for 30% and gill resistance 17% in swimming scombridae, the Thunniform locomotion mode.

2.3.1 Another Model for COT in Engineered Systems

While COT is a model that only accounts for straight line swimming, an AUV does not just swim in a straight line during a mission. An AUV can dive, surface, and maneuver while submerged. In addition, AUVs must transmit data and broadcast their location periodically while on mission. To account for this, Tiwari *et al.* [177] presents a model for COT in their unified framework to estimate the range of unmanned robots. This model is similar to the one presented by Phillips *et al.* [3], in that they decompose the energy consumed into a propulsion energy (E_P) and hotel energy (E_H). Their research mostly focuses on ground and air based platforms, but it can be extended to AUVs. The group separates propulsive energy (E_P) and hotel energy (E_H) with propulsive energy defined in terms of mechanical energy as

$$E_P = \frac{ME}{\eta_{Man}} = \frac{\mathbf{F}_{net} \cdot \mathbf{dx}}{\eta_{Man}} = \frac{[\mathbf{F}_{r,m} + \mathbf{F}_{x,r,m} + \mathbf{F}_{t,r,m} + \mathbf{F}_{v,r,m}] \cdot \mathbf{dx}}{\eta_{Man}}, \quad (2.10)$$

where \mathbf{F} signifies the net forces on the robot, $\mathbf{F}_{r,m}$ is the constant resistive force as a function of robot, r , and mission, m . $F(x, r, m)$ is the environment dependent force based on the current position, x . Tiwari *et al.* [177] define this force as the changing gravitational potential along with changing frictional force due to changing drag coefficient. $\mathbf{F}_{t,r,m}$ is the time dependent resistive force due to disturbances. $\mathbf{F}_{v,r,m}$ is the instantaneous operational velocity resistive force that varies with instantaneous velocity, v . The duration, t , is defined as

$$t = g(x, v, D, m)$$

as a function of position x , velocity v , mission m and duty cycle D . Tiwari *et al.* [177] define the duty cycle, D , as the % driving time or proportion of net mission time that the robot was moving, usually set to 0.5. The denomination η_{Man} is the net maneuvering efficiency of the robot as a percentage.

If the assumption is made that the robot traverses at a constant velocity, but with instantaneous acceleration, and fixed duty cycle, equation (2.10) becomes

$$E_P = \frac{[F_{r,m} + F_{v,r,m}]d}{\eta_{Man}} + \frac{[\mathbf{F}_{x,r,m} + \mathbf{F}_{x,v,D,r,m}] \cdot \mathbf{dx}}{\eta_{Man}}, \quad (2.11)$$

where d is the travel distance in meters. The hotel energy, E_H , is defined as

$$E_H = \frac{P_{Anc}d}{v_{avg}D}, \quad (2.12)$$

where v_{avg} is the average velocity in m/s and P_{anc} is

$$P_{anc} = P_{sense} + P_C = (s_0 + s_1 f_s) + (P_{comp} + P_{comms}). \quad (2.13)$$

Here scalars s_0 and s_1 are static power consumption [Watts] and operational energy consumption [J] coefficients, f_s is frequency of sensing [Hz], P_{comp} is the power consumed by the computer and P_{comms} is:

$$P_{comms} = k|Data|f_{comms}, \quad (2.14)$$

where k is a constant energy/data rate coefficient, $|Data|$ is the size of data gathered, and f_{comms} is the frequency [Hz] at which communication takes place. The total energy cost for the mission (\tilde{E}) in Joules is then given by

$$\tilde{E} = E_H + E_P, \quad (2.15)$$

which is similar to the model developed by Phillips and Haroutunian [3, 142, 69]. One difference of this model is that it quantifies the gathering of data as part of hotel power / hotel energy. An important note of this point is that it depends on the scientific payload (sensors and computers) onboard the AUV. Limitations aside, this model is generally good to apply to different mission applications after a payload has been chosen.

2.3.2 Cost of Transport in Natural swimmers

There have been few studies that give allometric relationships for COT of natural swimmers. One of the first studies performed by Schmidt-Nielsen [7] in 1972 compared the COT of flying, running, and swimming animals. The original work reported COT in terms of the amount of calories that the animal consumed while in motion. This is converted to the COT units given in the previous section by multiplying by the conversion factor of 1 Calorie = 0.00418 Joules. Performing this conversion and performing a power law fit of the data over the mass of the animal, known as displacement in naval application, is shown in Fig. 2.7.

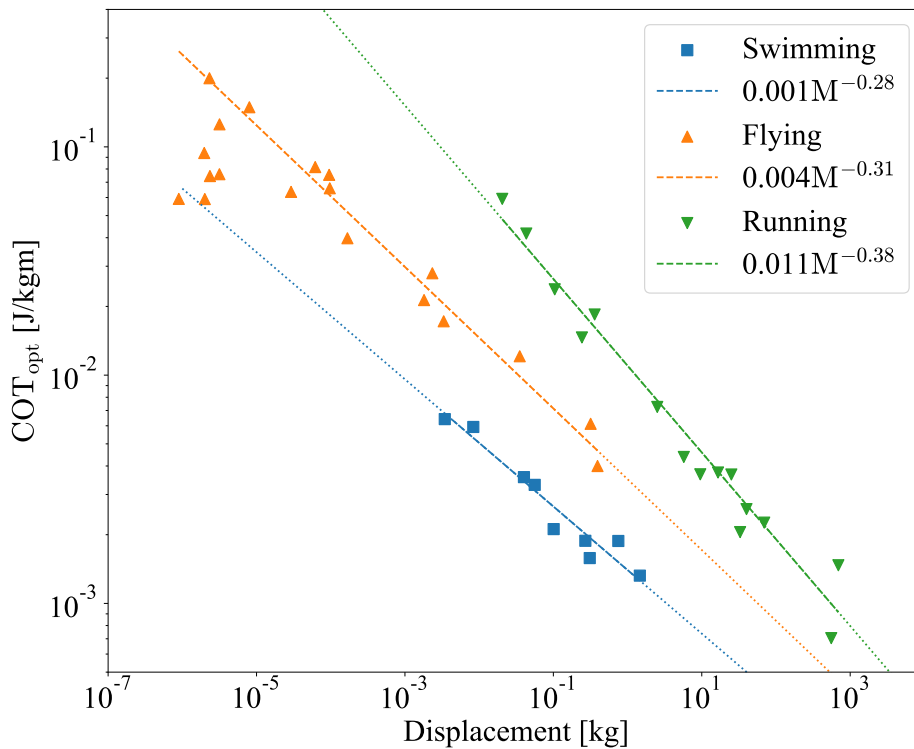


Fig. 2.7 COT versus body weight for flying, running, and swimming animals fit with an allometric power law. Original data is from Schmidt-Nielsen [7]. COT_{opt} is the COT at optimum velocity where COT is at a minimum.

The data from Schmidt-Nielsen shows that *running* scales the best out of the three locomotion modes in terms of energy consumption as body weight increases. The scaling for *swimming* shows that COT scales significantly worst than *running* animals and slightly worse than *flying* animals as indicated by the magnitude of the exponent. Videler and Nolet performed an analysis to estimate the COT at optimum swimming speed [8]. A subset of their data pertaining to submerged swimmers is given in Fig. 2.8 shows how the COT varies over the Reynolds Number and the body mass of the animal.

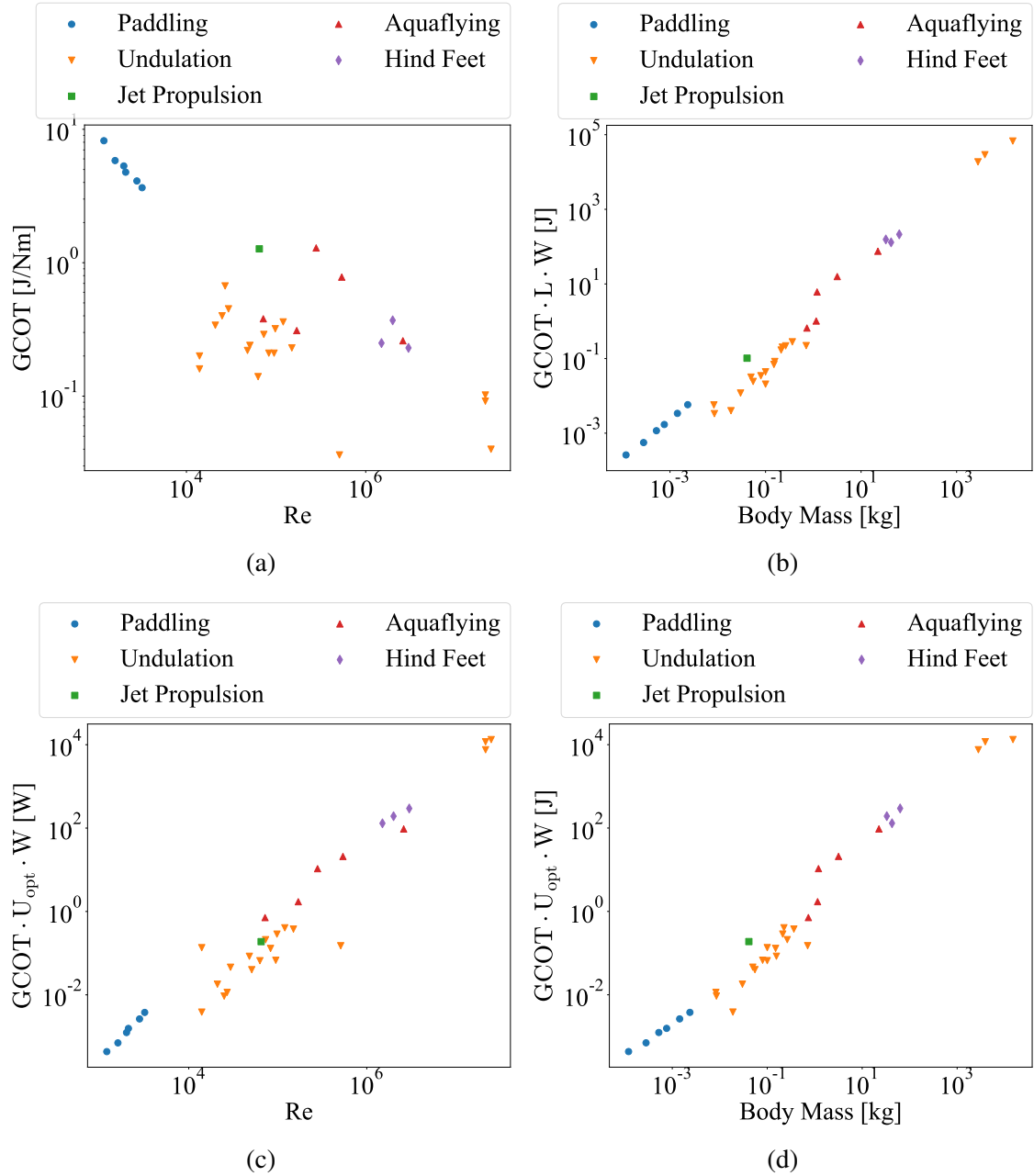


Fig. 2.8 **(a)** Dimensionless GCOT (J/Nm) versus Re number. **(b)** Energy needed to transport the body weight of a swimmer at the optimum speed over its body length related to body mass. **(c)** and **(d)** Energy needed to transport the the body weight of a swimmer at optimum speed related to Reynolds Number and body mass. Data from Videler and Nolet [8].

Videler and Nolet present two important allometric relationships for Fig. 2.8. The first is the variation of Gross Cost of Transport (GCOT), defined in [8] as:

$$GCOT = \frac{GMR}{M \cdot g \cdot u_{opt}}, \quad (2.16)$$

where GMR is AMR and calculated the same as (2.4), M is the mass of the animal, g is the gravitational acceleration constant, and u_{opt} is the optimal locomotion speed. This is multiplied by the length and weight of the animal L and W respectively. When plotted over the body mass on a log scale in Fig. 2.8b, the following relationship is observed

$$\begin{aligned} GCOT \cdot L \cdot W &= 3.3M^{1.05} \quad (J) \\ R^2 &= 0.99. \end{aligned} \quad (2.17)$$

The second allometric relationship is generalized by multiplying the GCOT by the optimum speed and body weight. In this form, shown in Figures 2.8c and 2.8d, two relationships arise from the data:

$$\begin{aligned} GCOT \cdot u_{opt} \cdot W &= 1.1Re^{1.12} \\ R^2 &= 0.91 \end{aligned} \quad (2.18)$$

$$\begin{aligned} GCOT \cdot u_{opt} \cdot W &= 3.2M^{1.01} \\ R^2 &= 0.99, \end{aligned} \quad (2.19)$$

where equation (2.18) applies to fish and turtles while equation (2.19) is for all other submerged swimmers. These results show that the energy consumption, at optimum swim speed, of fish and turtles is roughly 1.1 times body mass for fish and approximately 3.2 times body mass for other swimmers.

The decrease in energy consumption at higher Reynolds numbers, as shown in Fig. 2.8a, can be attributed to higher optimum swim speeds due to larger animals having a larger stride length. Another key conclusion from these observations is the comparison of animals with similar mass and length but different locomotion modes. For instance, Fig. 2.8b comparing the undulatory swimmer to the jet swimmer. A jet swimmer consumes roughly 4 times more energy than that of a similar sized undulatory swimmer [8]. These results show that there is variation among locomotion modes even of similarly sized animals which leads to the

question of which locomotion modes consume less energy at smaller or larger Reynolds numbers.

Another question is if different types of animals optimize their locomotion according to different lifestyles and if that effects their energy consumption. The animals presented in Videler and Nolet were classified based on their trophic levels. A trophic level is an integral count of the number of steps between the primary producers and that species [137]. The trophic level determines an animal's standing in their ecosystem's biomass pyramid and is indicative of their dietary preference. Trophic levels are divided into 5 different categories, with level 1 being the producers (plant life) and level 5 being the consumers with little natural predators (carnivorous fish and mammals). Table 2.4 gives a short description of each level and which animals fit into each level provided by FAO [137].

Table 2.4 Description of the trophic levels as defined by the Food and Agricultural Organization of the United Nations [137].

Level	Trophic class	Functional Group	Example
5	Top Predators	Piscivorous fish	dolphinfish, tuna, wahoo
4	Tertiary Consumers	Planktivorous fish	scads, herring, flyingfish
3	Secondary Consumers	Herbivorous zooplankton	predatory copepods, jellyfish, amphipods
2	Primary Consumers	Herbivorous zooplankton	Rotifers, filter-feeding copepods
1	Primary Producers	Phytoplankton	Diatoms, dinoflagellates

Figure 2.9 shows the data from Videler and Nolet categorized by trophic level. The figure shows a definite delineation between the different trophic levels. There is overlap between the trophic levels 3 and 4, but, in general, a higher trophic level corresponds to a higher COT at the same Reynolds number. The exception to this rule is animals in trophic level 2. This can be explained by the abundance of food in their environment so no need to optimize locomotion for COT. As higher trophic levels correspond to more predacious behavior, the animals are more optimized for burst and chase swimming than for steady foraging behaviours. The higher COT at the same Reynolds number may be explained by this.

2.3.3 Cost of Transport in Artificial Swimmers

One of the first studies on the COT of engineered systems was undertaken by Gabrielli and von Kármán [9] entitled "What Price Speed?" [9]. This is the first instance of the von Kármán-Gabrielli diagram which represents a comparison of the ratio between the tractive force and weight of the vehicle to speed of the vehicle as shown in Fig. 2.10.

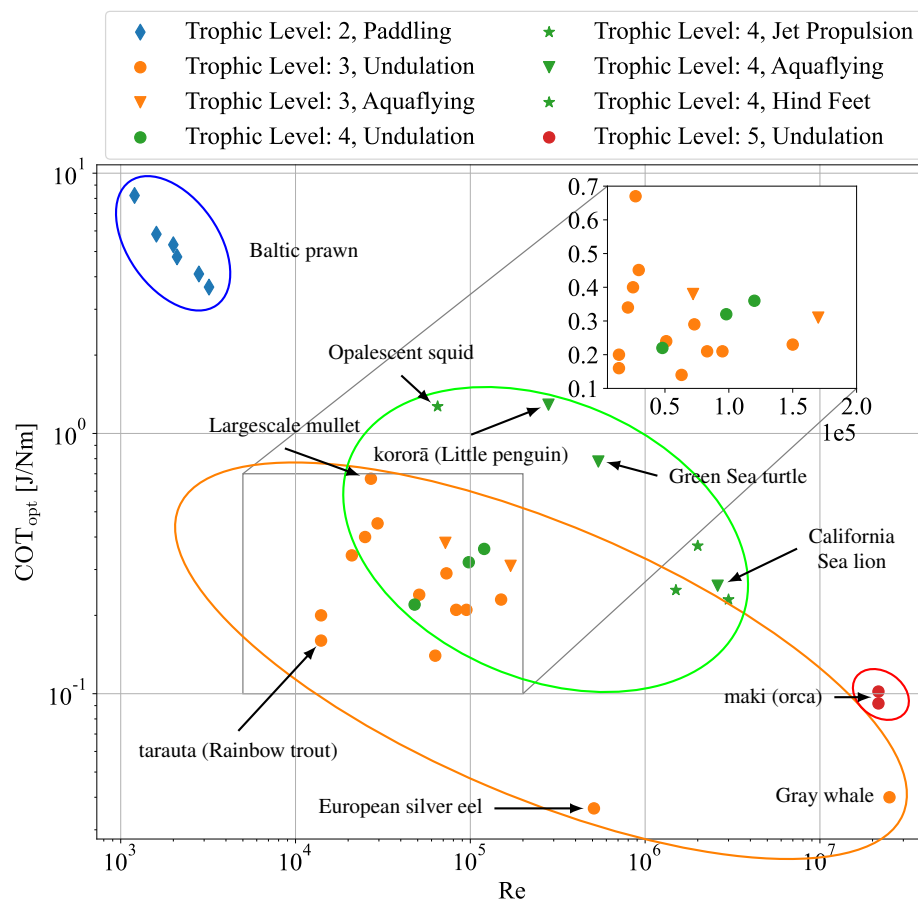


Fig. 2.9 COT for aquatic animals categorized in trophic levels as defined in Table 2.4. Ellipses cover relative area of each trophic level for this data set. COT_{opt} is the COT at U_{opt} which represents the velocity where COT is lowest.

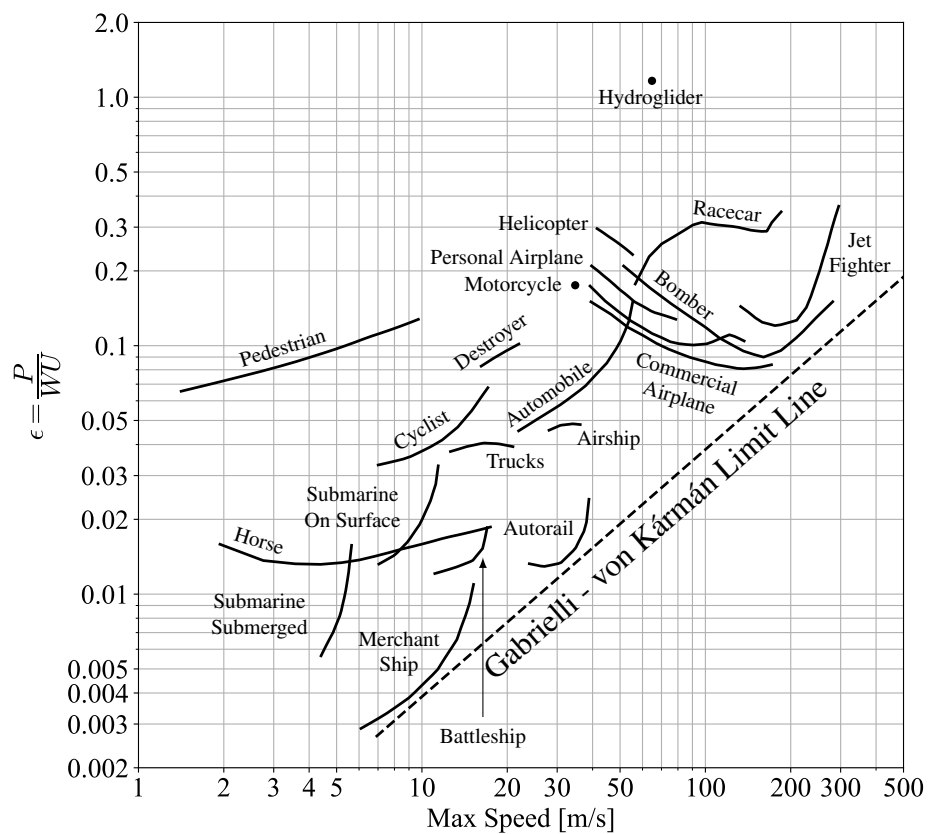


Fig. 2.10 Specific power of single vehicles versus maximum speed from before 1950. Re-drawn from [9].

Gabrielli and von Kármán [9] define the non-dimensional quantity ε as:

$$\varepsilon = \frac{P}{WU}, \quad (2.20)$$

where P is the maximum power, W is the gross weight of the vehicle, and U is the velocity. It should be noted that this specific power is equivalent to the COT metric presented in Section 2.3. Gabrielli and von Kármán's original study has since been revisited and updated over the years, most recently by Trancossi [178] and Bejan *et al.* [179, 180]. Relevant to these studies is the inclusion of payload mass into the specific power calculation. This was not done for the original study due to a lack of payload data [9].

Compared to commercial land and air vehicles, there is more limited information on the COT of AUVs. The only study more recently performed by Phillips and the team at Newcastle University [3], is significant in two ways; First, it compiled a significant amount of COT information for both natural and engineered systems [142]. Second, it presented a model in which both types of swimmers can be compared which is represented by the COT in (2.6). Their model was then applied to natural swimmers and conventional AUVs and the comparison is shown in Fig. 2.11.

The group developed scaling laws, similar to those in Section 2.3.2 for AUVs after removing the LR AUVs and Gliders. LR AUVs and Gliders were not included because, according to the author, they represent a different class of AUV. The COT of AUVs is given as

$$\text{COT}_{\text{opt}} = 1.813M^{-0.285}, \quad (2.21)$$

where M is the mass of the vehicle. This scaling is contrasted with that of Brett [74] for salmonoid fish:

$$\text{COT}_{\text{opt}} = 2.15M^{-0.25}, \quad (2.22)$$

and Williams [181] for marine mammals between 21kg and 15,000kg:

$$\text{COT}_{\text{opt}} = 7.79M^{-0.29}. \quad (2.23)$$

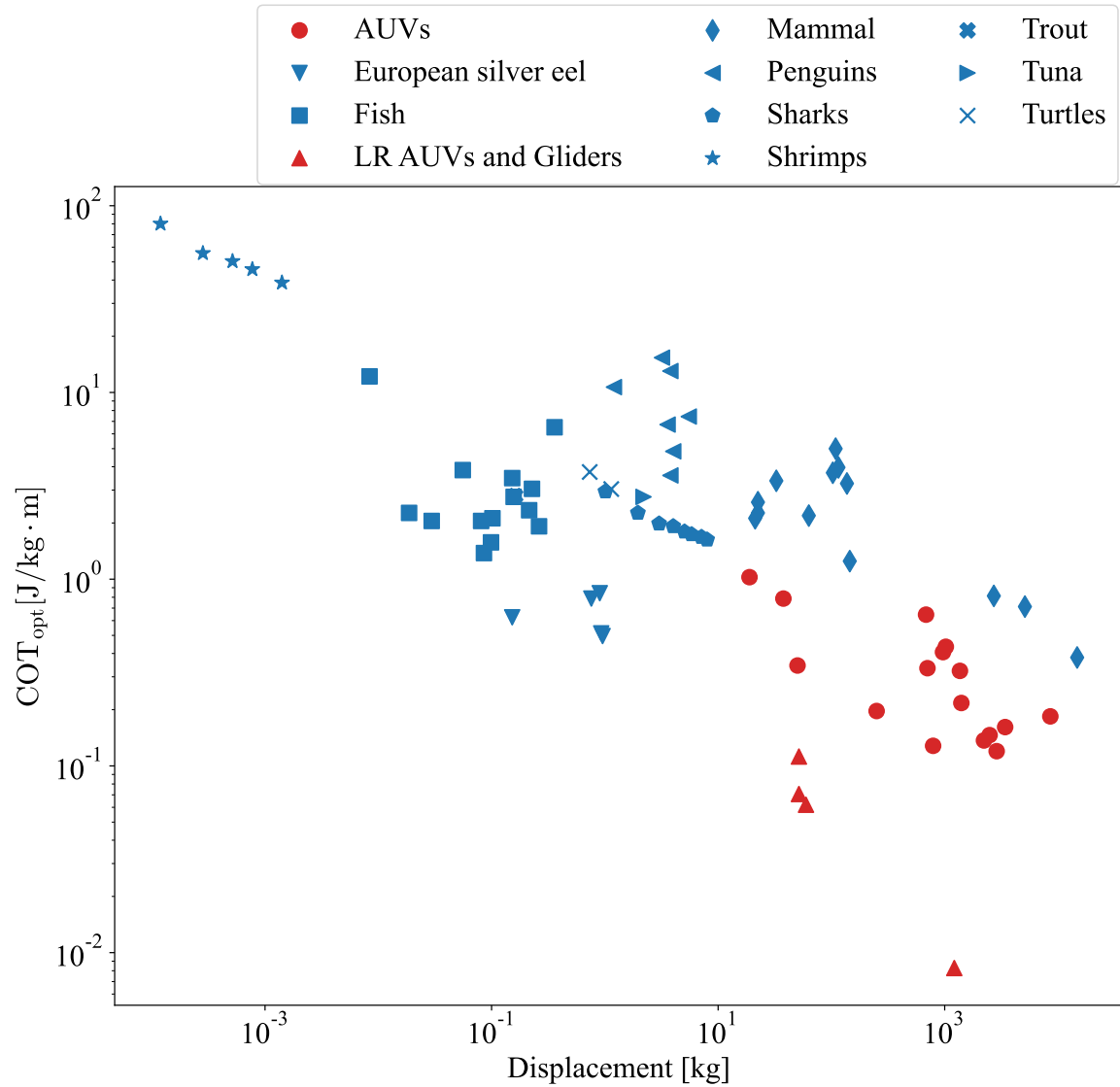


Fig. 2.11 Comparison of Cost of Transport versus displacement for natural and artificial swimmers. Data collected and collated by Phillips *et al.* [3]. COT_{opt} is the COT at U_{opt} which represents the velocity where COT is lowest.

The regression relationships show that AUVs and marine mammals are nearly identical, except for an offset. A significant finding of the groups study is that at the same displacement, AUVs have a lower COT than aquatic mammals. In addition, they have a steeper regression than that of salmonoid fish. Finally, the group found that glider type AUVs have significantly lower COT than even regular AUVs, which can be attributed to their operating mode.

The data raises the question of whether AUVs are more energy efficient than fish at high displacements. The regression lines given by (2.21) - (2.23) suggest that they are, but the intervals for the regression lines are limited and do not overlap. Furthermore, bio-inspired AUVs were not included in the original analysis. This is in part due to the research and non-commercial nature of bio-inspired platforms, but it is informative to include these platforms to understand how close engineered systems are to natural swimmers. The next part of this section expands on this original study with more data to determine the answer to this question.

In a similar methodology to Phillips and Videler, the COT of biological swimmers, conventional propeller AUVs, and bio-inspired robotic platforms is compared in Fig. 2.12. In cases where COT was not directly calculated, the current is calculated based on the battery capacity expressed in ampere hours and the endurance measured in hours. The operating voltage is then used to convert the current and voltage to a power so that the COT can be measured using the following relationship:

$$COT = \frac{P}{M \cdot U} = \frac{I \cdot V}{M \cdot U}, \quad (2.24)$$

where I is the amperage in amperes, V is the operating voltage in volts, M is the mass in kilograms, and U is the velocity in m/s. All data is fit with an allometric power law trend line except for non-carangiform bio-inspired swimmers. This is due to the limited amount of data available in published papers for these locomotion modes.

Figure 2.12 shows that conventional AUVs consume more energy as a biological swimmer with the same displacement. At higher displacements, conventional AUVs eventually overtake the biological swimmers for some designs. In general, gliders and long range AUVs require far less energy than biological swimmers for the same displacement, but, as Phillips points out, these are in a different class of AUV [3]. In general, undulation and paddling swimmers follow the same trend line with similar exponents. Interesting to note, hind feet swimmers are very similar to carangiform bio-inspired swimmers. Extending the trend lines shows that there is a cross-over point for bio-inspired carangiform robots with both conventional AUVs and undulating biological swimmers at higher displacements (masses).

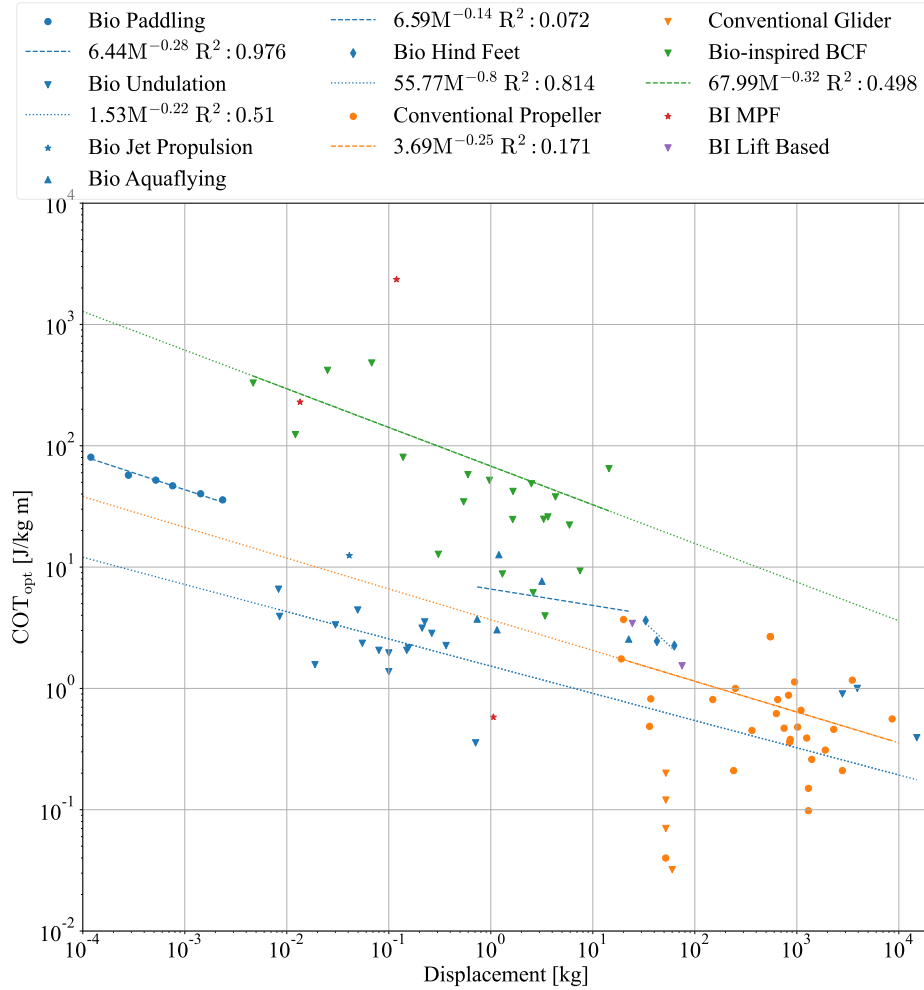


Fig. 2.12 COT of biological and artificial swimmers. Biological data is from Videler and Nolet [8]. Conventional Propeller and Glider information is taken from [10–15]. BI Carangiform data is from [16–27]. BI Ostraciiform data is from [28, 29]. BI Thunniform data is from [30–32]. BI Anguilliform data is from [33–35]. BI Sub-Carangiform data is from [36]. MPF data is from [37–39]. Lift based data is from [40, 41]. Best fit allometric trend line for power law fit is shown in legend. COT_{opt} is the COT at U_{opt} which represents the velocity where COT is lowest.

As Videler and Nolet have shown, the data is further represented as $COT \cdot W \cdot L$ to obtain a raw energy value and shown in Fig. 2.13. The correlation between the data and the trend lines is more accurate in this format. There is a clear cross-over point between the propeller driven AUVs and data for undulating fish at $\sim 3 \cdot 10^3$ kg displacement. Interestingly, there is also a cross-over point between the bio-inspired BCF robots and undulating swimmers at $\sim 6 \cdot 10^5$, not shown in figure for clarity. For reference, this is larger than an average Blue whale ($\sim 1 \cdot 10^5$ kg) and smaller than a Los Angeles class attack submarine ($\sim 6.08 \cdot 10^6$ kg).

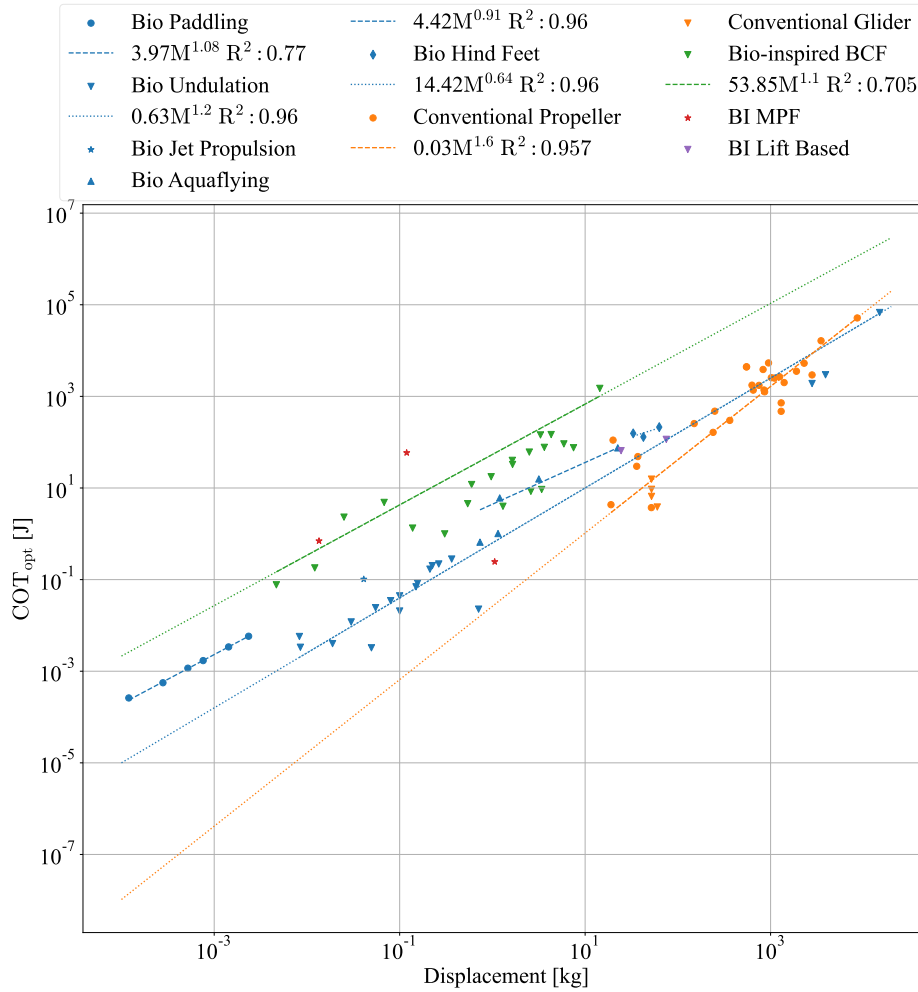


Fig. 2.13 COT of biological and artificial swimmers. Biological data is from Videler and Nolet [8]. Conventional Propeller and Glider information is taken from [10–15]. BI Carangiform data is from [16–27]. BI Ostraciiform data is from [28, 29]. BI Thunniform data is from [30–32]. BI Anguilliform data is from [33–35]. BI Sub-Carangiform data is from [36]. MPF data is from [37–39]. Lift based data is from [40, 41]. Best fit allometric trend line for power law fit is shown in legend. ε is the COT at U_{opt} which represents the velocity where COT is lowest.

The methodology of Gabrielli and von Kármán [9] is extended in Fig. 2.14 to illustrate the technological limits for current natural and artificial swimmers. The figure shows that AUVs have a lower technological limit than biological fish and that bio-inspired BCF platforms have the highest technological limit of the three categories. A simple explanation for this trend is given by Haroutunian [142] in that biological swimmers are optimized not just for efficient transportation whereas the emphasis for commercial AUVs is on extending mission times. This can be somewhat shown by the argument of different types of animals fitting into different trophic levels as discussed in Section 2.3.2 relating to Fig. 2.9.

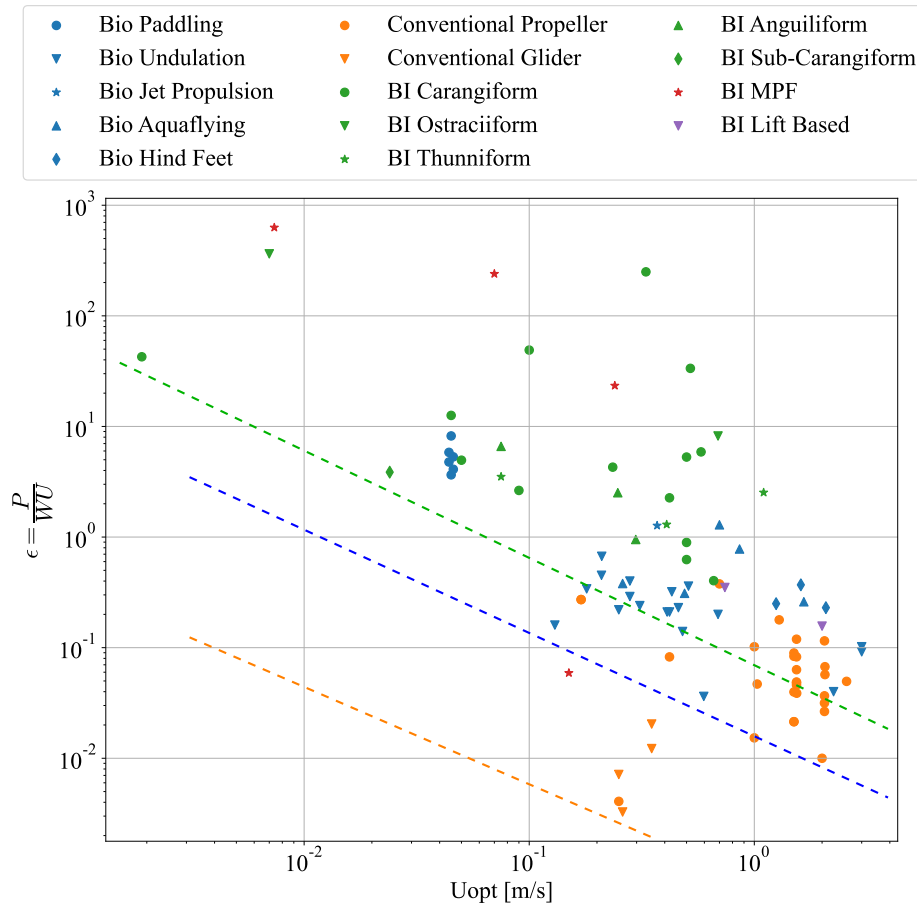


Fig. 2.14 Extension of the von Kármán-Gabrielli methodology to AUVs [9]. ϵ is the COT at U_{opt} which represents the velocity where COT is lowest.

2.4 Influence of Kinematic and Power Scaling

2.4.1 Kinematic Scaling

Another critical aspect of designing and building an artificial swimmer is its underlying kinematics. The kinematic parameters determine the speed of the actuators of a swimmer. The scaling factor relates this actuator speed to the Reynolds number of the swimmer. The Reynolds number provides the ratio of inertial to viscous forces for the fluid. It can also provide a characteristic length for the swimmer assuming a constant speed or provides a velocity speed when fixing the vehicle to a certain characteristic length. This is how the kinematic scaling of the swimmer relates velocity and length to actuation. The first such scaling study was done by Gazzola *et al.* [4]. The group introduced a new dimensionless parameter called the Swim number defined as:

$$Sw = \frac{\omega AL}{\nu}, \quad (2.25)$$

where ω is the frequency of undulation or flapping in rad/s, A is the tail beat amplitude peak-to-peak, L is the length of the animal, and ν is the kinematic viscosity of the fluid. The group refers to this as a transverse Reynolds number, which describes the relevant kinematics of the swimmer. This number is derived by multiplying the axial Reynolds number by the Strouhal number of the form $St = \omega A/U$ to get

$$Sw = Re \cdot St = \frac{UL}{\nu} \cdot \frac{\omega A}{U}, \quad (2.26)$$

where U is the velocity of the animal. It should be noted that the normal form for the Strouhal number is $St = fA/U$ and is converted via the well known relationship of $\omega = 2\pi f$.

The group performed a meta analysis on 3000 animals and found the following very simple scaling relationship between Re and Sw :

$$Re = \begin{cases} 0.03Sw^{1.31}, & Sw < 3 \cdot 10^3, \\ 0.4Sw^{1.02}, & Sw > 3 \cdot 10^3, \end{cases} \quad (2.27)$$

where $3 \cdot 10^3$ represents the critical transition point between laminar and turbulent regimes. As Fig. 2.15 shows, this scaling relationship holds true for a variety of swimming and flying animals at 6 decades of Reynolds scales.

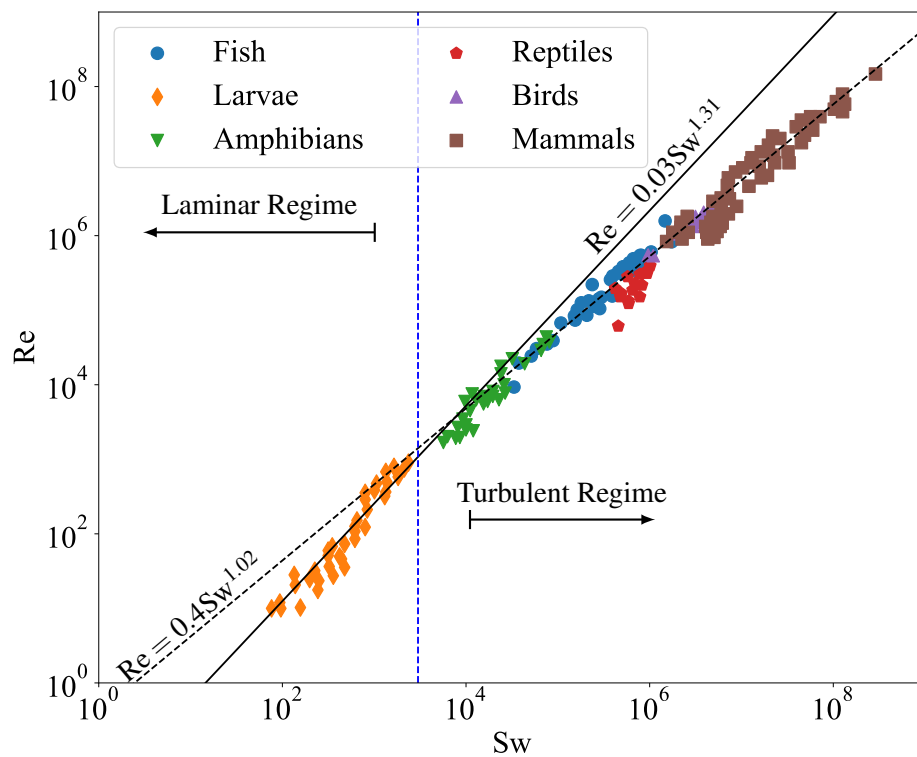


Fig. 2.15 Swim vs. Reynolds number for a collection of swimming and flying animals. Data for fish taken from Videler [42] and all other data from Gazzola *et al.* [4]. Figure adapted from Gazzola *et al* [4].

This scaling relationship is significant because it allows for the estimation of kinematic parameters of swimmers. For instance, if you have a swimmer of link L , swimming at velocity U , the frequency of undulation or flapping and amplitude (ωA) can be found by an algebraic arrangement of the scaling relationship.

A second significant finding of this group is that 2D and 3D simulations of fish swimming scale similarly when the same scaling procedure is applied, as shown in Figure 2.16. The group simulated both 2D and 3D fish swimming and determined the kinematics at self propulsion, when drag and thrust are equal ($C_D = C_T$). Applying their scaling relationship to their data and other 3D simulation data shows that both the 2D and 3D simulations scale with the same exponent but are offset by a constant. This finding implies that 2D simulations can be used in lieu of 3D simulations to compare the kinematic parameters of swimming fish. It also shows that 2D simulations can accurately solve for the swimming kinematics of the natural and artificial swimmers. This is significant because drag and thrust are based on the forces of the fluid on the body during locomotion. In general, 2D hydrodynamic simulations don't capture 3D effects such as vortices shed from the tips of fins. This data shows that, for the purposes of their scaling procedure, these forces can be neglected.

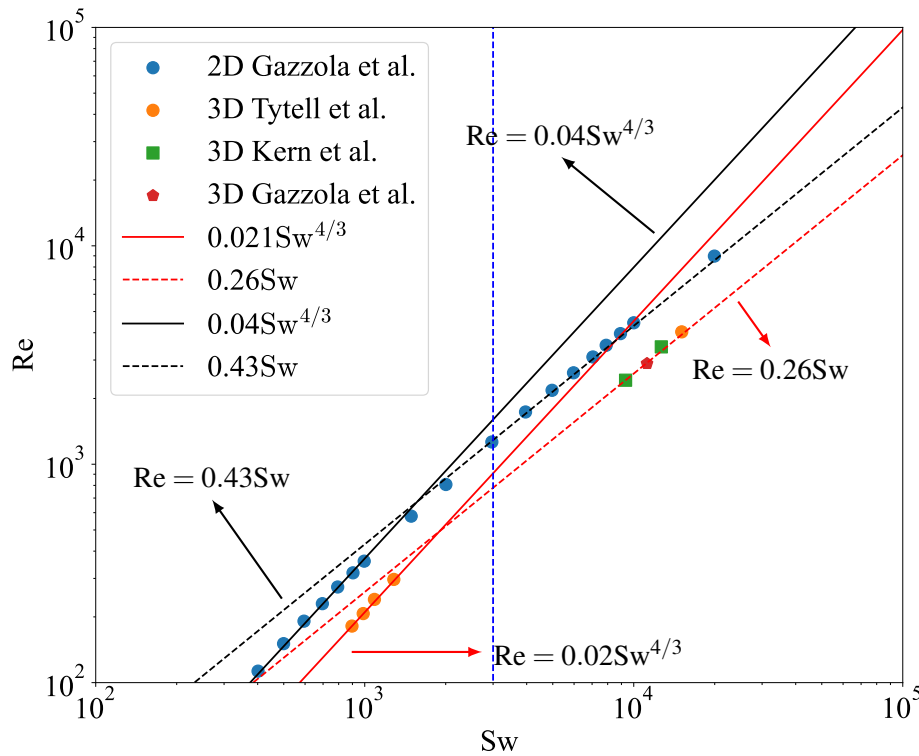


Fig. 2.16 Swim vs. Reynolds number for 2D and 3D simulations. The 2D and 3D simulations scale with the same exponent in both the laminar and turbulent regimes. Adapted from Gazzola *et al* [4].

Collected data allows for the estimation of COT and kinematic parameters for natural swimmers over several decades of Reynolds numbers. Data for artificial swimmers such as AUVs and bio-inspired platforms are limited to only a narrow range of Reynolds numbers. This research seeks to develop a model so that kinematic parameters and COT can be estimated for these platforms over a broader range of Reynolds numbers. In this way, biological and artificial platforms can be compared based on kinematic parameters and COT, or these parameters can be estimated if there is no existing platform. Scaling further allows for the determination of platforms that perform better than others. This will allow for the comparison of design features which may elucidate what makes a superior performing design over a poor performing one.

2.5 Power Scaling Laws

The first work on propulsion scaling laws for aquatic propulsion was performed by Gray in 1936 [145]. Gray used Blasius' model for flow over a flat plate to determine the drag over the animal [146] and the muscle power measured from human oarsmen to determine the amount of energy needed for dolphins to swim. This persisted for 24 years until Lighthill released his mathematical model for slender fish [182]. This model was revised in 1970 [183] and became known as Lighthill's Elongated Body Theory (LEBT) in 1971 [184].

LEBT says that the mean thrust of the tail's motion, $\langle T \rangle$, is related to the local motion of the swimmer's midline, $h(x, t)$, where x is the local coordinate down the body by the following equation:

$$\langle T \rangle = \left[\frac{m}{2} \left(\left(\frac{\partial h(x, t)}{\partial t} \right)^2 - U^2 \left(\frac{\partial h(x, t)}{\partial x} \right)^2 \right) \right]_{x=L}, \quad (2.28)$$

where L is the total length of the tail and the $\langle \cdot \rangle$ indicates the time averaged value over a tail beat cycle. m is the added mass at the tail tip given by

$$m = \frac{\pi \rho_w s^2}{4} \beta, \quad (2.29)$$

where ρ_w is the density of the fluid, s is the width of the tail, and β is the added mass coefficient, which is a function of the fish body cross-section as described in [183]. $h(x, t)$ is the motion of the fish body down the midline in the x -direction with time t . As discussed by Wu [185], employing a simple swimming energy conservation, the instantaneous power is

$$\langle P \rangle = \langle E \rangle + \langle T \rangle U, \quad (2.30)$$

where $\langle T \rangle$ is described in (2.28), $\langle P \rangle$ is given as

$$\langle P \rangle = mU \left[\frac{\partial h(x,t)}{\partial t} \left(\frac{\partial h(x,t)}{\partial t} + U \frac{\partial h(x,t)}{\partial x} \right) \right]_{x=L}, \quad (2.31)$$

and $\langle E \rangle$ is

$$\langle E \rangle = \frac{m}{2} U \left[\left(\frac{\partial h(x,t)}{\partial t} + U \frac{\partial h(x,t)}{\partial x} \right)^2 \right]_{x=L}. \quad (2.32)$$

LEBT gives the scaling of thrust, power, and energy with Reynolds number in the form of velocity, kinematic parameters in the form of the local coordinate body velocity, and body shape with the inclusion of β in m . Interestingly, this model is still used as a standard today even though the use of particle image velocimetry shows that it overestimates power consumption by up to 30% in some cases [186].

More recently, Yu *et al.* [43] performed 2D simulations at different Reynolds numbers of anguilliform and carangiform swimmers. The group derived an average thrust scaling relationship given as

$$\langle C_T \rangle = \xi_\lambda \left(1 - \frac{d_1}{\sqrt{Re}} \right) St^2 \left(1 - \frac{U}{c} \right) - \frac{d_2}{\sqrt{Re}} F_B, \quad (2.33)$$

where the angle brackets $\langle \cdot \rangle$ signifies the time averaged value. ξ_λ is the body slope factor which is different for anguilliform and carangiform swimmers.

$$\xi_\lambda = \begin{cases} \beta_1 \left(1 - \exp^{-\beta_2(\lambda - \lambda_0)/L} \right), & \text{anguilliform,} \\ \alpha_1 \left(\frac{\lambda}{L} \right)^2 + \alpha_2 \left(\frac{\lambda}{L} \right) + \alpha_3, & \text{carangiform,} \end{cases} \quad (2.34)$$

for $\alpha_1 = 3.13$, $\alpha_2 = -4.82$, and $\alpha_3 = 3.32$. $\beta_1 = 2.25$, $\beta_2 = 2.29$, and $\lambda_0/L = 0.34$. λ is the body wavelength, λ_0 is obtained using data fitting, and L is the body length of the swimmer. From equation (2.33), d_1 , and d_2 are different for anguilliform and carangiform swimmers:

$$\begin{cases} d_1 = 11.2, & d_2 = 3.94, & \text{anguilliform,} \\ d_1 = 5.6, & d_2 = 3.87, & \text{carangiform.} \end{cases} \quad (2.35)$$

St is the strouhal number

$$St = \frac{2Af}{U}, \quad (2.36)$$

where f is the frequency of tail, A is the amplitude of the tail, and U is the swimming velocity. c is the slip velocity which is the speed in which the tail undulates back and forth defined as $c = f\lambda$. The body form, F_B , in (2.33) is defined by Hoerner [158]:

$$F_B = 1 + 1.5\sigma^{1.5} + 7\sigma^3, \quad (2.37)$$

where σ is again different for the anguilliform and carangiform swimmers:

$$\sigma \approx \begin{cases} 2A/L, & \text{anguilliform,} \\ D/L, & \text{carangiform,} \end{cases} \quad (2.38)$$

where A is the amplitude as in (2.36), and D is the width of the swimmer.

The propulsion power is given by:

$$P_p = \langle C_T \rangle \frac{1}{2} \rho A_s U^3, \quad (2.39)$$

where ρ is the fluid density, A_s is the wetted surface area, and U is the speed of the swimmer. This model can be compared to experimental data of a swimming robot. Here the University of Virginia Tunabot developed by Zhu *et al.* [30] is chosen due to the availability of kinematic, thrust, and COT data for a range of Reynolds numbers. Furthermore, the hydrodynamics for this platform are similar to the biological analog as verified by particle image velocimetry. For this comparison, the carangiform model was compared with the ITTC method outlined in Section 2.3 and experimental results as shown in Fig. 2.17.

The model shows a negative thrust at lower Reynolds values which is caused by the second term in (2.33) dominating in this region which implies that body drag is dominating

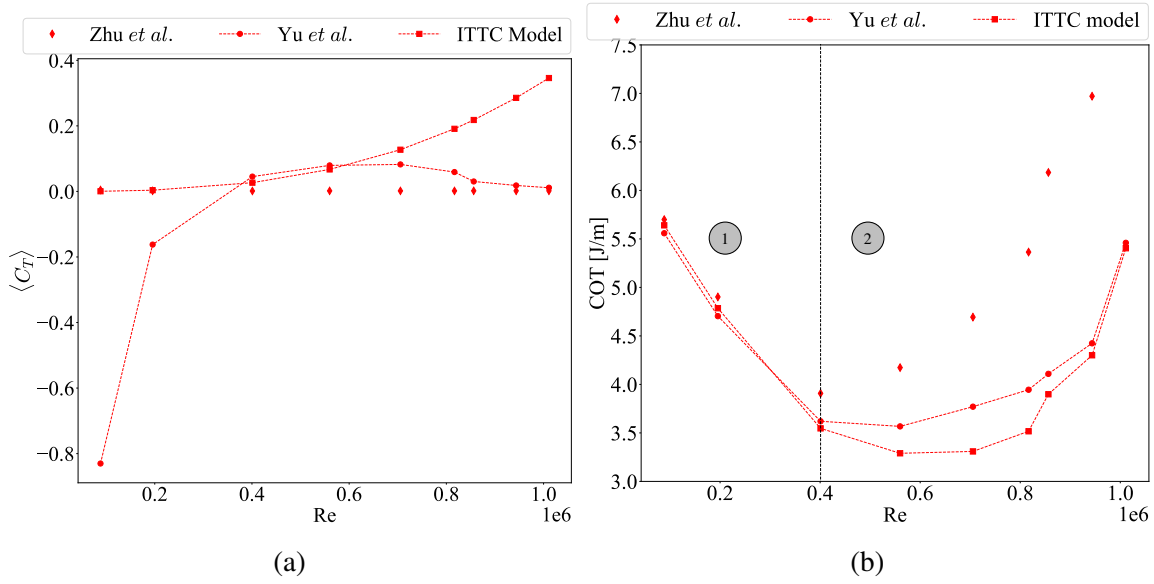


Fig. 2.17 (a) $\langle C_T \rangle$ and (b) COT versus Re for the University of Virginia Tunabot using the propulsion scaling model of Yu *et al.* [43]. Numbers in (b) represent the different regions of the Cost of Transport model. 1 is the region where hotel power is the dominate energy consumption of the vehicle and 2 is the region where propulsion power is the dominate energy sink.

in this region. Comparing this region to the same region in the COT plot, Fig. 2.17b, shows that this region corresponds to the region where hotel power is the dominant energy usage for the vehicle.

Region 2 of Fig. 2.17b shows the comparison between (2.33) and available experimental results. The comparison shows that the model underestimates the actual data but is closer to the actual data than the ITTC model in the intermediate Reynolds numbers. This indicates that the Yu *et al.* model captures extra hydrodynamic forces than the simple drag model presented by the ITTC. Neither model accurately estimates the experimental data, but as the Yu *et al.* model captures hydrodynamic forces for different locomotion modes, it can be used for a relative comparison between different swimming styles.

2.6 Hotel Power

It was shown in Section 2.3, that hotel power is the predominate energy consumption at lower velocities. It was also shown that the total output power can be decomposed into hotel power and propulsive power. Section 2.5 shows the scaling laws for propulsive power and it would be good to know the scaling of hotel power in order to get the entire picture of how

COT scales. To the author's knowledge, there has been no research into the scaling of hotel power for AUVs. This type of analysis is quite hard because hotel power depends on control systems, instrumentation, and scientific payload. In contrast, biological animals typically follow the same internal structures with similar density tissue. The following sections give a review of hotel power or BMR for biological animals and a first estimate of hotel power for AUVs given.

2.6.1 Hotel Power (BMR) For Biological Animals

As outlined in Section 2.3, hotel power is synonymous with BMR for biological animals. The conversion from BMR to a power in watts is given by

$$P_H = \frac{BMR \cdot CF \cdot M}{3600[\text{sec/h}]}, \quad (2.40)$$

where BMR is the base metabolic rate in $[mg\ O_2/kg \cdot h]$, CF is the conversion factor given in (2.4) $[J/mgO_2]$, M is the mass $[kg]$, and $3600\ [sec/h]$ is the conversion from hours to seconds. Figure 2.18 shows the hotel power for some biological animals and AUVs fit with a power law to give an allometric scaling equation.

For a meta analysis of BMR, the first well known study was done by Kleiber [187] and became known as Kleiber's law. Kleiber argues that metabolism can be formulated as:

$$BMR = C \cdot W^n [1 + \alpha(A - a) + \phi(s - S) + \dots], \quad (2.41)$$

where BMR is the basal metabolism, C is the coefficient of species and sex, W is the body weight, n is the exponent $2/3$ or $3/4$, a is the coefficient of age, A is the standard age, α is the actual age, ϕ is the coefficient of build, S is the standard specific stature, and s is the actual specific stature. Most researchers only include C , W , and n while disregarding everything in the brackets. This gives an allometric relationship for BMR to body mass that scales with exponent $2/3$ or $3/4$.

Work done by Eckert [188] shows that metabolic rates for various animals from unicellular to large endotherms (whales) scale with an exponent between 0.65 and 0.80. More recent work by Ballesteros *et al.* [138] gave a meta-analysis of both plants and animals to derive allometric relationships for BMR in $[kJ/h]$. Table 2.5 shows the allometric scaling laws developed by Ballesteros *et al.* The table indicates that most BMR scales near the original $3/4$

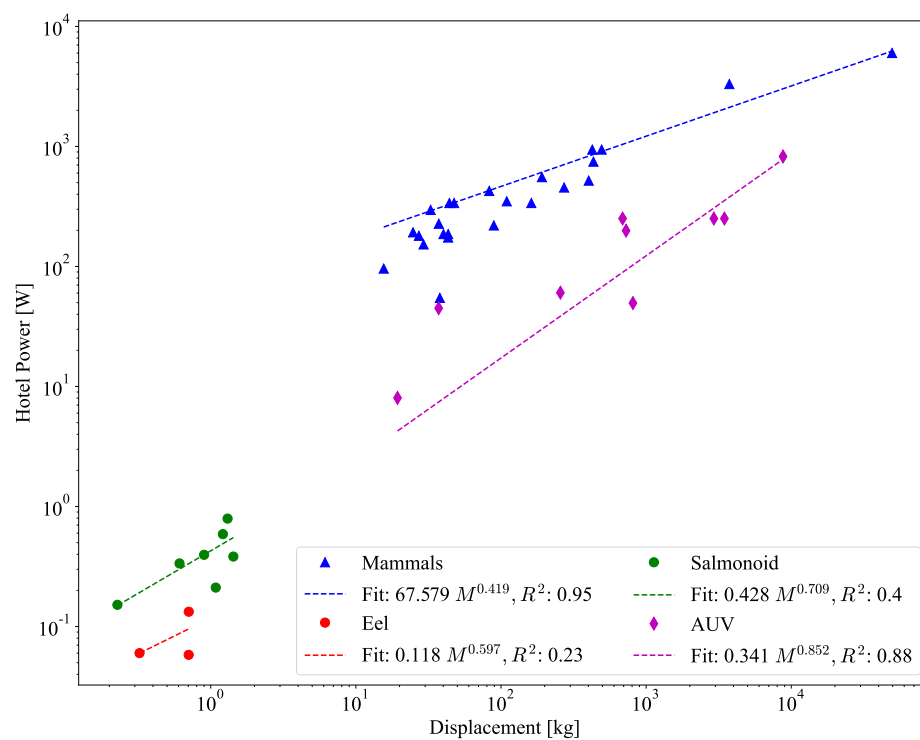


Fig. 2.18 Data collected in [3] and fit with a power-law curve to obtain the hotel power scaling for biological animals. Displacement in this case is the mass of the swimmer, assuming neutrally buoyant.

Table 2.5 Allometric scaling laws for BMR from meta analysis given by Table 1 in Ballestoros *et al.* [138]. The model used for all data reported is of the form $BMR = aM^k$ where M is the body mass of the animals. N is the number of animals in each data correlation.

Database	a	k	R^2
Mammals (all, N=637)	0.0692	0.72	0.952
Polar mammals (N=14)	0.1326	0.6928	0.988
Desert mammals (N=99)	0.0556	0.7393	0.969
Polar and Desert mammals (N=113)	0.0569	0.7468	0.969
Plants (N=89)	0.0053	0.81	0.957
Flying Birds (N=510)	0.143	0.657	0.884
Flightless Birds (all, N=22)	0.062	0.744	0.902
Flightless Birds (all, N=22)	0.062	0.744	0.902
Flightless Birds (without outliers, N=20)	0.041	0.805	0.986
Insects	0.007	0.832	0.604

rule that Kleiber derived but varies in exponent interval of 0.65 and 0.84. It also demonstrates that the slope coefficient varies significantly for each type of animal. Sw

Besides the analysis shown in Fig. 2.18, there have been no studies on allometric scaling for the hotel power of AUVs. There are various general robotic energy models, but the research in this area is relatively new and small [177]. In the following subsections, I present some scaling analysis and a model that can be used as an estimate of hotel power for AUVs with an emphasis on bio-inspired platforms. Referring back to Fig. 2.5, all component contributions to hotel power are shown in the non-shaded region inside the box. The figure shows a majority of hotel power is actuators and mechanical coupling. Other sources of hotel power include control systems, sensors, and scientific instruments. The following section gives a simple allometric scaling model for the most common actuation schemes for engineered systems.

Motors and Servo Scaling

Motor data from Maxon Motors [44] and servo data from Hitec RCD USA [45] were collected for a range of different sizes. Power draw was calculated using the nominal voltage and current as posted on the manufacturer specification sheet and using (2.24). Similar to COT and other scaling laws, all data is fit with a power law fit. Figure 2.19 shows the power and torque for motors and servos as they increase in mass.

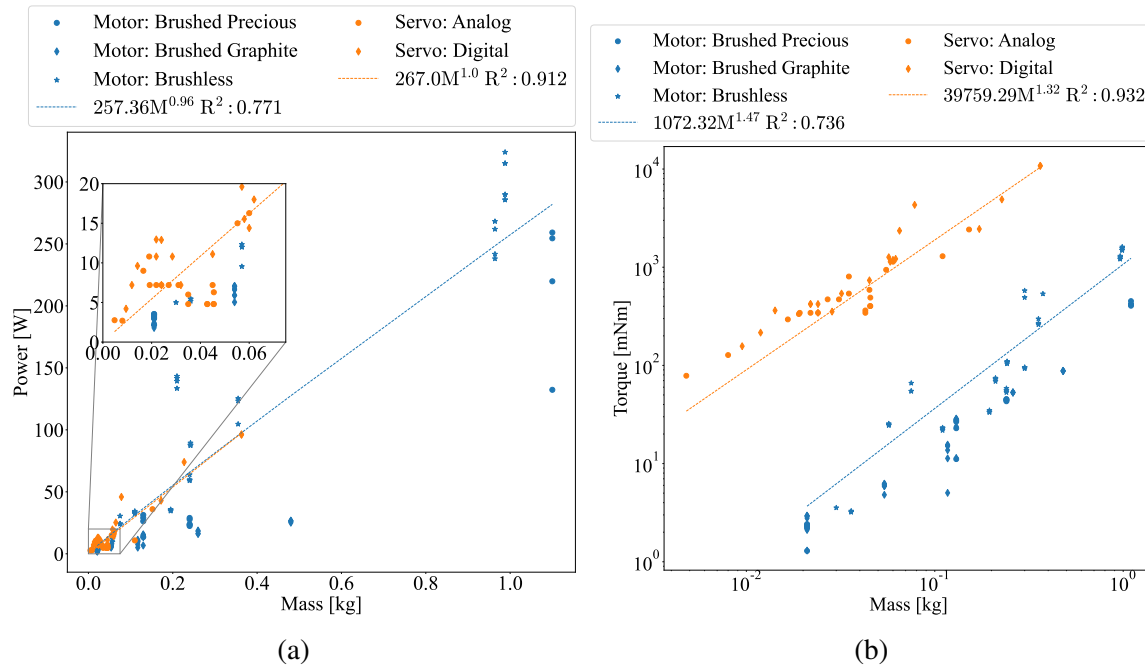


Fig. 2.19 (a) Actuator Power and (b) Actuator Torque versus actuator mass for Maxon motors from [44] and Hitec servos from [45].

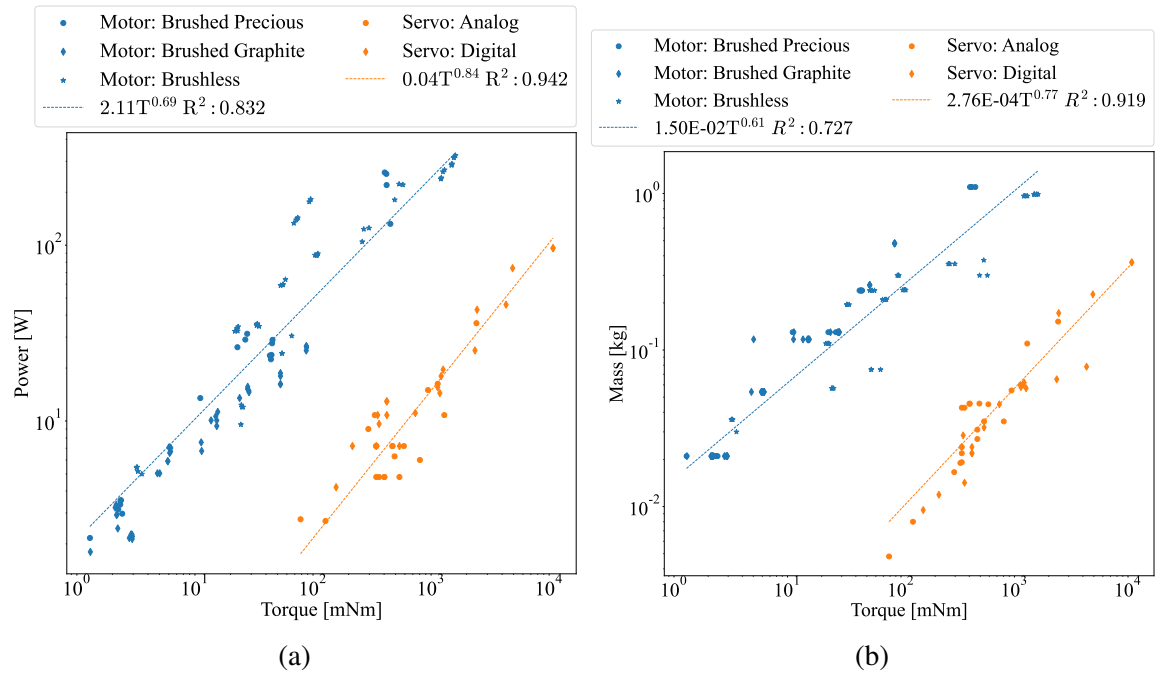


Fig. 2.20 Actuator power versus torque for Maxon motors from [44] and Hitec servos from [45].

Table 2.6 Allometric scaling laws for motor and servo data. The model used for all data reported is of the form $y = ax^k$ where y is the independent variable and x is the dependent variable.

Database	y	x	a	k	R ²
Motor	Power [W]	Mass [kg]	257.36	0.96	0.771
Servo	Power [W]	Mass [kg]	267.00	1.0	0.912
Motor	Torque [mNm]	Mass [kg]	1072.32	1.47	0.736
Servo	Torque [mNm]	Mass [kg]	39759.29	1.32	0.932
Motor	Power [W]	Torque [mNm]	2.11	0.69	0.832
Servo	Power [W]	Torque [mNm]	0.04	0.84	0.942
Motor	Mass [kg]	Torque [mNm]	$1.50e^{-02}$	0.61	0.727
Servo	Mass [kg]	Torque [mNm]	$2.76e^{-04}$	0.77	0.919

With the scaling laws shown in Table 2.6 and specifically Fig. 2.20, the required torque can be converted into a power draw and a mass. In the case of bio-inspired robotic platforms, multiple actuators might be implemented with the most common design being a servo actuator mounted to a link in a chain of links as shown by the Polish Naval Academy Cyberfish in Fig. 2.21 developed by Malec *et al.* [46].

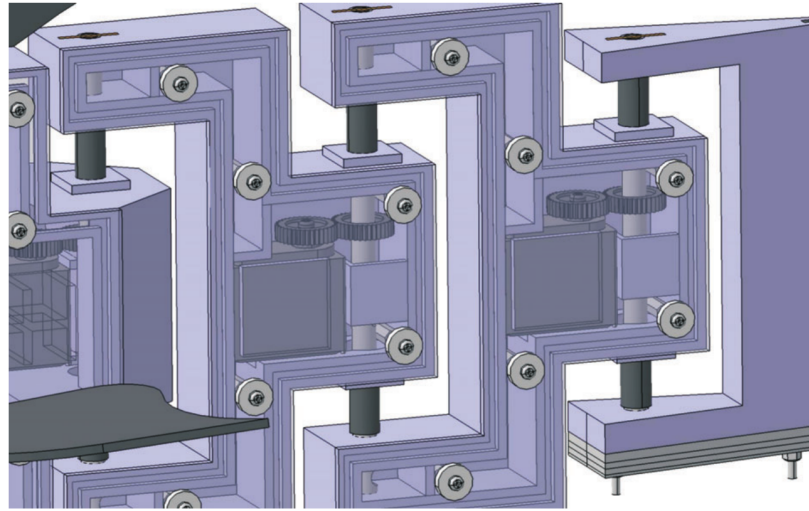


Fig. 2.21 CAD rendering of common servo linkage transmission system from Malec *et al.* [46, Fig. 3].

Actuator and Linkage Scaling

To understand how COT scales with linkages and actuators, the propulsion power for bio-inspired platforms is calculated using (2.6) and subtracted to the total COT. Table 2.7 shows the relevant data and platforms used in this analysis. Figure 2.22 displays the hotel power and propulsion power over the number of linkages and actuators. Each value was calculated at the optimal velocity if available or the reported velocity if an optimal velocity was not reported. The application to the Tunabot [30] shown in Fig. 2.17b shows that at this velocity, the true COT and the model (2.6) are fairly close.

To validate this methodology, two cases from Table 2.7 for which hotel power is available are compared. The cases for the Ho Chi Minh labriform developed by Ahn Pham *et al.* [38], which reported a hotel power of 0.085W, and the University of Glasgow RoboSalmon developed by Mazlan [36], which reported $P_H=3.0\text{W}$, are used. In both cases, the outlined methodology overestimates hotel power by 5% and 30% respectively. This suggests the following data should only be used as a comparison of the different platforms and not as the true hotel power.

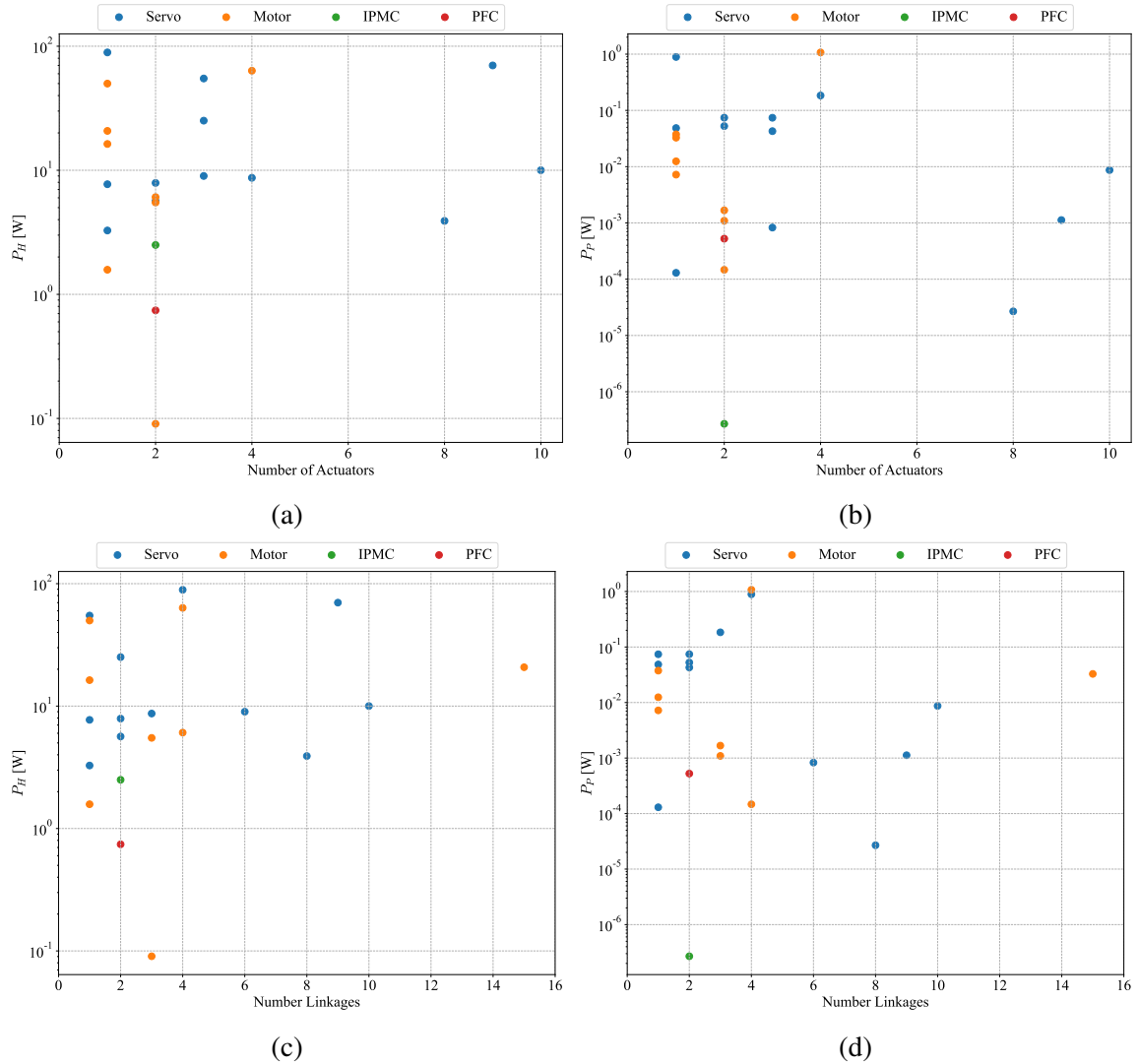


Fig. 2.22 (a) hotel power and (b) propulsion power for bio-inspired platforms plotted over the number of actuators. (c) hotel power and (d) propulsion power for bio-inspired platforms plotted over the number of linkages. Linkages and actuators were taken from pictures of platforms from published works.

Table 2.7 Comparison of calculated hotel power and propulsion power of different bio-inspired platforms and their actuator type.

Name	Reference	Locomotion	Actuator	Number Actuators	Number Linkages	Velocity [m/s]	Hotel Power [W]	Propulsion Power [W]
MIT Carangiform	[18]	Carangiform	Servo	1	1	0.10	3.27	$1.30e^{-4}$
Kyushu Carangiform	[16]	Carangiform	Motor	1	1	0.58	49.96	$3.75e^{-2}$
Beihang SPC-III	[31]	Thunniform	Servo	1	4	1.10	89.11	$8.91e^{-1}$
MIT SoFi	[21]	Carangiform	Motor	1	1	0.24	16.29	$7.22e^{-3}$
UV Tunabot	[30]	Thunniform	Motor	1	1	0.41	1.58	$1.25e^{-2}$
MAR	[33]	Anguilliform	Motor	1	15	0.30	20.77	$3.28e^{-2}$
Harvard Finbot	[28]	Ostraciiform	Servo	1	1	0.69	7.72	$4.85e^{-2}$
Ho Chi Minh Labriiform	[38]	Labriiform	Motor	2	3	0.15	0.09	$1.67e^{-3}$
Nanyang NAF-1	[17]	Carangiform	Motor	2	3	0.33	5.50	$1.09e^{-3}$
Nanyang Arawana	[25]	Carangiform	Motor	2	4	0.05	6.07	$1.47e^{-4}$
UV Robotic Manta ray	[37]	Rajiform	IPMC	2	2	0.07	2.50	$2.69e^{-7}$
EC PFC	[39]	Rajiform	PFC	2	2	0.24	0.74	$5.25e^{-4}$
Miro-7	[26]	Carangiform	Servo	2	2	0.50	5.65	$5.25e^{-2}$
Miro-9	[26]	Carangiform	Servo	2	2	0.50	7.91	$7.42e^{-2}$
CAS Robotic Shark	[23]	Carangiform	Servo	3	2	0.50	25.12	$4.28e^{-2}$
MIT RoboPike	[27]	Carangiform	Servo	3	6	0.09	9.00	$8.28e^{-4}$
Southwest State Carangiform	[20]	Carangiform	Servo	3	1	0.42	54.93	$7.40e^{-2}$
Madeline	[40]	Lift	Motor	4	4	0.74	63.44	1.07
Polish Naval Academy Cyberfish	[189]	Carangiform	Servo	4	3	0.66	8.69	$1.84e^{-1}$
RoboSalmon	[36]	Sub-Carangiform	Servo	8	8	0.02	3.91	$2.69e^{-5}$
NUTNU Mamba	[35]	Anguilliform	Servo	9	9	0.75	70.00	$1.12e^{-3}$
IMSat Artefact	[34]	Anguilliform	Servo	10	10	0.25	10.01	$8.68e^{-3}$

The data from Fig. 2.22 shows that the propulsion power is about 3 orders of magnitude less than that of the hotel power. For the case of the IPMC robotic manta ray from Chen *et al.* [37], the propulsion power is 6 orders of magnitude lower. Interestingly, the general trend is that hotel power remains fairly constant with the number of actuators and linkages. However, propulsion power seems to decrease the more actuators and linkages are used. An explanation for this is that the motion of fish undulation is optimized, therefore, the more linkages there are the closer the optimized undulation form is achieved.

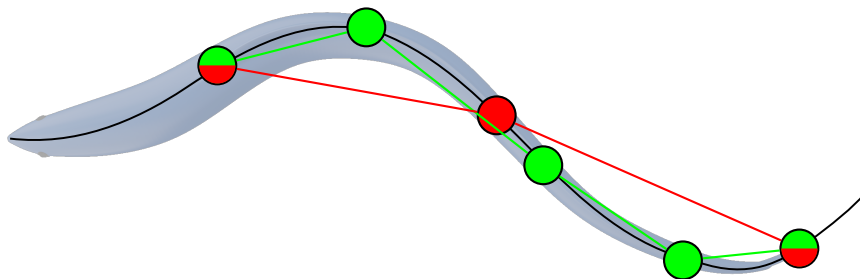


Fig. 2.23 Diagram showing that more linkages lead to a more optimized undulation form. Green circle and links show curvature for a 4 link tail. Red circle and links represent the same curvature with 2 links.

Sensor Scaling

Sensor scaling is difficult to quantify. It is highly dependent on scientific payload and mission requirements. This is compounded by the lack of published research for the power consumption of sensors and onboard electronics. Amory and Maehle are the only ones that have published the power consumption for onboard electronics on their micro AUV platform SEMBIO [139]. Table 2.8 shows the published power consumption for the components on the SEMBIO platform while either idle or actively being used.

Table 2.8 Data showing the power consumption from onboard electronics for the SEMBIO platform from Amory and Maehle [139, Table 2].

Component	Idle Mode [W]	Active Mode [W]
Arduino Mega	0.53	0.65
RPi Processor	1.25	1.86
Pixy Camera	0.12	0.74
GPS Unit	0.01	0.8
Camera Light	0.2	5.62
Compass kit		0.4
Pressure Sensor		0.01
PXFmini	0.13	0.41
XBee	0.27	0.97
Total	2.51	11.47

The table shows that one must consider an idle and active mode when estimating hotel power contribution from onboard electronics. While inactive, there is a decrease in power consumption of $\approx 22\%$. This also illustrates just how difficult it is to estimate hotel power. This is because it is entirely dependent on the payload of the vehicle and even a small change in equipment can drastically change hotel power consumption.

2.7 Are Biological Fins Better Than Engineered Thrusters?

The data so far has indicated that biological animals are marginally better than conventional propeller-driven AUVs and that bio-inspired platforms lag far behind in terms of energy consumption. So are biological fins better than engineered thrusters? Initial investigations into the hydrodynamics of engineered fins were performed by Triantafyllou *et al.* [190]. The group's investigation shows that oscillating foils in a uniform oncoming flow will develop thrust, given certain conditions. An interesting phenomenon to occur during thrust generation is the reverse Kármán Street. This is manifested by shedding vortices with alternating angular

velocities as shown in Fig. 2.3. A further finding of this research is that the optimal efficiency of the oscillating foils is reached in the Strouhal range of 0.25 to 0.35. These findings were confirmed with a more comprehensive parameter sweep done by Anderson *et al.* [191].

Interestingly, a meta study done by Gazzola *et al.* [4] showed that the preferred Strouhal number of swimming animals is 0.3 over several decades of Reynolds numbers. Figure 2.24 presents Gazzola's data with a clear distinction between the preferred Strouhal number of biological animals between lower and higher Reynolds numbers. At lower Reynolds numbers, the Strouhal number scales as $St = Re^{-1/4}$ but stays constant at 0.3 past a critical Reynolds number of $\approx 10^4$. These studies have led to the idea that creating a bio-inspired platform that operates at these Strouhal numbers will be inherently more efficient than current technologies, but experiments done by Palmisano *et al.* [47] show that thrusters are actually better in terms of thrust output and energy consumption when compared to engineered fins.

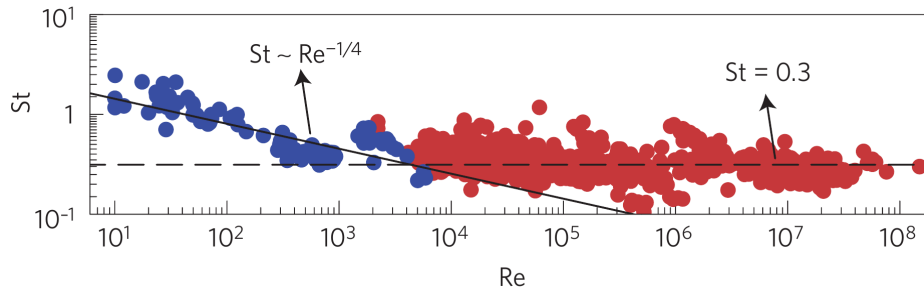


Fig. 2.24 Strouhal versus Reynolds number for 3000 swimming animals from Gazzola *et al.* [4, Fig. 2.b].

Palmisano *et al.* [47] at the Naval Research Laboratory tested small scale commercially available thrusters against their optimized fish fin and other bio-inspired fin data. Two key figures from their research are reproduced in Fig. 2.25 with power (P_{in}) being power consumption in Watts, generated thrust (T_{out}) in Newtons, and comparative efficiency (η) given by T_{out}/P_{in} . The data shows that commercially available thrusters outperform biomimetic fins in both thrust generation and comparative efficiency, but often require a higher power consumption. An additional note about these results is that these tests were conducted on an isolated test rig, and therefore, are not necessarily applicable to real-world locomotion and swimming scenarios.

2.7.1 Hybrid Biological Locomotion with Thrusters

A study that compares thrusters and the anguilliform swimming mode with an actual bio-robotic platform was performed by Kelasidi *et al.* [35]. The group presents 8 cases of an eel-like robot utilizing different propulsion mechanisms including only thrusters, only

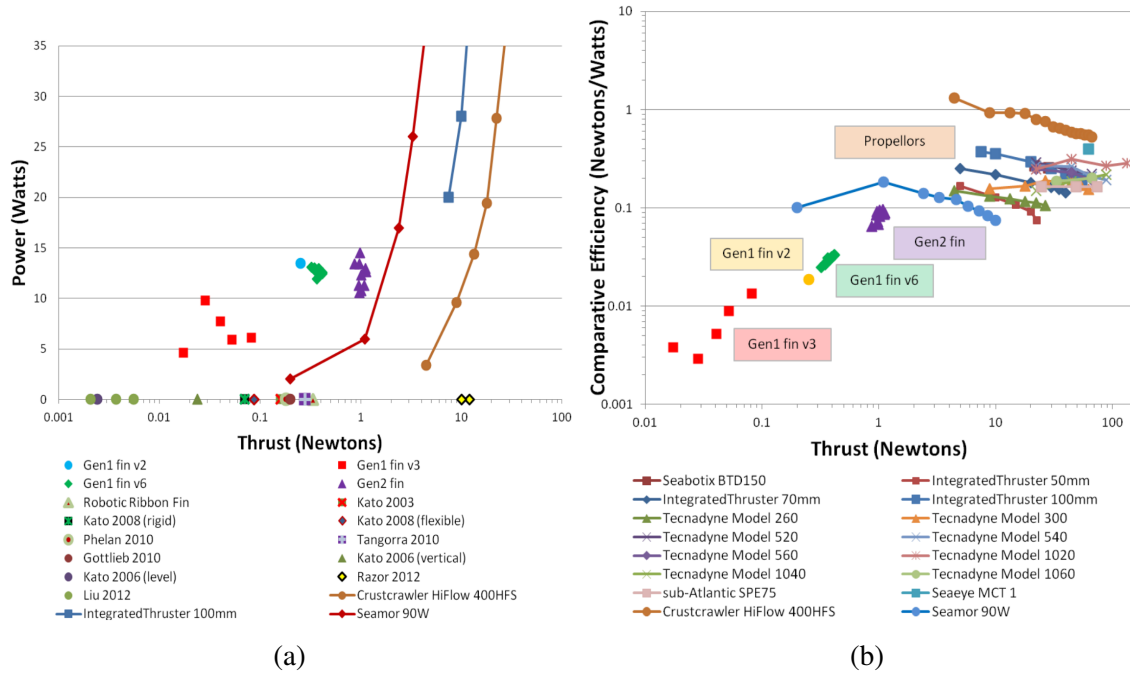


Fig. 2.25 (a) Power versus Thrust and (b) Comparative Efficiency versus Thrust showing the power and efficiency of commercially available thrusters compared to a biomimetic fin from Palmisano *et al.* [47, Figures 1 and 12].

anguilliform motion, and combinations of both types of propulsion. Five of these cases are reproduced in Fig. 2.26 and are defined as follows with u_c being the control input to the thrusters in percentage of total power:

- Case 1: $u_c = 60$
- Case 2: $u_c = 100$
- Case 6: $u_c = 60$ and anguilliform motion
- Case 7: $u_c = 100$ and anguilliform motion
- Case 8: $u_c = 0$ and anguilliform motion

The data shows that thrusters alone can achieve a higher forward velocity with less power consumption than only anguilliform motion. This is illustrated by the comparison of Case 1 and Case 8. Interestingly, Case 6 shows that implementing anguilliform motion with thrusters causes a decrease in kinematic performance and increase in energy consumption. Kelesadi *et al.* [35] does not address this result, but an increase in average power is expected for running thrusters and servo motors at the same time. The results are reported for average forward velocity so it is reasonable to assume that the undulation of the robot with the thrusters causes the robot to move in an S-turn pattern which would decrease the average forward velocity.

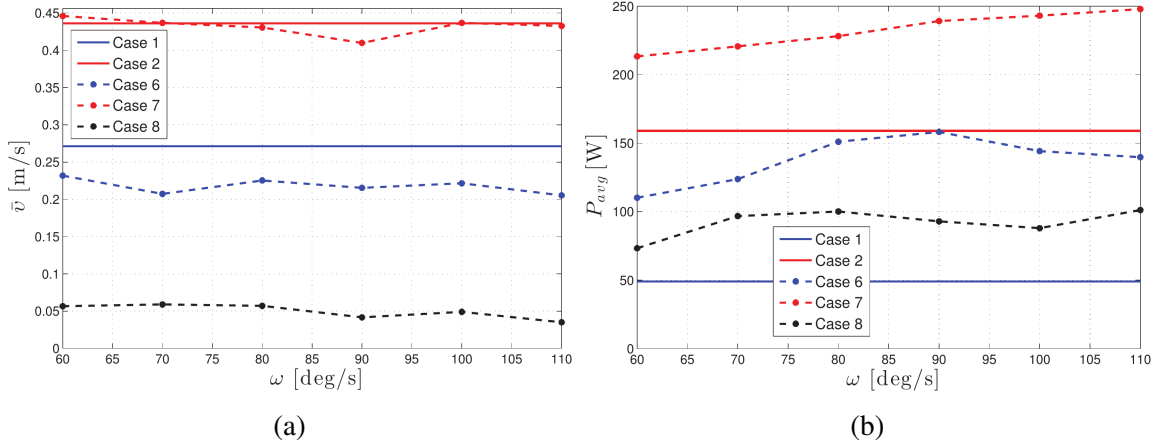


Fig. 2.26 (a) velocity versus undulation speed and (b) average power versus undulation speed showing different cases for an eel-like robot with thrusters from Kelasidi *et al.* [35, Figures 11.e and 11.f].

2.7.2 Where Does Biological Design Excel?

The data thus far is contrary to the consensus that building a bio-inspired robot makes for a more efficient swimmer. The next logical question would be what is the benefit of building an AUV with bio-inspired design features. This question is addressed by Fish [68] where the comparison of minimum turning radius in meters and maximum turn rate in degrees per second is made. Figure 2.27 shows the comparison between conventional AUVs, bio-inspired platforms, and biological animals from published research with the original data from Fish [68] also included.

Figure 2.27 highlights an attractive feature of biological animals and bio-inspired vehicles. Biological animals have a smaller turn radius in terms of body length and can turn at a greater rate than conventional AUVs. This trend extends to bio-inspired platforms in terms of a smaller turn radius. Turn rate for conventional AUVs is limited, so a proper analysis cannot be performed in that regard, but the data shows that, in general, bio-inspired platforms are within the cluster of their biological animals. The data thus far shows that conventional AUVs have the advantage in terms of straight line swimming, but that natural and bio-inspired vehicles are more maneuverable.

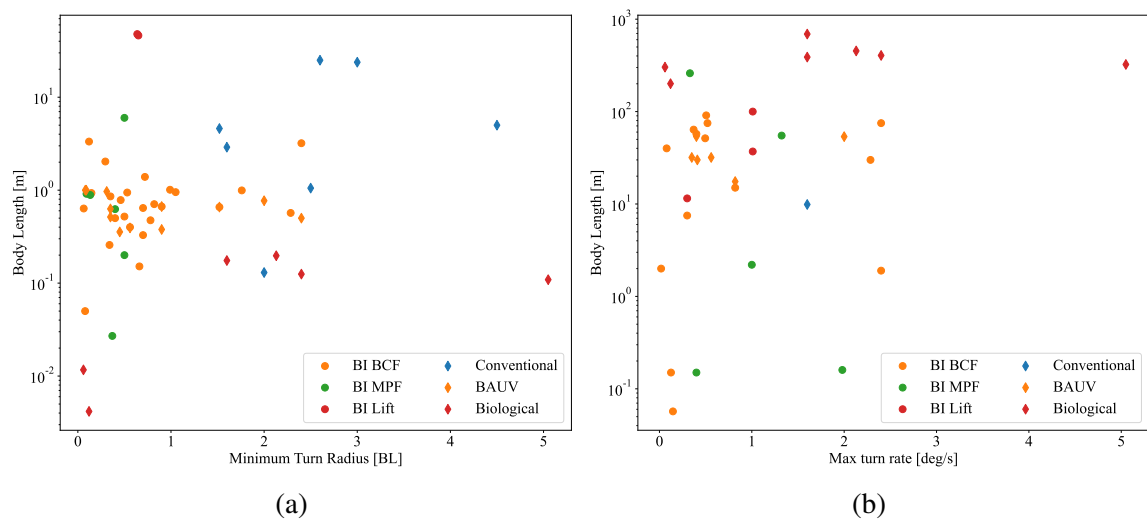


Fig. 2.27 (a) Length versus minimum turn radius and (b) length versus maximum turn rate for conventional AUVs, bio-inspired robotic platforms, and biological animals. Turn Radius BI BCF data from [19, 20, 23–27, 31, 34, 48–62]. BI MPF data from [63–66]. BI Lift data from [67]. All diamond data collected by Fish [68].

2.8 Concluding Remarks

This chapter gave an extensive overview of kinematic and COT scaling laws for biological swimmers, conventional AUVs, and bio-inspired robotic platforms. It explains how the total power in the COT calculation can be broken up into hotel power, P_H , and propulsion power, P_P . Propulsive scaling laws were reviewed for different modes of fish swimming. There is a lack of analytical scaling models pertaining to the propulsion of AUVs. In contrast, there is a substantial body of work and data for the energy consumption, both P_H and P_P , of natural swimmers including laboratory tests [8] and tests on wild animals [72]. While there is a lack of propulsive scaling laws for AUVs, there is enough COT data to construct the scaling laws directly. The same is not true for bio-inspired platforms, so a piece-wise approach was developed.

This chapter addresses the gap in scaling laws for hotel power in conventional AUVs and bio-inspired platforms. Scaling models for actuators and linkages were presented. It was found that the number of linkages had a minimal effect on the P_H and P_P of a platform. However, this may also be attributed to the wide variety of engineered designs. Unlike their biological counterparts, bio-inspired platforms are mainly used for research and are not optimized for efficiency. This results in a wide variety of design and power consumption. The data indicates that the scaling of hotel power is difficult to quantify as it depends entirely on the type of scientific, electronic, and actuation systems onboard the vehicle.

Finally, the relative thrust and efficiency of biological fins versus thrusters is discussed. Several experiments indicate that thrusters perform better than fins in terms of straight line swimming, but at the cost of maneuverability. Fins and flexible bodies allow for faster turning with a smaller turn radius. This indicates that a bio-inspired AUV may be more beneficial over conventional AUVs for certain applications.

Chapter 3

Better Estimates of Surface Area

This chapter was published in Coe and Gutschmidt [159].

3.1 Introduction

As discussed in the Chapter 1.1, there needs to be a reliable way to estimate the physical parameters (surface area, volume, mass, length) of vehicles to account for different mission payloads. Each of the physical parameters can be determined as a function of length so a methodology for scaling a variety of biological and bio-inspired platforms was developed using computer vision. The ability to scale platforms allows for a direct comparison between platforms at different Reynolds/length scales when coupled with the scaling laws discussed in Chapter 2. Furthermore, the methodology developed in this chapter allows for the scaling of platforms, existent or non-existent, based on the amount of payload needed for a mission which can inform the size of the overall design for that particular payload.

There is very little information on allometric relationships for length/mass versus surface area and volume of various aquatic animals, AUVs, or artificial swimmers. The animal-based literature mainly focuses on aquaculture and the colonization of lice on farmed fish. The current method for accurate measurement of the surface area of fish is an adapted wrap method presented by O'Shea *et al.* [71]. The animal is anesthetized, the fins are dissected, and the body of the fish is wrapped in paper. The paper is cut such that all the edges are flush with each other. Once cut, the paper is laid flat on graph paper and the surface area is measured [192, 71, 193]. Another method is to wrap the body with cotton strings at 2 mm increments along the body length as done in Ling *et al.* [194]. The strings are then measured and translated to 2D coordinates to get the surface area. Similarly to the method described by O'Shea *et al.*, this method has some interpolation error associated with it and is not as accurate as the previous wrap method. In the realm of engineering systems, there have not

been any studies that show allometric relationships between length/mass and surface area. Chapter 3 gives the description of a computer vision based algorithm that overcomes this lack of data.

There has been considerable research done in the realm of fish classification and estimation of physical dimensions by computer vision. Traditional image segmentation and volume estimation methods are presented by Siswantoro *et al.*; these authors employed k-means clustering and the Sobel operator [195]. Balaban *et al.* [196] measured Alaska pollock (*Theragra chalcogramma*) by taking side and top view images and then estimating the body contour as a b-spline. This was then used to calculate the volume. In contrast, Rantung *et al.* [70] measured the length, height, and width through side and top views of the fish and used these measurements as inputs that divide the fish into discrete elliptical discs. Prior to using this algorithm, the camera was calibrated such that the length and height of each pixel was known.

With the recent increase in computational power, there has been increased emphasis on the use of convolution neural networks (CNN) for this task. Yang *et al.* [197] provides a review of the use of deep learning techniques in fish farming. The group shows that using CNNs for image segmentation and estimation of fish parameters can achieve between 0.2% and 5% accuracy. A disadvantage of these models is that they require a lot of data to train and are less accurate when trained with a limited data set. Additionally, these models are valid only for the species they are trained on [197–199]. While there are large datasets available to train CNNs [200], this study’s focus is on bio-mimetic robots and AUVs, which are not always shaped as fish. For this reason, we elected to employ a simpler model that gives complete manual control over determining the contours and fit parameters versus being a “black box”. Furthermore, these methods are not strictly used to measure fish surface area as would be needed for the COT model.

A method not based on computer vision is derived by Murphy and Haroutunian and requires the length and mass of the animal or engineered system. This method derives an equivalent diameter as an input to a prolate spheroid approximation used in calculating the surface area [142, 201, 3]. This particular approach is beneficial because much of the literature on fish species only provide their length and mass. Rantung *et al.* also presents this prolate spheroid method, but without the equivalent diameter derivations [70].

The contact methods described by O’Shea *et al.* and Ling *et al.* require the animal to be physically present and anesthetized. In many cases, obtaining a specimen is difficult, and applying this method to AUVs and artificial swimmers is impractical. The contact-less methods described in Murphy and Haroutunian and Rantung *et al.* are more appropriate when trying to synthesize data from specimens that are not physically present. These methods have

the drawback that some of the physical dimensions, such as length, mass, width, and height, are needed beforehand to obtain a relatively accurate measurement.

This chapter introduces a novel computer vision methodology, named Ika-Fit, that reduces the body geometry of marine animals to a polynomial and a NACA airfoil. This methodology is compared to other methods from the prolate spheroid approximation already discussed to other computer vision methodologies. The importance of this methodology is shown in relation to the estimation of the surface area, volume, and cost of transport.

The method outlined in this research is called the Ika-Fit method. “Ika” is the Māori word for fish, and the research involves using computer vision to fit contours to various natural and artificial fish. The formulation for this is included in Appendix B.

3.2 Results

3.2.1 Validation with 3D Scanned Model

The 3D mesh of the scanned salmon shown in Fig. B.1 was imported into the open-source software Blender and dimensions were consistently scaled from lengths of 0.1 to 100 m. Surface area and volume were evaluated within Blender software using the 3D print add-on. Validation of the surface area and volume are given in Fig. 3.1.

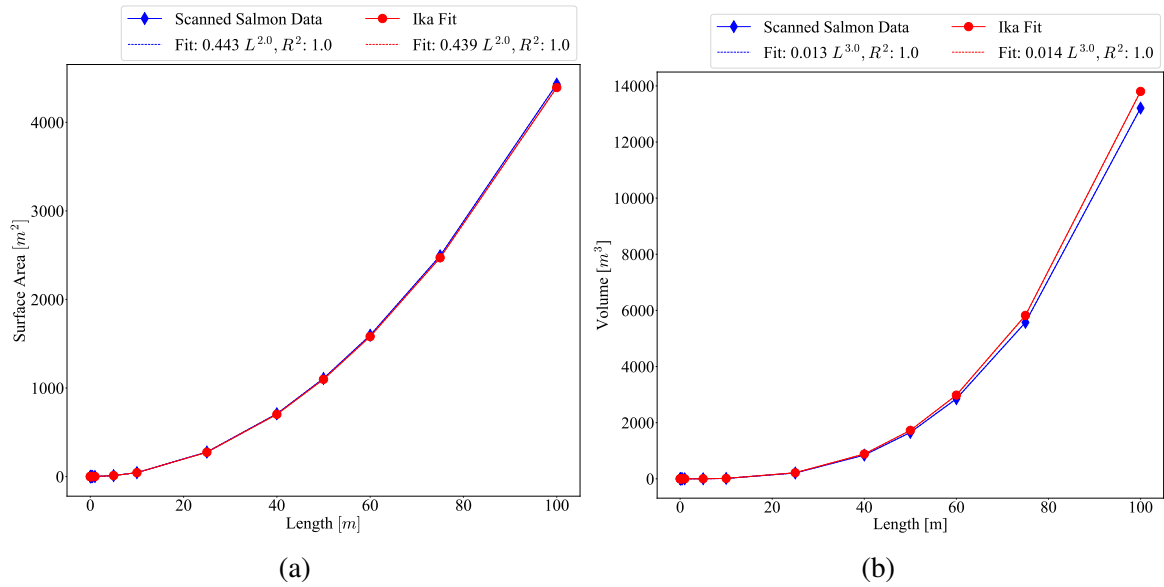


Fig. 3.1 (a,b) Validation of Ika-Fit (a) surface area and (b) volume algorithm with scanned salmon data scaled using Blender software.

The figures show reasonable agreement with the measured data through a large interval of length scales. This data further validates the method's ability to scale surface area and volume with the model's length. An extremely significant finding is that the volume scales in a similar manner. This allows for the calculation of mass by assuming that the animal or robot is neutrally buoyant in water. Then, the mass property is deduced:

$$M_{platform} = \rho_{water} \cdot V_{platform}, \quad (3.1)$$

where ρ_{water} is taken to be the average density of water, $1025 \frac{\text{kg}}{\text{m}^3}$. A dissection of fish by Haroutunian [142] shows that the fins account for approximately 1% of the total mass of the fish, so they are excluded from the volume calculation.

The error between different datasets is reported as root mean square difference (RMSD) and mean absolute error (MAE) using the following equations:

$$RMSD = \sqrt{\frac{\sum_{n=1}^N (\hat{y}_t - y_t)^2}{N}}, \quad (3.2)$$

$$MAE = \frac{\sum_{n=1}^N |\hat{y}_t - y_t|}{N}, \quad (3.3)$$

where \hat{y}_t is the estimated value, y_t is the published data, and N is the number of data points. This formulation is used for the following analysis and in the rest of this paper.

3.2.2 Validation with Biological Animals

Four species were used in the validation of the mass: Atlantic salmon (*Salmo Salar*), Atlantic cod (*Gadus Morhua*), killer whale (*Orcinus orca*), and European silver eel (*Anguilla Anguilla*). Data from O'Shea *et al.* [71] were utilized for the Atlantic salmon and cod. For the European silver eel, the length-weight relationship from Froese and Pauly [202] was adopted. The length-weight relationship used for the killer whale is given in Bigg and Wolman [203]. No images were available for the killer whale or silver eel, so a 3D model was constructed in the open source software Blender based on reference pictures. Figure 3.2 shows the mass calculated by the Ika-Fit method compared to published data.

The figure shows that there is good agreement between the data and the estimation from the algorithm. Table 3.1 shows the error of the Ika-Fit method with the length-weight relationships of each species.

These species were chosen according to the available data of both physical morphology and COT. In the case of the killer whale and European Silver Eel, a 3D model constructed

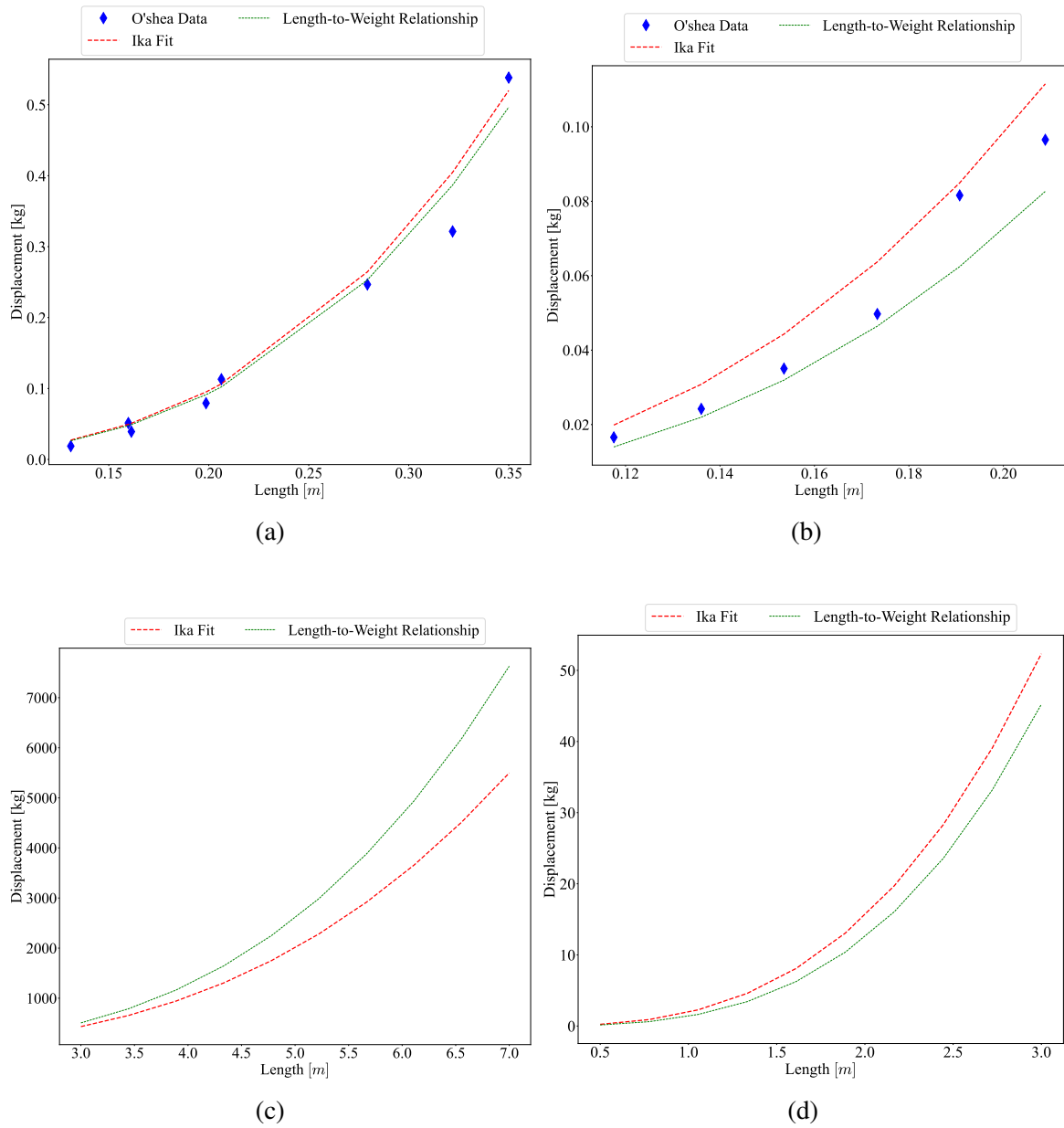


Fig. 3.2 (a) The calculated mass versus published data for (a) Atlantic salmon, (b) Atlantic cod, (c) killer whale, and (d) European silver eel.

Table 3.1 RMSD and MAE of the Ika-Fit method of four animal species. Error is given as a percentage of the Ika-Fit estimate versus published data.

Error	Salmon	Cod	Killer Whale	European Silver Eel
RMSD	0.032	0.010	172.328	3.655
MAE	0.020	0.009	161.478	2.810

based on reference images was used instead of images of the actual specimen, due to unavailability. For this reason, the fit algorithm does not have as precise of a top body shape because dimensions had to be estimated. These results further show the flexibility of the proposed method as a 3D model can be constructed from reference images. The results are similar to published data of multiple different samples of the same species.

Comparison with Other Algorithms

As discussed in Section B.0.2 and presented in Table B.2, there are three other methods that our results are compared against. Figure 3.3 displays the comparison between the methods on the 3D scanned model. The Ika-Fit method shows better agreement with the scaled data than comparison methods. For reference, a power-law fit for each method is provided, as is convention with allometric data, and the error for each method is shown in Table 3.2.

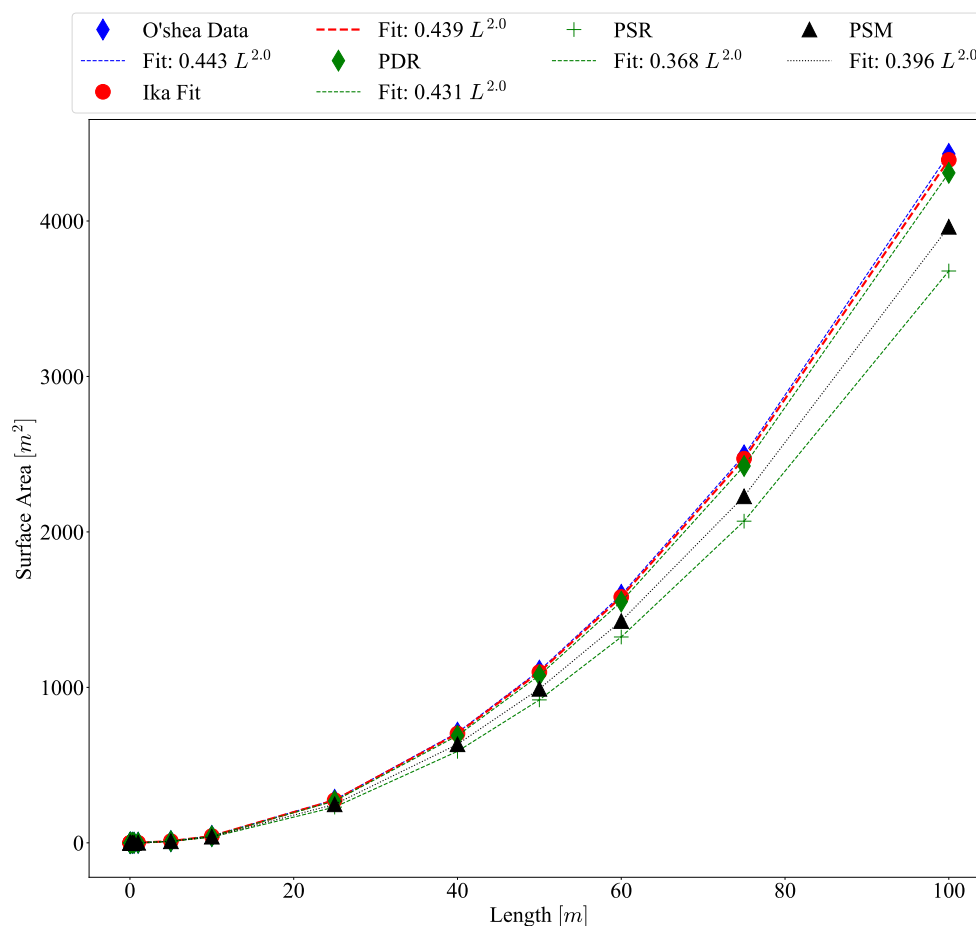


Fig. 3.3 Comparison of surface area of all algorithms with the scanned salmon model that has been scaled inside Blender.

Table 3.2 RMSD and MAE of compared methods with laser-scanned salmon. Error is given as a percentage of the estimated algorithm value versus the direct value given by the 3D model.

Error	IF	PD	PSR	PSM
RMSD	12.25	38.98	241.13	150.10
MAE	6.14	19.54	120.87	75.23

The comparison shows that the developed Ika-Fit method outperforms the other methods in terms of accuracy. The PDR method gives the next most accurate estimation of surface area. These two methods are similar in that they both partition the specimen into ellipse discs, with the difference being the NACA fit of the top of the specimen, as well as the inclusion of fin surface area.

To compare all methods to biological data for surface area, data from O'Shea *et al.* were used with images of Atlantic cod (*Gadus morhua*) and Atlantic salmon (*Salmo salar*) [71]. A high-precision 3D scan of a king salmon is presented as a further validation. Due to the king salmon and Atlantic salmon not being the same species but part of the same family, a scaling factor was formed to scale the contours to be closer to the Atlantic salmon. No top images of Atlantic salmon were available, nor was a real fish available for purchase and laser scanning for proper dimensions.

For comparison with published methods, the PDR, PSR, and PSM methods are included in the analysis. It should be noted that the PDR and PSR methods use the contour data from the computer vision algorithm. This is because height and width data was not reported in the published research. The mass needed for the PSM method was directly taken from the data given in O'shea *et al.* [71], since those values were reported. Tables 3.3 and 3.4 give the relative error of all approximation methods with the data of O'Shea *et al.*

Table 3.3 RMSD and MAE of compared methods with Atlantic salmon data from O'Shea *et al.* [71]. Error is given as a percentage of the estimated algorithm value versus the direct value given by the 3D model.

Error	IF	PD	PSR	PSM
RMSD	0.0025	0.0035	0.0066	0.0082
MAE	0.0016	0.0029	0.0057	0.0074

To determine the effect of the fin surface area, the same fin area ratio was added to the final surface area of the other methods. Figure 3.4 shows that the addition of the fin area improves the accuracy of all the other methods.

Table 3.4 RMSD and MAE of compared methods with Atlantic cod data from O'Shea et al. [71]. Error is given as a percentage of the estimated algorithm value versus the direct value given by the 3D model.

Error	IF	PD	PSR	PSM
RMSD	0.0013	0.0024	0.0010	0.0050
MAE	0.0011	0.0020	0.0009	0.0045

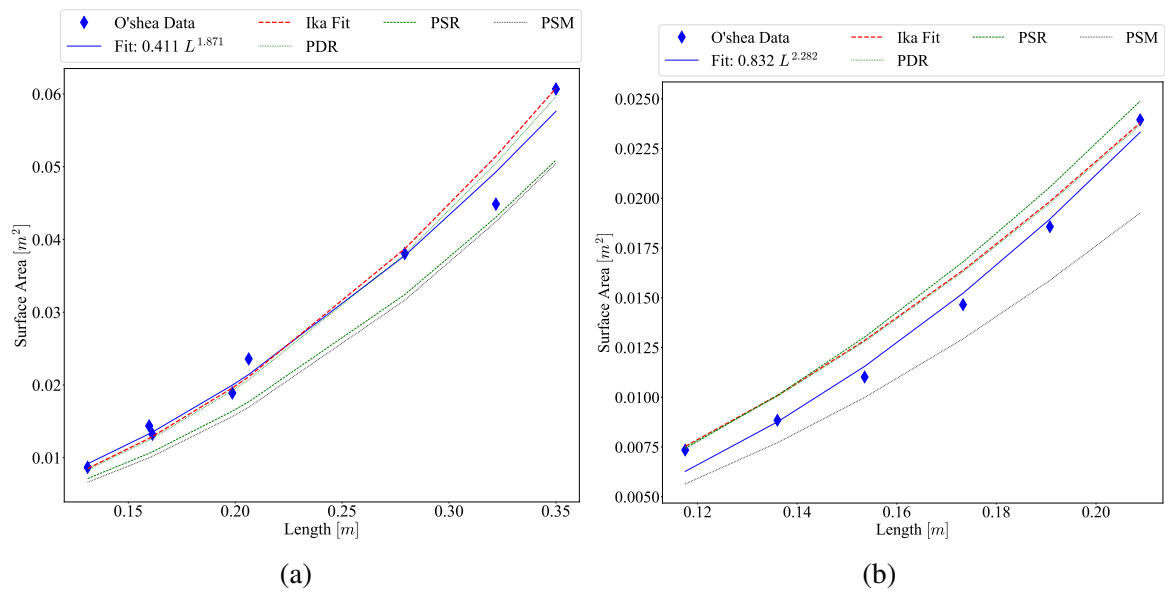


Fig. 3.4 Comparison of algorithms with the fin area added to the final surface area. (a) Scaled Atlantic salmon data and (b) Atlantic cod data.

The data shows that the Ika-Fit method and the PDR perform similarly to each other when the fin area ratio is added. This is expected, as the difference between the two methods is treating the top and bottom of the animal as separate partitions, as well as the NACA airfoil fit and the inclusion of the fin area ratio. The PSR and PSM methods perform well in the case of the Atlantic cod, but is not consistent when comparing it to the Atlantic salmon data.

To determine the breakdown of the two prolate spheroid methods, the equivalent diameter was calculated and superimposed on the side and top view contours of the king salmon. As shown in Fig. 3.5, the equivalent diameter underestimates the side and top contour. This leads to the prolate spheroid surface area being underestimated, wha shown in the comparison data of Figures 3.4 and 3.6.

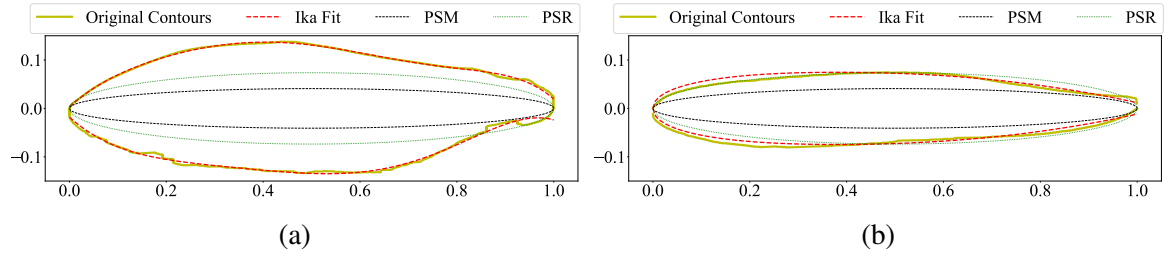


Fig. 3.5 Equivalent diameter ellipse superimposed on fit contour. **(a)** Side and **(b)** top views of king salmon. Width used as input into the ellipse of Rantung et al. was taken from the top contour output from the computer vision operations and not from the NACA airfoil fit.

A study was conducted to determine the relative time difference between the wrap method presented by O'Shea *et al.* [71] and the comparison algorithms. Table 3.5 lists the time it takes for each algorithm to perform operations in the following order: importing images, performing computer vision operations, fitting all contours, finding minimum and maximum points, fitting the NACA airfoil, and calculating the surface area. The PSM method does not require computer vision operations; therefore, the time for this is defining variables and estimating the surface area.

Robotic Data

To extend the algorithm to bio-inspired robotic models, a database of 81 conventional AUVs and 139 bio-robots was compiled. Of the bio-inspired robots, 80 operate with body-caudal fin (BCF) propulsion, 35 use median-paired fin (MPF) swimming mode, 13 use lift-based propulsion, and the remaining robots are inspired by other propulsion modes. Out of the 139 bio-inspired robots, only 4 reported their surface area, and their relevant parameters are summarized in Table 3.6.

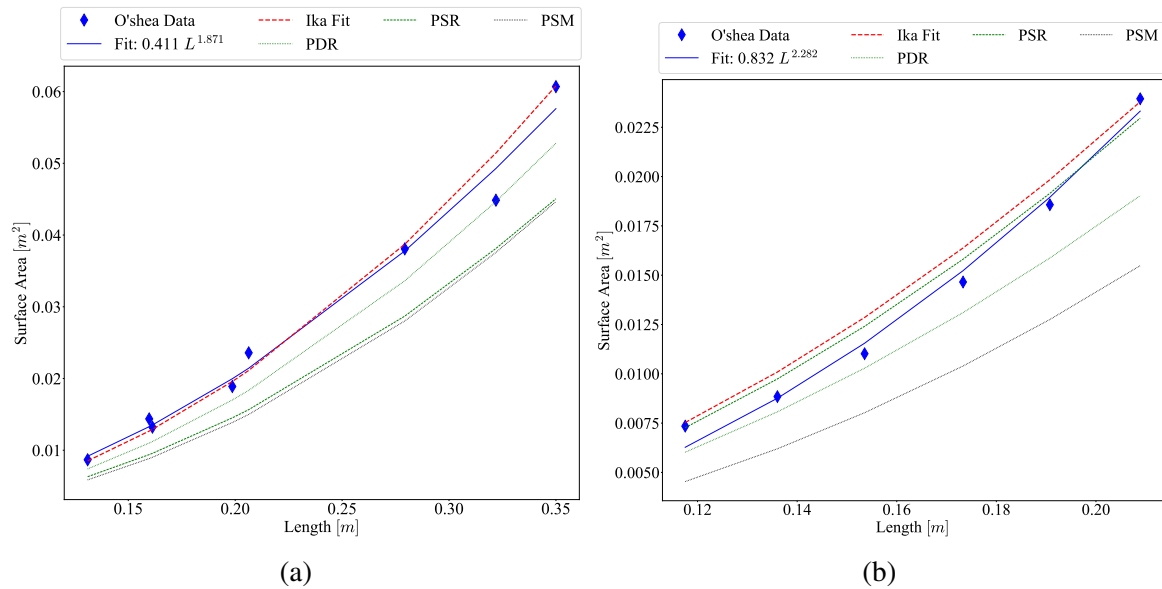


Fig. 3.6 Comparison plot between the surface area estimation algorithm, prolate spheroid method [69, 70], partition disc method [70], and O'Shea regression lines [71]. (a) Unscaled Atlantic salmon data, and (b) scaled Atlantic salmon data.

Table 3.5 Relative time difference between methods. Time was measured from the image input to the value output.

Method	Time Difference [s]
O'Shea et al. wrap method	<i>laborious</i>
Ika-Fit	1.3925
PD	1.3540
PSR	1.3384
PSM	0.0004

Table 3.6 Bio-inspired platforms used as validation in this research. Relevant parameters and citations are given.

Platform Name	Total Length [m]	Standard Length [m]	Width [m]
Harvard Beihang Mackerel [204]	0.588	0.548	0.080
MIT Carangiform [18]	0.148	0.127	0.025
NYU iDevice [205]	0.066	0.0452	0.019
NRL 4-Fin [206]	0.438	0.438	0.089
	Height [m]	Mass [kg]	Surface Area [m ²]
Harvard Beihang Mackerel [204]	0.95	2.79	0.137
MIT Carangiform [18]	0.043	0.068	0.013
NYU iDevice [205]	0.021	0.009	0.0032
NRL 4-Fin [206]	0.089	0.178	2.9

The results of the Ika-Fit method on these bio-robotic models are shown in Fig. 3.7. The figure shows that the Ika-Fit method performs the same as or better than other methods. Interestingly, the PSM method performs poorly for the last two bio-robotic models. An explanation of this is due to the formulation of the equivalent diameter, which is only a function of length and mass of the animal/robot. In the case of the Naval Research Laboratories four-fin platform, the mass is close to that of the Harvard Beihang Mackerel but the surface area is nearly double.

The mass of the robots was calculated by multiplying the volume by water density. The PSM method performed the most effectively due to the method formulation that guarantees the same mass is output when multiplied by the density of water. This test highlights a limitation of the Ika-Fit method when applied to engineered systems like bio-mimetic robots. In cases where the designers stayed as close to the biological analog as possible, as in [18, 204], the Ika-Fit method is the best. In cases where the biological analog is not followed, as in [205, 206], the Ika-Fit method does not perform well in estimating mass. This can be resolved by calculating an equivalent mass factor based on the published and calculated mass that can account for the different design parameters, such as:

$$M_{eq} = \frac{M_{published}}{M_{calculated}} \quad (3.4)$$

3.2.3 Application to Cost of Transport

The primary purpose of an accurate parameter measurement is the application to COT in (2.6) to obtain an estimation of the propulsion power. This is only half of the COT equation, and the hotel power must also be estimated. For the purpose of the following analysis, the hotel

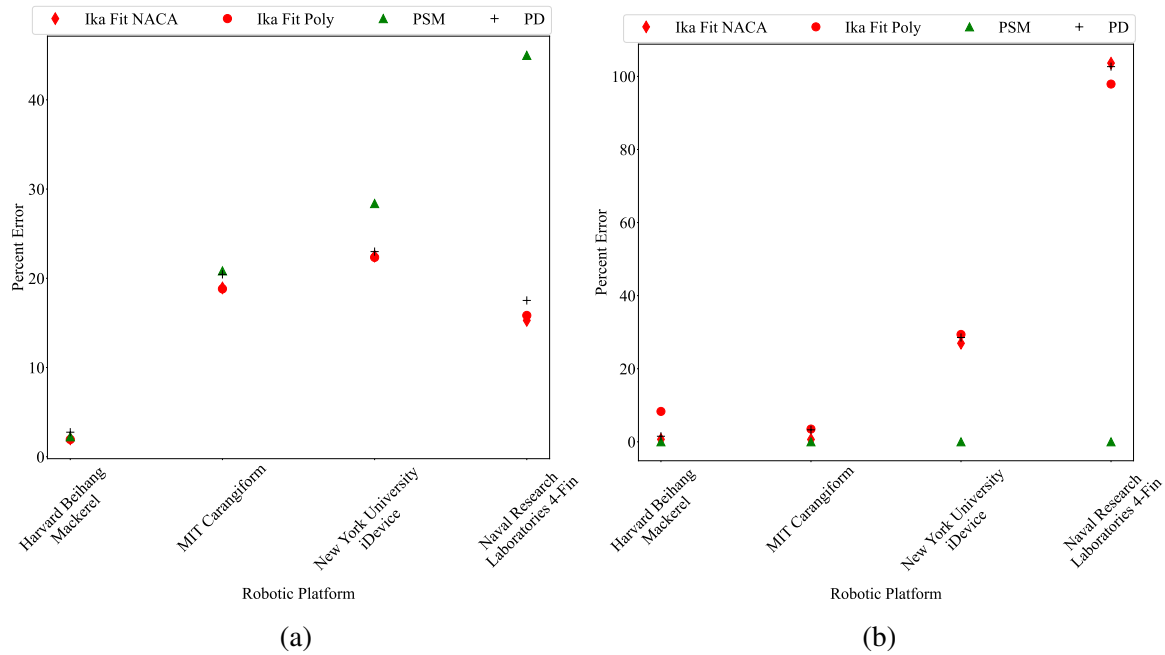


Fig. 3.7 (a) Comparison of surface area and (b) mass for bio-robotic platforms. Ika-Fit NACA and Ika-Fit Poly are this study's method using the NACA fit described in Section B.0.1, and Ika-Fit Poly is the top contour's fit with a fourth degree polynomial in a similar way to the side contours.

power of biological animals is estimated using data collected by [3], and a power-law curve fit is applied to obtain an allometric relationship, as shown in Figure 2.18. The relationships derived from this data are used in conjunction with the Ika-Fit method and the PSM method outlined in [3] to compare the derived COT model against aquatic animals. The PSM method is included to give a comparison as to the relative importance of the estimation method for physical parameters of the animal within the complete COT formulation. Figure 3.8 shows that the two methods converge in region 1 and slowly diverge as velocities increase. This is to be expected as hotel power dominates the model in this region, and so the estimation of physical parameters is less relevant. As the propulsive power dominates in region 2, the data shows that the two estimations diverge and do not follow the same trend lines. This is particularly apparent in the data for the European silver eel, where the Ika-Fit method seems to follow the data more closely than the PSM method.

There is a significant difference in the dominant hotel power region for the two salmonoid specimens. The method used to calculate hotel power was the power-law fit for salmonoid data from Figure 2.18 and the results suggest that a hotel power for each specific species should be derived separately from each other. This also shows the importance of an accurate

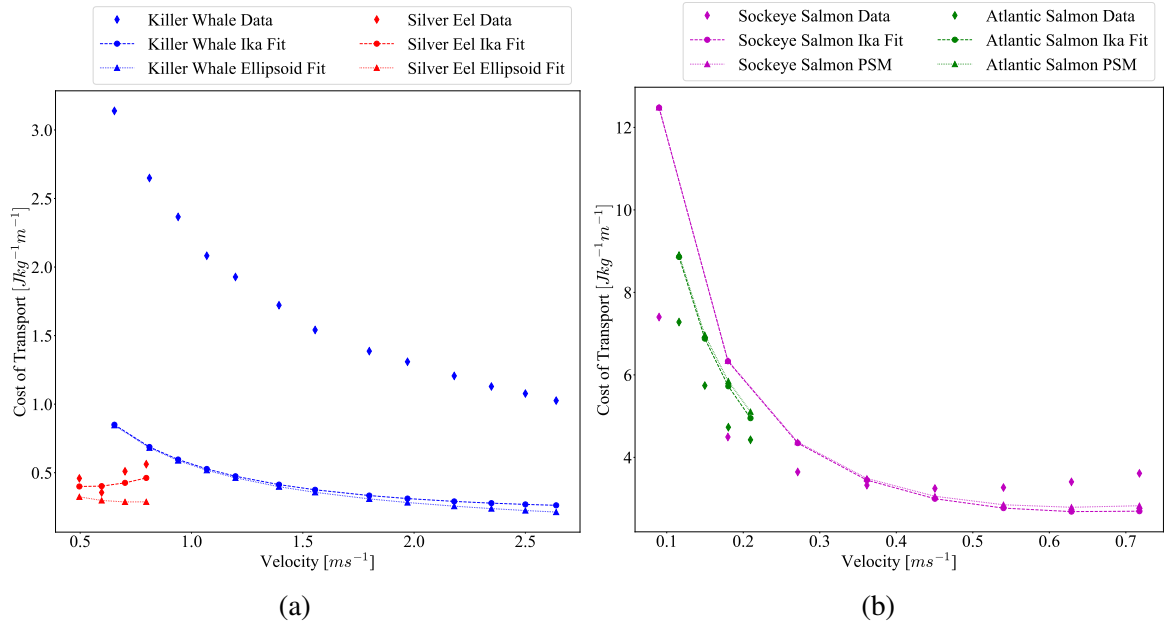


Fig. 3.8 Comparison of the Ika-Fit and PSM method in calculating COT with published data. **(a)** Killer whale data from [72] and European silver eel data from [73]. **(b)** Sockeye salmon data from [74] and Atlantic salmon data from [75].

hotel power estimate for animals and vehicles operating in region 1, as there is a significant gap between the data and calculated data as hotel power becomes dominant.

There is a significant difference between the data and the model of (2.6) at higher velocities, as is most clearly seen in the data for the killer whale specimen. As an animal or robot swims, there are three sources of drag, as explained by Magnuson [156]: friction and form drag, gill resistance, and induced drag. The ITTC method, explained in Section 3.1 and used to determine C_D in (2.6), only gives the friction and form drag. In other words, it is the resistance of the animal/vehicle when towed. Data estimated on skipjack tuna swimming at sustained speeds of 66 cm s^{-1} showed that induced drag accounts for 30% and gill resistance accounts for 17% of total drag [156]. The induced drag arises from wake effects and vortices generated from the propulsor interacting with the water.

The biological data itself has some assumptions built into it that may contribute to the disparity in data. Specifically, the killer whale data is of wild animals swimming in open water. As noted by the author, the calculations assumed straight line swimming, and swim speeds were not corrected for the effect of tidal currents. These could not be made, since the orientation of the whale with respect to the tidal currents was difficult to determine [72].

In an effort to limit uncertainty from the analysis and to test the efficacy of the COT model, the Ika-Fit method was compared to bio-robotic data. This approach was taken

because it removes most uncertainties due to electrical power output, or the combination of hotel power and propulsion power, being measured exactly. This approach is also applicable to velocity, mass, and physical dimensions. To determine hotel power for these robots, total power is plotted at each velocity. The data is then extrapolated with a curve fit to velocity $U = 0$. The total power at velocity $U = 0$ is determined to be the hotel power of the artificial swimmer. This is analogous to what is done for biological swimmers [74]. Out of the 139 bio-mimetic robots surveyed, 3 groups reported COT results and the comparison is shown in Figure 3.9. The RoboSalmon and Knifebot both reported total power for a range of velocities. The UVTunabot only reported COT, and total power was calculated by multiplying COT by mass and velocity at that point. The extrapolation method described above was used in all cases.

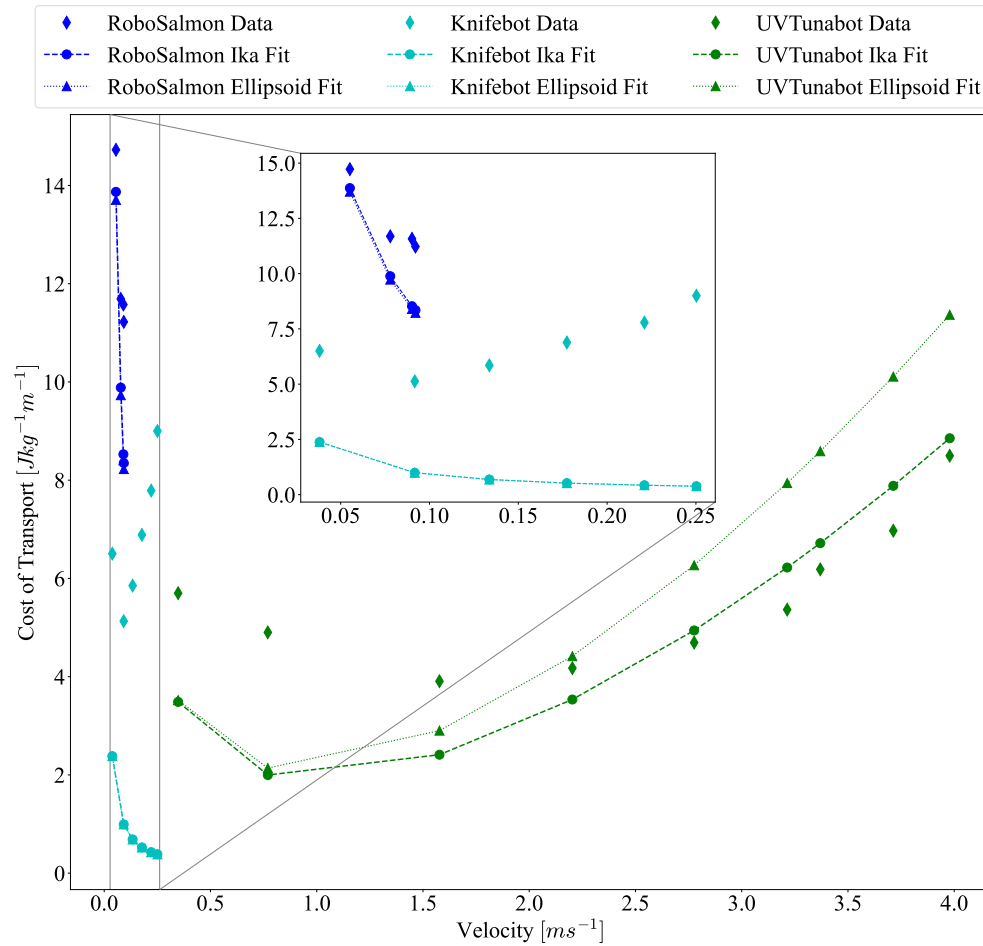


Fig. 3.9 Comparison of the Ika-Fit and PSM method in calculating COT with published bio-mimetic robot data. RoboSalmon data from [36], Knifebot data from [76], and UVTunabot data from [30]. The dashed black line marks the demarcation of regions 1 and 2 for the UVTunabot.

It is critical to note the different modes of locomotion encompassed by the data and how it relates to the COT model. The RoboSalmon operates in sub-carangiform mode, which means that the rear 50% of the entire body undergoes undulation. The UVTunabot operates in thunniform locomotion, in which only the tail or aft third of the fish undergoes oscillation. The Knifebot uses gymnotiform locomotion similar to that of a black ghost knifefish. This robot has a fin that runs the length of the body and oscillates in a sinusoidal pattern to create locomotion.

Data in Fig. 3.9 shows that the developed method and COT model have the same drawbacks, as discussed in Fig. 3.8. The published COT is significantly greater than the modeled COT for the non-thunniform swimmers. The UVTunabot keeps pace with the model, because the body of the robot is mostly rigid while swimming. Interestingly, there is a cross-over point in the UVTunabot data in region 2 at around 2.25 ms^{-1} that is not present in the other bio-robotic platforms or the biological data comparison of Fig. 3.8. It is expected that the model of (2.6) would underestimate values in region 2 as explained above, but that is not the case here. Comparing the velocities of the robot data and biological data of Fig. 3.8 shows that the highest biological speed reported is around 2.75 ms^{-1} .

Another consideration is the efficiency of the actuator and propulsor, which varies with velocity and was reported only for [76, 36]. For the UVTunabot data presented in Fig. 3.9, the actuation and propulsion efficiency for the UVTunabot was estimated to be 0.64, with the DC motor having an efficiency of 0.8 and the propulsor having an efficiency of 0.8. Propulsor efficiency was chosen because the reported Strouhal number is between 0.2–0.4, which is shown to reach propulsion efficiencies of up to 0.9 [175]. The actuation mechanism was a DC motor that has efficiencies in the range of 0.6–0.9, as shown in Table 2.3. As seen in Fig. 2.6d, efficiencies have the marked effect of either flattening or rounding the COT curve in region 2.

Comparing the Ika-Fit algorithm to the PSM algorithm for the bio-robotic data, there is a significant divergence for the UVTunabot data; this divergence is less so for the other two platforms. Out of the three robots shown, the UVTunabot is the least like a prolate spheroid, with the max height being more than 50% the length. This implies that the PSM method is overestimating the surface area and shows how the approach taken by the Ika-Fit method is more accurate.

The results outlined above show how sensitive the COT model is to input parameters. In addition, the COT model lacks the formulation to include complex induced drag and wake effects. The model works well when less of the body is undergoing undulation for propulsion. This suggests that it is a suitable model to use when evaluating ships, AUVs, and submarines, as was its original purpose. To apply this model to biological animals or bio-mimetic robots,

there needs to be further research into propulsion power at different swimming speeds, and therefore different Reynolds numbers, for the various modes of locomotion.

For input parameters to (2.6), the Ika-Fit method performs more accurately than comparison methods at estimating physical parameters, including the surface area, volume, mass, and slenderness ratio. This method has the added benefit of not requiring a physical specimen and performs the estimation with only the length of the specimen as an input parameter. This is beneficial because it allows for the scaling of a platform in order to give an estimate of the COT at different length scales or Reynolds numbers. This is further useful if the only information provided about a platform is the length.

A limitation of the Ika-Fit method comes from the estimation of mass from the volume and density of a specimen. When estimating the mass of engineered systems such as bio-robotics vehicles, as in Figure 3.7b, the estimated mass does not perform well in specific use cases when there is a large discrepancy between the lumped mass and center of gravity versus a more natural mass distribution. This implies that the platform is larger than it needs to be, but such a platform may be needed based on payload requirements. Referring to Table 3.6, it can be seen that the Harvard Beihang Mackerel and the NRL four-fin have similar lengths and mass, but the surface area, and volume by analogy, of the NRL four-fin is almost twice as much as the Harvard Beihang Mackerel. This would cause the mass estimation to increase, since it is based on volume.

Another limitation to the proposed algorithm is the need for two views of the platform for an accurate measurement of the platform's parameters. Often, orthogonal views of the platform are missing from publications. The Ika-Fit method is limited by assuming the "unknown" view by either estimating the view with a NACA profile or constructing a 3D model based on other reference pictures, as done with the European silver eel in Section 3.2. A summary of pros and cons of the Ika-Fit method is given in Table 3.7.

Table 3.7 Pros and Cons of the using the Ika-Fit method.

Pros	Accurate estimation of physical parameters
	Good estimation of mass based on volume and density
	Ability to get an estimate at any length scale
	Only length needed with no other knowledge or physical specimen needed
Cons	Needs side and top views of platform
	Not accurate for a large mismatch between size and mass
	Cross section of platform needs to be circular or elliptical-like

When the Ika-Fit method is applied to (2.6), there is a gap between published and calculated data. This is caused by (2.6) not including other forms of drag present in swimming

animals. The cross-over point at higher velocities is an area of further research in order to formulate a more accurate model for propulsion power. Another finding is that the COT model is very sensitive to hotel power, and so a better estimation for unknown quantities is needed. Shortcomings aside, (2.6) gives a consistent and easy to calculate method to compare with published COT data for natural and artificial swimmers. Further research needs to be done to develop a more accurate COT model that addresses the discrepancies found in this research.

3.3 Concluding Remarks

In this work, a novel methodology for determining physical parameters and COT of artificial and natural swimmers when no COT data exists is developed by extending the core COT model of Phillips *et al.* [3, 69]. The developed Ika-Fit method uses image segmentation techniques and ellipsoid estimations to estimate the surface area and volume of engineered systems. These parameters are used to estimate the COT with a simplified model.

The Ika-Fit method shows high accuracy when measuring the surface area, volume, and mass of fish species and fish-shaped robotic platforms. An accuracy of within 20% for surface area for engineered systems that are not fish-shaped can also be obtained. The estimated COT is shown to underestimate when compared to real-world data. This can be explained partly by the COT model used not including induced drag from wake effects. More research is needed to improve the COT model to reflect real-world data.

This method is useful to a designer of underwater vehicles when mission time and energy efficiency are critical factors for specific applications. The COT for many different platforms can be estimated to obtain a relative energy efficiency in the design phase. The comparison of COT can elucidate whether it is beneficial to design a fish-like underwater vehicle or a propeller-driven AUV for long distance missions. This method can also be useful for biologists who want a way to perform a non-contact estimation of COT for natural swimmers.

Finally, the method can be used to accurately estimate the physical parameters of a design at any length scale. Estimating physical parameters is useful to a designer when considering the amount of payload needed for a mission. This is because a parameter like volume can be estimated and the length, surface area, and mass is readily available. As discussed later in Chapter 6 Section 5.5, this methodology is particularly useful in determining the relative cost of a vehicle and especially when comparing a design to existing platforms.

Chapter 4

Do Computers Dream of Swimming Fish?

4.1 Introduction

In Chapter 2, it was shown that utilizing bio-inspired propulsion does not necessarily mean better performance in terms of energy consumption. In most straight line swimming applications, a conventional AUV performs at or better than biological animals. A benefit to biological animals is that they have superior maneuverability with smaller and faster turn radii. If a roboticist wants to design a bio-inspired robotic fish that is optimized for minimum energy consumption, it is critical to understand what locomotion mode is most efficient. In this chapter, computational fluid dynamics (CFD) is utilized on four different body-caudal fin (BCF) swimmers and the scaling of output power for each swimmer is established.

There are a number of numerical studies of swimming fish, from analytical and potential flow models [207–210] to CFD simulations [43, 211, 79, 212–216], and even utilizing simple vortex panel methods [217–219]. There have been many studies of NACA airfoil using overset meshes and specifically OpenFOAM®'s implementation [220–225]. As pointed out by Chandar [225], there are errors caused by the interpolation of the overset mesh to the background mesh when compared to a single mesh, but the solutions are comparable to Arbitrary Mesh Interface and Generalized Grid Interface techniques [224, 225]. The interpolation error is dependent on the interpolation method with the inverse distance method having the highest error and the polynomial interpolant having the least [225]. There have been studies of a stationary airfoil at steady state conditions [224] and some studies of undulating airfoils using overset meshes [226–228]. Existing research is usually confined within a specific range of Strouhal and Reynolds numbers. The only study undertaken for multiple Strouhal and Reynolds numbers was performed recently by Yu *et al.* [43]. Yu *et al.* covers Reynolds numbers between 10^3 - 10^4 , but the Reynolds numbers of interest for this research encompasses numbers from 10^3 - 10^7 .

In this research, the existing solid body motion solver from OpenFOAM[®] Computational Fluid Dynamics library (CFD) is rewritten for use with an overset grid. Overset grids, or chimera grids, are CFD meshes that have more than one computational domain which are joined together by an interpolation region, explained in more detail in Section 4.3. Moreover, I perform simulations for a variety of Strouhal and Reynolds numbers to understand how fish propulsion scales through several decades of Reynolds numbers.

All of the above studies focus mainly on anguilliform and carangiform motion. This research is interested in four types of BCF motion; anguilliform, carangiform, thunniform, and ostraciiform, as discussed in Section 4.2. These four modes are crucial in understanding what bio-robotic designs may be more efficient and roughly translates to a number of linkages for a given propulsion design. The next section gives a review on how fish motion is modeled, the concept of an overset mesh, and how the OpenFOAM[®] library is modified to simulate fish motion. Specific findings from performing the simulations and results are discussed in the last Section of this chapter.

4.2 BCF Motion

Among BCF swimmers, shown in Fig. 2.2 of Section 2.2, the carangiform motion is most popular for bio-inspired robots according to Table 2.1 of the same Section. To understand the difference between the different types of BCF swimming modes, a visual comparison of the BCF swimmers is given in Fig. 4.1. BCF propulsion is characterized by a undulating wave that travels down the body. The difference between the different BCF swim modes is the amount of the body that contributes to propulsion. The far left of Fig. 4.1 is the anguilliform mode. This mode has the whole body undulating versus the Ostraciiform on the far right of Fig. 4.1, where only the last 15% of the body oscillates. The modes become increasingly oscillatory from left to right.

The motion of the fish can be described as a prescribed deformation of the midline, shown in Fig. 4.1, in the transverse direction. It is convention to write this motion as $h(x, t)$, where x is a coordinate down the midline of the swimmer and t is the time. Tytell [186] provides kinematic data for anguilliform motion using high speed film and DPIV of American eels. Videler [8] proposed a model using the high speed film technique much earlier for subcarangiform and carangiform swimmers. For thunniform motion, Dewar [229] was able to perform a high speed film study on tuna.

In general, Videler and Nolet [8] describe the modeling BCF locomotion by fitting an envelope function using the least squares method with a backward traveling wave. The general form of this equation is

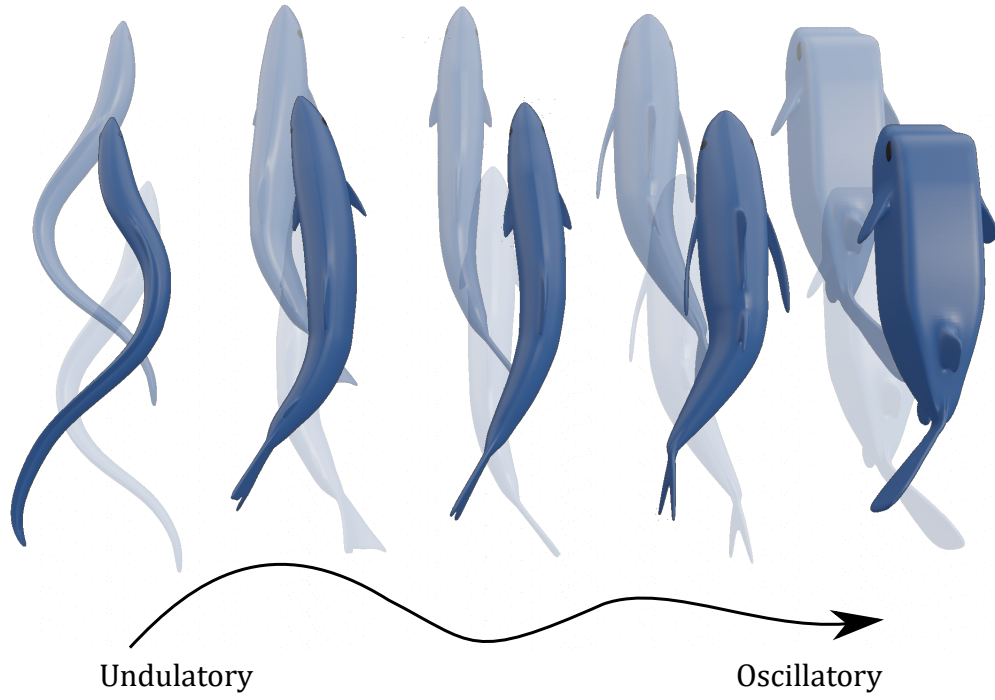


Fig. 4.1 Comparison of one cycle of the all BCF swimmers.

$$h(x, t) = (c_0 + c_1x + c_2x^2) \sin(kx - \omega t), \quad (4.1)$$

where $h(x, t)$ is the motion down the fish midline, signified by the x coordinate at time t , shown in Fig. 4.2a. c_0 , c_1 , and c_2 are the least squares fit coefficients. k is the wave number that describes the amplitude of undulation given by

$$k = \frac{2\pi}{\lambda}, \quad (4.2)$$

where λ is the wavelength. ω is the frequency of undulation or oscillation given

$$\omega = \frac{2\pi V}{\lambda}, \quad (4.3)$$

where V is the body wave speed, and t is time. A simple way to fit these coefficients to an envelope are presented by Xia [230] by formulating the coefficients into a matrix and multiplying by the inverse which is described as follows:

$$\begin{bmatrix} 1 & x_h & (x_h)^2 \\ 1 & x_c & (x_c)^2 \\ 1 & x_t & (x_t)^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} A_h \\ A_c \\ A_t \end{bmatrix}, \quad (4.4)$$

where A_h is the the amplitude of the snout, A_c is the amplitude of the mass center, and A_t is the tail amplitude at the peduncle. The location of the snout is x_h , the location of the mass center is x_c , and the location of the tail peduncle as x_t . The coefficients of equation (4.1) can be solved by solving the set of linear equations or by inverse matrix multiplication. Different literature suggest different coefficients for the envelopes depending on the motion types.

Maertens *et al.* [211] proposed a different format for equation (4.1) that takes on the form:

$$h(x, t) = a \left(1 + c_1(x - 1) + c_2(x^2 - 1) \right) \sin(kx - \omega t), \quad (4.5)$$

where a is half the peak-to-peak amplitude of the tail. This formulation allows for the adjustment of the tail amplitude without resolving the coefficients. Coefficients used in (4.5) will be discussed in the following sections and (4.5) will be used for each type of motion. For clarity, a shape function in terms of x is given as $A(x)$ which represents the amplitude function. In this form, each equation for the fish midline takes the form of

$$h(x, t) = A(x) \sin(kx - \omega t) \quad (4.6)$$

where k is defined in (4.2) and ω is defined in (4.3).

4.2.1 Anguilliform Motion

From the kinematic data given by Tytell [186], the amplitude for anguilliform swimmers can be modeled as

$$A(x) = ae^{\alpha \left(\frac{x}{L} - 1 \right)} \quad (4.7)$$

where a is the tail beat amplitude, α is the amplitude growth rate, x is the contour length down the midline of the swimmer starting at the head, and L is the body length. Here a large α gives a rapid increase in undulation towards the tail and a small α implies more undulation

near the head. Figure 4.2b shows the time progression of an anguilliform swimmer's midline motion as defined in (4.7).

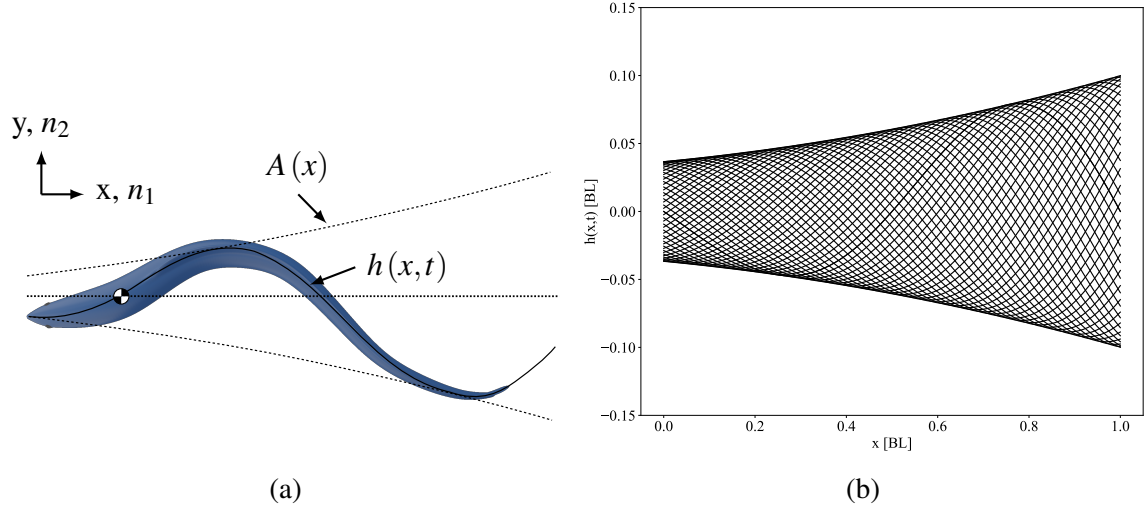


Fig. 4.2 (a) Diagram showing the envelope and motion of an anguilliform swimmer. (b) Midline motion of an anguilliform swimmer at different times steps for a is set to 0.1, α is set to 1.

The literature around numerical experiments of anguilliform swimmers shows that a reasonable approximation for modeling anguilliform swimming is to set A to 0.1 and α to 1 [43, 211, 212, 231, 232]. While only [211, 212] used the data from Tytell directly for their simulations, the others performed a least squares fit on the envelope and adapted it to the polynomial envelope proposed by Videler and Nolet [8]. In the notation proposed by Maertens *et al.* [211], the coefficients from the least squares fit are

$$a = 0.1 \quad c_1 = 0.323 \quad c_2 = 0.310. \quad (4.8)$$

These coefficients are analogous to setting A to 0.1 and α to 1 and a comparison of the two amplitudes is provided in Fig. 4.3. There are two options available to model the amplitude of anguilliform motion. The first is to use the carangiform motion function and the coefficients in (4.8). The second option is to use the anguilliform motion function that implements the equation from Tytell. Translating these equations to an airfoil for visualization purposes is shown in figure 4.4.

4.2.2 Carangiform Motion

Carangiform and sub-carangiform motion is the most common form of BCF locomotion. This mode is characterized by the undulation of approximately 25 to 75% of the total body and

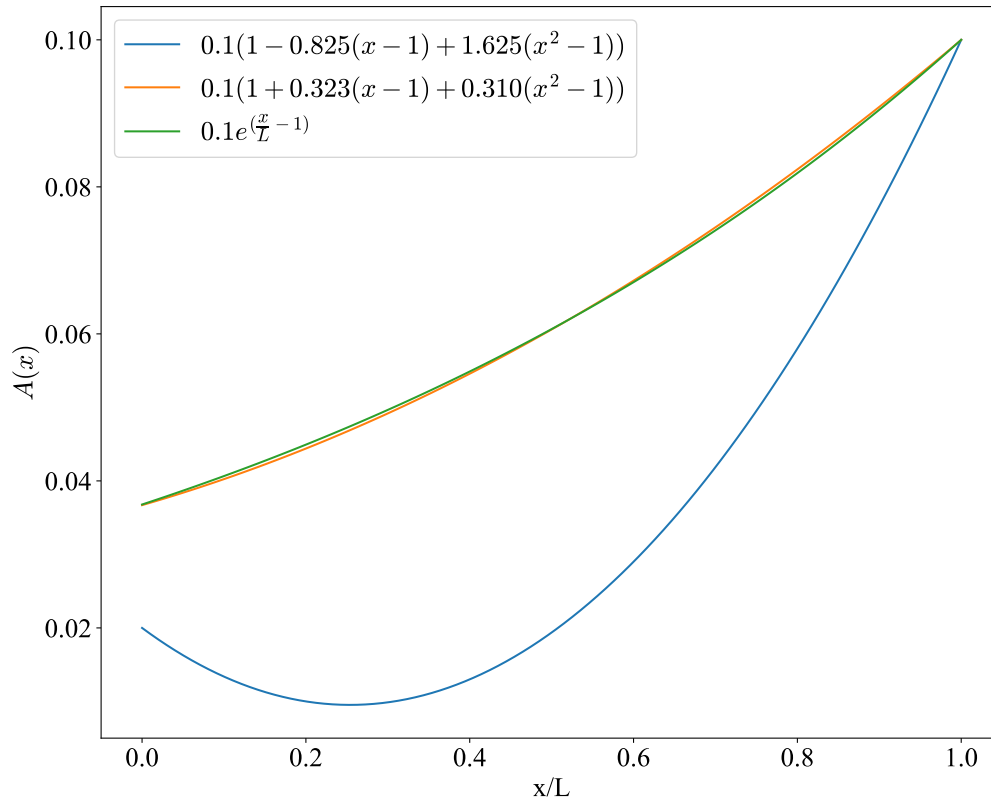


Fig. 4.3 Comparison of shape functions for anguilliform and carangiform mode. Amplitude function given in legend.

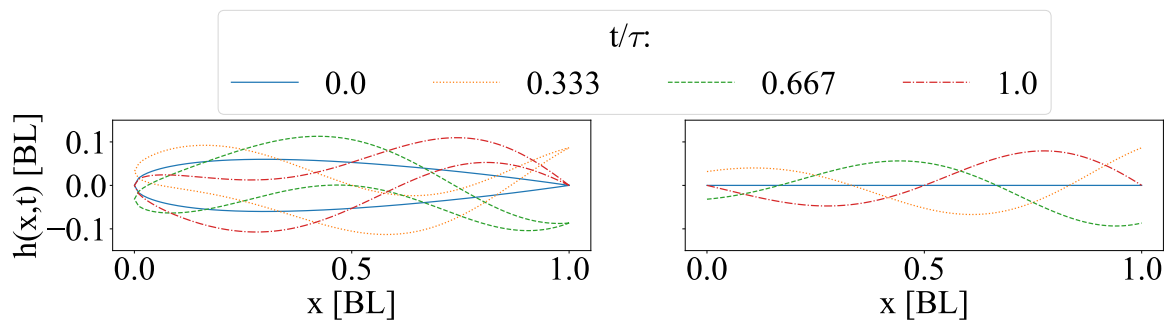


Fig. 4.4 Visualization showing (a) shape of an airfoil swimming and (b) shape of the midline during anguilliform motion. t/τ represents the time (t) over 1 tail beat (τ).

caudal fin [233]. The amplitude envelope for sub-carangiform and carangiform swimmers was first given by Videler and Hess [234] using high speed images of fish swimming. The midline motion can be modeled as

$$A(x) = a \left(1 + c_1(x - 1) + c_2(x^2 - 1) \right), \quad (4.9)$$

with the coefficients used for the amplitude envelope as

$$c_1 = -0.825 \quad c_2 = 1.625, \quad (4.10)$$

where k and ω are as used before. In a majority of numerical studies for carangiform motion, $\frac{\lambda}{L}$ is set to 1.0, where L is the length of the foil, and $\frac{V}{\lambda}$ are grouped into a frequency of undulation f with units Hz [43, 211, 212, 215, 235].

From data collected by Videler [42], the tail amplitude varies between $0.05L$ and $0.14L$ with an average value of $0.09L$. The average wave length of the fish λ is $0.92L$ with a standard deviation of $0.10L$. Figure 4.5 shows the traveling motion wave and the midline motion at multiple time steps.

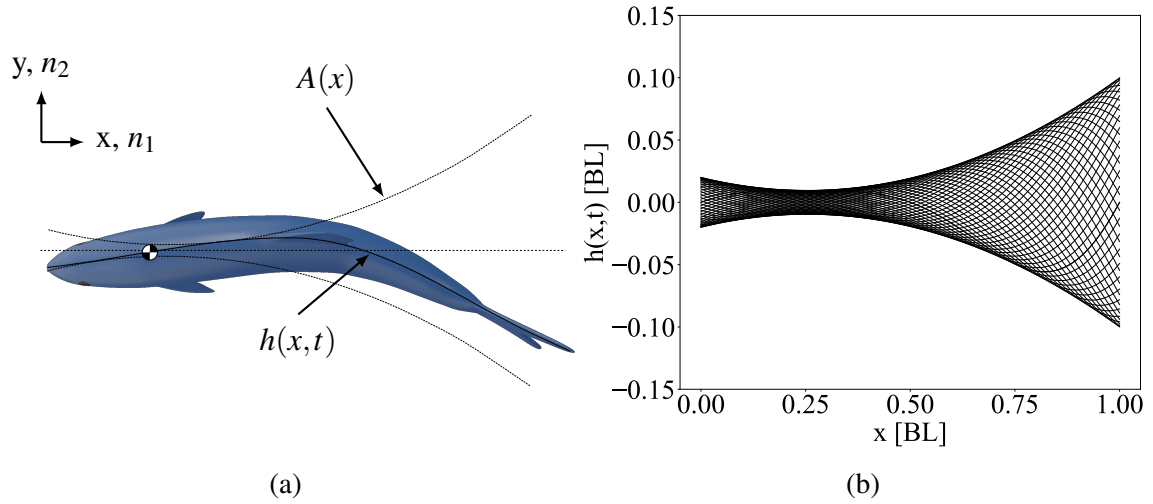


Fig. 4.5 (a) Diagram showing the envelope and motion of an carangiform swimmer. (b) Midline motion of an carangiform swimmer at different times steps. a is 0.1 and coefficients of (4.10)

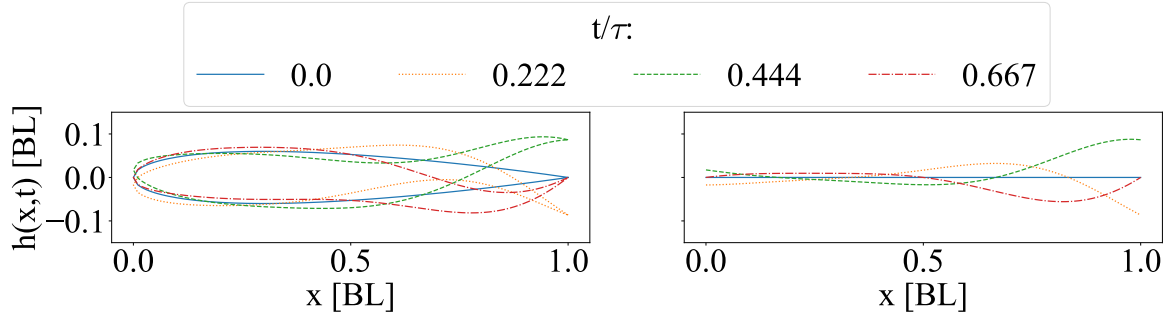


Fig. 4.6 Visualization showing (a) shape of an airfoil swimming and (b) shape of the midline during carangiform motion. t/τ represents the time (t) over 1 tail beat (τ).

4.2.3 Thunniform Motion

Thunniform motion is widely considered the most efficient form of fish locomotion [148, 235]. This mode is characterized by 90% of the propulsive force being generated by the caudal fin with the rest being generated by body undulations [235]. The caudal fin is modeled as pitching foil with angle $\phi(t)$, as done by Barrett [148] and Zhu *et al.* [30]. The body undulations are modeled the same as carangiform motion (4.5) which makes the total body motion given as

$$h(x, t) = A(x) \sin(kx - \omega t), \quad (4.11)$$

$$\phi(t) = \phi_{max} \sin(kx_{pivot} - \omega t + \psi), \quad (4.12)$$

where c_1 , c_2 , k , and ω are defined the same as (4.5) and (4.10). ϕ_{max} is the maximum pitching angle and ψ is the phase angle, and x_{pivot} is defined by the user but represents the pivot point of the caudal fin normalized by the fish body length.

The $h(x_{pivot}, t)$ position is the lateral deflection of the pivot point and is given as

$$h(x_{pivot}, t) = a(1 + c_1(x_{pivot} - 1) + c_2(x_{pivot}^2 - 1)) \sin(kx_{pivot} - \omega t), \quad (4.13)$$

and the midline deflection of the caudal fin is given by

$$h(x_{foil}, t) = h(x_{pivot}, t) + x_{pivot} \tan \phi(t) \quad (4.14)$$

Fig. 4.7 shows the diagram of thunniform motion and the midline displacement and a visualization of the airfoil undergoing this type of motion is given in Fig. 4.8.

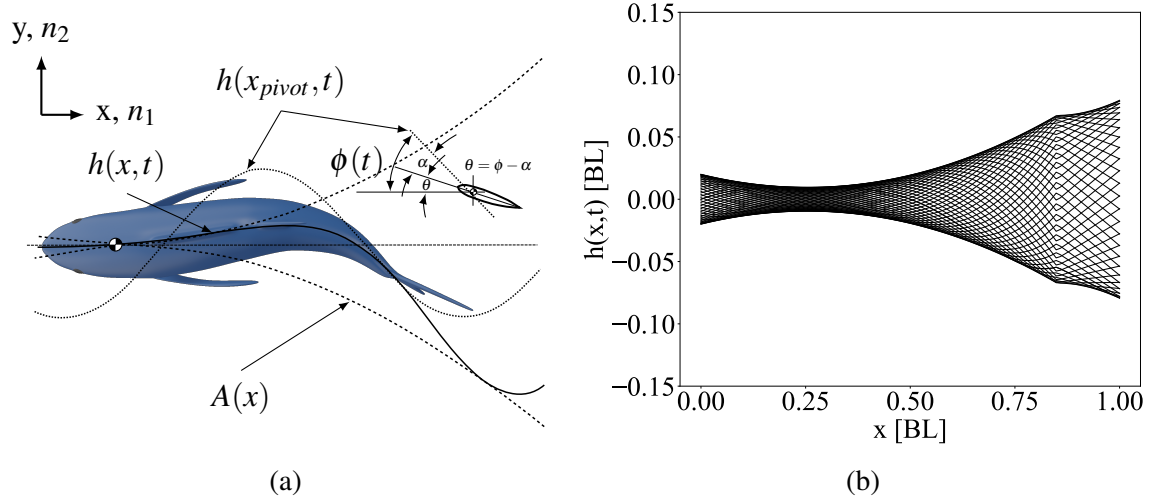


Fig. 4.7 (a) Diagram showing the envelope and motion of an thunniform swimmer. The pitching foil is shown inset to the right. (b) Midline motion of an thunniform swimmer at different times steps. a is 0.1 and coefficients of (4.10)

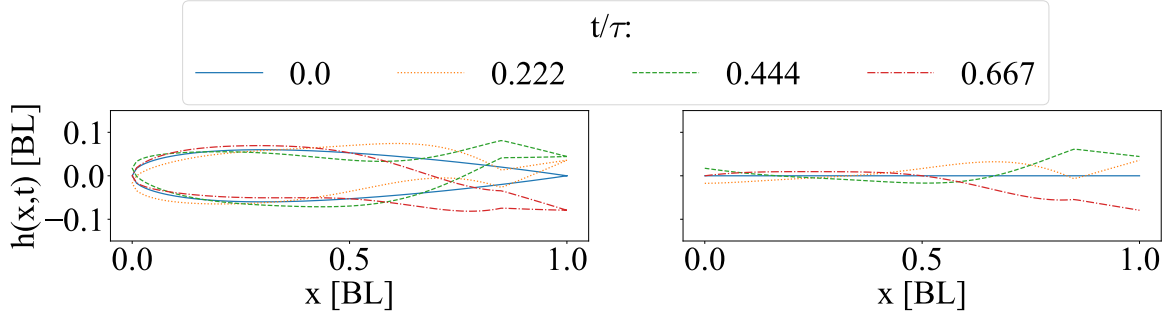


Fig. 4.8 Visualization showing (a) shape of an airfoil swimming and (b) shape of the midline during carangiform motion. t/τ represents the time (t) over 1 tail beat (τ).

4.2.4 Ostraciiform Motion

Ostraciiform motion is characterized by a flapping of just the caudal fin. Fish using this locomotion mode normally utilize MPF propulsion, but it is popular among roboticists as it has an easily implemented mechanism for propulsion. The motion of the caudal fin is given in (4.14) with the parameters being the same. To account for the motion of the head due to inertia, a slope function given by the lateral deflection of the head and pivot point is used. $h(x_{head}, t)$ and $h(x_{pivot}, t)$ are given by

$$h(x_{head} = 0, t) = A(0) \sin(0 - \omega t), \quad (4.15)$$

$$h(x_{pivot}, t) = A(x_{pivot}) \sin(kx_{pivot} - \omega t), \quad (4.16)$$

where the lateral deflection of the body can be solved using the point slope form for a linear line

$$h(x_{body}, t) = \frac{h(x_{pivot}, t) - h(0, t)}{(x_{pivot})}x + h(0, t). \quad (4.17)$$

Fig. 4.9 shows the implementation of this model onto the Ostraciiform swimmer and Fig. 4.10 shows the deflection of the airfoil.

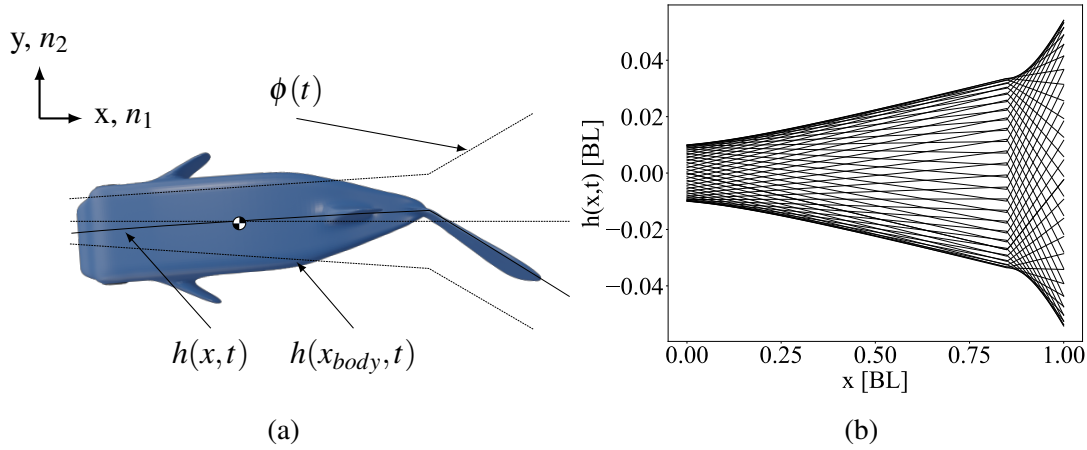


Fig. 4.9 (a) Diagram showing the envelope for the body and tail. (b) Midline motion of an ostraciiform swimmer at different times steps.

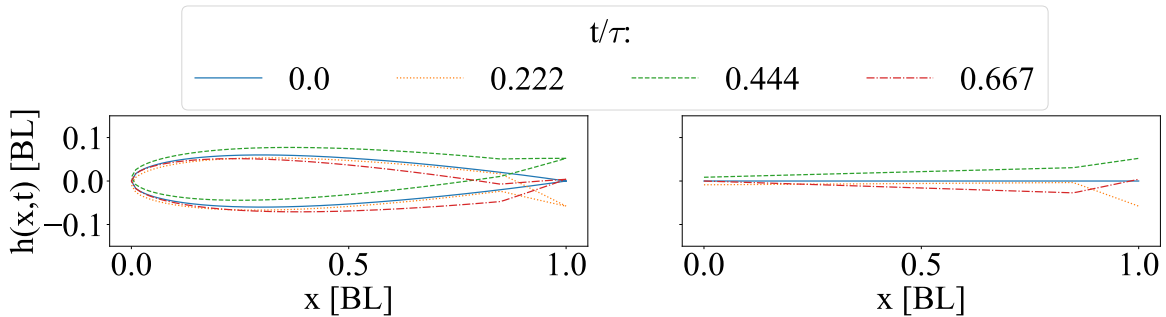


Fig. 4.10 Visualization showing (a) shape of an airfoil swimming and (b) shape of the midline during carangiform motion. t/τ represents the time (t) over 1 tail beat (τ).

4.3 Overset Grid

Much of the CFD simulations involving fish locomotion utilize a body fitted mesh. This is a single mesh that conforms to the body of the swimmer and expands into the free stream. In contrast, this research utilizes an overset mesh. Overset meshes involve 2 or more mesh that are merged on top of each other [221, 222]. The meshes are considered as layers and a stencil is used to interpolate the values of one mesh to the other. An overview of the type of cells is given in the OpenFOAM® documentation [144] and are given as:

- Donor cells: cells that provide values
- Acceptor cells: cells whose value gets set from interpolation
- Hole cells: unreachable or inactive cells

Overset mesh was chosen due to patch motion for body fitted meshes not working as well when there is not a large domain to diffuse the movement of the mesh nodes. The diffusion equation causes negative volume where the meshes close to the swimmer overlap with neighboring meshes. The author notes that this could be achieved with point patch movement using a spring like motion solver or remeshing techniques as would be done in Fluent [223]. This research seeks to perform simulations without any extra tools and therefore an overset mesh is the best solution.

There are a number of interpolation methods that can be used to interpolate from one mesh to another. This research uses the inverse distance method for overset interpolation. Tisovska gives a description of how the inverse distance method is implemented in OpenFOAM® [236]. The procedure for interpolation first determines the weights to each cell center by summing the inverse distances as follows:

$$S = \sum_i^n \frac{1}{|d_i|}, \quad (4.18)$$

where n is the total number of donors and d_i is the distance from the donor cell center to the center of the acceptor cell. The weights for each cell can then be calculated using:

$$w_i = \frac{1}{|d_i|S}, \quad (4.19)$$

where w_i is the weight of the i -th cell on the acceptor cell. The interpolated value is then calculated using:

$$\phi = \sum_i^n w_i \phi_i, \quad (4.20)$$

where ϕ is the field to be interpolated and n is the number of donors, as in (4.18). Figure 4.11 shows a schematic of this cell interpolation.

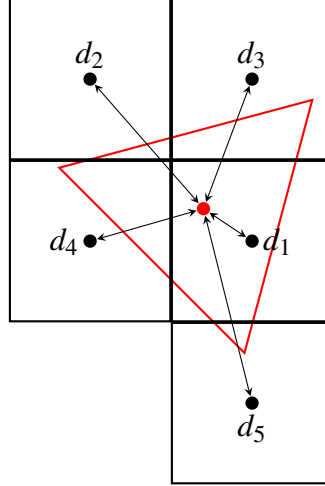


Fig. 4.11 Schematic showing overset interpolation from the source grid (donor cells in black) to the destination grid (acceptor cell in red).

Using this scheme, there are three types of cells types that are mapped to the mesh. From OpenFOAM[®] documentation [144] these are:

- calculated: cells where the equations are solved
- interpolated: cells that get their value interpolated from the solution cells
- holes: cells which values are not used

Figure 4.12 shows the "airfoil" mesh with the overset cell types colored. Blue cells are calculated, white cells are interpolated, and red cells are holes.

4.4 Computational Domain

The computational domain is made up of the background mesh region, 2 refinement regions, and an overset region. The overset mesh is generated using Ansys meshing software [237] by manually defining the cell sizes in each region. In Section 4.9, a comparison is conducted with a mesh created with GMSH [238]. This is a new comparison in the context of fish swimming. The background mesh, or fluid domain, is generated using OpenFOAM[®] blockMesh utility with the refinement regions made by running the refineMesh utility twice with different bounds. This has the effect of creating 2 mesh regions around the airfoil with finer resolutions. Figure 4.13 shows a schematic diagram of the entire computational domain.

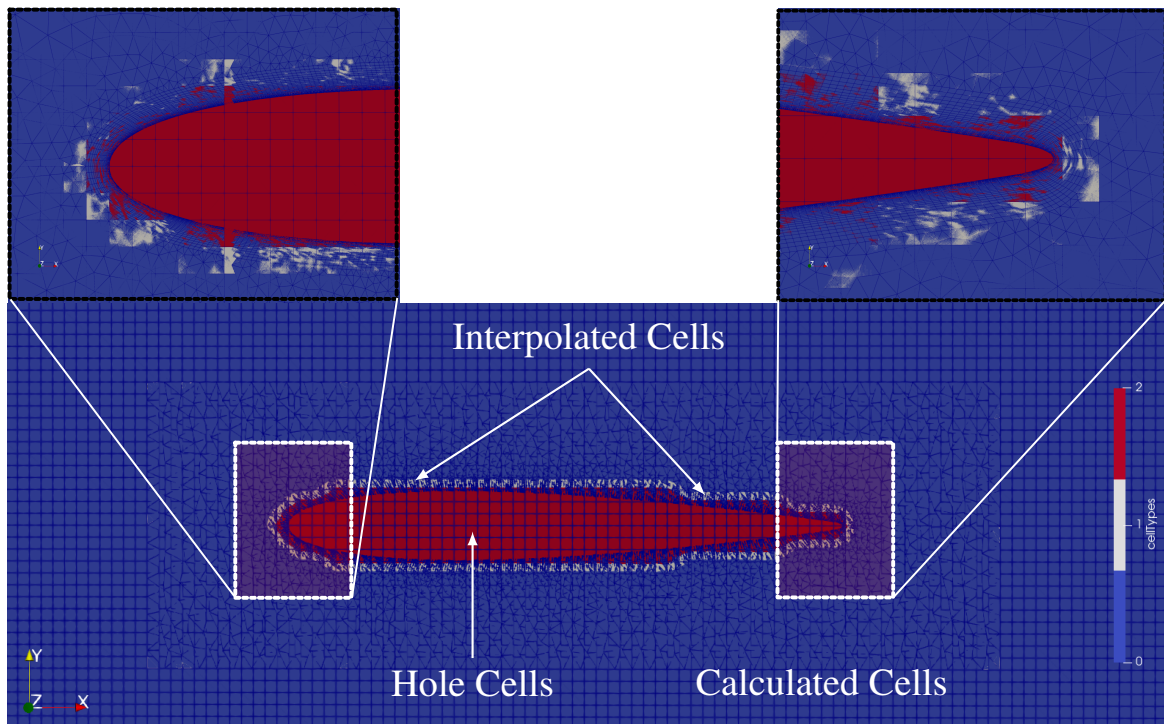


Fig. 4.12 Overset mesh showing the types of cells as rendered by Paraview [77]. Blue cells represent the calculated cells, red cells are the holes, and white cells represent the interpolation region. The number assigned to the cells is given by the cell types colormap, 2.0 for holes, 1.0 for interpolated, and 0.0 for calculated. Red cells outside the fish airfoil are cut from the background mesh and are an artifact from the rendering of the two meshes on top of each other.

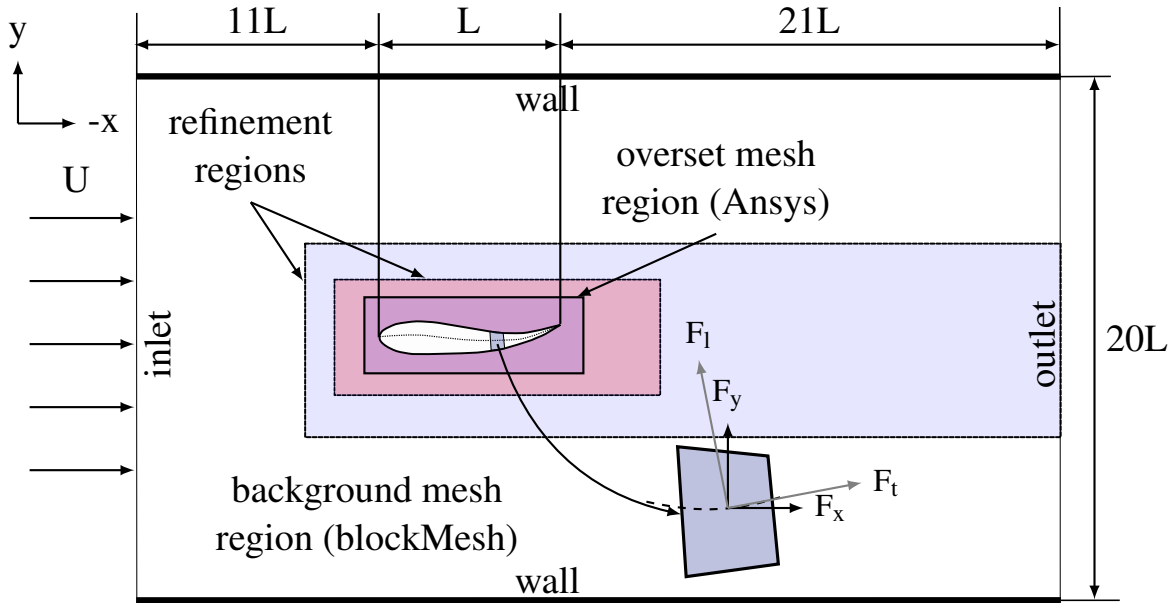


Fig. 4.13 Schematic of computational fluid domain. Overset region is shown around the "airfoil", 2 refinement regions are used as shaded in red and blue. Note that the coordinate system is set so that $-x$ is in the direction of flow as is customary for fish simulations.

Guidelines for overset mesh generation say that the cell size from the overset region to the background mesh should be the same size [239, 221]. For this reason, the overset mesh was constructed first. The background mesh with refinement regions was changed to match as closely to the size of the overset mesh as possible. Since the overset mesh is made of tetrahedral elements and the background mesh is constructed using hexahedral elements, the area of each cell in the overset region is roughly 2 of the background refined region. Following the guidelines of Guerrero [78], the overset mesh is built to ensure that there are more than 3 layers of elements above between the hole region and the background mesh. A mesh independence study is provided in Section 4.9.3.

Following the guidelines for external flow simulations given by Goetten *et al.* [240], the trailing edge of the foil is made round and the "airfoil" is forced to be 250 elements for the top and the bottom. Goetten *et al.* further offers guidelines on the extent of the fluid domain. The group recommends having longitudinal and lateral domain extents greater than 200 reference lengths [240]. Many of the current undulating airfoil simulations do not follow the recommended guidelines for their simulation. For example, a common validation case given by Dong and Lu [79] has domain extents of $[-2, 25]$ chord lengths in the x direction and $[-6, 6]$ in the y -direction. In lieu of using extents greater than 200 chord lengths, the current domain is $[-11, 22]$ in the x direction and $[-10, 10]$ in the y direction. The lateral

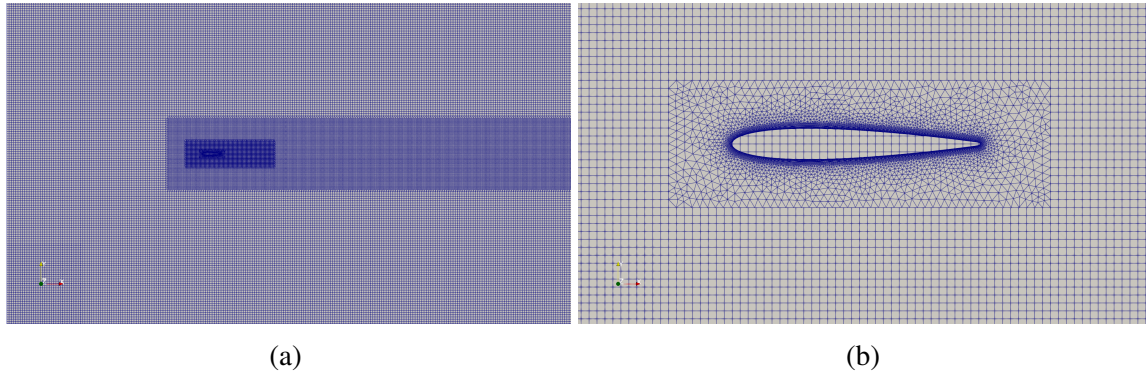


Fig. 4.14 (a) Overview of the mesh with background mesh and two refined regions encompassing the overset mesh and (b) a zoomed view of the overset mesh showing the interpolation region of the overset to the background mesh being near the same size.

dimensions result in a blockage ratio of 0.6%. The reason for a smaller domain is that the number of mesh elements increases by 30 times when comparing to a domain with extents of 200 chord lengths giving an element count of around ≈ 4.2 million severely increasing the computational demand. A summary of the final grid is given in Table 4.1 that includes the dimensions of the mesh as well as mesh metrics. More information on mesh metrics is provided by Knupp [241].

Table 4.1 Summary of the mesh composition and metrics for the overset and background mesh. Metrics were given by OpenFOAM[®] checkMesh utility with allGeometry and allTopology flags turned on.

Component	Size	
Overset domain [mm]	20	
Airfoil edges (top/bottom) [divisions]	250	
Inflation first layer [mm]	$5.0e^{-02}$	
Number of inflation layers	35	
Inflation growth rate	1.12	
Metric	Airfoil Mesh	Combined Mesh
Number cells	22,556	167,420
Min aspect ratio	80.20	80.20
Max skewness	1.68	1.68
Max non-orthogonality	40.78	40.78
Average non-orthogonality	15.91	6.18

4.5 Ika-Flow Implementation

To implement the motion solver in OpenFOAM[®], the solid body motion solver is rewritten to give the prescribed motion of fish as described in Section 4.2. The solid body motion solver is chosen because it properly determines the overset mesh zone from the background mesh. The changes to the solid body motion class are shown in figure 4.16 using the unified modeling language (UML) [242].

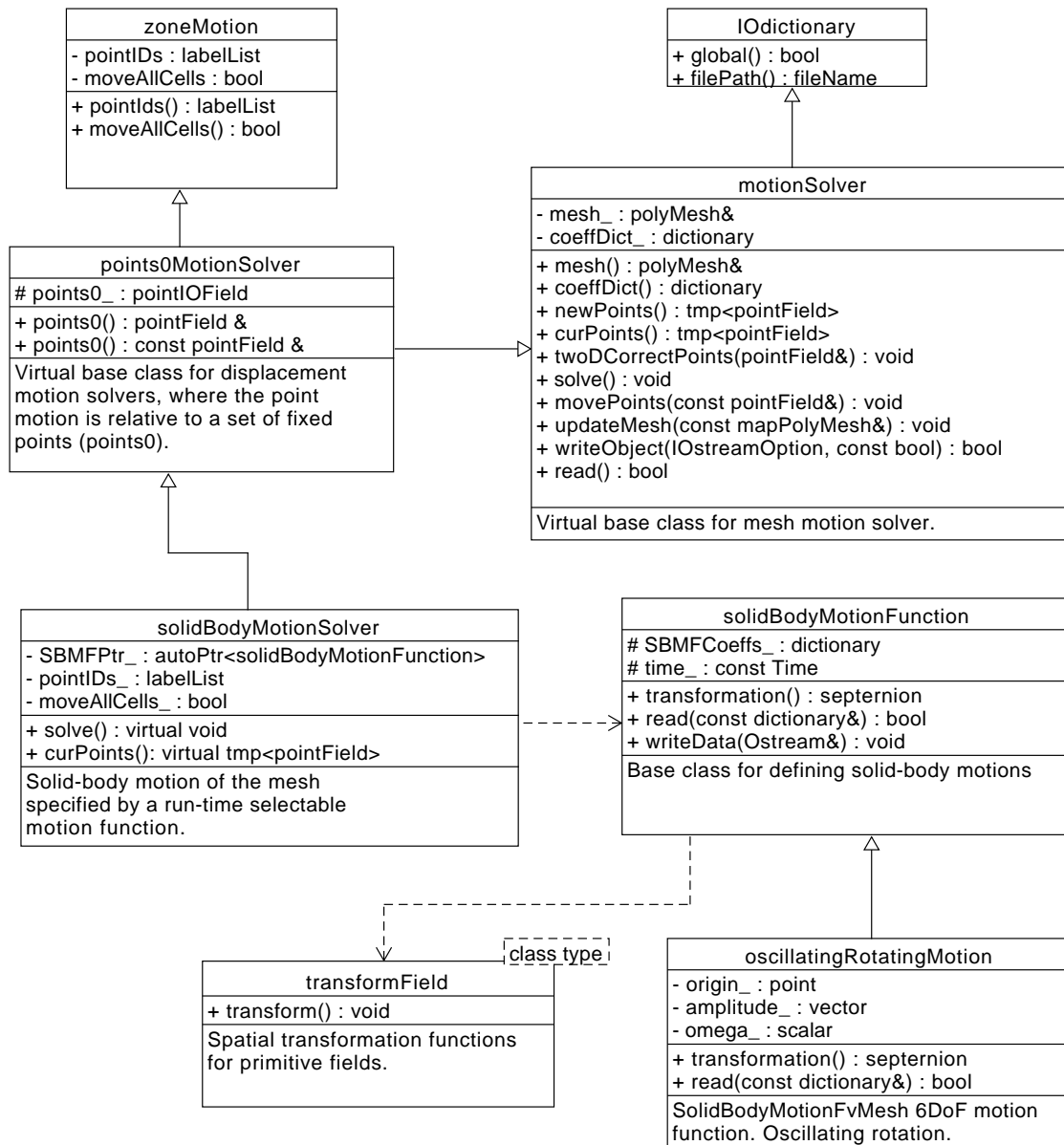


Fig. 4.15 Original Solid Body class in UML

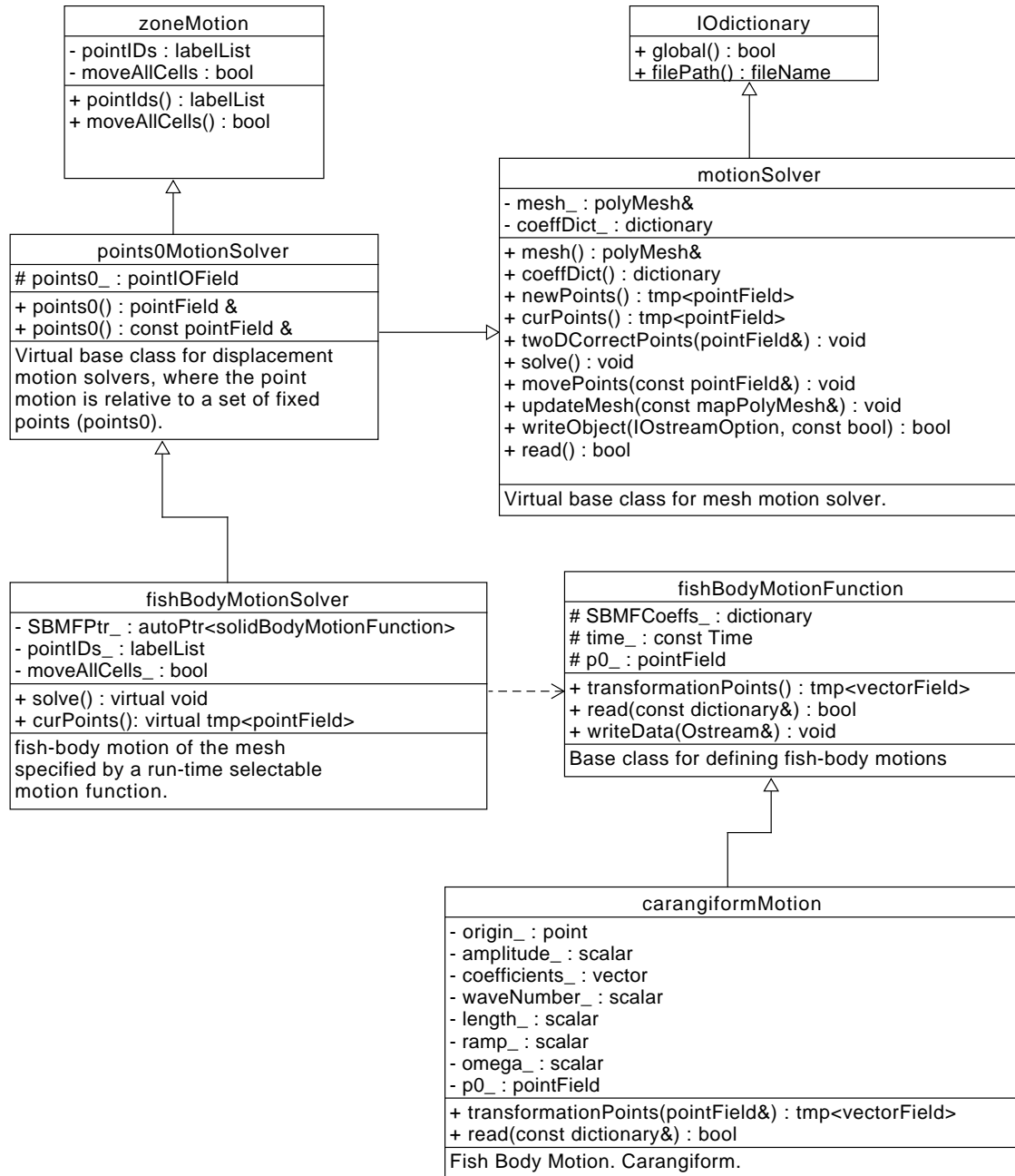


Fig. 4.16 Modified fish body class in UML. The new class does not call the transformField class, but instead provides it's own transformation class inside appropriate motion function. This can be seen when comparing the oscillatingRotatingMotion class to the carangiformMotion class.

Listing 4.1 Ika-Flow motion solver showing the Thunniform motion.

```

Foam::tmp<Foam::pointField>
Foam::fishBodyMotionFunctions::thunniformMotion::
transformationPoints(pointField& p0) const
{
    const scalar tm = time_.value();

    if (delay_ <= tm)
    {
        scalar t = tm - delay_;

        scalar maxAngle = maxAngle_ * 2 * M_PI / 180;
        scalar phase = phase_ * 2 * M_PI / 180;

    forAll(p0, pointI)
    {
        const scalar x = (p0[pointI].component(0)-origin_[0])/length_;
        const scalar y = p0[pointI].component(1)-origin_[1];
        const scalar z = p0[pointI].component(2)-origin_[2];

        scalar yr = 0;

        // check if the x-coordinate is larger than the pivot
        if (x >= pivot_)
        {
            //new value by equation
            const scalar xPivot = x - pivot_;

            const scalar localAmplitude = amplitude_ * (1 + (coefficients_[0]*(pivot_-1)) + (coefficients_[1]*(pivot_*pivot_-1)));
            const scalar yEnd = localAmplitude * sin(waveNumber_*pivot_ - omega_*t + phase);

            // calculate the tail angle
            const scalar thetaT = maxAngle * sin(waveNumber_*pivot_ - omega_*t + phase);

            yr = y + yEnd + xPivot * tan(thetaT) * length_;
        }
        else
        {
            // normal carangiform body equation
            const scalar localAmplitude = amplitude_ * (1 + (coefficients_[0]*(x-1)) + (coefficients_[1]*(x*x-1)));
            yr = y + localAmplitude * sin(waveNumber_*x - omega_*t) * length_;
        }
        p0[pointI] = vector(x, yr, z);
    }
    return p0;
}
else
{
    return p0;
}
}

```

Figure 4.16 shows that the call to `transformField` is replaced by a custom fish motion method. The solid body motion solver was used because it allows for the accurate selection of the overset grid points which is facilitated by the `points0MotionSolver` and `zoneMotion` classes. The motion implementation in the `transformationPoints` method is given as follows:

All variables are provided in the `dynamicMeshDict` dictionary file within the constant directory of the simulation case. Coefficients and variables for anguilliform and carangiform swimming modes are provided by well established research [42, 186, 211]. Thunniform and ostraciform coefficients and parameters are based on the robotic operating parameters given by Zhu *et al.* [30]. Table 4.2 gives a summary of the entries used in the dynamic mesh dict for this research and to generate Fig. 4.22.

The ramp entry pertains to a logistics function applied to the tail end of the overset mesh described as

Table 4.2 Summary of dynamicMeshDict entries for all four BCF locomotion modes. All angles are in degrees.

Entry	Anguilliform	Carangiform	Ostraciiform	Thunniform
origin	(0.0 0.0 0.0)	(0.0 0.0 0.0)	(0.0 0.0 0.0)	(0.0 0.0 0.0)
amplitude	0.10	0.10	0.10	0.10
coefficients	(0.323 0.310 0.0)	(-0.825 1.625 0.0)	(-0.825 1.625 0.0)	(-0.825 1.625 0.0)
waveNumber	6.28	6.28	6.28	6.28
length	1.00	1.00	1.00	1.00
ramp	10.00	10.00	10.00	10.00
omega	12.57	12.57	12.57	12.57
delay	5.00	5.00	5.00	5.00
pivot			0.85	0.85
maxAngle			16.00	16.00
phaseAngle			90.00	90.00

$$0.5 - 0.5 \tan(\text{ramp} \cdot x - (\text{ramp} + 9)), \quad (4.21)$$

where ramp is the entry in the dynamicMeshDict and x is the x-coordinate of the mesh. This is used to limit the movement tail end of the mesh to reduce mesh skewness problems. This function is not implemented in the thunniform and ostraciiform functions as the mesh skewness is linear in the tail region for these two swimming modes. The right oversight mesh in Fig. 4.17 shows an overdamped mesh where the left mesh shows the mesh used for this research. A ramp factor of 10 was experimentally found to be the optimal value in terms of limiting mesh skewness in the tail region without interfering with tail movement.

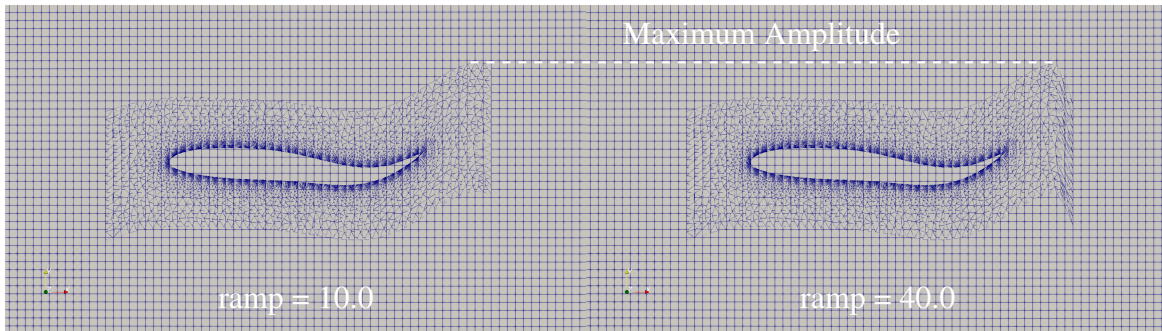


Fig. 4.17 Two meshes showing the effect of the damping factor, controlled by the ramp entry in the dynamicMeshDict. The left mesh shows a ramp factor of 10.0 and the right mesh shows a ramp factor of 40.0.

4.6 Numerical Setup

For the simulations performed in this research, the 2003 variant of the $k-\omega$ shear stress turbulence (SST) model presented by Menter *et al.* [243] is used. This model has been widely used for previous fish CFD simulations [43, 79, 211–216] and has shown superior results compared to other Unsteady Reynolds Averaged Navier Stokes models such as $k-\epsilon$ and Spalart-Allmaras in a study on vertical axis wind turbine blades by Meana-Fernández *et al.* [244]. In Section 4.9, there is a comparison between three well known turbulence models: the $k-\omega$ SST model, the $k-k_L-\omega$ given by Fürst *et al.* [245], and the Spalart-Allmaras model given by Spalart and Allmaras [246]. The Spalart-Allmaras model is a well used model for external aerodynamic flows and the $k-k_L-\omega$ model is a recently developed model that shows accurate results especially in transitional flows [245, 247]. To the author's knowledge, there has not been another comparison of these three models in the context of fish swimming. A derivation of all models is provided in Appendix C.

Guerrero [78] provides guidelines for performing overset simulations in OpenFOAM®. It states that there should be at least 5 PIMPLE iterations for dynamic meshes. PIMPLE iterations refers to the number of iterative marching steps performed after the PISO algorithm. This is controlled in the fvSolution file as the nOuterCorrectors entry. An overview of the difference between the PISO and PIMPLE algorithm is given in Fig. 4.23. Table 4.3 gives the schemes used in the fvSchemes file for this research. For overset mesh, it's recommended that the turbulent values (k , ω , ν_t) be interpolated explicitly [78]. Boundary conditions are calculated using OpenFOAM® initialization guidance [144] and a derivation is included in the Appendix C.5.

The temporal control of the simulation was performed using the adjustable time step feature and keeping the max Courant number below 0.5. The Courant–Friedrichs–Lewy, CFL , condition was introduced by Courant, Friedrichs, and Lewy in 1928 and later translated to English in [248]. It can be written as:

$$CFL = \frac{U \Delta t}{\Delta x} \leq CFL^{max}, \quad (4.22)$$

where U is the fluid velocity, Δt is the time step, and Δx is the smallest dimension of a mesh cell in the direction of fluid flow. In practice, this CFL number is normally kept to be less than 1.0 which means that the fluid cannot jump the smallest mesh cell within a timestep. A comparison of different maximum Courant numbers, CFL^{max} , is given in Section 4.9.

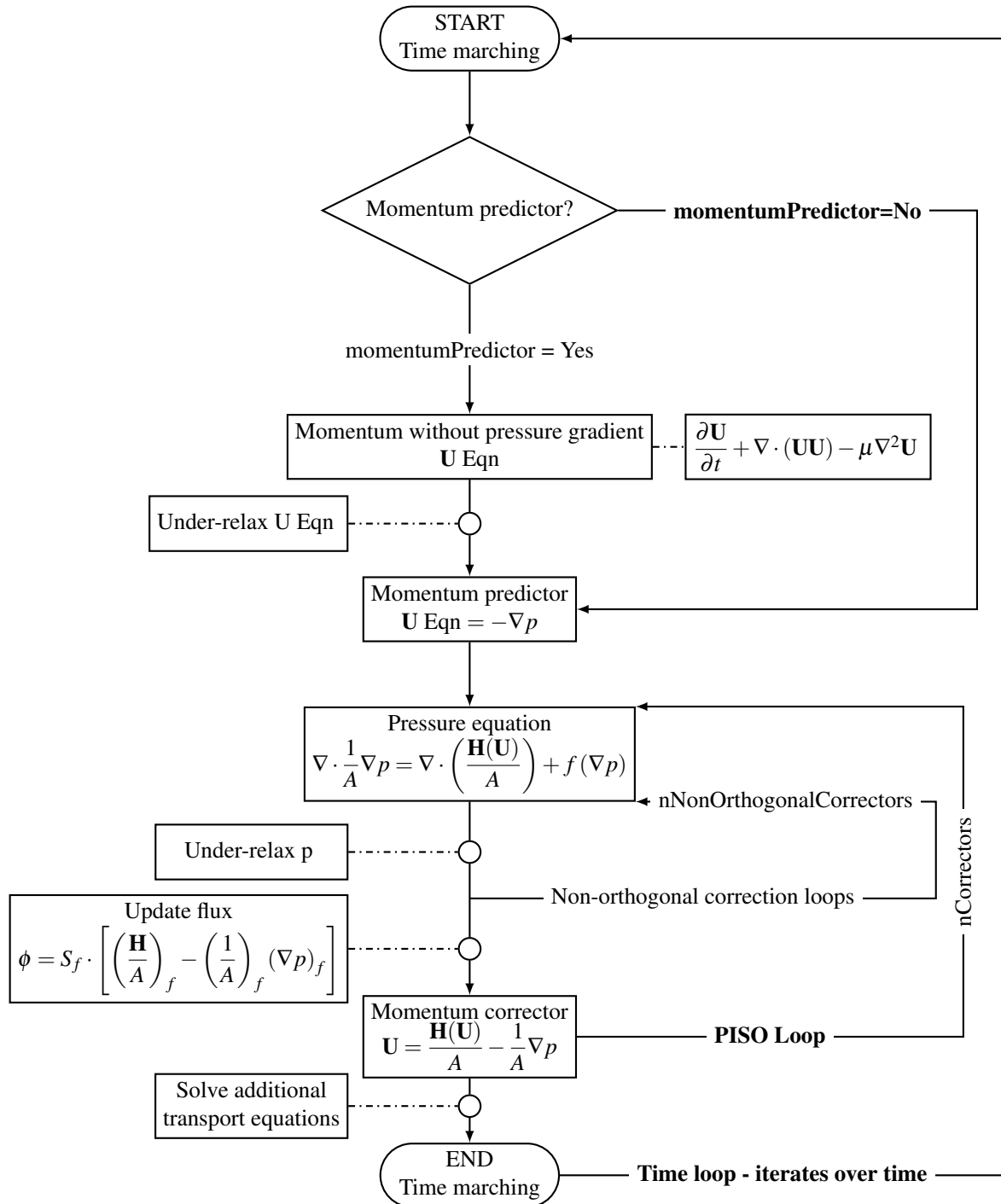


Fig. 4.18 PISO algorithm with non-iterative marching

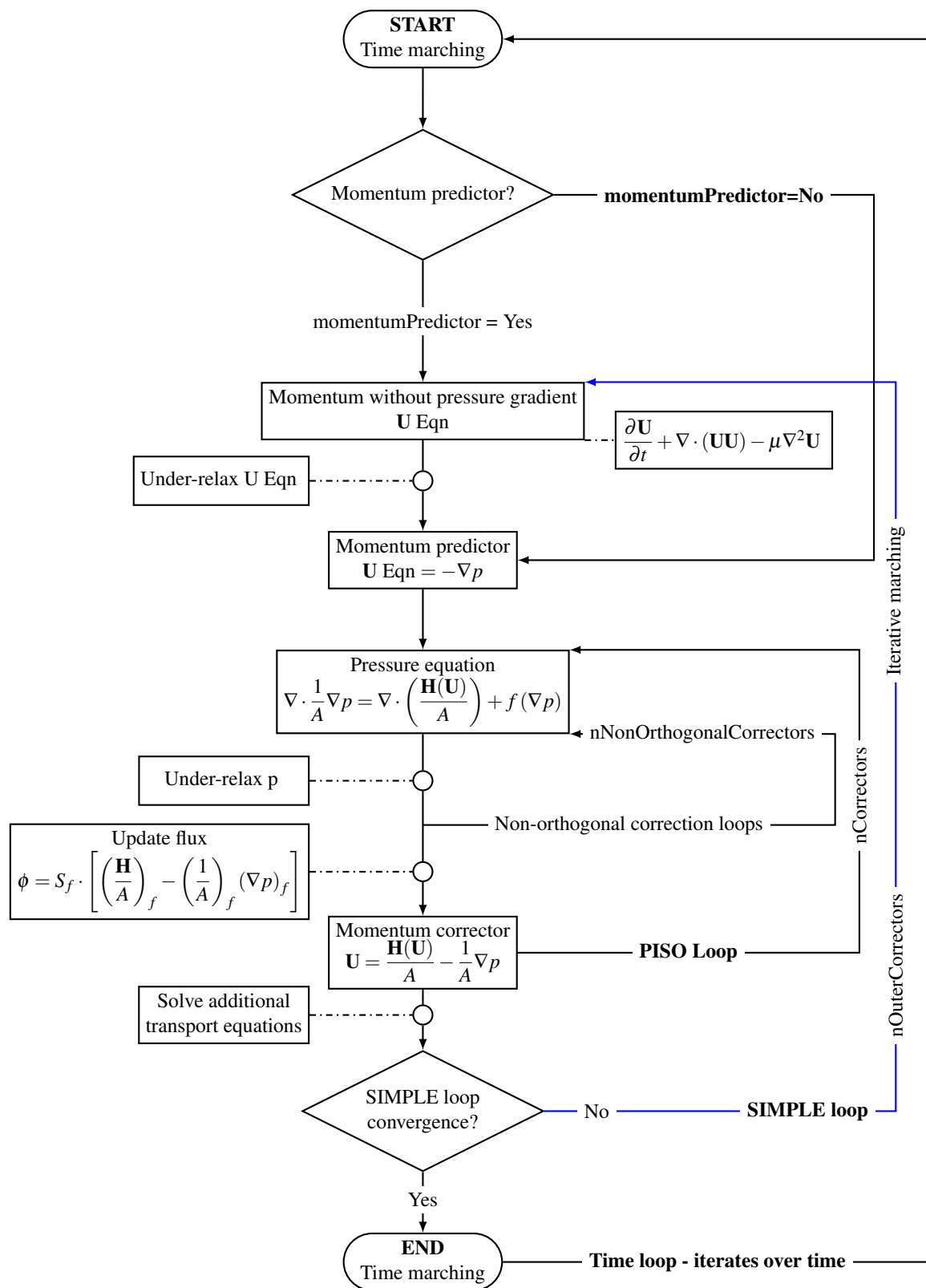


Fig. 4.19 PISO algorithm with iterative marching (PIMPLE). Adapted from Guerrero [78]

Table 4.3 Numerical schemes and their respective term. Schemes are limited to second order except for time scheme. x_j represents the partial derivative where 1 – 3 is a Cartesian coordinate direction (i.e.) $1 = \partial_x, 2 = \partial_y, 3 = \partial_z$.

Term		Scheme	OpenFOAM Entry
transient	$\frac{\partial u_i}{\partial t}$	1st order implicit	Euler
gradients	$\frac{\partial x_j}{\partial x_j}$	1st order cell limited	cellLimited Gauss linear 1
advective U	$u_j \frac{\partial u_i}{\partial x_j}$	2nd order single limiter	Gauss linearUpwindV
advective turbulence	$u_j \frac{\partial (k \omega)}{\partial x_j}$	2nd order	Gauss linearUpwind
laplacian	$\frac{\partial^2}{\partial x_j^2}$	2nd order limited	Gauss linear limited 1
interpolation	$\phi P \phi L$	2nd order linear	linear
surface normal gradients	$\frac{\partial}{\partial n}$	explicit non-orthogonal corrected	limited 1
wallDistance	$\nabla^2 \psi$	Poisson Equation	Poisson

4.7 Forces, Power, and Efficiencies

The Froude efficiency, η , for constant speed inline swimming is adopted from Tytell and Lauder [186]:

$$\eta = \frac{P_{out}}{P_{in}} = \frac{\langle T \rangle U}{(\langle T \rangle U + \langle P_L \rangle)}, \quad (4.23)$$

where the angle brackets $\langle \cdot \rangle$ signifies the time overage over the swimming cycle, T is the thrust force, U is the steady swimming speed, and P_L is the average power loss over a swimming cycle that is due to lateral undulations. For the force decomposition, I adopt the notation of Borazjani and Sotiropoulos [212] for a hydrodynamic force in the x-direction. Assuming a cartesian coordinate system with unit normal vectors n_x , n_y with the subscript denoting the direction, as defined in Figures 4.2, 4.5, 4.7, and 4.9. The hydrodynamic force is a summation of the thrust force, $T(t)$, and drag force, $D(t)$ over time given by

$$F(t) = T(t) - D(t) = \int_A (-pn_x + \tau_{xj}n_j)dA, \quad (4.24)$$

where n_j is the j th component of the unit normal on dA , A denotes an integration across the surface of the swimmer, p is the pressure force, n_x is the normal vector in the x direction,

τ_{ij} is the viscous stress tensor, and double subscripts indicate a summation of all components. This composition has the added benefit of being positive or negative depending on its contribution to either drag $D(t)$ or thrust $T(t)$. As in Borazjani and Sotiropoulos [212], in this form, the separate contributions of lift and drag can be decomposed as follows:

$$T(t) = T_p + T_v = \frac{1}{2} \left(\int_A -pn_x dA + \left| \int_A pn_x dA \right| \right) + \frac{1}{2} \left(\int_A \tau_{xj} n_{jx} dA + \left| \int_A \tau_{xj} n_{jx} dA \right| \right) \quad (4.25)$$

$$-D(t) = -(D_p + D_v) = \frac{1}{2} \left(\int_A -pn_x dA - \left| \int_A pn_x dA \right| \right) + \frac{1}{2} \left(\int_A \tau_{xj} n_{jl} dA - \left| \int_A \tau_{xj} n_{jx} dA \right| \right). \quad (4.26)$$

The power loss due to lateral undulations, P_L , is given by the following:

$$P_L = \int -pn_y \dot{h} dA + \int \tau_{yj} n_j \dot{h} dA, \quad (4.27)$$

where \dot{h} is the time derivative of the lateral displacement $h(x, t)$. As Borazjani and Sotiropoulos point out, the Froude efficiency defined in (4.23) can only be applied when the thrust and drag force is balanced, which is referred to as self propelled swimming [212]. This is due to the fish either accelerating or decelerating when the two forces are not balanced meaning that the freestream velocity U is no longer constant.

Maertens *et al.* [211, 249] suggests that in lieu of the Froude efficiency, the quasi propulsion efficiency be used. I adopt this efficiency because it is widely used in naval architecture, takes into account the towed resistance of the animal, and does not become 0 when thrust and drag are balanced (i.e. $\langle T \rangle = 0$) which is more intuitive. As Maertens *et al.* have defined, the more useful power definition, P_{out} , is given by

$$P_{out} = \overline{T}U_s + RU_s, \quad (4.28)$$

where R is the drag of the body with no movement, also called towed resistance, at the same reference speed U_s . Note that during self propulsion when drag and thrust are equal to 0, P_{out}

is equal to only the towed resistance, RU_s . The power needed for the swimmer to self propel is the addition of the thrust power, $\langle T \rangle U_s$, and the lateral power, P_L , given as:

$$P_{total} = \langle T \rangle U_s + \int -pn_2 \dot{h} dA + \int \tau_{2j} n_j \dot{h} dA. \quad (4.29)$$

The quasi-propulsion efficiency is thus defined as:

$$\eta_{QP} = \frac{P_{out}}{P_{in}} = \frac{(\langle T \rangle + R) U_s}{\langle T \rangle U_s + \langle P_L \rangle}, \quad (4.30)$$

where this equation simplifies to the normal Froude efficiency of equation (4.23) during self propulsion for $\langle T \rangle = 0$. The force coefficient is used to non-dimensionalize the force in the x-direction as:

$$C_D = \frac{D}{\frac{1}{2} \rho U^2 L} \quad (4.31)$$

$$C_T = \frac{T}{\frac{1}{2} \rho U^2 L}, \quad (4.32)$$

where U and L are defined as before. Since $C_T = -C_D$ and $F(t)$ is already defined in terms of lift and drag, the hydrodynamic coefficient is defined as

$$C_F = \frac{F}{\frac{1}{2} \rho U^2 L}, \quad (4.33)$$

which will be positive or negative depending on the average thrust or drag is dominate over a tail beat cycle. The power is non-dimensionalized in the following way

$$C_{power} = \frac{P}{\frac{1}{2} \rho U^3 L}, \quad (4.34)$$

where L represents the same reference area as in (4.31).

4.8 Post-processing

Post-processing is needed for the results of a simulation because of the presence of high frequency noise. To filter this noise, a convolve function using a Hanning filter as described by Harris *et al.* [250] is used on the raw force output. The Hanning window is defined as:

$$w(n) = 0.5 - 0.5 \cos\left(\frac{2\pi n}{M-1}\right) \quad 0 \leq n \leq M-1, \quad (4.35)$$

where n is the length of the input array and M is the length of the kernel, which is odd for the implementation in this research. Figure 4.20 shows the raw data with the filtered data superimposed.

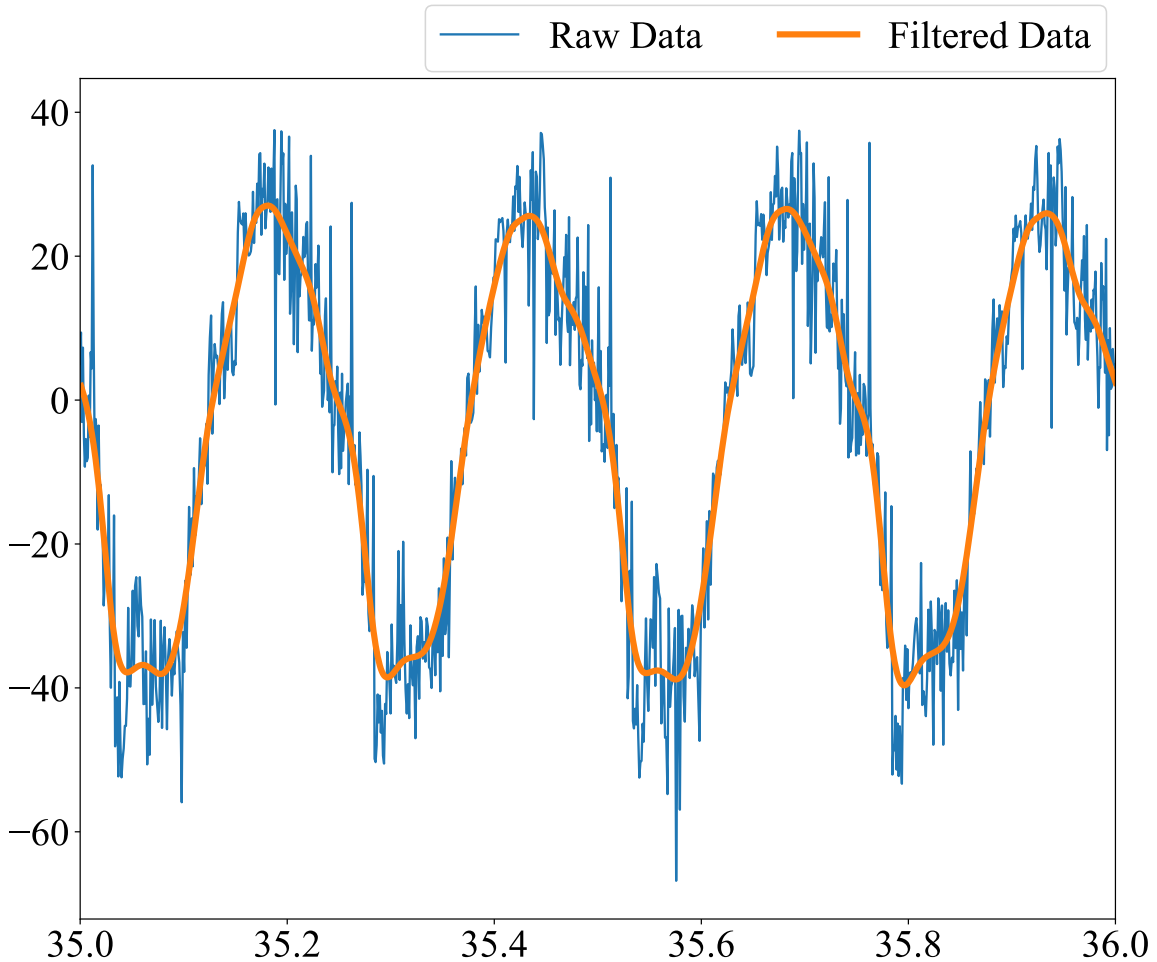


Fig. 4.20 Comparison of forces in the x-direction before and after filtering using a Hanning filter.

The use of the adjustable time step and limiting the simulation to a CFL number gives a non-uniform temporal output for the raw forces. Uniform temporal sampling was performed using the Scipy [251] linear interpolation method in order to ensure that time averaging could be taken using the arithmetic mean and to decrease the overall amount of data.

Post-processing the wake is a common technique when simulation fish motion. The purpose of this is to ensure that a reverse Kármán street is developed as discussed in Section 2.2. The accepted method for wake post-processing is the use of the Q -criterion introduced by Hunt *et al.* [252]. The Q -criterion indicates regions of vortices where the second invariant of the velocity gradient tensor $\nabla \mathbf{u}$ is positive. OpenFOAM®'s implementation of Q for incompressible flow is given as [144]:

$$Q = \frac{1}{2} \left[(\text{tr}(\nabla \mathbf{u}))^2 - \text{tr}(\nabla \mathbf{u} \cdot \nabla \mathbf{u}) \right], \quad (4.36)$$

where tr indicates a function that gives the trace of a second rank tensor defined in OpenFOAM®'s tensor class. An example of the Q criterion for carangiform swimming at a Strouhal number of 40 is given in Fig. 4.21. The figure shows the body bound vortices being shed into a reverse Kármán street as is observed for carangiform swimming.

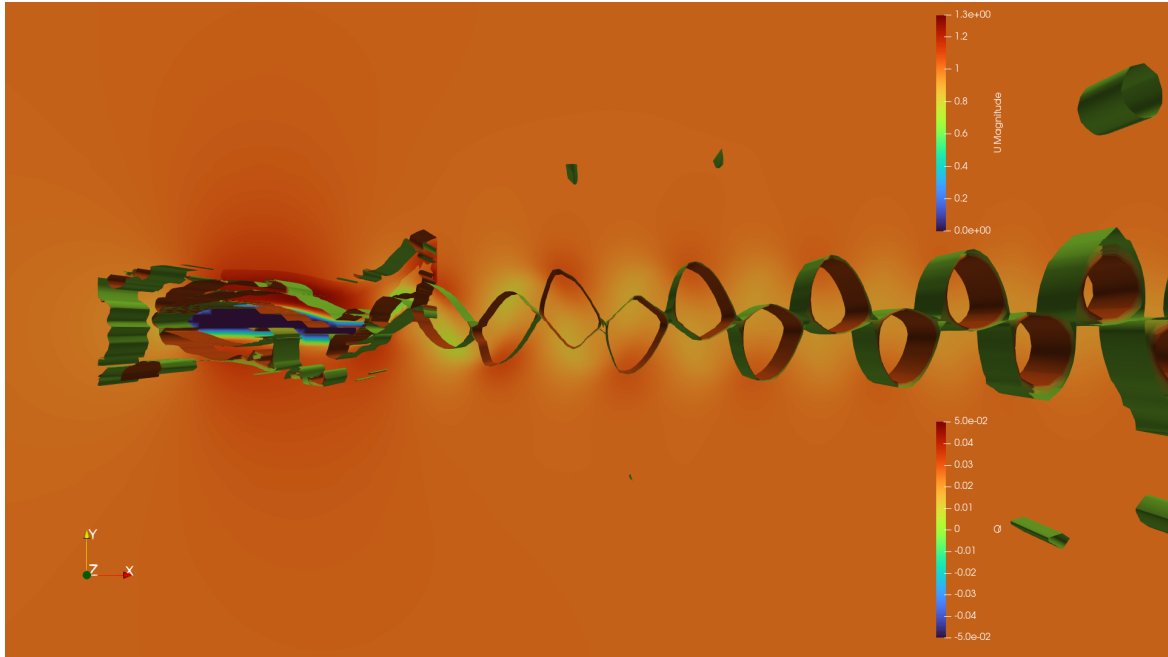


Fig. 4.21 Q criterion for carangiform swimming at a Strouhal of 40. Simulation flow time is 60 seconds. 3D Circular vortices are superimposed onto a velocity contour. Rendering is done in ParaView [77].

4.9 Limitations and Comparisons

This section compares the following simulation parameters: number of PIMPLE iterations, the use of a time delay, the max Courant number, different overset meshes, and three different turbulence models. The simulation results as well as the clock time (real time to finish a simulation) are used as a performance metric. This is useful in order to determine which parameters can be changed to speed up simulation times without effecting results and also how each parameter effects results. In the following comparisons, the time averaged values are the ones being compared and are shown as dashed lines.

A common limitation with simulating fish undulation movement is that the simulation requires many cycles of undulation ≈ 75 in order to reach a steady state. This is also a phenomenon that is seen in Borazjani and Sotiropoulos [212] and is not covered by most existing fish CFD literature. Figure 4.22 shows the lift and drag for all four BCF modes of locomotion run for 60 seconds of flow time run at a strouhal number of 40 which represents the fastest tail beat amplitude used in this research.

Figure 4.22 show that anguilliform and ostraciiform swimming modes reach steady-state at near the same flow time at ≈ 40 seconds corresponding to 80 tail beat cycles. Carangiform and Thunniform similarly reach steady-state at ≈ 25 seconds corresponding to 50 tail beat cycles. Interestingly, the anguilliform swim mode shows the least amount of drag while the thunniform shows the most. The only difference between the thunniform and carangiform swim modes is a rigid tail 85% down the body that pitches with a phase offset. This means that the rigid tail and phase offset contribute a lot to the overall drag. It would be expected that the thunniform drag is closer to the carangiform drag. However, the tail phase offset causes an abrupt change in flow direction which can contribute to drag.

The lift plots, Fig. 4.22d show that the lift is nearly the same for all four locomotion modes. It is expected that lift hovers around 0 for a symmetric airfoil so this is consistent with previous works. The largest amplitude oscillations occur with the ostraciiform motion due to the airfoil effectively pitching its angle of attack while having a phase offset trailing edge. This pitching motion is what causes the large amplitude oscillations versus the other three BCF modes. In order to determine the number of computer hours for each simulation, the simulation times are given in Table 4.4. Interestingly, the ostraciiform mode took almost twice as long as the other three modes which can be explained by the solvers taking more iterations to reach their tolerance in the ostraciiform case than in the anguilliform case. It is not clear why this occurs and more research would be needed to determine the exact reason behind this, but that is out of scope of this research.

Comparing the drag plots for the BCF swimmers, it is seen that the thunniform has around twice the amount of drag and the ostraciiform has about 3 times the amount of lift. In

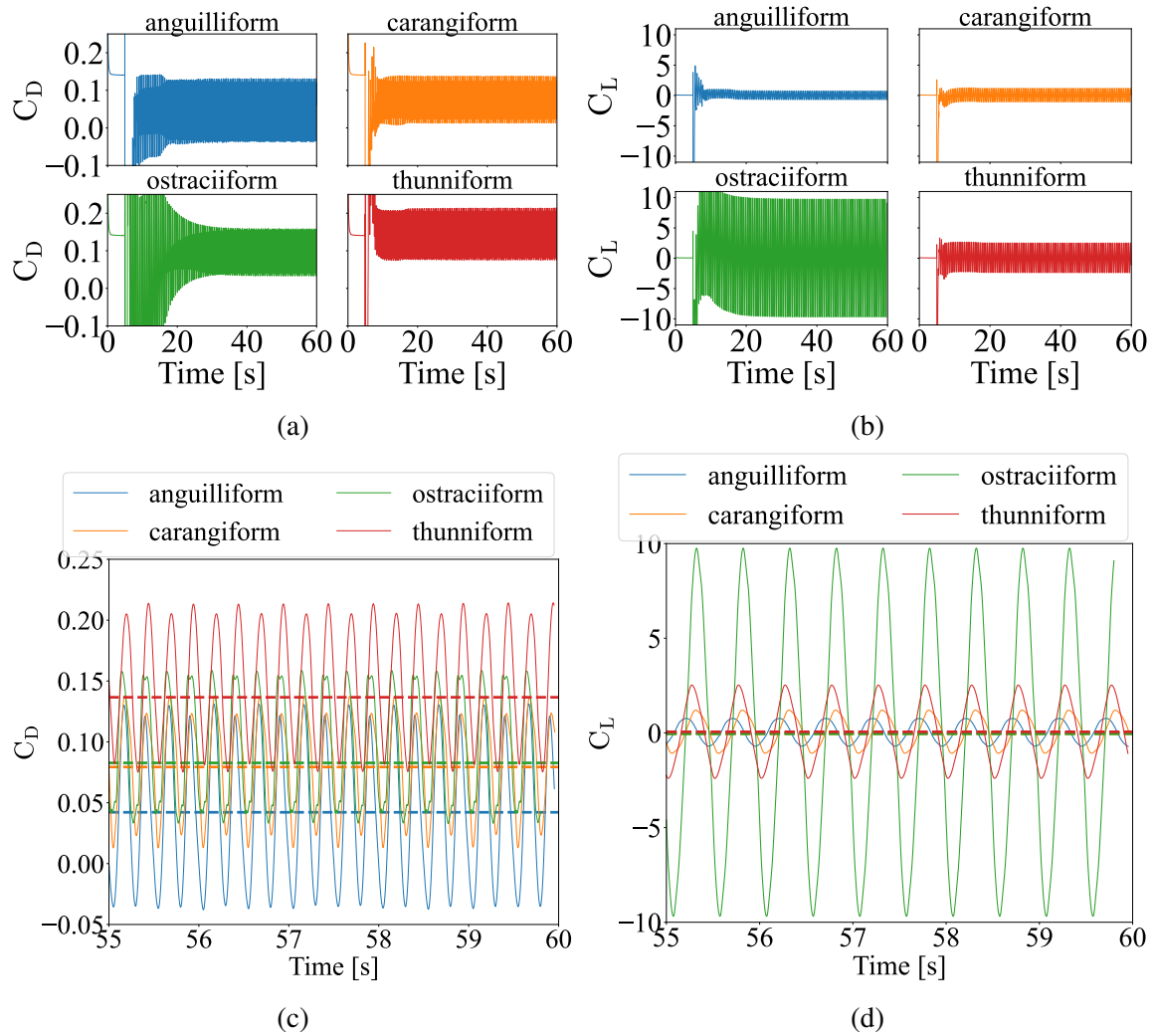


Fig. 4.22 (a) C_D , (b) C_L , (c) C_D zoomed in, and (d) C_L zoomed in for all four BCF swimming modes for 60 seconds at a strouhal number of 40 using the PISO algorithm (1 nOuterCorrectors).

nature, thunniform is considered the most efficient locomotion form for BCF swimmers [176, 30]. For these simulations, the parameters of all the swimmers were kept the same as the carangiform swimmer in order to establish a one-to-one comparison for the power scaling laws that are discussed in Section 4.10. The author acknowledges that non-optimal parameters were used for these two swimming modes and that is a cause for the higher lift and drag plots for both the thunniform and ostraciiform modes.

Table 4.4 Summary of simulation times reaching a quasi steady-state condition and completion of 60 seconds of flow time using 8 cores.

Case	Steady-State [s]	Clock [s]	Clock [hr]	Clock [day]
anguilliform	40.0	213322	59.26	2.47
carangiform	25.0	185032	51.40	2.14
ostraciiform	40.0	350270	97.30	4.05
thunniform	25.0	207611	57.67	2.40

4.9.1 Pimple Iterations

As discussed in Section 4.6, the time marching scheme used in this research is the PISO or PIMPLE algorithm. Figure 4.23 shows a comparison of the number of time marching steps in the PIMPLE algorithm, shown in Fig. 4.19 of Section 4.6. In OpenFOAM®, the number of time marching steps is controlled by the `nOuterCorrectors` entry in the `fvSolution` file under the system folder. The figure shows that after around 45 seconds of flow time, all simulations reach a steady state oscillation at close to the same time. Interestingly, the 5 and 25 iteration cases follow a similar trajectory with the 25 iteration case having an abnormal divergence at around 30 seconds. This indicates that steady state is independent of the number of time march iterations.

Comparing the steady state oscillations of each case, all cases have similar average C_L . C_D shows the 1 iteration and 25 iteration cases having nearly the same average drag, with the 5 iteration case being slightly greater. In both drag and lift, the 25 iteration case has a larger envelope of oscillation. This may be caused by better convergence of the steady state solution compared to the 5 iteration case. The 1 iteration case is similar to the 5 iteration case, but with an offset which may indicate that using time marching is not beneficial in this case unless using a larger number of iterations.

To determine the most appropriate number of iterations, the simulation time for each case is shown in Table 4.5. All simulations were performed on the same 32 core virtual machine using 8 cores. This is true for all reported times related to this research.

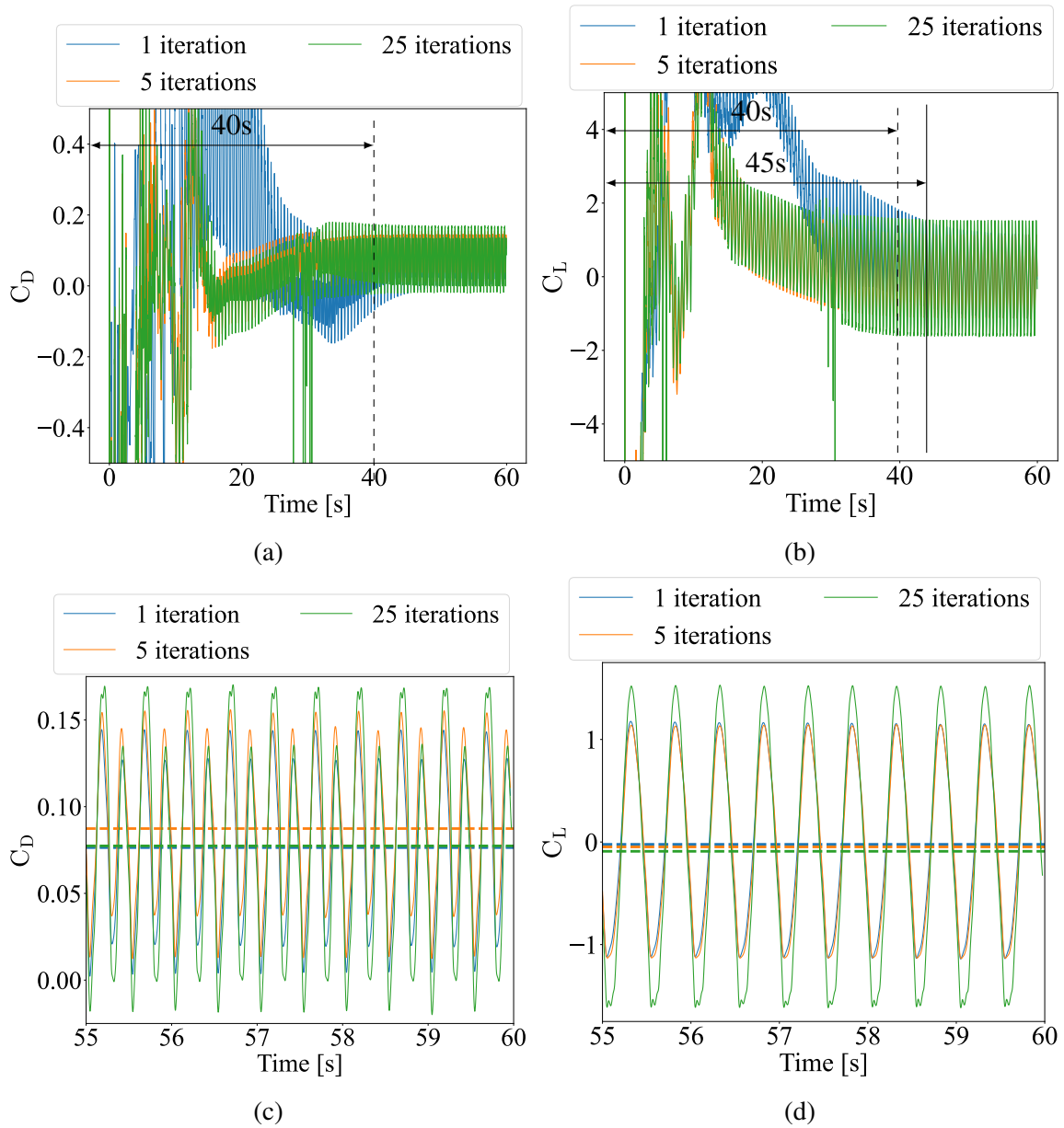


Fig. 4.23 (a) C_D , (b) C_L , (c) C_D zoomed in, and (d) C_L zoomed in of carangiform swimming for 60 seconds utilizing 1, 5, and 25 pimpleFoam iterations. Simulations are performed at Reynolds number $5 \cdot 10^3$ and at Strouhal number of 40.

Table 4.5 Summary of simulation times for 1, 5, and 25 pimple iterations (nOuterCorrectors).

Case	Clock Time [s]	Clock Time [hr]	Clock Time [day]
1	223838	62.18	2.59
5	453748	126.04	5.25
25	1133369	314.82	13.12

Table 4.5 shows that the run time for each simulation increases as expected. Due to the amount of simulations needed to be performed and the amount of resources available, the remainder of this research uses the PISO algorithm or 1 pimple iteration.

4.9.2 Time Delay

The amount of time it takes to reach steady state is a limitation on computational resources. To mitigate this, a time delay for fish movement was coded into the Ika-Flow solver package. The reasoning behind introducing a time delay is that the solution stabilizes to a steady-state value with no movements and then movement begins, therefore, there would not be large fluctuations in flow as seen in the beginning of Fig. 4.23. Figure 4.24 shows the same simulation with a delay of 5 and 15 seconds.

Interestingly, Fig. 4.24 indicates that C_D does reach steady state sooner with a time delay but C_L seems to be invariant of delay. In both cases, the time delay does stabilize the forces quicker versus the case with no time delay shown in Fig. 4.23. This has the added benefit of reducing simulation time as shown in Table 4.6.

Table 4.6 Summary of simulation times the same simulation with no delay, a 5 second delay, and a 15 second delay.

Case	Steady-state [s]	Clock Time [s]	Clock Time [hr]	Clock Time [day]
No Delay	40	453748.00	126.04	5.25
5s	20	338924.00	94.15	3.92
15s	35	337924.00	93.87	3.91

As the times for each simulation show, the clock time for the delayed cases is about 1.5 days shorter than for the non delayed case. The reason for the simulation time for the 15s delayed case being shorter than the 5s delayed case is that the simulation spent more flow time with zero movement which is faster numerically. For the rest of the simulations, a delay time of 2 seconds was chosen due to being when the non-movement simulation stabilizes. The force data is then taken after 35 seconds of flow time for 5 tail beat cycles. This allows for the solution to reach steady state and gives enough cycles for averaging.

4.9.3 Grid Independence and Validation

To the author's knowledge, there are no 2D overset studies involving fish motion. Previously referenced overset studies of NACA 0012 airfoils are with no motion [222–225] or only look at pitching motion [221]. Another limitation with the existing research is that there is no

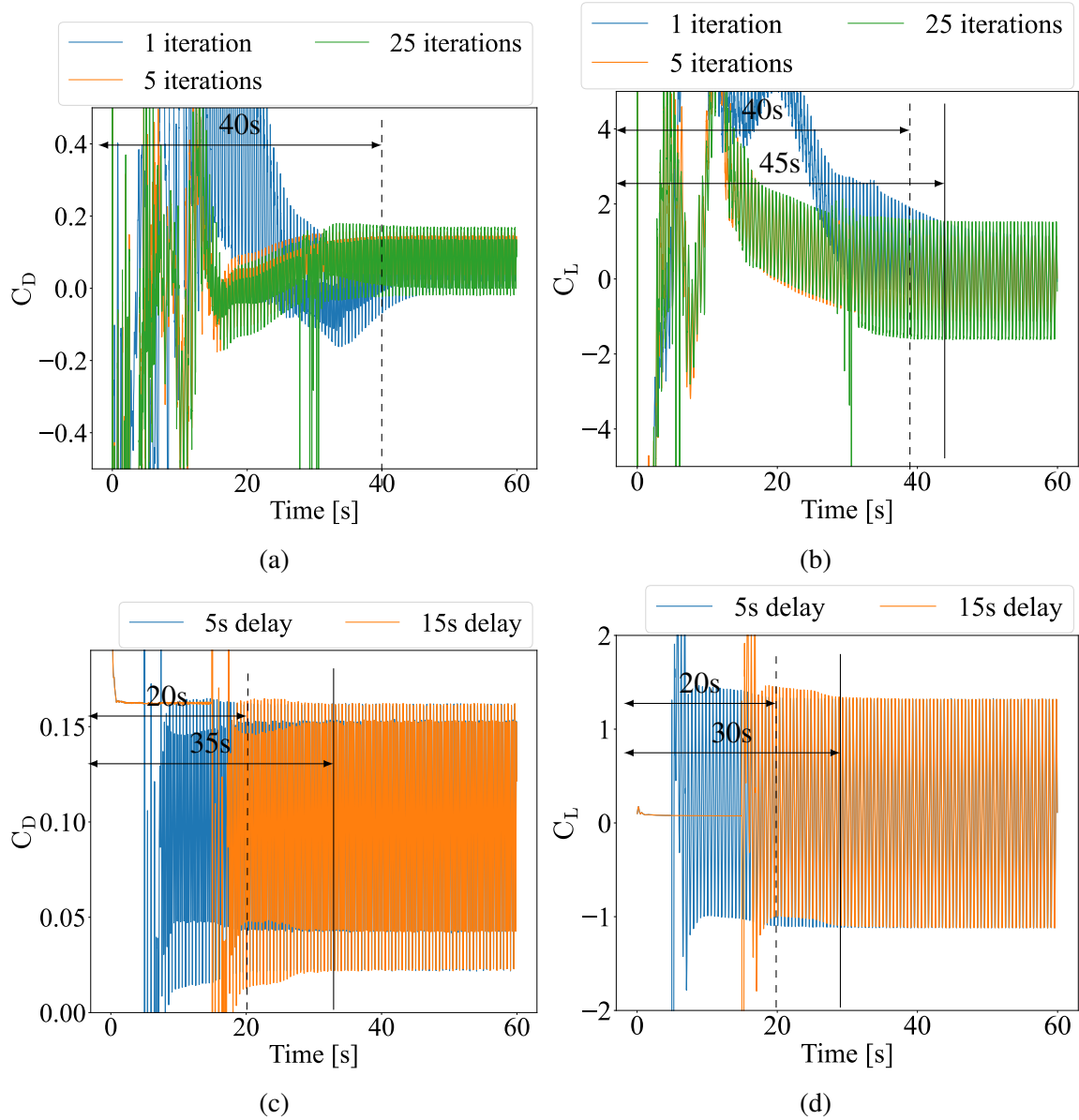


Fig. 4.24 (a) C_D , (b) C_L for carangiform swimming for 60 seconds utilizing 1, 5, and 25 pimpleFoam iterations. (c) C_D and (d) C_L for the same simulation with a delay of 5 and 15 seconds. Simulations were performed at Reynolds number $5 \cdot 10^3$ and at a Strouhal number of 40.

research at Reynolds numbers higher than 10^4 . I choose to validate the simulations using two studies done by Dong and Lu [79] and Yu *et al* [43]. This validation is combined with a grid independence study in accordance with the guidance of Celik *et al.* [253].

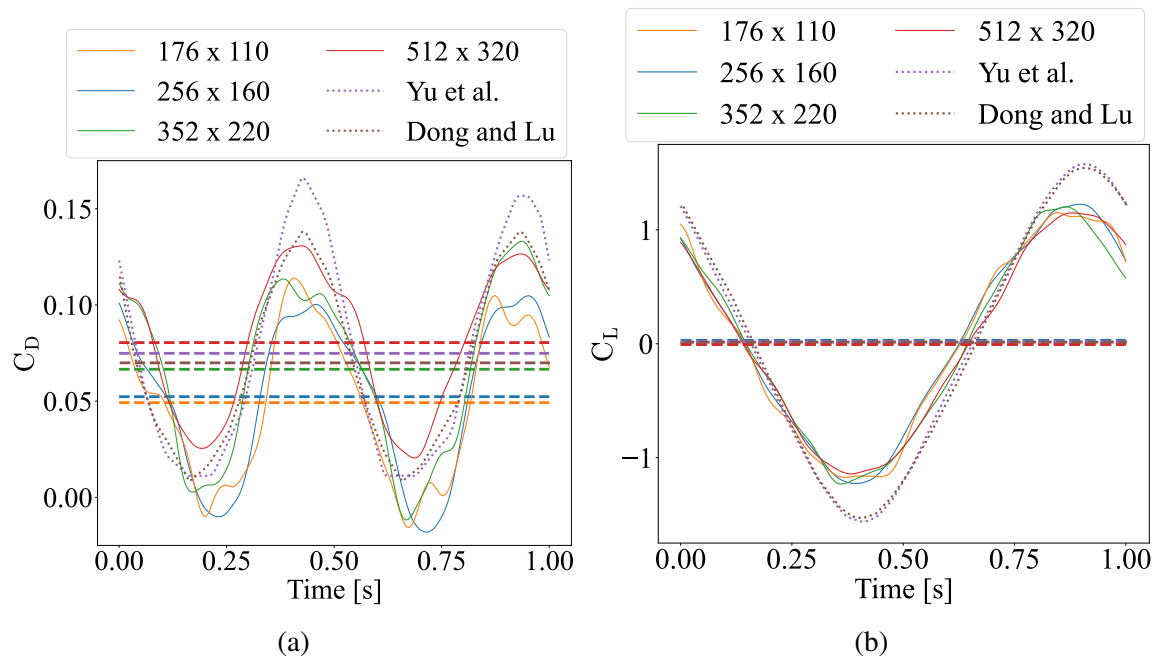


Fig. 4.25 (a) C_D and (b) C_L for the same simulation data from Dong and Lu [79] and Yu *et al.* [43]. Simulations performed at a Reynolds number of $5 \cdot 10^3$ at a Strouhal number of 40.0 with only the carangiform swim mode. Dashed lines represent the time averaged values over the tail beat cycles.

Table 4.7 Summary of simulations for increasing grid sizes.

Dimensions	No. Elements	Clock Time [s]	Clock Time [hr]	Clock Time [day]	Drag Error [%]
176 x 110	51508	200568.00	55.71	2.32	34.67
256 x 160	85760	338924.00	94.15	3.92	32.00
352 x 220	143952	720498.00	200.14	8.34	6.67
512 x 320	282002	1203972.00	334.44	13.93	5.33

Figure 4.25 shows variations between the drag plots while negligible difference in lift data. For reference, the coarsest mesh is $\approx 39\%$ different from the finest mesh. Looking at Table 4.7, the difference between the second mesh and the next finest mesh is 4.42 days of simulation time. When looking at the time averaged values, shown as dashed lines in Fig. 4.25, there is significant variation in C_D but C_L is near zero. C_L near zero is expected as a normal symmetric airfoil would have 0 lift at an angle of attack at 0, or free stream velocity only in the x-direction. The error between the body-fitted results of Dong and Lu [79] and

Yu *et al.* [43] can be explained by the overset interpolation error discussed in Section 4.3. The grid chosen for investigations in this research is the 3rd grid on Table 4.7. This mesh has a longer computational time than the first 2 grids, but is much closer to the validation cases than the first 2 grids.

4.9.4 The Effect of Max Courant Number

As discussed in Section 4.6, the simulation temporal discretization is controlled via the maximum Courant number. In theory, the benefit of using the PIMPLE algorithm allows for the use of a higher Courant number in simulations thus lowering simulation time. Here we compare three Courant numbers: 0.5, 1.0, 2.0. These simulations are run at a Reynold's number of $5e03$ and a Strouhal number of 40.0 with the number of PIMPLE iterations set to 5. The number of PIMPLE iterations (nOuterCorrectors) must be greater than 1 (the PISO algorithm) to take advantage of the time marching method. Figure 4.26 shows the results from this comparison.

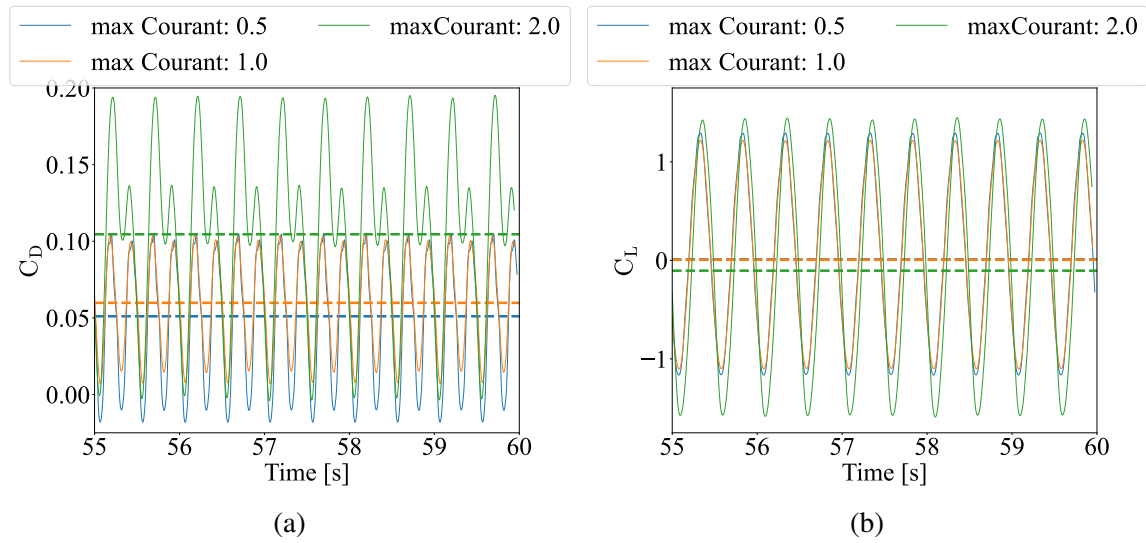


Fig. 4.26 (a) C_D and (b) C_L for max Courant numbers of 0.5, 1.0, and 2.0. Simulations performed at Reynolds numbers $5 \cdot 10^3$ and Strouhal number of 40.

The data shows a second oscillation occurs with the 2.0 max Courant number that is not present at lower Courant numbers. Furthermore, the drag increases with the higher Courant number which is explained when examining the residuals of the simulations. By the 5th pimple iteration for each time step, the 0.5 Courant simulations converge to an order of magnitude smaller than that of the 2.0 Courant number with the 1.0 Courant case somewhere in between. These results might improve with a higher number of PIMPLE iterations, but

this is out of scope for this research, and the simulation times shown in Table 4.8 support using the smaller Courant number versus more PIMPLE iterations.

Table 4.8 Summary of simulations for increasing grid sizes.

Max Courant	Clock Time [s]	Clock Time [hr]	Clock Time [day]
0.5	326658.00	90.74	3.78
1.0	299850.00	83.29	3.47
2.0	314158.00	87.27	3.64

4.9.5 Ansys vs. GMSH mesher

The proprietary Ansys Meshing software [237] and open-source mesher GMSH [238] were both used to create the overset mesh. Mesh settings were kept as described in Table 4.1. Figure 4.27 shows the two meshes with a comparison of the lift and drag. Figure 4.27a-4.27b shows that fluent is able to capture the rounded trailing edge better than GMSH. Refinement points were used in GMSH in order to refine the trailing edge, but had a limited effect on the trailing edge.

The data shows that there is significant variance in the drag with a small difference in the lift. The residuals show that the two meshes converge to similar residuals, but the Ansys mesh converges faster than the GMSH mesh. This is most likely a direct result of the mesh resolution at the trailing edge of the airfoil. Interestingly, Table 4.9 indicates that the GMSH mesh takes ≈ 1 day more in simulation time. The Ansys mesh results are closer to the validation case presented in Section 4.9.3, and with the trailing edge being refined better and the simulation time saving, the fluent mesh is used for the rest of this research. A mesh using snappyHexMesh was also investigated, but there were problems with resolving the inflation layers and so it was not used.

Table 4.9 Summary of simulation times for two different meshing programs.

Mesher	Clock Time [s]	Clock Time [hr]	Clock Time [day]
Ansys	338924.00	94.15	3.92
GMSH	429006.00	119.17	4.97

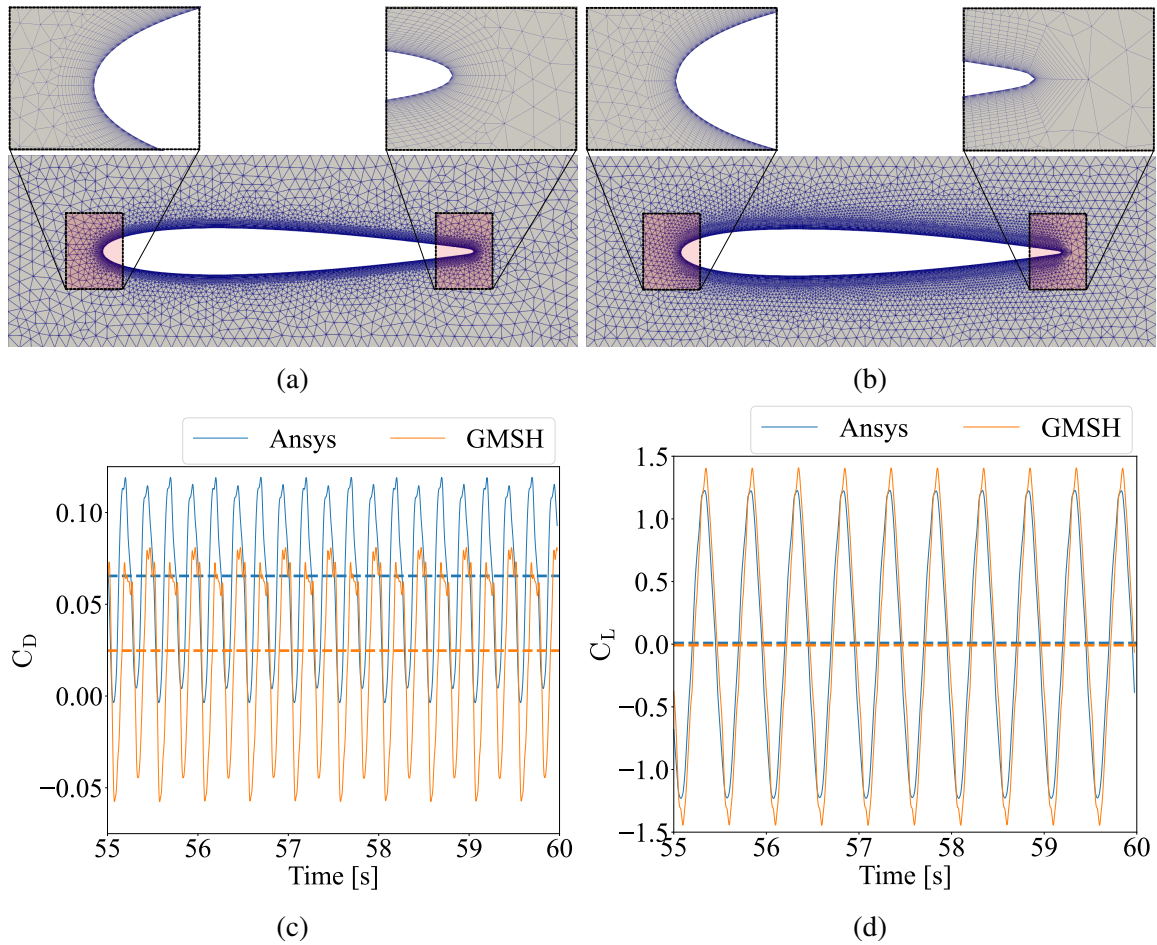


Fig. 4.27 (a) Ansys and (b) GMSH mesh generated using the same parameters in Table 4.1. (c) C_D and (d) C_L zoomed in for both meshes after steady state has been reached.

4.9.6 Turbulence Models

A comparison is made between three turbulence models; the k - ω SST [243], the k - k_L - ω [245], and the Spalart-Allmaras (S-A) model [246]. The purpose of this comparison is to determine if one model is computationally faster than the other and if their results are comparable. The models are tested at two Reynolds numbers: $5 \cdot 10^3$ and $4 \cdot 10^5$. All three are linear eddy viscosity models using the Reynolds-averaged Navier-Stokes (RANS) equation with the difference being the number of additional transport equations [254]. A derivation of the models as implemented in OpenFOAM[®] is given in Appendix C and are briefly described below.

The Reynolds Averaged Navier-Stokes (RANS) equation in tensor notation is

$$\left(\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} \right) = - \frac{\partial \bar{P}}{\rho \partial x_i} + \bar{g}_i + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial}{\rho \partial x_j} \overline{u'_i u'_j}, \quad (4.37)$$

and various RANS turbulence models relate the unknown components of the stress tensor to mean flow quantities. The quantity $\tau_{ij} = -\overline{u'_i u'_j}$ is known as the Reynolds stress tensor. Eddy viscosity models relate the Reynolds stress components to the mean rate of the strain tensor. This is performed through the Boussinesq hypothesis [255] that relates the Reynolds stresses to the mean velocity gradients

$$-\overline{u'_i u'_j} = \nu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij} \quad (4.38)$$

where δ_{ij} is the Kronecker delta function. This is a special function where $\delta_{ij} = 1$ if $i = j$ and 0 if $i \neq j$. ν_t is the kinetic eddy viscosity and k is the kinetic energy defined as $k = \frac{\overline{u_i u_i}}{2}$.

Spalart-Allmaras model

The S-A model adds one additional transport equation where the kinematic eddy viscosity, $\tilde{\nu}$, is related to the local mean vorticity [246, 254] and is described by

$$\begin{aligned} \frac{D}{Dt}(\rho \tilde{\nu}) = \nabla \cdot (\rho D_{\tilde{\nu}} \tilde{\nu}) + \frac{C_{b2}}{\sigma_{\nu_t}} \rho |\nabla \tilde{\nu}|^2 + C_{b1} \rho \tilde{S} \tilde{\nu} (1 - f_{t2}) \\ - \left(C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2} \right) \rho \frac{\tilde{\nu}^2}{\tilde{d}^2} + S_{\tilde{\nu}}, \end{aligned} \quad (4.39)$$

where \tilde{S} is the local mean vorticity, \tilde{d} is the distance to the solid wall (the swimmer in this case) and the rest of the terms are coefficients given in Table C.3.

k- ω SST model

The two equation k- ω SST model presented by Menter *et al.* [243] is a combination of the Wilcox k- ω model and the k- ϵ model with a blending function. It adds two additional transport equations with the kinetic eddy viscosity defined as

$$\nu_t = \frac{k}{\omega}. \quad (4.40)$$

The turbulence specific dissipation rate (ω) is given as

$$\begin{aligned} \frac{D}{Dt}(\rho\omega) = \nabla \cdot (\rho D_\omega \nabla \omega) + \frac{\rho\gamma G}{\nu} - \frac{2}{3}\rho\gamma\omega(\nabla \cdot \mathbf{u}) - \rho\beta\omega^2 \\ - \rho(F_1 - 1)CD_{k\omega} + S_\omega, \end{aligned} \quad (4.41)$$

and the turbulence kinetic energy (k) as

$$\frac{D}{Dt}(\rho k) = \nabla \cdot (\rho D_k \nabla k) + \rho G - \frac{2}{3}\rho k(\nabla \cdot \mathbf{u}) - \rho\beta^*\omega k + S_k, \quad (4.42)$$

where F_1 is a blending function given by (C.14) and the coefficients are defined in Table C.1.

k-k_L- ω model

The three equation k-k_L- ω model is given by Fürst *et al.* [245] adds three extra transport equations. The specific dissipation rate is

$$\begin{aligned} \frac{D}{Dt}(\omega) = \nabla \cdot (D_\omega \nabla \omega) + C_{w1}P_{kt}\frac{\omega}{k_t} - \left(1.0 - \frac{C_{wR}}{f_w}\right)k_l(R_{bp} + R_{nat})\frac{\omega}{k_t} \\ - C_{w2}f_w^2\omega^2 + C_{w3}f_\omega\alpha_t f_w^2\frac{k_t^{0.5}}{y^3} \end{aligned} \quad (4.43)$$

The laminar turbulent kinetic energy is given as:

$$\frac{D}{Dt}(k_l) = \nabla \cdot (\nu \nabla k_l) + P_{kl} - R_{bp} + R_{nat} + D_l. \quad (4.44)$$

The turbulent kinetic energy is given as:

$$\frac{D}{Dt}(k_t) = \nabla \cdot (D_k \nabla k_t) + P_{kt} + (R_{bp} + R_{nat})k_l - \omega + D_t, \quad (4.45)$$

and the coefficients are described in Table C.2. All simulations were performed at a Stouhal number of 40 and at two different Reynolds numbers representing the laminar region and the transitional region; $5 \cdot 10^3$ and $4 \cdot 10^5$.

Comparisons

Figure 4.28 shows the comparison between the three turbulence models at different Reynolds numbers. The second Reynolds number, $4 \cdot 10^5$, is chosen due to being in the transitional region for flat plate flow [256]. At lower Reynolds numbers, all three models perform nearly the same, but in the transitional regime, the models diverge in drag while being consistent in lift. Interestingly, higher Reynolds numbers show more oscillations in the forces than at lower Reynolds numbers. This may be caused by the resolution of the grid normal to the surface of the surface, known as the y^+ value. For reference, the grid has a y^+ value of less than 1 for Reynolds numbers up to $Re = 4 \cdot 10^7$ from flat plate theory discussed in Appendix C.5 and Schlichting [256].

To the author's knowledge, there are no studies of fish swimming at Reynolds numbers exceeding $1 \cdot 10^5$ and therefore there is no data to validate these simulations. Experimental data on a rigid NACA 0012 airfoil at a Reynolds number of $Re = 6 \cdot 10^6$ are given in McCrosky [140]. McCrosky's data shows that lift is nearly 0 and drag is about 0.0085 for an airfoil at 0 angle of attack. Here I also use the XFOIL program developed by Drela [257], which can be used to determine the lift and drag at the specific Reynolds number of interest. It should be noted that XFOIL uses panel methods and the airfoil is not undulating for the XFOIL calculations. Table 4.10 shows the average lift and drag for the three turbulence models, XFOIL, and data from McCroskey [140].

Table 4.10 Lift C_L and drag C_D for three different turbulence models, XFOIL, and data from McCrosky [140] at Reynolds numbers of 5e03 and 4e05.

Model	Reynolds Number	C_D	C_L
k – ω SST	$5 \cdot 10^3$	0.0656	0.0056
k – k_L – ω	$5 \cdot 10^3$	0.0649	0.0045
Spallart-Allmaras	$5 \cdot 10^3$	0.0654	0.0004
k – ω SST	$4 \cdot 10^5$	0.0101	0.0072
k – k_L – ω	$4 \cdot 10^5$	0.0039	0.0057
Spalart-Allmaras	$4 \cdot 10^5$	0.0137	0.0065
XFOIL	$4 \cdot 10^5$	0.0062	0.0000
McCrosky [140]	$6 \cdot 10^6$	0.0085	0.0000

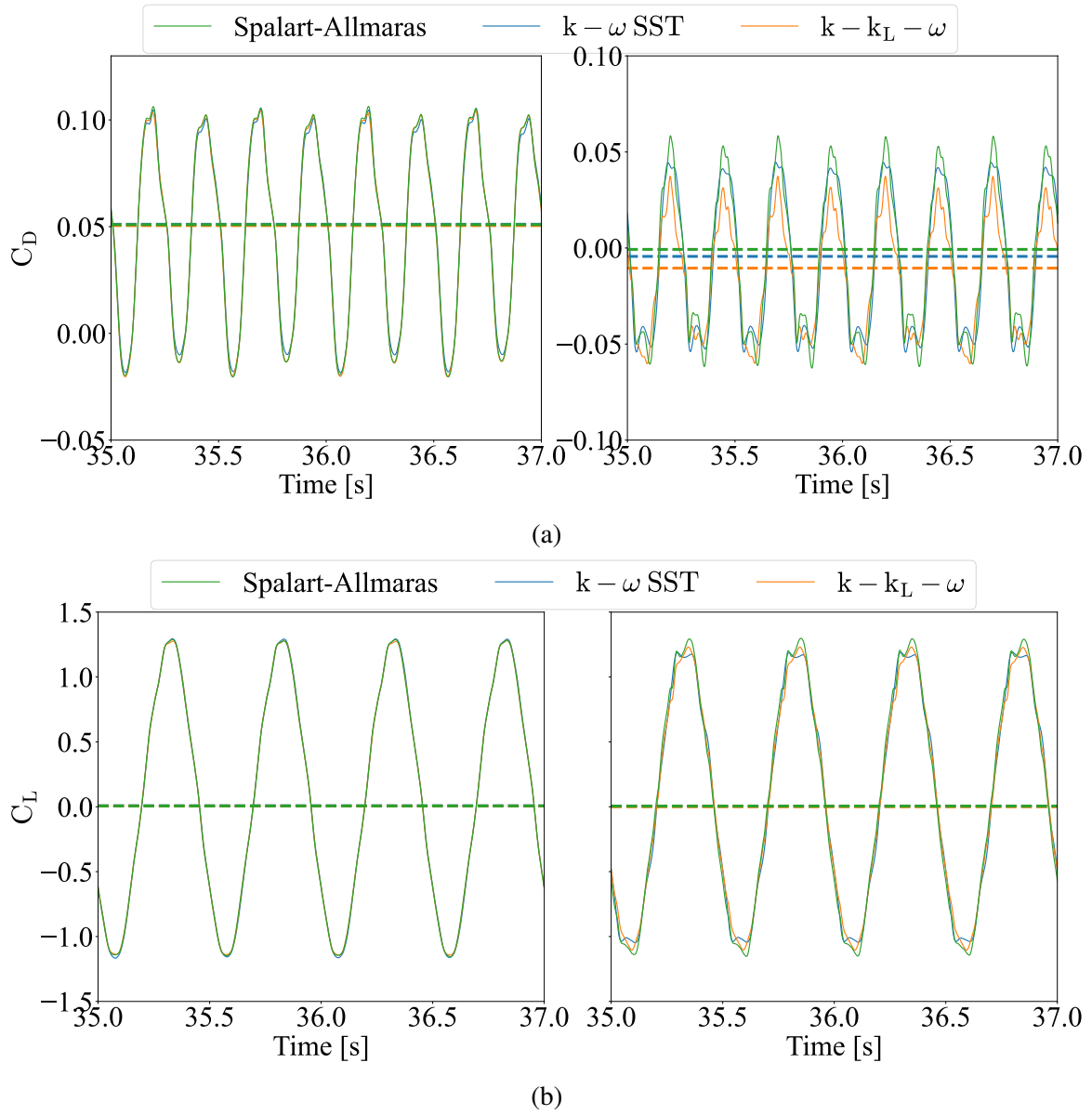


Fig. 4.28 (a) C_D and (b) C_L for different turbulence models. Simulations performed at Reynolds numbers $5e03$ (left plots) and $4e05$ (right plots) with a Strouhal number of 40.0.

Comparing the simulation data, the trend is a decreasing drag for an increasing Reynolds number. For the higher Reynold's case, the $k - \omega$ SST and Spalart-Allmaras model are significantly higher than the $k - k_L - \omega$ model. The expected trend for an airfoil is that drag will decrease as Reynolds number increases as given by empirical data in Hoerner [158]. All models follow this expected trend but $k - k_L - \omega$ model decreases significantly more than the others. It is difficult to discern which model is more accurate and further simulations will be needed to be done in future research. Discrepancies aside, the $k - \omega$ SST model has been successfully used in a wide variety of engineering cases including fish hydrodynamics [43, 79, 211–216] which increases the confidence in this model. In order to assess if there are any time savings with the different models, the simulation times are presented in Table 4.11.

Table 4.11 Summary of simulation times for three different turbulence models at $Re = [5 \cdot 10^3, 4 \cdot 10^5]$.

Simulation	Reynolds Number	Clock Time [s]	Clock Time [hr]	Clock Time [day]
$k - \omega$ SST	$5 \cdot 10^3$	220480.00	61.24	2.56
$k - k_L - \omega$	$5 \cdot 10^3$	214672.00	59.63	2.48
Spallart-Allmaras	$5 \cdot 10^3$	216337.00	60.09	2.50
$k - \omega$ SST	$4 \cdot 10^5$	335537.00	93.20	3.88
$k - k_L - \omega$	$4 \cdot 10^5$	310976.00	86.38	3.60
Spallart-Allmaras	$4 \cdot 10^5$	302305.00	83.97	3.50

Table 4.11 show that all three turbulence models have similar computational times for low Reynolds numbers, but in higher Reynolds number applications, the Spalart-Allmaras model performs the most efficiently. This is consistent with a one equation model versus a two or three equation model as there are less equations to solve. Comparing the other two models, the $k - k_L - \omega$ is faster than the $k - \omega$ SST, but for reasons mentioned above, this research is utilizing the $k - \omega$ SST model.

4.10 Simulation Results

Simulations were performed for all motion cases for the following Reynolds and Strouhal numbers:

$$Re = 5.00 \cdot 10^3, 4 \cdot [10^4 - 10^7]$$

$$St = [0.1, 0.2, 0.3, 0.4].$$

The Strouhal numbers were chosen as above a Reynolds number of $4 \cdot 10^5$, biological animals settle near a Strouhal number of 0.3, as shown in the meta-study by Gazzola *et*

al. [4]. A positive thrust coefficient, C_T , represents the condition in which the swimmer is producing more thrust than fluid drag on the body. The point at which the swimmer passes from negative thrust to positive thrust, where $C_D = C_T = 0$, is called the self-propulsion value, herein called the self-propelled Strouhal (SPS).

Power is calculated using (4.29) and the condition of self propulsion is given by linear interpolation of values and finding the zero crossing of the thrust force. For the ostraciiform and thunniform cases that do not cross the self propulsion line, the curves are fit to a 3rd degree polynomial and the zero crossing is found using this curve fit. Figure 4.29 shows the drag / thrust for anguilliform and carangiform swimmers and Fig. 4.30 shows the drag /thrust for ostraciiform and thunniform modes. Table 4.12 gives the Strouhal numbers for the self propulsion condition at each Reynolds number.

Table 4.12 Strouhal numbers that satisfy the self-propulsion (SPS) condition at tested Reynolds numbers.

Re Form	Anguilliform	Carangiform	Ostraciiform	Thunniform
$5 \cdot 10^3$	0.333	0.378	0.840	4.252
$4 \cdot 10^4$	0.267	0.300	0.303	0.973
$4 \cdot 10^5$	0.245	0.276	0.267	0.562
$4 \cdot 10^6$	0.279	0.264	0.245	0.373
$4 \cdot 10^7$	0.258	0.258	0.233	0.272

The data in Figures 4.29 and 4.30 shows that the highest SPS happens during the viscous regime and become consistent around 0.3 at higher Reynolds numbers which can be seen further in the scaling laws for SPS shown in Fig. 4.31. This is consistent with the meta analysis done by Gazzola *et al.* [4] shown in Fig. 2.24 where below the transition number of $\approx 5 \cdot 10^3$, the SPS scales as $\sim Re^{-1/4}$. Intuitively, a smaller swimmer is represented at lower Reynolds numbers will have to undulate at a faster speed, thus a higher Strouhal number, to keep up the same speed as of a larger swimmer. Figure 4.31 shows the piece-wise scaling of the SPS for 5 decades of Reynolds numbers. The larger Reynolds numbers have a scaling exponent close to 0 except for the thunniform locomotion mode. This means that the SPS is constant for $Re \geq 4 \cdot 10^4$ except for the thunniform mode. Thunniform locomotion was modeled using a combination of information from Barret [176] and Zhu *et al.* [30] with parameters to match the other cases. This was intentionally done in order to obtain a direct comparison between the locomotion modes and it is acknowledge that the thunniform and ostraciiform has room for improvement, as discussed in Section 4.9.

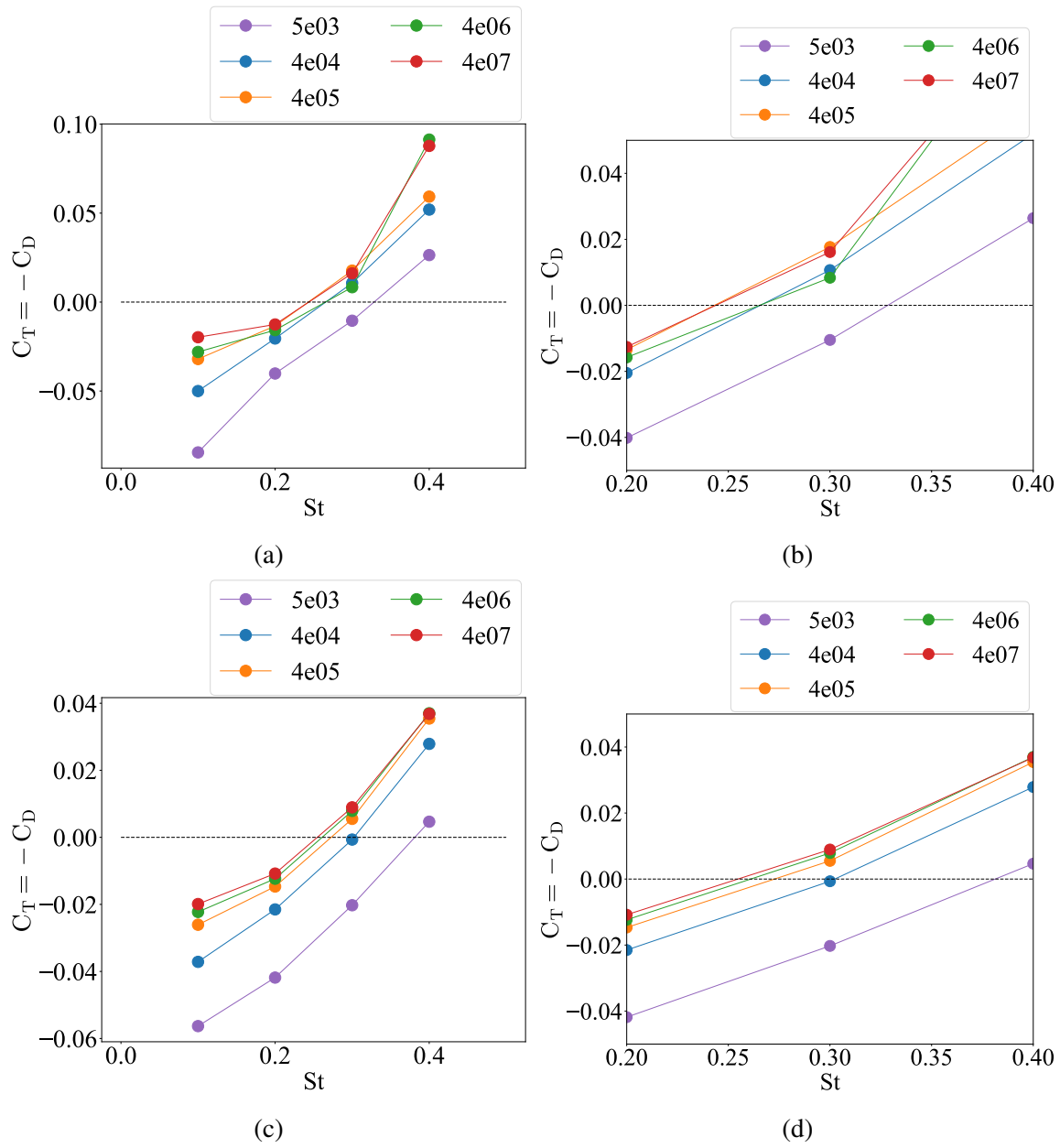


Fig. 4.29 (a) Anguilliform drag / thrust coefficient and (b) zoomed in of (a) to show zero crossing. (c) Carangiform drag / thrust coefficient and (d) zoomed in of (c) to show zero crossing.

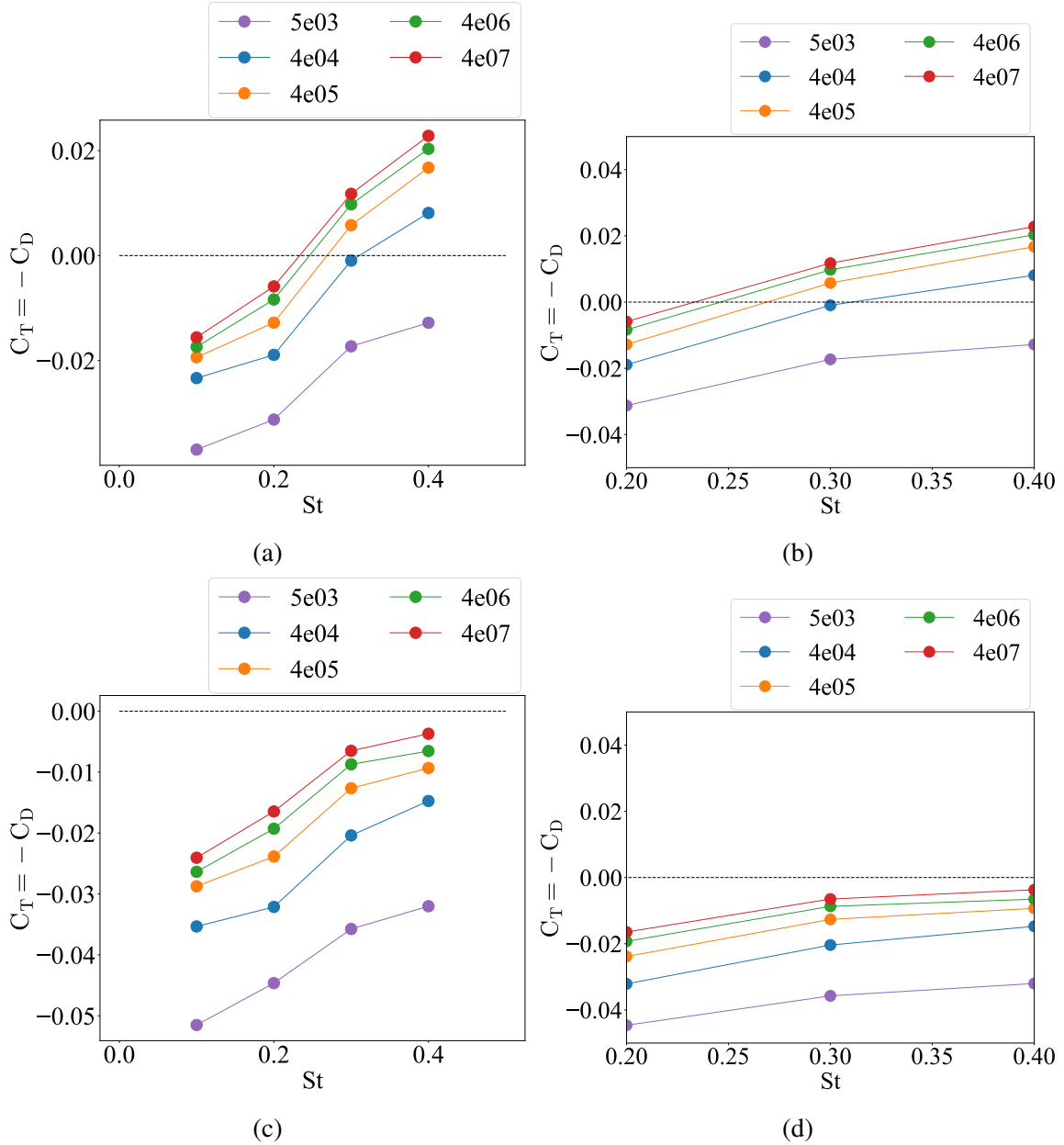


Fig. 4.30 (a) Ostraciiform drag / thrust coefficient, (b) zoomed in of (a) to show zero crossing, (c) thunniform drag / thrust coefficient, and (d) zoomed in of (c) to show zero crossing. Note that the $5 \cdot 10^3$ case for the ostraciiform and all cases for thunniform mode never pass the zero line. In these cases, a third degree polynomial was fit and the self propulsion case was taken from that fit.

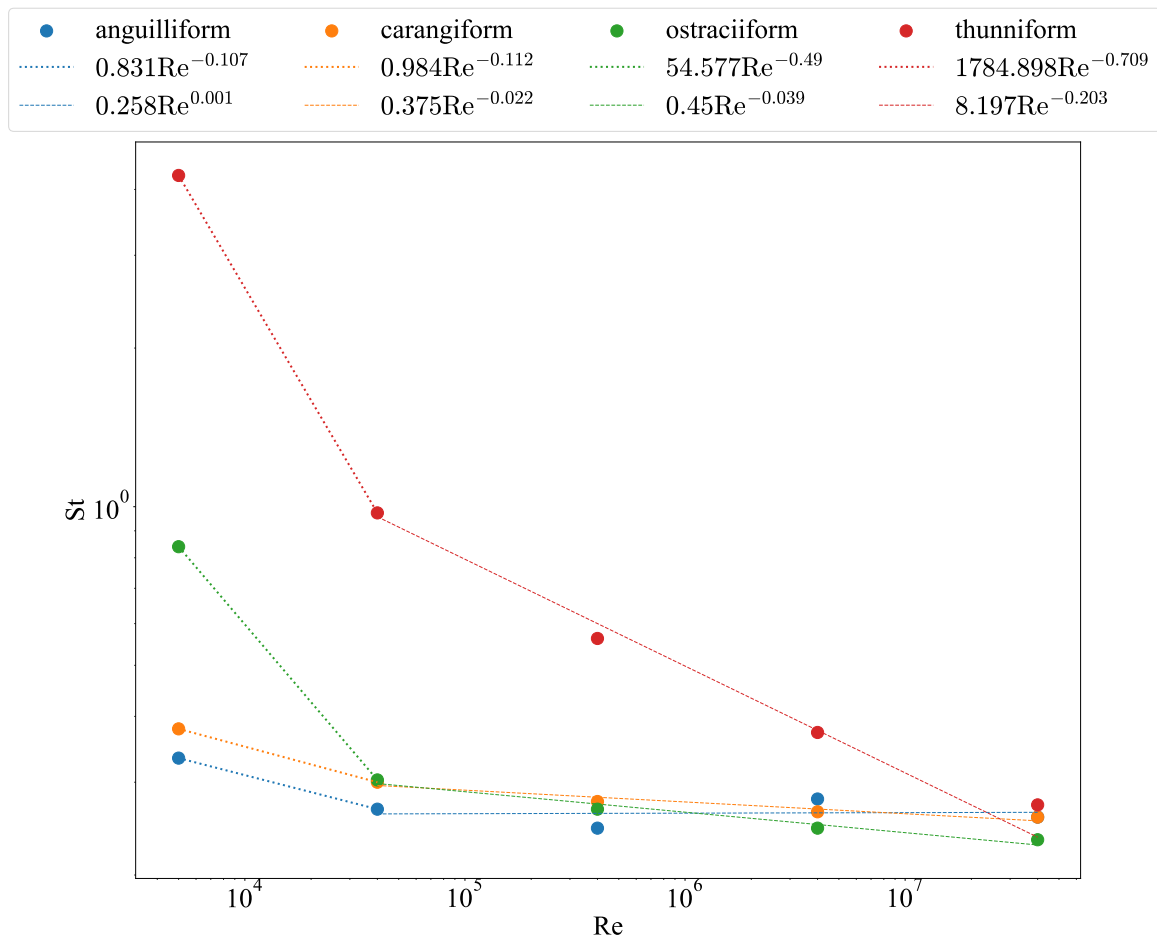


Fig. 4.31 Scaling of self propelled Strouhal numbers for all four locomotion modes. Note that the scaling is broken into two parts consistent with experimental observations given by Gazolla *et al.* [4]. The divergent scaling occurs at the transition from laminar to turbulent regimes.

Figure. 4.32 shows that carangiform motion requires less power overall with anguilliform, thunniform, and ostraciiform using increasing amounts of total power. The value of power for each Reynold number is taken at the SPS to give a piece-wise scaling relationship which is shown in Fig. 4.33 and summarized in Table 4.13. The power scaling shows consistency with the scaling laws given by Yu *et al.* [43], presented in Section 2.5, where an anguilliform and carangiform swimmer with the same parameters has the anguilliform motion requiring slightly more power than carangiform motion, which is consistent with experimental observations and simulations performed by Tytell [186].

Table 4.13 Summary of power Scaling for natural and artificial swimmers. All equations are of the form ax^b where x is the dependent variable and Reynolds number in for all data in this table.

Swimmer	a	b	R ²
Re $\leq 4 \cdot 10^4$			
Anguilliform	180.295	-0.254	1.0
Carangiform	576.943	-0.421	1.0
Ostraciiform	110.323	-0.065	1.0
Thunniform	108.442	0.095	1.0
Re $\geq 4 \cdot 10^4$			
Anguilliform	15.203	-0.025	0.95
Carangiform	6.064	0.005	0.98
Ostraciiform	51.1	0.007	0.98
Thunniform	26.153	0.038	0.98

The data in Figs. 4.29 and 4.30 shows that anguilliform and carangiform perform better than ostraciiform and thunniform in terms of power and requiring less thrust to overcome drag for higher Reynolds numbers. This is signified by having a lower Strouhal, St , crossover from negative thrust to positive thrust. Figure 4.33 shows that both ostraciiform and thunniform require far more energy exerted onto the fluid in order to obtain self-propulsion.

4.10.1 Pressure Coefficient

To understand the reason for this phenomenon, the pressure coefficient is investigated for a swimming cycle. The pressure coefficient, C_p is the non-dimensionalized pressure on the swimmer is

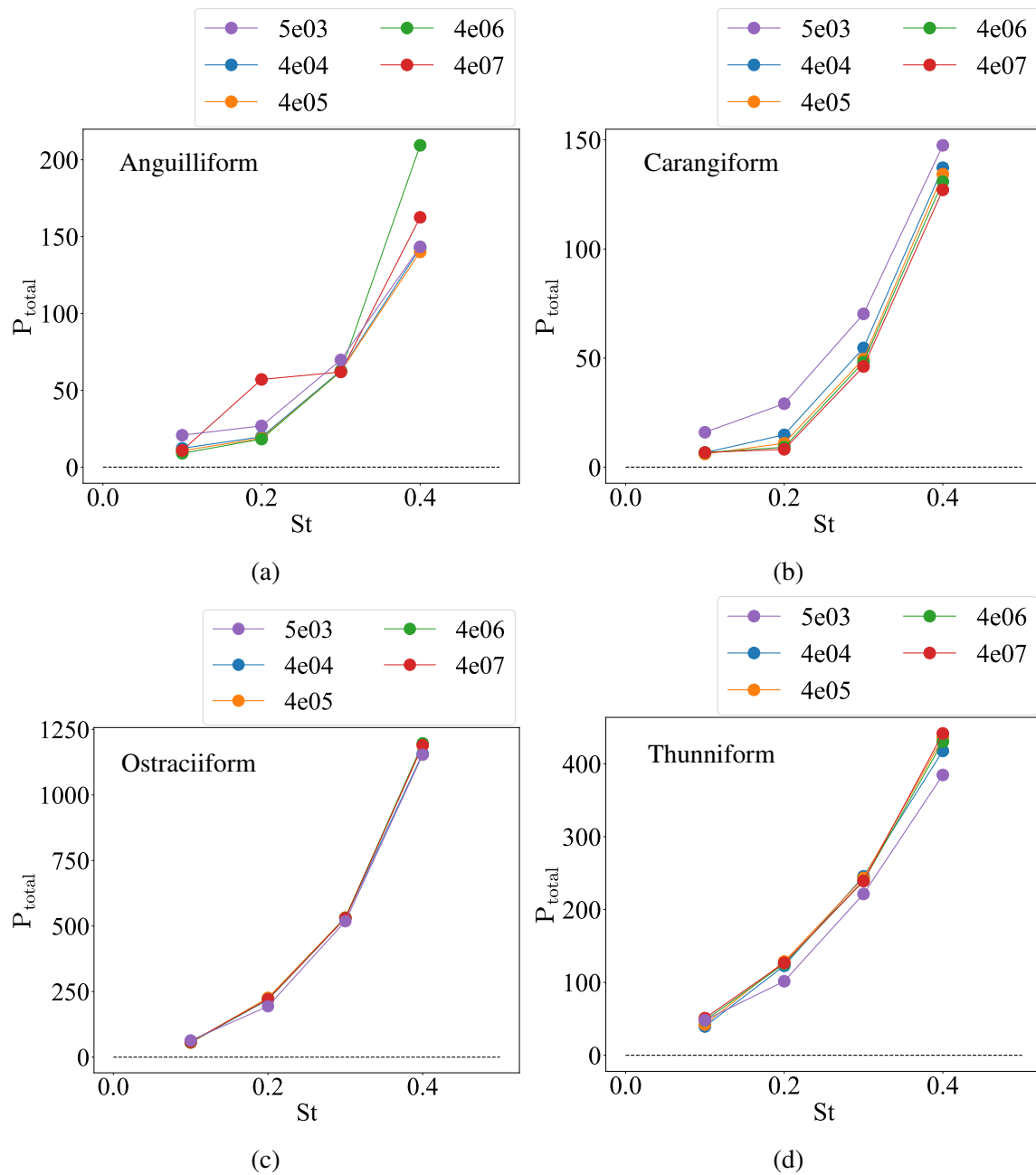


Fig. 4.32 (a) Anguilliform, (b) carangiform, (c) ostraciiform, and (d) thunniform total power.

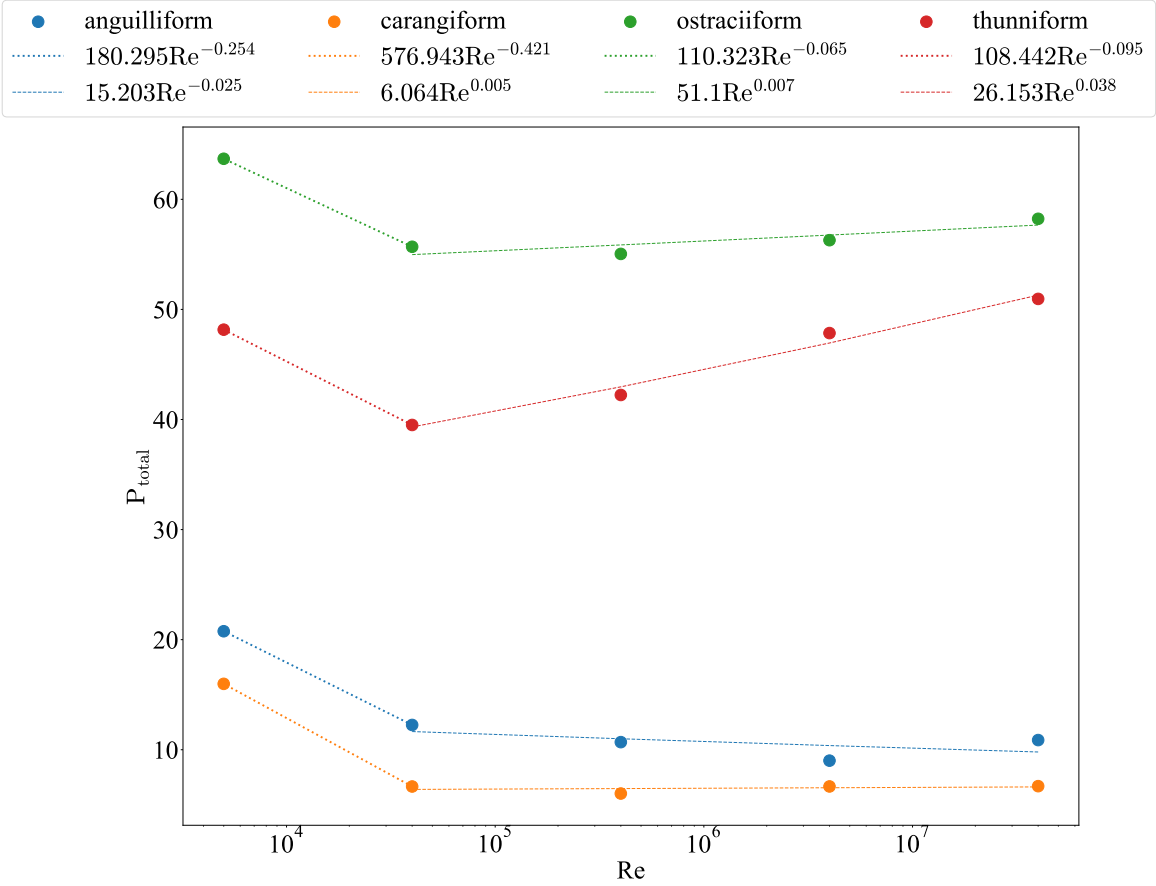


Fig. 4.33 Total power scaling for each locomotion mode.

$$C_P = \frac{(P_{incident} - P_{freestream})}{0.5\rho U^2}, \quad (4.46)$$

where $P_{incident}$ is the pressure on the swimmer, $P_{freestream}$ is the far field pressure, ρ is the fluid density, and U is the free stream velocity.

Figs. 4.34 and 4.35 show the swimmer body position and corresponding C_P values along the chord length done at Reynolds number of $5 \cdot 10^3$ and Strouhal number of 40. The figures show that throughout a swim cycle, the ostraciiform and thunniform locomotion modes have larger pressures incident on the swimmer, with the thunniform's pressure distribution being relatively smaller than the ostraciiform's. This gives a direct correlation between the pressure distribution on the swimmer and the amount of power needed for thrust. An interesting observation is that the ostraciiform mode does not have a leading edge pressure distribution that increases exponentially like the other three modes. This is a result of this mode being in a pure pitching motion versus an undulating mode of the other three which effectively provides a more blunt leading edge.

A further observation made in Figs. 4.34 and 4.35 is the abrupt change in pressure distribution at the trailing edge for the ostraciiform and thunniform cases. This is in contrast to the smoother transition seen in the anguilliform and carangiform cases. This is caused by the change in geometry for the pitching tail which gives a drastic change in fluid direction versus the smoothly transitioning tails. This causes a high pressure region on one side of the tail that is not present in the anguilliform and carangiform cases. This high pressure region adds to the total power needed to overcome the drag force for self propulsion.

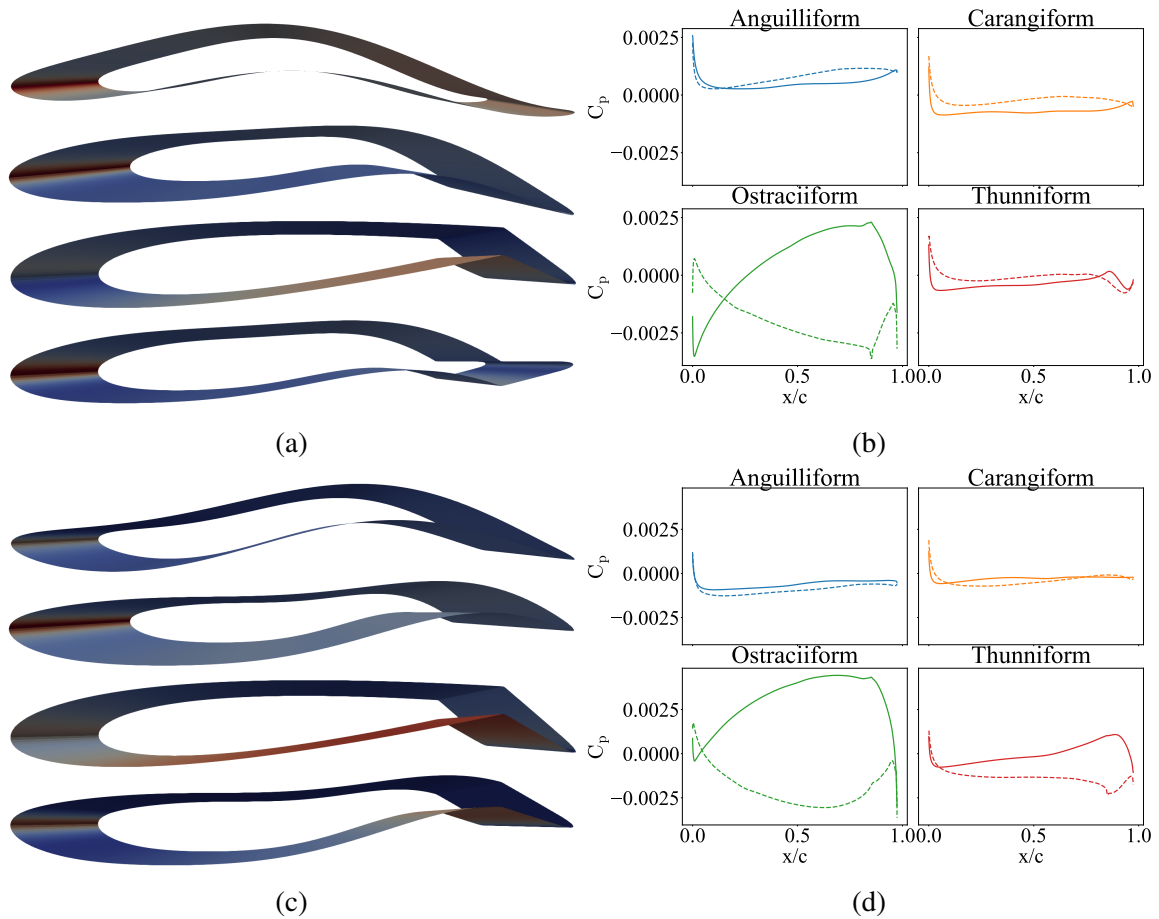


Fig. 4.34 (a), (c) Swimmer at first half of cycle time, $\frac{t}{\tau} = 0.25, 0.5$. Anguilliform, carangiform, ostraciiform, and thunniform modes from top to bottom. (b), (d) upper and lower pressure coefficient, C_p , for each locomotion mode plotted over the chord length. Simulations performed at Reynolds number of $5 \cdot 10^3$ and Strouhal number of 40. Solid lines represent the lower airfoil and dashed lines represent the upper airfoil.

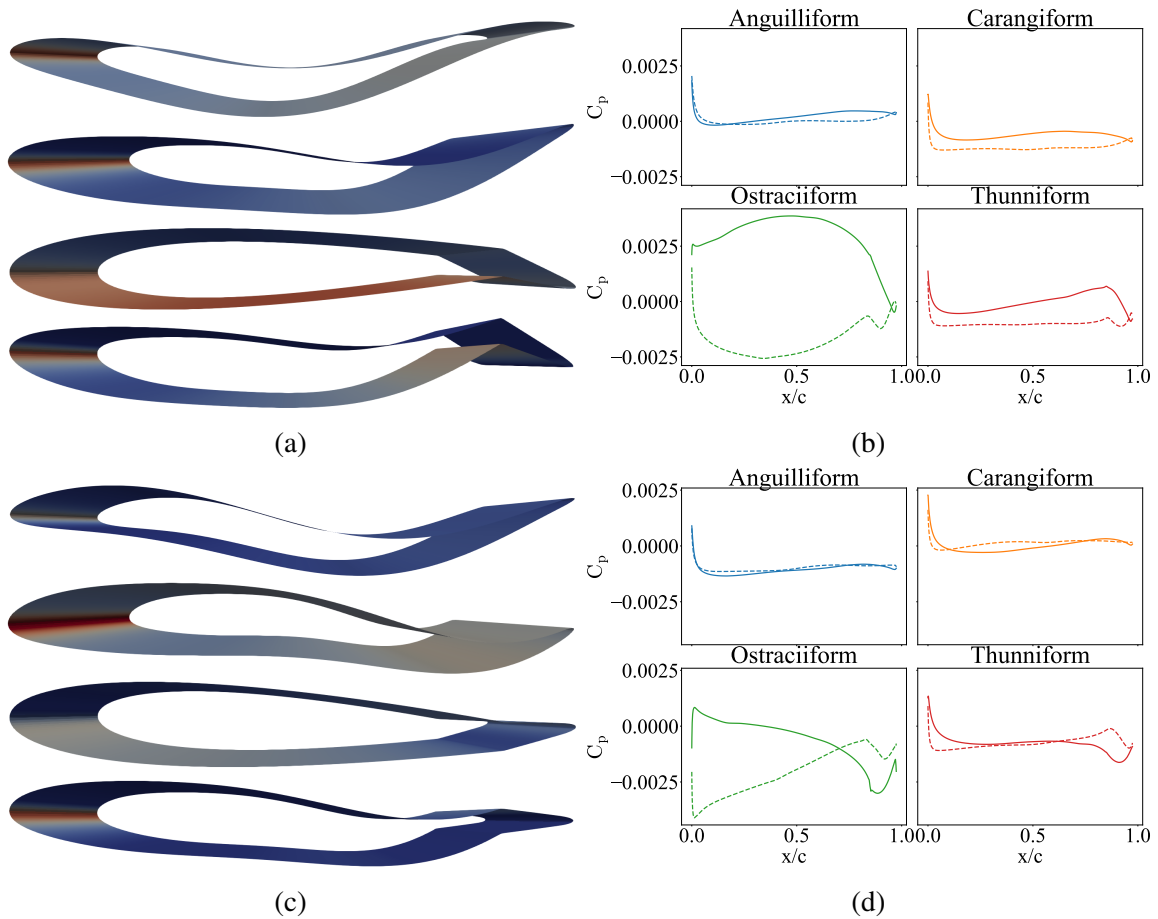


Fig. 4.35 (a), (c) Swimmer at first half of cycle time, $\frac{t}{\tau} = 0.75, 0.5$. Anguilliform, carangiform, ostraciiform, and thunniform modes from top to bottom. (b), (d) upper and lower pressure coefficient, C_p , for each locomotion mode plotted over the chord length. Simulations performed at Reynolds number of $5 \cdot 10^3$ and Strouhal number of 40. Solid lines represent the lower airfoil and dashed lines represent the upper airfoil.

4.10.2 Wake

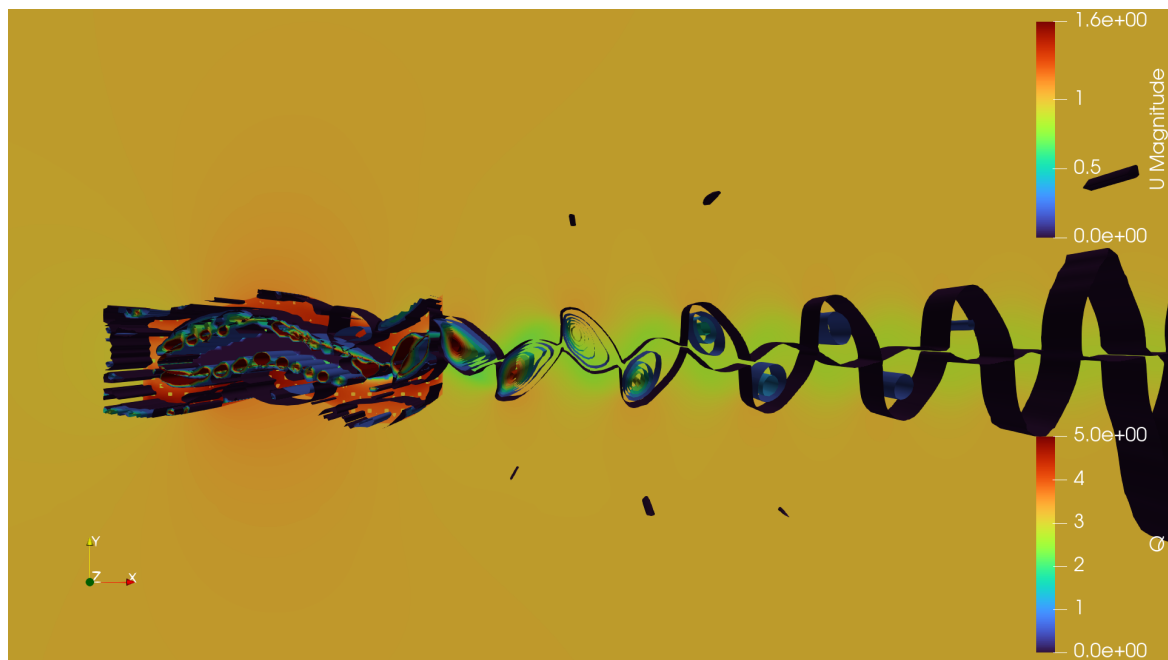
Another source of drag on the swimmer is the induced drag from the wake. Figures 4.36 and 4.37 show the velocity field and q-criterion for all four locomotion modes. Simulations here are the same as shown in the pressure coefficient plots. They are performed at Reynolds number of $5 \cdot 10^3$ and Strouhal number of 40 at a simulation time of 45 seconds. Interestingly, the q-criteria indicates that the anguilliform and carangiform cases have vortices that are more elliptical than that of the ostraciiform and, to a lesser extent, the thunniform. This can be explained by the presence of body undulation in the anguilliform, carangiform, and thunniform cases that is not present in the ostraciiform case. A further observation is that the wake of the cases in Fig. 4.37 is not as wide as the wake in Fig. 4.36. This means that the wake for the later cases dissipates into the free stream more than the first two cases. This may be causing increased drag, as the wake is not dissipating as fast.

The scaling laws presented in Fig. 4.33 provide evidence to the idea that COT is a direct result of locomotion mode and kinematic parameters. Considering the hotel power for each simulated swimmer to be the same, the COT is directly proportional to the power output since the geometry remained the same throughout each simulation. In all locomotion cases, the spread of data is very small with an excellent correlation to the power law fit. In the case of the anguilliform and carangiform, there are virtually no shifts in Reynolds numbers for several decades. This shows that length and velocity have very little effect on the output power of the swimmer. It can be inferred that the COT is more influenced by locomotion mode.

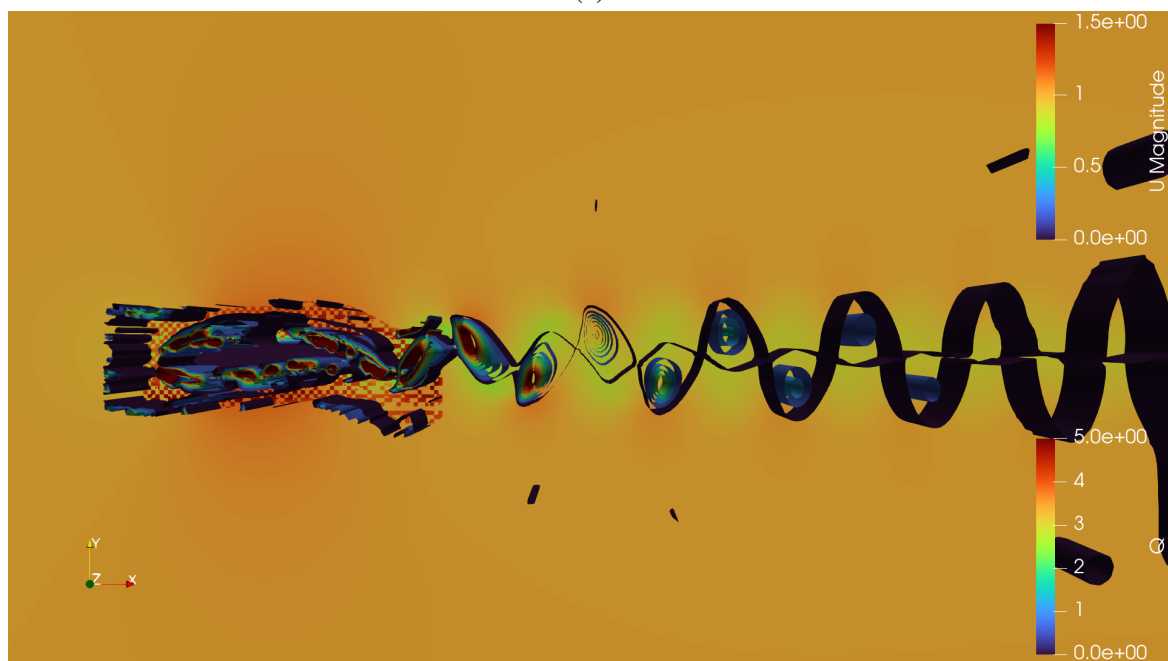
4.10.3 Optimum Linkages

The scaling laws derived last section, Section 4.10, can be roughly translated to the number of linkages and actuators in a bio-inspired vehicles by determining the relationship between locomotion modes, linkages, and actuators. Figure 4.38 shows the linkages and actuators for BCF bio-inspired vehicles. The data shows that Anguilliform motion requires the largest number of linkages at an average of 10. Carangiform and Sub-Carangiform modes require the next highest number of linkages at an average of 4. Finally, thunniform and ostraciiform require nearly the same number of linkages at an average of 2 and 1, respectively. Interestingly, the general trend is a linear relationship between the number of actuators and number of linkages with the number of actuators being around $N+1$ where N is the number of linkages.

From the data presented in Figures 4.33 and 4.38, it can be concluded that Anguilliform roughly translates to around 10 linkages, Carangiform and Sub-Carangiform to 4 linkages,

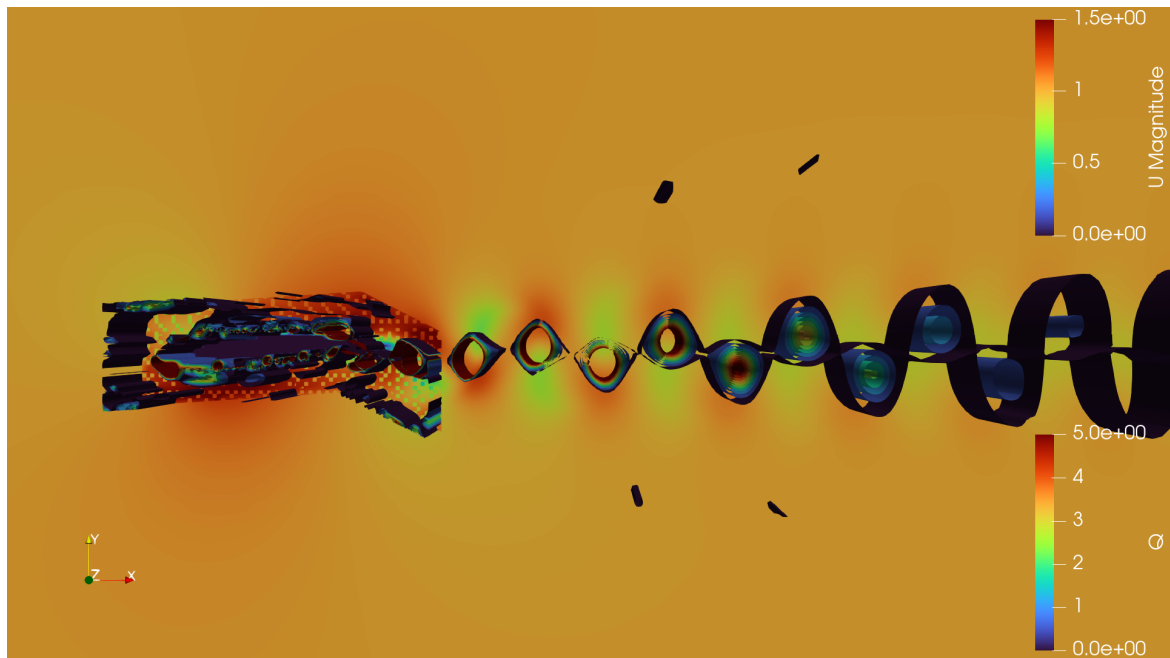


(a)

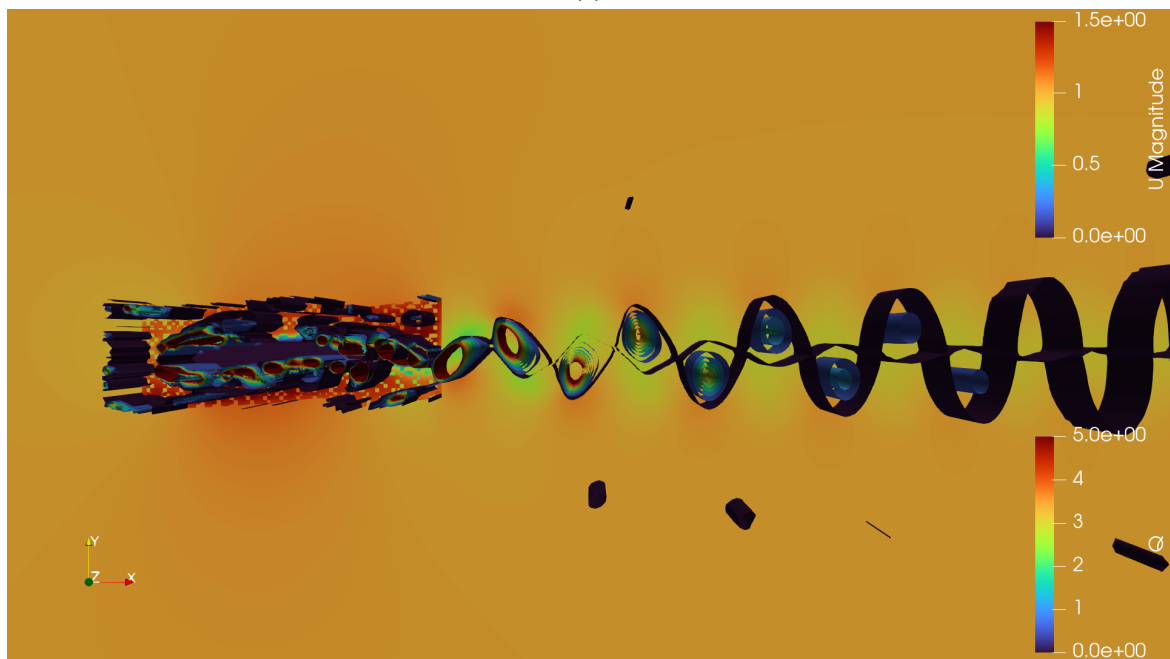


(b)

Fig. 4.36 Velocity magnitude and q-criterion for (a) anguilliform and (b) carangiform locomotion modes. Vortices show the characteristic reverse Kármán street present in biological swimmers.



(a)



(b)

Fig. 4.37 Velocity magnitude and q-criterion for (a) ostraciiform and (b) thunniform locomotion modes. Vortices show the characteristic reverse Kármán street present in biological swimmers.

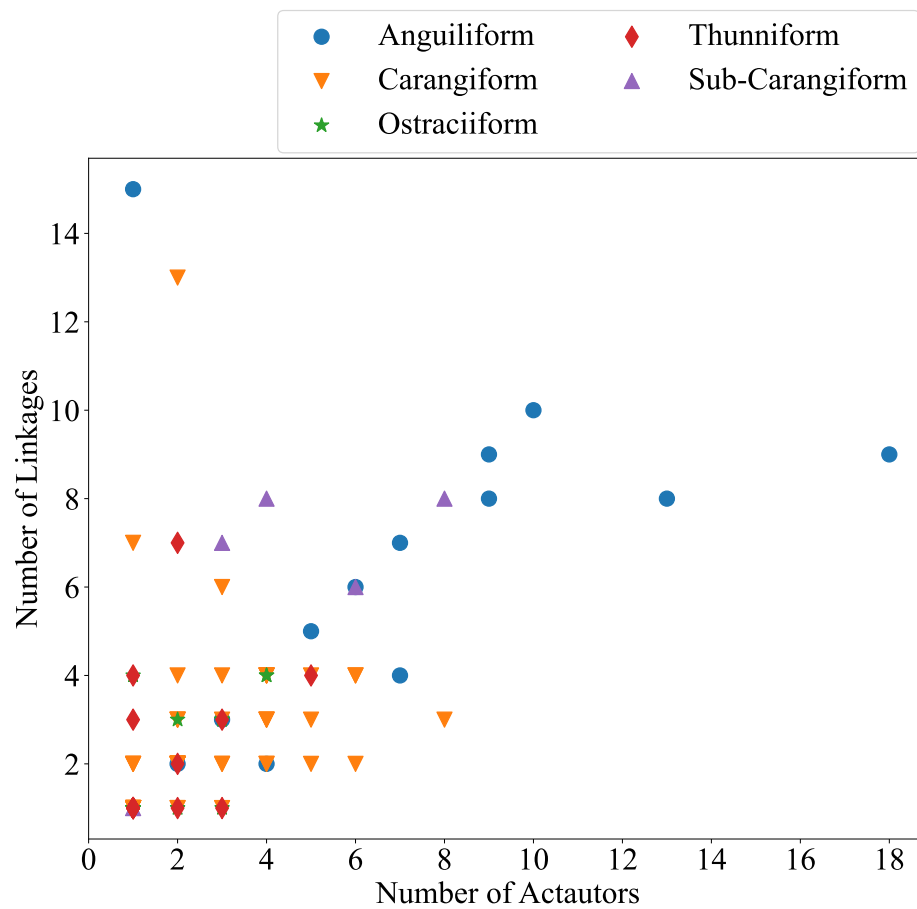


Fig. 4.38 Linkages and actuators for bio-inspired BCF vehicles.

Ostraciiform to 1 linkage, and Thunniform to 2 linkages. This means that there is actually a benefit to using more linkages to match the prescribed motion of fish swimming when designing a bio-inspired robot. The data presented here and the discussion given in Section 2.6.1 shows that there is an optimized number of ≈ 4 linkages.

4.11 Concluding Remarks

A motion library to simulate the BCF swimming modes of fish is presented in Section 4.5. The library is derived from parts of OpenFOAM®'s existing solid body motion library. This library was specifically chosen for its integration with the overset mesh framework within the larger OpenFOAM® library. It is shown that each swimming mode takes many cycles in order to reach a steady-state and that requiring a delay to first stabilize the calculations can be effective at limiting the amount of flow time needed. A comparison between grid sizes as well as overset meshes and body-fitted meshes is described in Section 4.9.3. The comparison shows that overset meshes provide similar results to body-fit meshes with a slight offset that is explained through interpolation error [223]. This is significant because overset meshes for fish motion are much more stable than body-fitted meshes, especially when there is no re-meshing available as in commercial packages like Ansys.

Section 4.9.5 shows a comparison between meshes generated using two different mesh generators: the commercial Ansys mesher and the open-source GMSH. There is a significant effect on drag with how the curvature of the trailing edge of the "airfoil" is captured. The Ansys mesher does a better job in capturing the curvature and having a smooth expansion rate into the free stream mesh. This results in a higher drag being captured by the Ansys mesh versus the GMSH mesh. More refinement of the GMSH mesh might result in a better drag result, but, for simplicity, the Ansys mesh is used for all simulations.

I have also presented a comparison with three popular turbulence models for low and transitional Reynolds numbers. At low Reynold's numbers, all three turbulence models perform similarly, but tend to diverge in the transitional regime. As there is no experimental or simulation data currently available to validate these models with, I have chosen to use the $k - \omega$ SST as it is widely used for CFD regarding fish. More research should be done in this area to determine which is the most suitable model for this specific use case.

Results for drag and power consumption show a clear delineation between the four locomotion modes. It is found that carangiform mode consumes the least amount of power than the other modes which is consistent with current literature. Figs. 4.34 and 4.35 in Section 4.10.1 show that output power is directly correlated with the coefficient of pressure distribution on the swimmer. A larger pressure distribution overall amounts to a higher power

needed to overcome body drag. While these results are not surprising, the pressure coefficient plots indicate an abrupt change in tail geometry when comparing the pitching tails with the smooth transitioning tails. This abrupt change in tail geometry causes a high pressure zone which drastically increases the power needed. Optimization can be done to decrease the amount of power required for the ostraciiform and thunniform locomotion modes. However, the data presented thus far is a comparison with the same driving parameters. A more rigorous study in 3D is needed to understand the coupling between locomotion modes, pressure distribution, and the amount of power needed to overcome body drag forces.

Finally, power scaling laws for output power are derived for the four locomotion modes. These modes show a clear delineation between the scaling of power from the viscous regime to the inertial regime. This is also consistent with a meta-analysis of biological fish performed by Gazzola *et al.* [4]. Importantly, it is found that the energy output for ever increasing Reynolds numbers is nearly constant. This shows that energy output is more of a function of locomotion mode than length or velocity.

Chapter 5

How Do Things Scale?

5.1 Introduction

Given the variety of designs and in engineered systems, and more specifically in bio-inspired swimmers, a direct equation that relates the performances of both natural and artificial swimmers at various Reynolds scales is difficult. This difficulty is further enhanced by the lack of published data pertaining to kinematic and energetic performance characteristics for conventional and bio-inspired AUVs. In lieu of a model relating all swimmers, it is helpful to determine how things scale for several decades of Reynolds numbers. A scaling model allows for a relative comparison between different swimmers, both artificial and natural. This scaling approach further allows for the identification of design features that are common to high performing platforms, as well as, acting as a baseline performance metric for designs of non-existent platforms. This chapter derives the scaling laws for both natural and artificial swimmers in terms of energetic and kinematic performance, as well as cost. It also discusses selected design aspects that are present in both well performing and under performing platforms.

5.2 Energetic Scaling of Natural and Artificial Swimmers

Energetic scaling of AUVs has already been reviewed in Section 2.3, where the emphasis was laid on the displacement of the vehicle to fit with the current literature and not the Reynolds Number. However, Fig. 5.1 shows the scaling of biological, bio-inspired, and conventional AUVs for several decades of Reynolds numbers and a summary is provided in Table 5.1. The Reynolds number was calculated using the length of the vehicle and the optimum speed, if provided. When the optimum speed was not reported, the maximum speed was selected.

Engineered systems show a wide range, particularly bio-inspired BCF platforms. This can be partially explained by the applications of these platforms as discussed in the next chapter. These platforms are mainly used for research into fish swimming without a focus on energy consumption and efficiency. This is particularly salient when comparing the spread of data to biological animals or conventional AUVs, which are optimized to increased mission times or for certain trophic levels as discussed in Fig. 2.9 of Section 2.3.2.

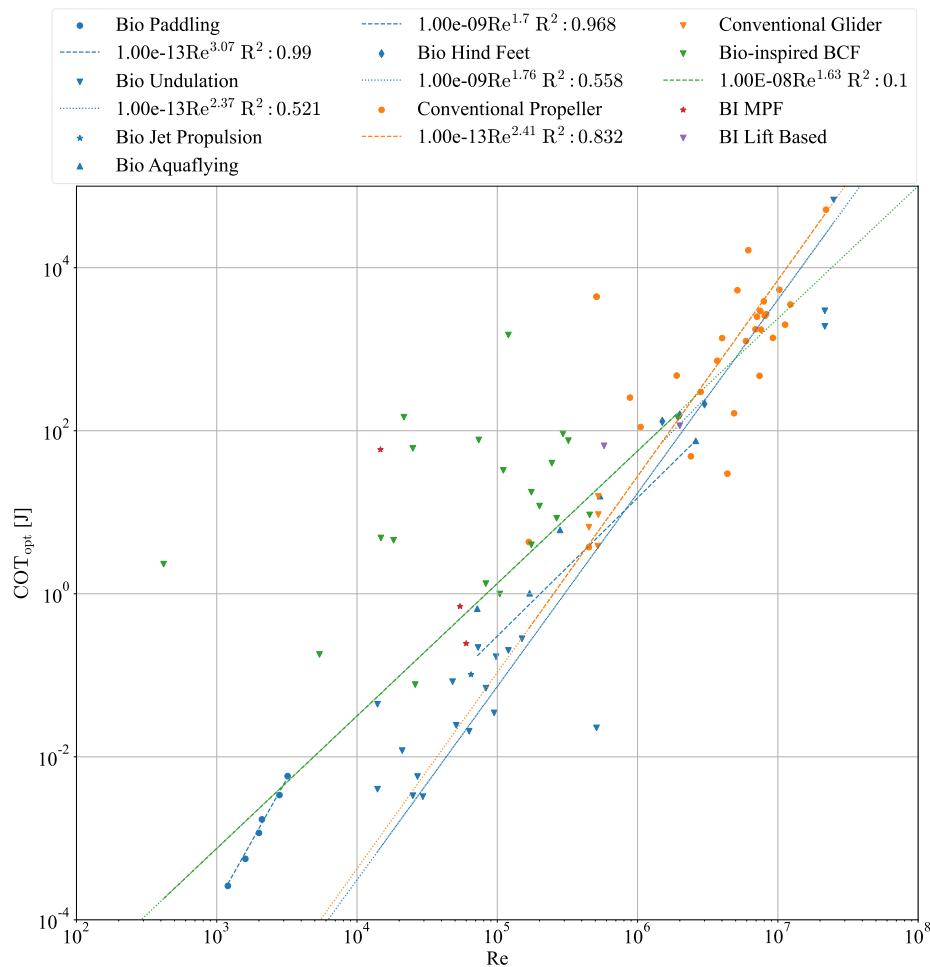


Fig. 5.1 COT of biological and artificial swimmers. Biological data is from Videler and Nolet [8]. Conventional Propeller and Glider information is taken from [10–15]. BI Carangiform data is from [16–27]. BI Ostraciiform data is from [28, 29]. BI Thunniform data is from [30–32]. BI Anguilliform data is from [33–35]. BI Sub-Carangiform data is from [36]. MPF data is from [37–39]. Lift based data is from [40, 41]. Best fit allometric trend line for power law fit is shown in legend underneath their corresponding data set.

The data in Fig. 5.1 shows that conventional AUVs and undulating biological animals are nearly the same for 4 decades of Reynolds numbers, with the amplitude and exponent of the scaling being similar. Interestingly, the spread for undulating species of natural swimmers

Table 5.1 Summary of COT Scaling with Reynolds Number for natural and artificial swimmers. All equations are of the form ax^b where x is the dependent variable and in this case is Reynolds number (Re).

Swimmer	a	b	R^2
Bio Paddling[8]	$1 \cdot 10^{-13}$	3.07	0.990
Bio Undulation[8]	$1 \cdot 10^{-13}$	2.37	0.521
Bio Aquaflaying[8]	$1 \cdot 10^{-9}$	1.70	0.968
Bio Hind Feet[8]	$1 \cdot 10^{-9}$	1.76	0.558
Bio-inspired AUVs	$1 \cdot 10^{-8}$	1.63	0.1
Conventional AUVs	$1 \cdot 10^{-13}$	2.41	0.832

is quite large. This can be explained as a result of different testing methodologies and the difficulties of standardized laboratory tests on animals that are several meters long. Another reason for this spread is the variety of body geometries in biological animals. As discussed in the review of COT literature, Fig. 2.6 of Section 2.3, body geometry has a noticeable effect on the COT. A further observation about this data is that the spread of data for the same species is quite low, even at different Reynolds numbers. Fig. 5.1 shows this in the biological padding and aqua flying data set. The COT is related more to locomotion mode rather than length or velocity as discussed with the power scaling plots, Fig. 4.33 in Section 4.10. The same species is assumed to have a similar body geometry and locomotion mode, therefore a direct comparison of the effect of Reynolds number can be made.

The data for bio-inspired BCF platforms shows a large spread with a weak correlation to the power law fit versus the natural swimmers and conventional AUVs. This large spread of data is in contrast to the power law scaling correlation when the same data is plotted versus displacement as in Fig. 2.13 of Section 2.3.3. Fig. 2.13 is reproduced here in Fig. 5.2 for convenience. Here there is a much better correlation in the data with the allometric power law fit.

5.3 Kinematic Scaling of Bio-inspired Robotic Platforms

Following the development of the scaling relationship, (2.27) in Section 2.4, Cha *et al.* [258] performed a similar analysis for bio-derived BCF robots. In this research, a larger set of bio-inspired robotic platforms is examined, and a comparison with biological swimmers is made. Figure 5.3 shows only the fish data presented in Fig. 2.15 and bio-inspired robotic data for BCF, MPF, and Lift-based swimmers. In instances where the amplitude was not explicitly given, an estimation was made based on the maximum tail angle, assumed to be in

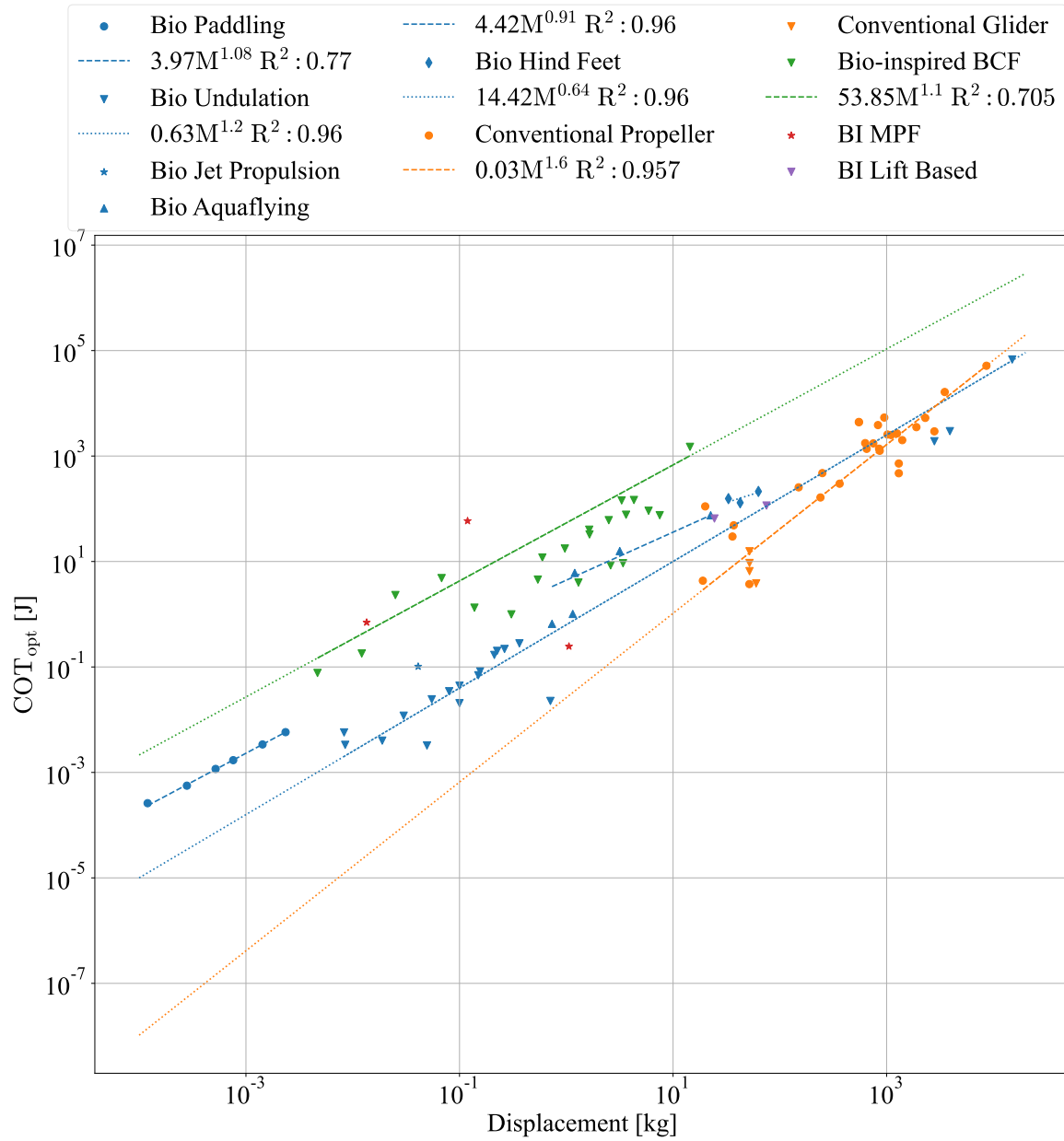


Fig. 5.2 COT of biological and artificial swimmers. Biological data is from Videler and Nolet [8]. Conventional Propeller and Glider information is taken from [10–15]. BI Carangiform data is from [16–27]. BI Ostraciiform data is from [28, 29]. BI Thunniform data is from [30–32]. BI Anguilliform data is from [33–35]. BI Sub-Carangiform data is from [36]. MPF data is from [37–39]. Lift based data is from [40, 41]. Best fit allometric trend line for power law fit is shown in legend underneath their corresponding data set.

the body fixed reference frame. The trigonometric relationship for a right triangle is used to determine the maximum deflection of the tail in relation to the body. The amplitude is given in units of body lengths. For lift and MPF based swimmers, the fin amplitude is calculated using the fin as the reference length and then divided by the body length.

The data shows a large spread of data of bio-inspired platforms compared to biological data, which is expected of engineered systems. There is a weak correlation in both the BCF and MPF data in the turbulent region as follows:

$$\text{Re} = \begin{cases} 5.34\text{Sw}^{0.76} & R^2 : 0.342 & \text{BCF,} \\ 10.24\text{Sw}^{0.73} & R^2 : 0.407 & \text{MPF.} \end{cases} \quad (5.1)$$

Furthermore, in most cases, bio-inspired robots at the same Reynolds number have a lower swim number than their biological counterparts. This provides evidence that the actuation scheme used in bio-inspired platforms cannot undulate as fast or with the same amplitude as an equivalent natural swimmer. The data in Fig. 5.3 is further discretized into motor/servo driven vehicles and vehicles that use smart actuators. The comparison between these two actuation schemes shows that a few motor driven platforms do perform as well as their biological counterparts. However, no smart actuator platforms perform as well, except for one platform in the viscous regime.

The scaling relationship defined in (2.27) can be used to determine the kinematic parameters of bio-inspired robots in the following way: recall from (2.26) that there are 4 control parameters (ω , A , L , U) with L being shared between Sw and Re . Note that μ is a fixed property of the fluid that the robot operates in. Suppose there is a bio-inspired robot with body length at $L = 0.3$ m and the speed of the robot to be 1 m/s which determines the Re number to be 300,000. For undulating fish, the Sw gives the relationship of $\omega A \approx 1.918$ [m/s]. From the data from Videler [42], reproduced in Appendix A, we can fix the amplitude to 0.1 body lengths which makes $A = 0.2$ [BL]. This means we need a linkage and actuator system that can achieve ≈ 32 rad/s or ≈ 5 Hz.

This metric can further be used to determine how viable a biologically-inspired robotic design is. At the minimum, the kinematic performance of the design should be able to reach the scaling laws for bio-inspired platforms, (5.1), but would ideally be able to perform up to the scaling for biological swimmers, (2.27). If a particular actuator scheme and design cannot meet these performance criteria, then the design should be reconsidered. Furthermore, these scaling laws can be used in the next-generation of platforms as a baseline performance metric.

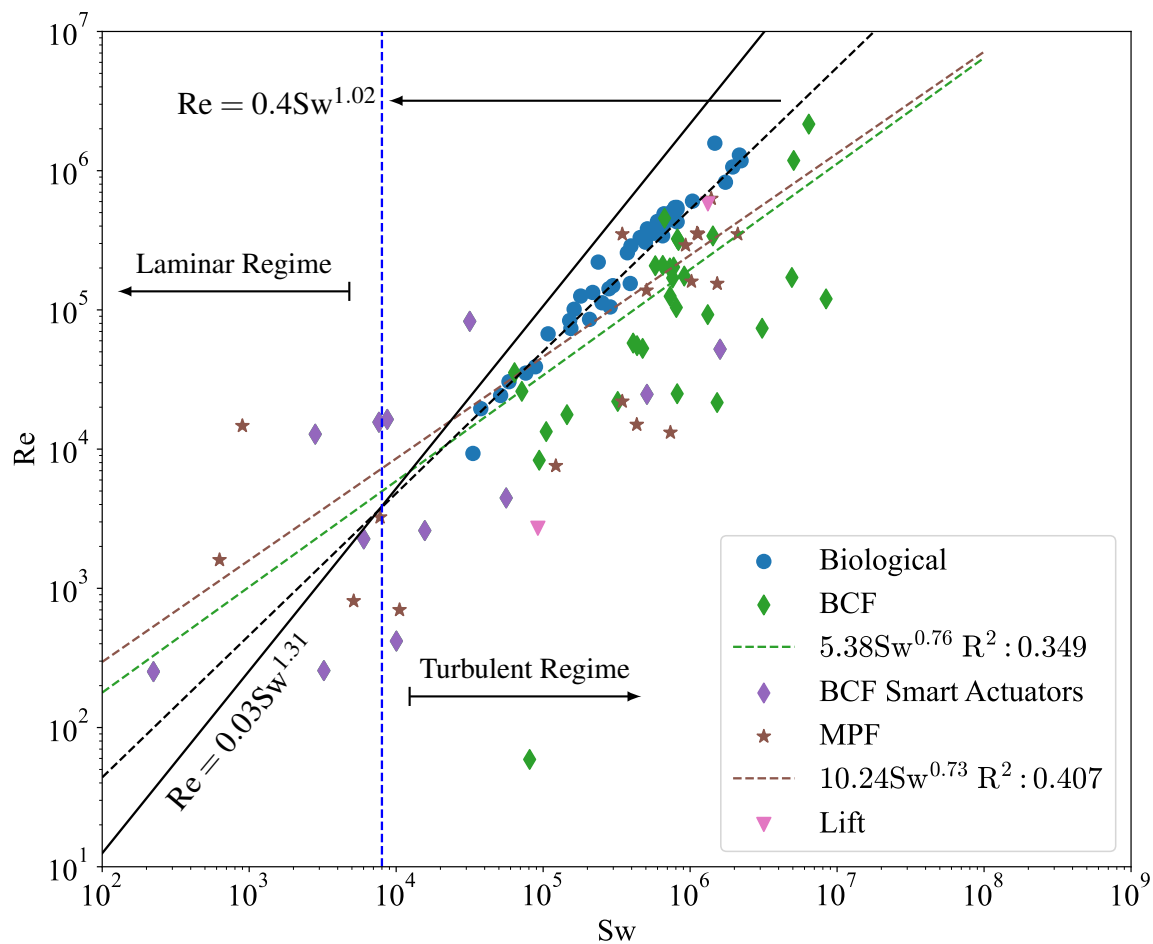


Fig. 5.3 Swim vs. Reynolds number for a collection of swimming animals and bio-inspired robotic platforms. Data for fish taken from Videler [42]. BCF data taken from [16, 17, 19, 27, 28, 30, 36, 53, 57, 80–110]. MPF data taken from [37, 111–127]. Lift data taken from [40, 128]. Best fit allometric trend line for power law fit is shown in legend underneath their corresponding data set.

Combining the COT and kinematic metrics, we can compare different bioinspired AUV platforms. The correlation between good and bad performers for both metrics is provided in Fig. 5.4. As a note, the best performing bio-inspired platforms in both metrics are the carangiform BCF mode Polish Naval Academy Cyberfish described in Morawski *et al.* [19] and the lift-based Madeline robot developed by Long *et al.* [67]. The University of Essex iSplash Micro developed by Clapham and Hu [84] performs well in terms of swimming performance, but not in terms of energy consumption.

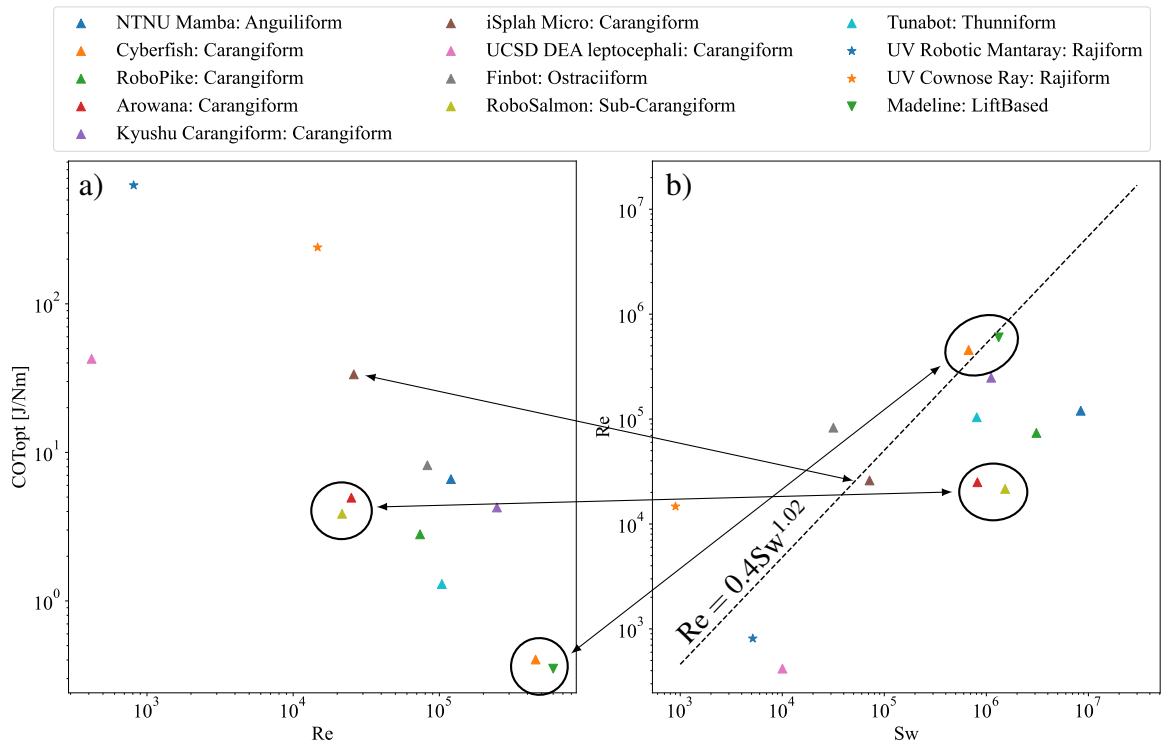


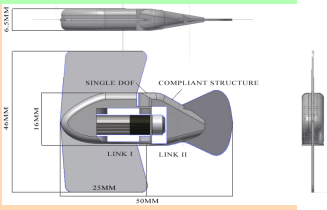
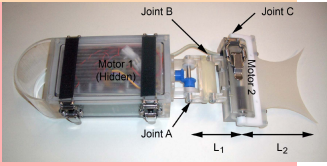
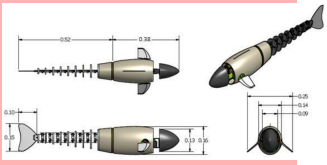


Fig. 5.4 Optimum COT versus Reynolds number and Reynolds number versus Swim number for selected AUV platforms. Arrows give the correlation between the same platforms on the different plots. Black dashed line is the swim number scaling line from Gazzolla *et al.* [4].

There are a few design cues that show up in the most successful performers, Table 5.2. The first is the use of a rigid hull versus a more flexible body, as shown in the Cyberfish, Madeline, and iSplash Micro vehicles. In these three designs, the body is mostly rigid and has a minimal amount of changes in body geometry between the body, linkages, and fins. We can see this clearly when comparing the Cyberfish to the Arowana design. The Arowana has a drastic cutoff of body geometry between the body, linkages, and fins that is not present in the Cyberfish. As discussed in Fig.4.33 in Section 4.10, drastic changes in body geometry cause a high pressure zone which increases the body drag and thus the power needed to

Table 5.2 Published Images of designs from Fig. 5.4.

Name	Affiliation	Performance	Image
Cyberfish [19]	Polish Naval Academy	Good kinematic and COT	
Madeline / Rhex [67]	Vassar College	Good kinematic and COT	
Isplash Micro [84]	University of Essex	Good kinematic Bad COT	
Arowana [17]	Nanyang Technology University	Bad kinematic and COT	
RoboSalmon [36]	University of Glasgow	Bad kinematic and COT	

overcome this drag. It is preferable to have a smooth transition along the body to negate any high pressure zones along the body.

The other design feature present here is the side cross section being in the shape of an airfoil. Looking at the Cyberfish, Madeline, and Isplash Micro platforms, the side view is shaped to look more like a fish which resembles the cross section of an airfoil. The Cyberfish in particular has a top down cross section of an airfoil. It performs better in terms of energy consumption due to the more streamline cross-section and the lack of bluntness. This is in contrast to the Arowana and RoboSalmon which are more blunt-bodied than the high-performing platforms.

5.4 Adapting Swim Number to Conventional AUVs

To assess the relevance of this scaling methodology to conventional propeller AUVs, a new dimensionless number is presented that is functionally similar to that of the Swim number, the Propulsion Number J_w . Unlike undulating fish or bio-inspired robots, conventional AUVs do not have a Strohahl number, but a similar dimensionless number can be found in the advance ratio, defined as

$$J = \frac{U}{\omega D}, \quad (5.2)$$

where U is the vehicle's velocity in meters per second, ω is the frequency of propeller in revolutions per seconds, and D is the diameter of the propeller in meters. Similarly to equation (2.26), I define the following Propulsion number J_w as:

$$J_w = \frac{Re}{J} = \frac{UL}{\nu} \cdot \frac{\omega D}{U} = \frac{\omega DL}{\nu}. \quad (5.3)$$

The Reynolds number and J_w are plotted against each other in Fig. 5.5. The data for the Wayamba is fit separately because it is a flatfish style AUV with two main thrusters. This arguably puts it into a different class than the other torpedo type, single thruster AUVs. In both cases, the scaling line for the turbulent region from Gazzola *et al.* [4] is provided to show a relative performance. The scaling relation for the Reynolds number with Swim number, Sw , presented in (2.27) of Section 2.4, to get an idea of the relative performance of AUVs versus biological swimmers. It should be noted that these two numbers are functionally the same but do not give a one-to-one comparison between the kinematics of each class of swimmer. The Sw trend line is only provided for reference.

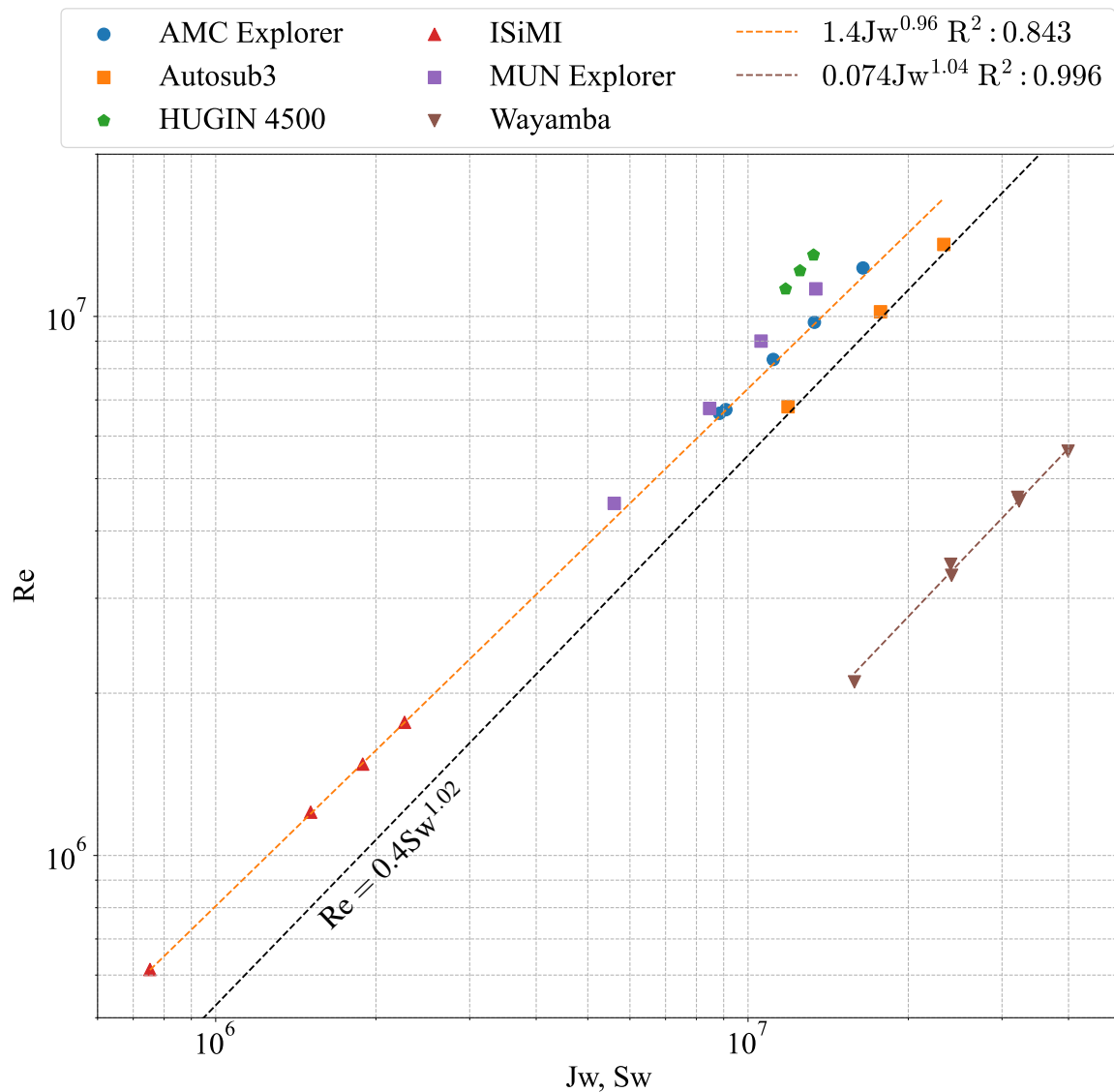


Fig. 5.5 Swim vs. Reynolds number for a collection of AUVs. Data taken from the following: AMC Explorer [129], MUN Explorer [130], HUGIN 4500 [131], Autosub3 [132], IM and Wayamba [133]. Black dashed line is the scaling relationship derived in Gazzola *et al.* [4]. Orange dashed line is the scaling relationship of the torpedo shaped single thruster AUVs and the brown dashed line is the scaling line for the flatfish type AUV, Wayamba.

Similar to the procedure for bio-inspired AUVs, the scaling relation of the propeller number, $Re = 1.4Jw^{0.96}$, can be used to assist in designing a new platform or comparing existing platforms to each other. When designing a new platform, the length and kinematic viscosity are known. The mission dictates the velocity for the AUV so the speed is known. The propulsion number, Jw , is calculated from the Reynolds number and an equation for the diameter of the propeller and frequency of rotation becomes

$$\omega D = \frac{Jw v}{L}. \quad (5.4)$$

Propeller diameter is functionally the same as the amplitude in Sw , but unlike biological swimmers, there is no average value for propeller diameter. Fig. 5.6 gives the diameter of each AUV plotted against the propeller (prop) diameter. Excluding the ISiMI and Wayamba, which is of a different design than torpedo style AUVs, the general rule for commercial AUVs is that the prop diameter is between 70-93% of the diameter of the AUV. So if we have an AUV like the Hugin 4500 which is 6 meters long, 1 meter in diameter, and has an optimum speed of 2.06 m/s. This gives a Reynolds number of $1.24 \cdot 10^7$. Rearranging the Propulsion number scaling law for Jw , $Jw = \left(\frac{Re}{1.40} \right)^{1/0.96}$, gives a Propulsion number of 17,249,011.76 which is substituted into (5.4) to provide the relation $\omega D = 2.87$ [m/s]. The diameter of the propeller can be designed to be any number so the limiting factor will be how fast the actuation scheme can turn the prop. In this case, the diameter of the propeller is known to be 0.7m which means that the motor will need to be able to spin at 4.1 rpm. The benefit of an approach such as this is that it is valid for all Reynolds numbers and therefore it can be used to determine the performance of a design if it was made larger or smaller.

Similarly to the Swim number with biological animals and bio-inspired AUVs, the Propulsion number can be used as a metric for the kinematic performance of an AUV. If a specific design cannot satisfy the scaling law, then parameters should be adjusted. It should be noted that there is a limited data set for the derivation of the Propulsion number. This is due to the lack of published designs for commercial AUVs, especially propeller diameter, velocity, and propeller angular frequency. Further research is needed to compile data and run experiments on a larger set of commercial AUVs for a better correlation with available data.

One caveat with the Propulsion number is that it is functionally the same as the Swim number, Sw , but it does not provide a one-to-one relationship between different platforms like the Sw does with biological and bio-inspired platforms. The Propulsion number does, however, allow for the calculation of kinematic parameters, such as the angular frequency of actuation ω . In this case, the angular frequency is defined the same for biological animals

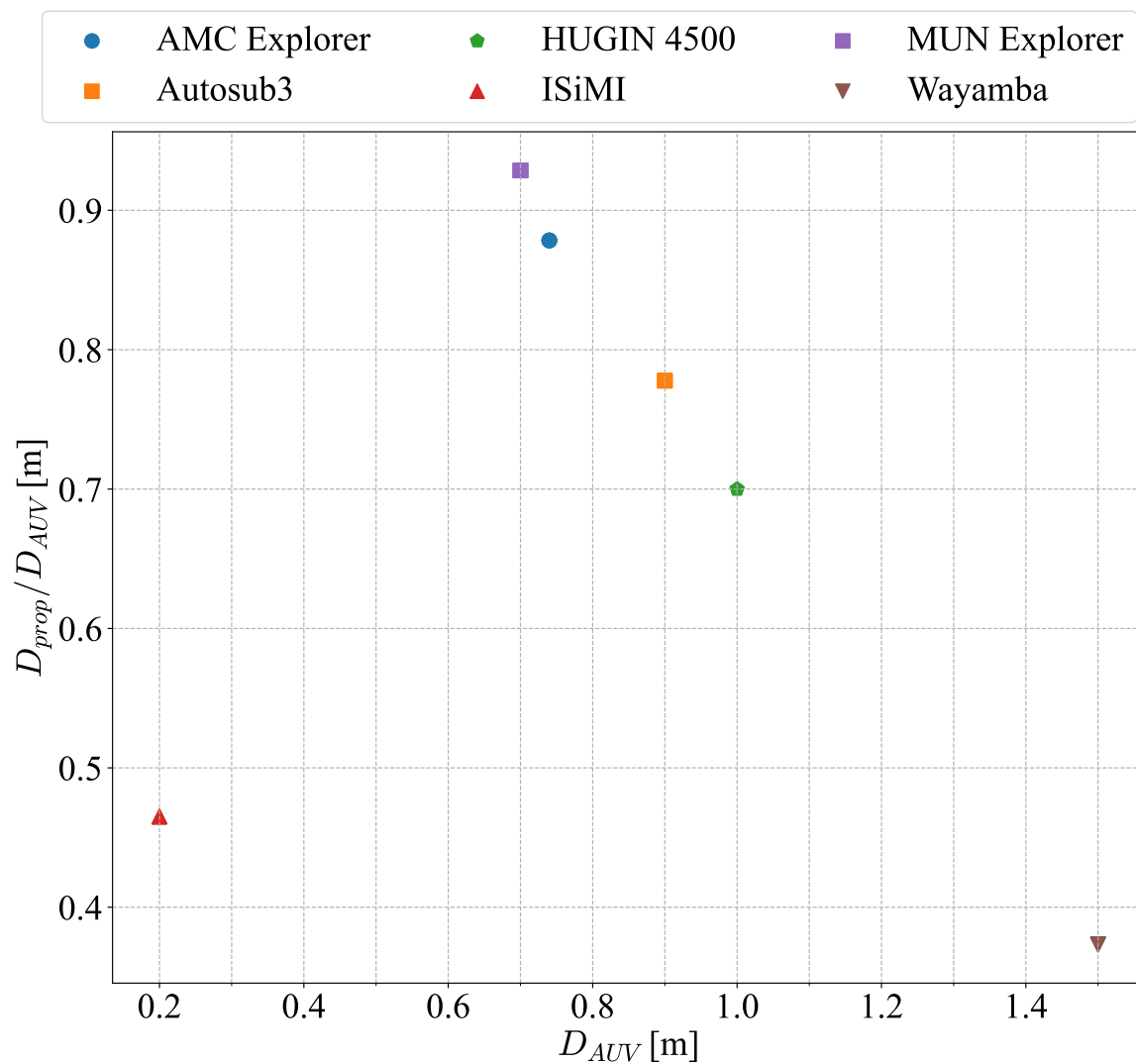


Fig. 5.6 Diameter of AUVs versus their propeller diameter. Data taken from the following: AMC Explorer [129], MUN Explorer [130], HUGIN 4500 [131], Autosub3 [132], IM and Wayamba [133].

and conventional AUVs. This provides a relative comparison of how fast the actuators of a design would need to be for a particular speed or Reynolds number. This would allow for a one-to-one comparison of at least the kinematic performance between platforms.

5.5 Cost Scaling of AUVs

There is very little analysis comparing the cost of developed platforms to existing ones. Before understanding the cost associated with making an AUV, the anatomy, payload, and modular design must be discussed. Henthorn *et al.* [135] describes the Monterey Bay Aquarium Research Institute Dorado Class AUV and a schematic is given in Fig. 5.7.

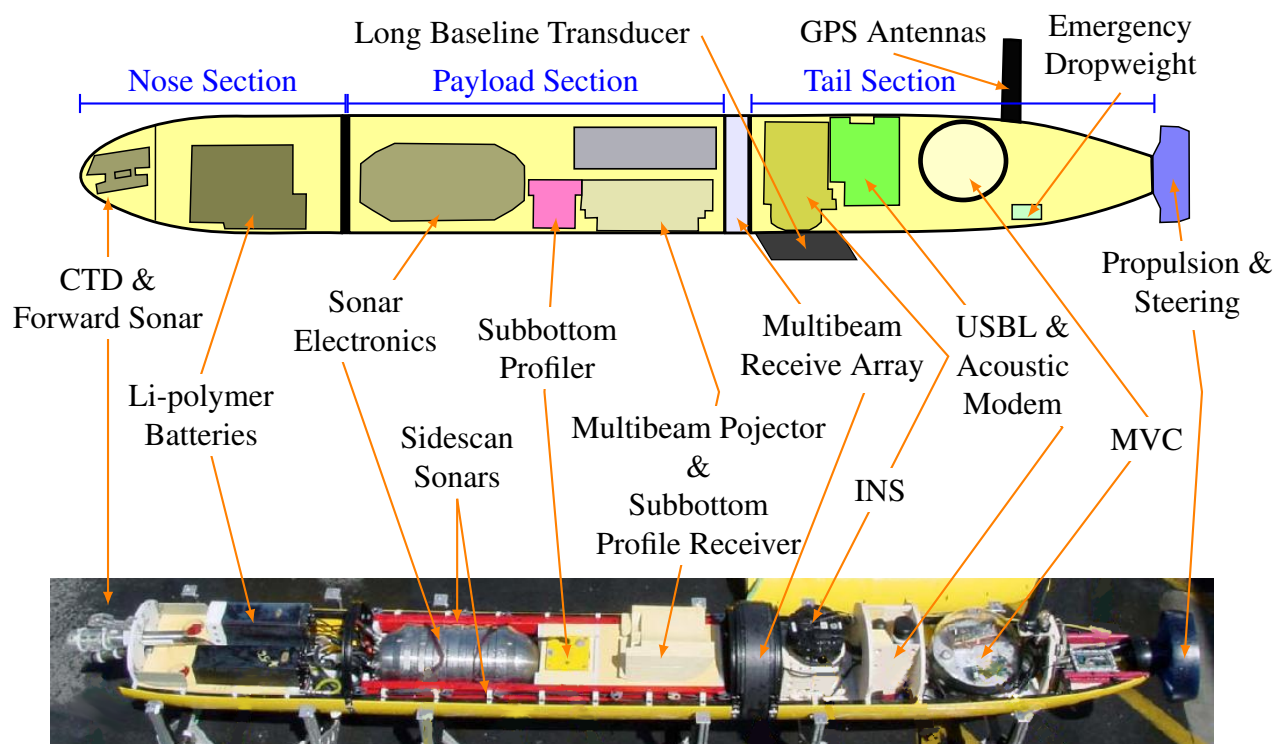


Fig. 5.7 Schematic of the interior layout of Dorado Class AUV adapted from a combination of schematics given in Sanchez *et al.* [134] and Henthorn *et al.* [135]. Real picture given in Henthorn *et al.* [135].

The dimensions of the Dorado Class (MBARI) AUV pictured in Fig. 5.7 provides a reference for what is needed to produce an AUV with a mission of ocean surveying. For reference, the MBARI AUV has a length of 5.24 meters and a diameter of 0.54 meters. For this analysis the cost of sensors and computers is fixed. The majority of cost is then the structural components (hull, buoyancy foam, tooling) and actuators (motors, servos, etc.).

Structurally, the hull of an AUV has to withstand hydrostatic pressure when at depth on a mission which is given as:

$$P_{hydrostatic} = \rho gh, \quad (5.5)$$

where ρ is the density of the fluid, g is the gravitational acceleration 9.80665 m/s^2 , and h is the depth of the fluid column. The density of saltwater for 30% salinity at 10°C is given by the International Towing Tank Committee [259] as 1027.00 kg/m^3 . To determine a suitable hydro-static pressure, the depth for conventional and bio-inspired AUVs is presented in Fig. 5.8. The figure shows that bio-inspired AUVs only reach a depth of 100 m while conventional AUVs cluster around 1,000-2,000 m and 6,000 m. Six thousand meters is the depth of ocean before reaching oceanic trenches as given by the National Oceanic and Atmospheric Administration data [260] which makes it an ideal depth to design for. The hydro-static pressure that the vehicle must withstand is then 594.41 atmospheres. The most common materials used for the structure of AUVs is reviewed in Griffiths [141] and Table 5.3 gives these materials and specifics. Costs for each material are provided by cost catalogs from the Formula SAE student competitions [261].

Another important structural component is the foam used to make the AUV neutrally buoyant in water. The equation for the volume of foam needed is given in Griffiths [141] by:

$$V_{foam} = \frac{M_{excess}}{|\rho_b - \rho_{sw}|}, \quad (5.6)$$

where M_{excess} is the excess weight in kilograms, ρ_b is the density buoyancy system in kg/m^3 , and ρ_{sw} is the density of sea water in kg/m^3 . Typical buoyancy foams with maximum operating depth and density are shown in Table 5.3.

To obtain a relative idea of the cost scaling for different vehicles, it is assumed that the surface area of the vehicle is directly proportional to the cost of the AUV. This analysis only considers the cost of the outer hull materials and assumes the same thickness of material for each vehicle. The surface area in terms of displacement weight and length is given in Fig. 5.9. Surface area was calculated using the equivalent spheroid method discussed in Section B.0.2, (B.9). This method was used due to not needing images for the platform. The data in Fig. 5.9 shows a strong correlation between surface area and both weight and length. Interestingly, the surface area increases at a greater rate with length than it does with weight.

The cost of actuators adds to the total cost of the AUV. Motor data was gathered from Maxon Inc. [44] and servo data from Hitec Inc. [45]. Price is converted to United States

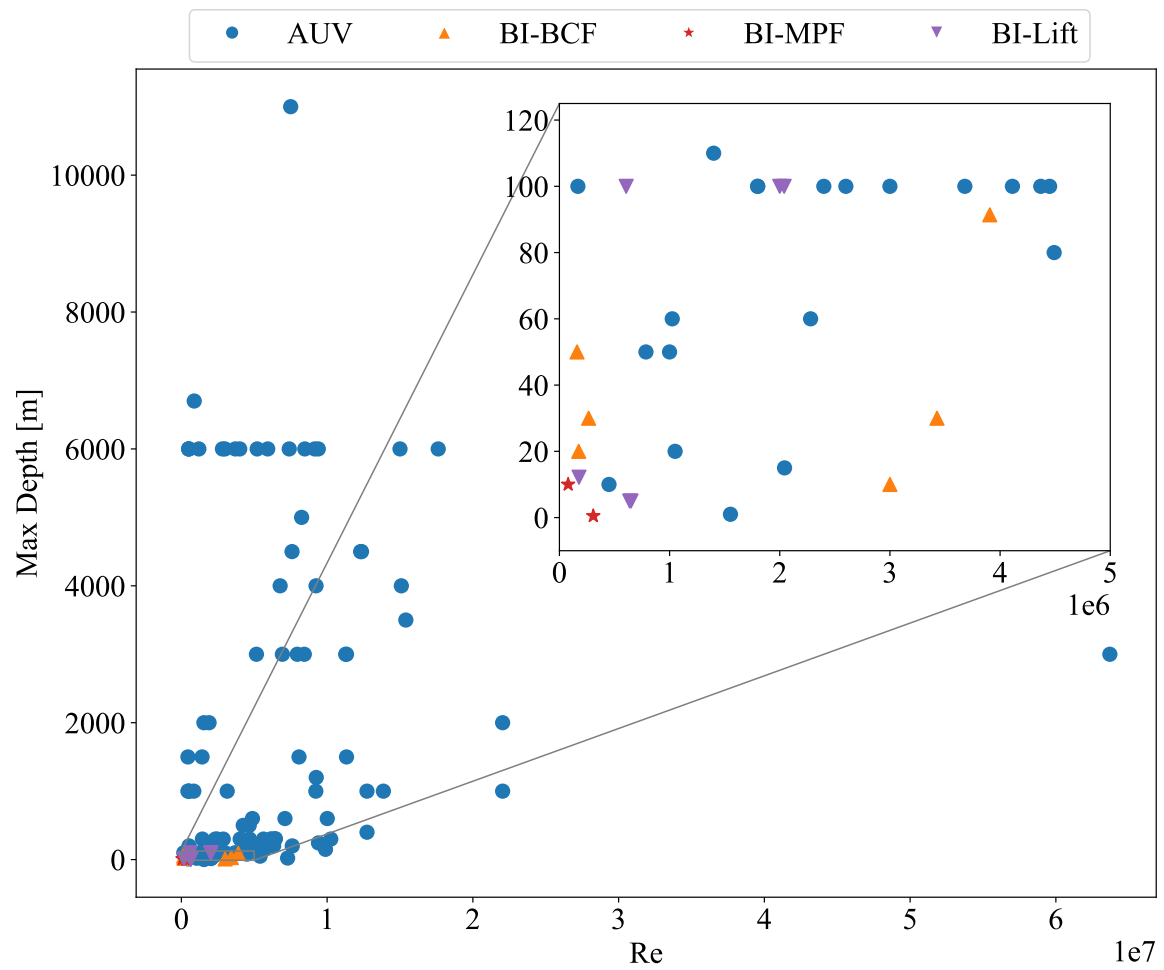


Fig. 5.8 Maximum depth versus Reynolds Number for numerous conventional AUVs (blue circle), bio-inspired BCF (orange triangle), bio-inspired MPF (red star), and bio-inspired lift based (upside down purple triangle).

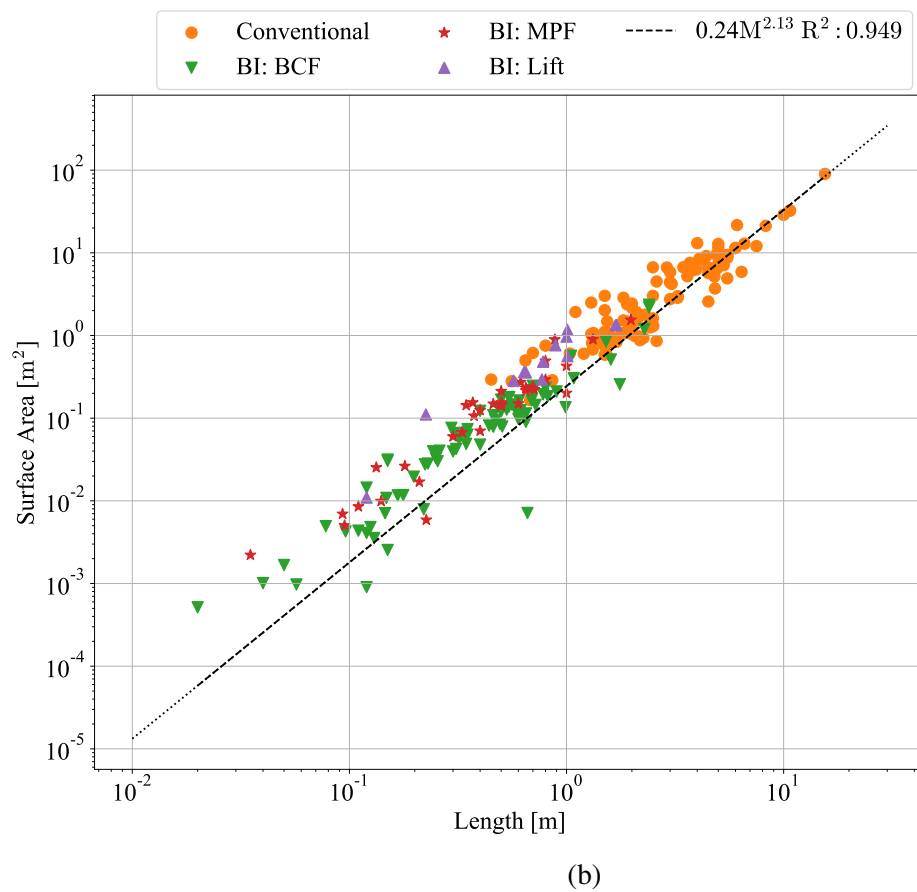
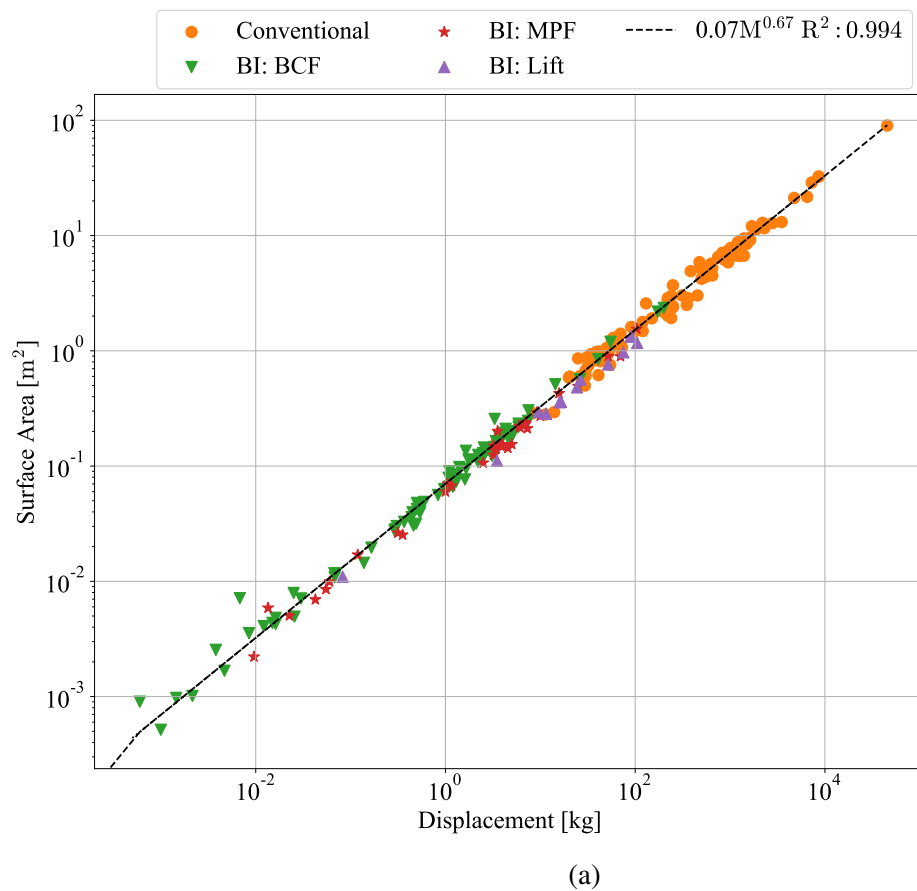
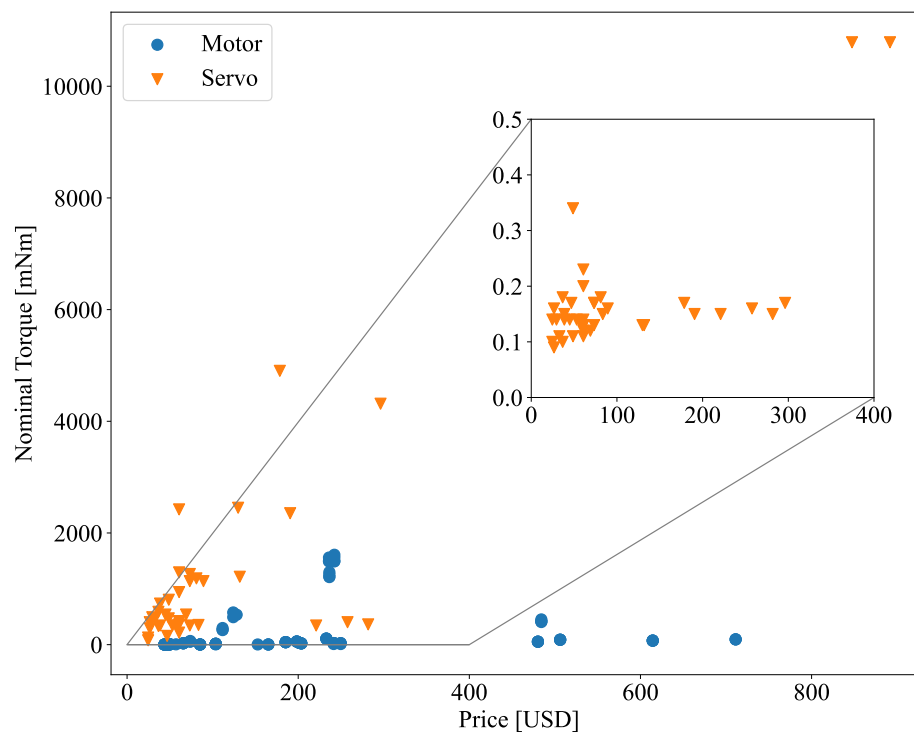


Fig. 5.9 Surface area versus (a) displacement and (b) length for all conventional and bio-inspired AUVs.

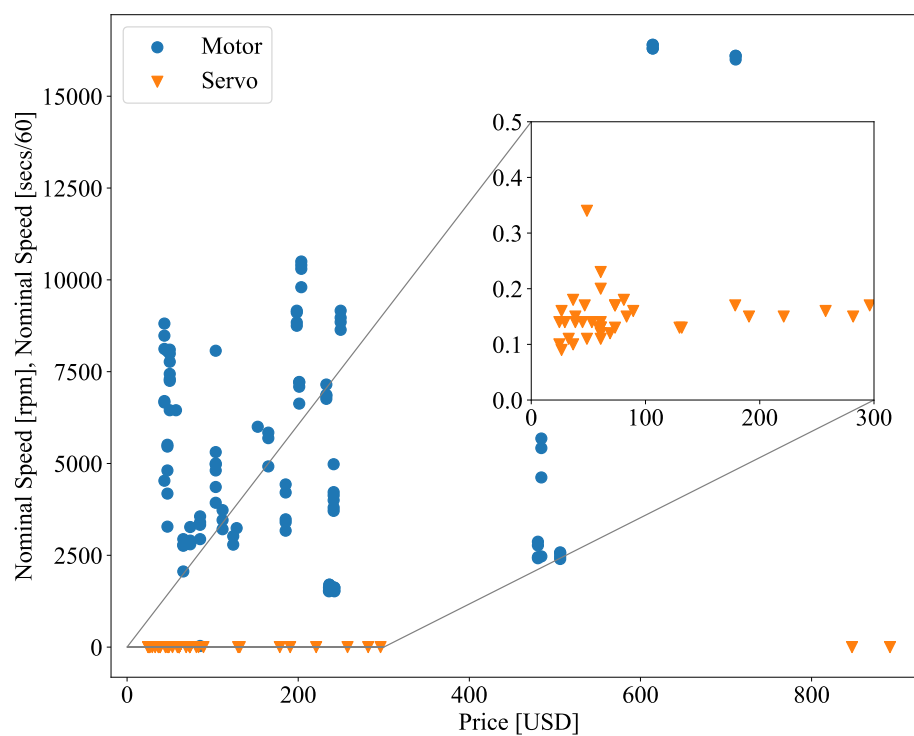
Table 5.3 Specifics of common AUV materials adapted from [141]. GFRP is Glass Fiber Reinforced Polymer and CFRP is Carbon Fiber Reinforced Polymer. Costs are provided in units of USD. Costs are given for raw material and alloys of metals may have different prices.

Material	Yield Strength [MPa]	Ultimate Strength [MPa]	Modulus [GPa]	Density [kg/m ³]	Cost [\$/kg]
Aluminium 6082	240	280	70	2700	4.20
Aluminium HDA 89	420	500	70	2800	4.20
Stainless steel 316	208	540	203	7960	2.25
Stainless steel 431	739	880	210	7830	2.25
Titanium IMI 115	200	290	110	4510	22.00
Titanium IMI 318	830	900	125	4420	22.00
GFRP pultruded section (tensile)	—	290	18	1800	100.00
GFRP pultruded section (flexural)	—	110	14	1800	100.00
CFRP Unidirectional (0°)	—	840	190	1550	200.00
CFRP Unidirectional (90°)	—	42	6.9	1550	200.00
Foam Material	Max. Operating Depth [m]	Density [kg/m ³]			
CRP Marine polyurethane foam (intermittent immersion)	200	0.20			
CRP Marine polyurethane foam (long term immersion)	200	0.33			
CRP Marine co-polymer foam (long term immersion)	500	0.38			
CRP Marine co-polymer foam (intermittent immersion)	800	0.38			
Balmoral BF/SE/60 Syntactic composite foam	300 – 600	0.32 – 0.38			
Balmoral BF/SE/62 Syntactic composite foam	1200 – 1800	0.45 – 0.517			
Balmoral BF/SE/72 Syntactic composite foam	2100 – 3000	0.51 – 0.54			
Balmoral BF/SE/75 Syntactic composite foam	5400 – 6700	0.60 – 0.64			

Dollar (USD) and plotted versus torque and speed in Fig. 5.10. The motor data does not indicate a scaling relationship in the form used thus far, but shows clustering around certain price points. A majority of the motors and servos being offered are below 200 USD and high torque brushless motors and servos are ≈ 250 USD. This price point also includes faster nominal speed motors. This is beneficial since the motor is routed through a reduction gearbox before it delivers the power to the shaft which is connected to the prop.



(a)



(b)

Fig. 5.10 Motor price versus (a) nominal torque and (b) nominal speed for Maxon motors from Maxon Inc. [44].

5.6 Concluding Remarks

Various scaling relationships for power consumption, kinematic performance, number of linkages, and cost are given in this chapter. A new dimensionless number was derived, the Propulsion number, to connect the kinematic scaling of natural and bio-inspired swimmers to conventional AUVs. An analysis of the kinematic and COT performance for all platforms was performed to determine a relative performance difference in these metrics. Conventional AUVs exceed other AUV platforms in terms of power consumption and kinematic performance.

Besides kinematic and COT scaling, the scaling of cost was quantified for the first time. The argument was made that all other things being equal, the overall cost of an AUV is mainly associated with the structural materials for a specific displacement or length. There is a small difference between different types of AUVs at the same displacement or length in terms of surface area which roughly corresponds to the materials used. It is important to note that different platforms do scale at different rates when scaling their length and displacement, as shown in the next chapter. In general, the amount of materials needed to construct an AUV scales more with length than with displacement.

Chapter 6

An Algorithm to Bring It All Together

6.1 Introduction

The scaling laws derived in the previous chapters are used to give a relative comparison between biological swimmers, bio-inspired vehicles, and conventional AUVs. This chapter gives a framework of how they can be utilized to design an AUV from the application requirements to the actual vehicle. Figure 6.1 shows the general framework for the design of an AUV. Each box in the framework has an associated sub diagram that will be explained in the following sections.

A case study of optimally performing bio-inspired and conventional AUVs is presented alongside the explanation. From Fig. 5.4, the three platforms chosen are the Cyberfish developed by Morawski *et al.* [19] at the Polish Naval Academy, Madeline or Rhex developed by Long *et al.* [67] at the Vassar College Interdisciplinary Robotics Research Laboratory, and the HUGIN 4500 developed by Kongsberg Maritime [15]. These platforms were chosen due to the availability of data and all three platforms performed the best in both kinematic and energetics performance. Furthermore, the Madeline and Cyberfish perform very similarly to each other both in kinematics and energetics as shown in Fig. 5.4 of Section 5.3.

6.2 Application Requirements

Application requirements define what type of vehicle will need to be designed. For instance, if a survey is needed of a cave, the vehicle might emphasize maneuvering in tight quarters. This could be done with a conventional AUV using multiple thrusters in a configuration like the BlueROV tethered underwater vehicle [262], but by using a bio-inspired vehicle it could be accomplished using the existing locomotion mode instead of adding thrusters. Another aspect

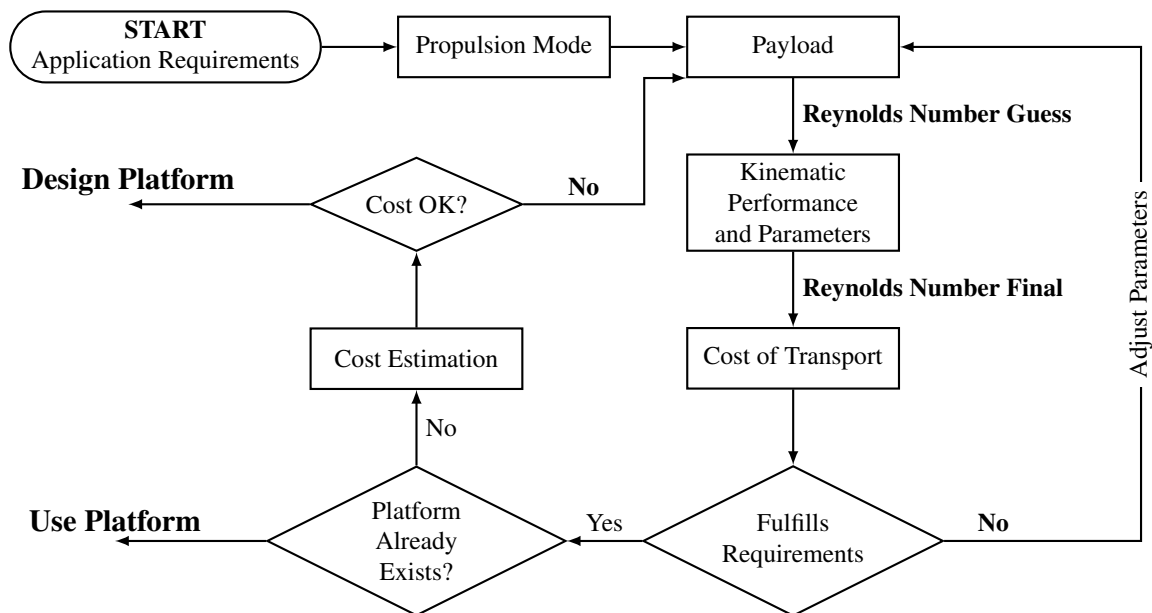


Fig. 6.1 Schematic of the overall diagram to design an Autonomous Underwater Vehicle. Each block is comprised of sub block diagrams.

that should be considered is the most efficient locomotion mode for a particular application. As pointed out by Palmisano *et al.* [163], there is a velocity at which all fish, even BCF swimmers, change locomotion modes from MPF to BCF. This point also corresponds to the lowest cost of transport and highest swimming efficiency. This means that for lower velocity applications, it may be preferable to use one locomotion mode over another.

Application requirements also define the type of payload needed for the platform. If the mission is to survey a water column, the payload will require the sensors and electronics that support that mission. The payload further determines the size that the AUV needs to be in order to accommodate everything needed for a specific application. Tables A.13 through A.17, in Appendix A, give the typical applications for AUVs that have a corresponding Cost of Transport associated with it in Tables A.3 through A.11. These can be used to determine a general size and platform that can be used for specific applications. In this chapter, a case study is presented about ocean surveying at a depth near ocean trenches.

6.3 Propulsion Modes

As discussed in the Section 6.2, the specific application or mission may benefit from a certain propulsion mode. Figure 6.2 gives an overview of the different choices of propulsion modes. The choice of propulsion mode determines the actuation scheme. Most conventional

AUVs utilize a single thrust at the aft end of the vehicle that is run by a single motor and gearbox [162]. A glider uses a buoyancy engine that takes on water and moves a battery in order to make the vehicle negatively or positively buoyant to move up and down the water column [162]. For bio-inspired propulsion, control schemes used for different types of bio-mimetic underwater propulsion is given by Low [263]. The general rule for this type of propulsion requires a series of linkages and multiple actuators. The control scheme needed for these propulsion modes utilizes a central pattern generator to create the propulsive wave or flapping needed to propel the vehicle. This means that a more complicated control scheme should be factored into the design of the AUV.

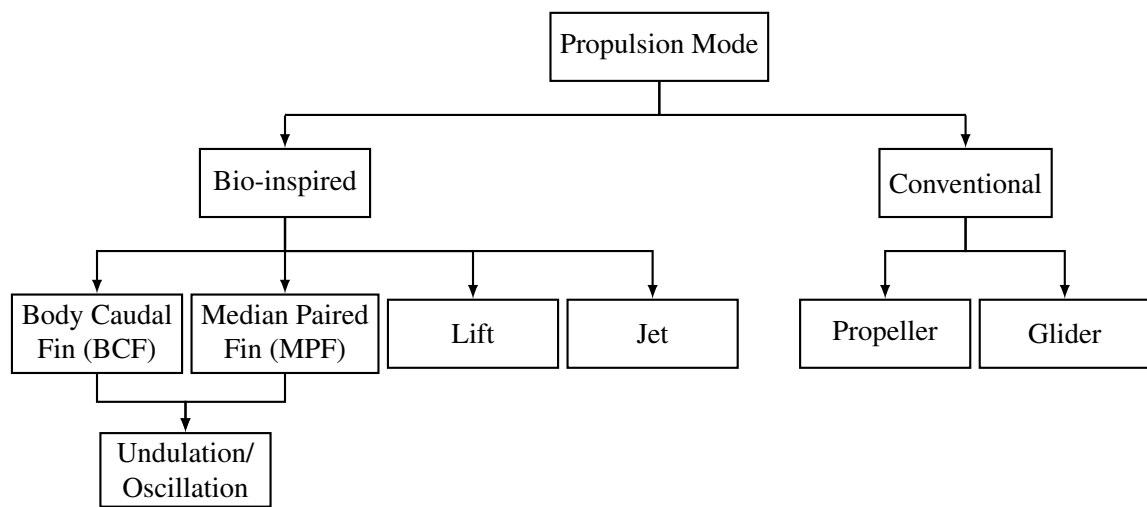


Fig. 6.2 Schematic of Propulsion modes.

In terms of the case study, the propulsion mode for the Cyberfish is bio-inspired, BCF, and undulation. Madeline has a bio-inspired and lift-based propulsion system. Lastly, the Hugin 4500 is conventional and propeller-driven.

6.4 Payload

The payload of an AUV is highly dependent on the type of mission, actuators, and hull materials. The overall volume needed internally and overall weight of the AUV is a result of the payload required for a mission. Because not all platforms are large enough to accommodate the payload, the methodology discussed in Chapter 3 was developed. This methodology allows for the scaling of any platform, conventional and bio-inspired, and the estimation of the surface area, volume, and corresponding length. The length is critical in this case because it determines kinematic parameters and the Reynolds number for the COT estimation.

Figure 6.3 gives the overall framework for determining the payload. The scientific payload includes the sensors and sampling equipment required for a mission. This could range from a mapping sensor, to a Doppler velocimeter, and the associated payload controller and electronics as described in Henthorn *et al.* [135]. The control systems for the AUV itself, separate from the payload electronics, include the communication and drive systems. This is the payload that is needed to actuate the thrusters and controls the heading of the AUV during a mission. I consider the hull a part of the payload in this instance because the materials used contribute to the overall weight and depth of the vehicle. The depth needed for a mission will inform about the material necessary to withstand hydrodynamic pressure, as discussed in Section 5.5, and the density of this material will determine the overall dry weight of the vehicle and the amount of buoyancy foam needed to be used.

Actuators are an essential part of an AUV and determine what type of payload is carried. AUV's run off of battery power, therefore, if an actuator requires AC power, an inverter is necessary, which adds another payload. Furthermore, motors are internal to the AUV and coupled to the propeller, thus taking up internal space to the AUV. Lastly, buoyancy is required in order to make an AUV have a certain orientation while submerged, generally neutrally buoyant. According to Section 5.5, this is achieved by using buoyancy foam which will occupy a certain amount of space within the AUV as well as depend on overall payload.

For the case study, the payload is assumed to be the same for scientific payload, control, and hull. The actuation and buoyancy system is different because the buoyancy foam will be different for each platform. The hull is assumed to be made of rigid materials and the actuators utilize DC voltage. The Hugin 4500 utilizes a motor and gearbox for actuation while the Madeline platform uses 4 motors and 4 gearboxes for each flipper. The Cyberfish is powered by 4 servo motors and 3 linkages.

To normalize designs for this analysis, it is assumed that the volume of each design matches the MBARI Dorado (MBARI) AUV, Fig. 5.7 in Section 5.5, in order to fit all equipment necessary for a mission and that the width and height should be at a minimum of 0.533 meters to accommodate the larger sensors and electronics. Table 6.1 gives the images of the three designs and their original and scaled dimensions. The methodology and procedure presented in Chapter 3, Section B.0.2 is used to determine the surface area and volume of the vehicle at lengths ranging from 0.1 to 100 m. The volume is matched to the MBARI AUV and the corresponding dimensions for the vehicle are checked to ensure the minimum dimensions are met. If this is not the case then the most limiting dimension, for example the height on the Madeline platform, is scaled appropriately. This allows for the minimum length which will be used for the remainder of this analysis.

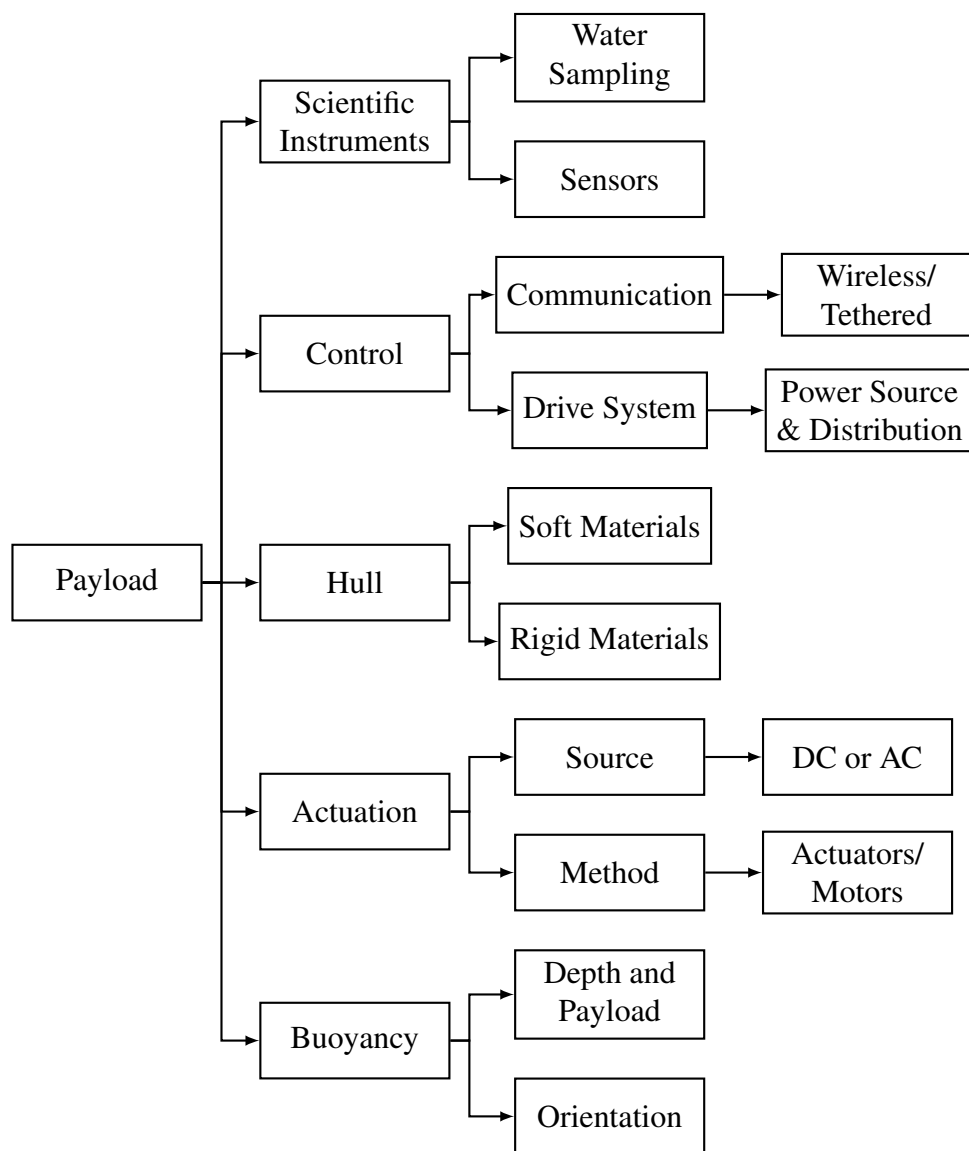





Fig. 6.3 Schematic of payload for an AUV.

Table 6.1 Published Images of designs used in cost scaling analysis. Numbers in parenthesis is the scaled value by the most limiting dimension.

Name	Affiliation	Length [m]	Width [m]	Height [m]	Displacement [kg]	Image
Cyberfish [19]	Polish Naval Academy	0.69 (3.36)	0.22 (1.07)	0.23 (1.12)	3.4 (16.6)	
Madeline / Rhex [67]	Vassar College	0.78 (3.66)	0.44 (1.83)	0.13 (0.54)	24.4 (114.5)	
Hugin 4500 [15]	Krongsberg Maritime	6.00 (3.77)	1.00 (0.63)	1.00 (0.63)	1900 (1192)	

6.5 Kinematic Performance and Parameters

The determination of kinematic parameters is critical to the design of an AUV. This gives insight into how fast a particular vehicle swims and how long a mission will take. In Section 5.3, the kinematic scaling of bio-inspired robotic platforms is given and in Section 5.4, the kinematic scaling for conventional AUVs is derived. Table 6.2 gives a summary of all the scaling relationships given in this research.

Table 6.2 Summary of Kinematic Scaling for natural and artificial swimmers. All equations are of the form ax^b where x is the dependent variable.

Swimmer	x	Re	a	b	R^2
Biological[4]	Sw	$< 6 \cdot 10^3$	0.03	1.31	0.95
Biological[4]	Sw	$\geq 6 \cdot 10^3$	0.40	1.02	0.99
Bio-inspired BCF	Sw	$\geq 6 \cdot 10^3$	5.14	0.76	0.34
Bio-inspired MPF	Sw	$\geq 6 \cdot 10^3$	10.24	0.73	0.41
Conventional	Jw	$\geq 5 \cdot 10^5$	1.40	0.96	0.84

The estimation procedure for a biological and bio-inspired platform is to rearrange the Swim number for the parameter of interest. In general, the amplitude is fixed to be 0.2 body lengths peak-to-peak for BCF swimmers. The length was determined using the size of the payload as discussed in the previous section, Section 6.4. Important in this is that the Swim number is coupled with the Reynolds number. This means that the desired vehicle speed is included in the Swim number. The missing parameter is the speed at which the propulsor

needs to actuate for the given velocity. The desired speed is given by the Swim number in the form of angular velocity, ω of the actuators by the following equation

$$\omega = \frac{S_w v}{AL}, \quad (6.1)$$

where S_w is the Swim number, v is the kinematic viscosity of the fluid, A is the amplitude of the propulsor, and L is the length of the animal or vehicle. A similar procedure can be performed for conventional AUVs using the Propulsion number derived in Section 5.4. Again, the Reynolds number gives the desired velocity of the vehicle, the diameter of the thruster is fixed and the length is determined in the previous section. The angular speed for the thruster is then given by:

$$\omega = \frac{J_w v}{DL}, \quad (6.2)$$

where J_w is the Propulsion number, (5.3) in Section 5.4, v is the kinematic viscosity of the fluid, D is the diameter of the thruster, and L is the length of the vehicle. Here the length and speed can be adjusted to suit the requirements for the actuation scheme and a final Reynolds number is found.

The kinematic performance and parameters process also serves two other purposes: finding similar platforms and quantifying kinematic performance. In the instance where there are multiple platforms that meet the propulsion mode and payload requirements, the kinematic performance scaling line gives an indication about the merits of certain designs over others. In this case, the best performing design should be chosen to go forward in the algorithm, but multiple designs might perform similarly. The second function of quantifying kinematic performance is to see where a design measures up to other designs. If a particular design that meets all previous requirements does not meet the minimal scaling line, then that design is not as well optimized as it could be.

The parameters for the scaled case study designs are given in table 6.3 shows the values of the kinematic parameters. Here, the velocity is set to a optimum velocity, U_{opt} , of 1.54 to match that of the MBARI AUV, and the kinematic viscosity, ν , is set to $1 \cdot 10^{-6}$. Amplitude is given as a scaled to the body length and also provided in table 6.3. There was not enough data to derive a kinematic scaling factor for the lift-based AUVs, therefore, the scaling factor for MPF vehicles is used instead as these vehicles are the closest to lift-based platforms.

The analysis describes an interesting story. The angular velocity, ω , for the bio-inspired platform is up to 2 orders of magnitude more than the conventional case. In the case of the

Table 6.3 Summary of the kinematic scaling parameters for the case study designs.

Swimmer	Length [m]	Re	A / D [m]	Sw	Jw	ω
Cyberfish	3.36	$5.17 \cdot 10^6$	0.27	$7.92 \cdot 10^7$		87.30
Madeline	3.66	$5.64 \cdot 10^6$	0.10	$7.31 \cdot 10^7$		200.00
Hugin 4500	3.77	$5.81 \cdot 10^6$	0.41		$7.82 \cdot 10^6$	5.06

servo driven Cyberfish, this may not be achievable with servo motors as shown in Fig. 5.10b, but a custom motor and gearbox could be used as a substitute. This means that for the same speed, the two bio-inspired platforms would need to actuate much faster than the conventional AUV, meaning that the Hugin 4500 has a better kinematic performance.

6.6 Cost of Transport

The COT of a natural and artificial swimmer gives an indication of how long it will be able to perform a specific mission. The COT is based on the kinematic performance, Reynolds number, and mass of the swimmer which is why it cannot be estimated prior to the previous blocks. COT is a complex metric and varies greatly from platform to platform, but the simple equation given by Phillips *et al.* [3] and discussed in Section 2.3 gives a way to compare different platforms to each other relatively easily. In order to obtain a real estimate of the cost of transport, complex fluid structure interactions in 3D must be performed for the specific AUV design of interest. The scaling of AUV platforms derived in Section 2.3.3 can also be used to determine the COT for existing platforms. Table 6.4 gives a summary of these scaling relationships with respect to vehicle weight and Reynolds number.

When deciding between BCF bio-inspired locomotion modes, the scaling laws derived in Section 4.10 can be used as a stand-in for the COT. Assuming that the hotel power will be the same for each locomotion mode, the propulsion power will give an indication of the most effective performance at different Reynolds numbers. Table 4.13 gives a summary of the power scaling for all four BCF locomotion types and it can be seen that in most cases, carangiform locomotion is the most efficient in terms of power scaling.

As with the Kinematic performance scaling in the previous section, the other function of the COT estimation is to determine the most suitable platform. This is because multiple platforms can satisfy all the previous blocks in the framework. This block also determines the relative optimization of a particular design. The scaling line represents the base line for performance and if a platform meets or exceeds this line it is considered a successful design, otherwise, cues can be taken from platforms that do exceed this line.

Table 6.4 Summary of COT Scaling for natural and artificial swimmers. All equations are of the form ax^b where x is the dependent variable. COT is in units of [J].

Swimmer	x	a	b	R ²
Bio Paddling[8]	Mass [kg]	3.97	1.08	0.77
Bio Undulation[8]	Mass [kg]	0.63	1.20	0.96
Bio Aquaflaying[8]	Mass [kg]	4.42	0.91	0.96
Bio Hind Feet[8]	Mass [kg]	14.42	0.64	0.96
Bio-inspired	Mass [kg]	53.85	1.1	0.705
Conventional	Mass [kg]	0.03	1.60	0.96
Bio Paddling[8]	Re	$1 \cdot 10^{-13}$	3.07	0.99
Bio Undulation[8]	Re	$1 \cdot 10^{-13}$	2.37	0.52
Bio Aquaflaying[8]	Re	$1 \cdot 10^{-9}$	1.70	0.97
Bio Hind Feet[8]	Re	$1 \cdot 10^{-9}$	1.76	0.56
Bio-inspired	Re	$1 \cdot 10^{-8}$	1.71	0.40
Conventional	Re	$1 \cdot 10^{-13}$	2.41	0.83

Applying the COT scaling laws to the case study design is given in table 6.5. In the case of the lift-based Madeline platform, the BCF carangiform scaling law is used as there is not enough data to properly correlate the lift-based data. The lift-based data is clustered with the bio-inspired carangiform data so the correlation should be similar. The analysis shows that the Hugin 4500 requires substantially less energy at optimum velocity. From the scaling law exponents, the power consumption from the bio-inspired vehicles should be closer to conventional AUVs at much higher displacements and Reynolds numbers. As discussed in the previous section, Section 6.5, this higher energy consumption may be a result of the poor kinematic performance of the two bio-inspired platforms.

Table 6.5 Summary of the COT scaling parameters for the case study designs.

Swimmer	Length [m]	Re	Mass [kg]	COT _{opt} [J]
Cyberfish	3.36	$5.17 \cdot 10^6$	334.32	47960.41
Madeline	3.66	$5.64 \cdot 10^6$	438.06	64564.12
Hugin 4500	3.77	$5.81 \cdot 10^6$	481.033	71.59

6.7 Cost Estimation

Cost estimation is outlined in Section 5.5 and is dependent on each design geometry and materials so there are no scaling laws but there are some guidelines derived from the analysis.

In general, it is more cost effective to manufacture a smaller AUV and there is no correlation between the size of the AUV and the depth it can reach. Due to the size and dimensions of the payload for a particular application, width and height are the limiting factors. Therefore, a design that has more of a fusilform or torpedo shape is preferable to a flatfish style.

Here, it is assumed that all three designs in the case study will travel to 6000 meters depth and will carry the same payload. Taking the MBARI AUV, described in Henthorn *et al.* [135], as the basis for the type of payload the mission requires, the volume of each design is scaled to accommodate the payload as discussed in Section 6.4. This allows for a simple comparison between surface area and actuators to be the determining factor in the cost of each design relative to the other.

Assuming the use of the same material used for the AUV structure and the Balmoral BF/SE/75 Syntactic composite foam from table 5.3 in Section 5.5 as the buoyancy foam, a cost analysis of the three designs in table 6.1 can be made by applying the surface area and volume estimation methodology described in Chapter 3. Figure 6.4 shows the surface area for the three designs as they range from lengths of 0 to 100 meters with the inset plot showing the surface area given by the parameters listed in table 6.1. Surface area is useful because it gives a relative amount of material that will be needed to make the hull of the vessel. We see from Fig. 6.4 that the Cyberfish and Hugin 4500 are around the same surface area at the scaled dimensions and Madeline has a surface area of around 3 times as much meaning that it would require 3 times the amount of materials to construct. The scaling laws derived also show that the Hugin 4500 scales the most efficiently cost-wise from the lower amplitude and that the Madeline scales ≈ 2.7 times more than the Hugin platform and ≈ 2.2 times more than the Cyberfish.

To calculate the amount of buoyancy foam needed, the excess mass needs to be found. I derive here a scaling relationship between length and mass for the different types of AUVs as shown in Fig. 6.5. For this analysis, only the conventional AUV relationship is used because the bio-inspired vehicles are not built with sensors and equipment normally present in conventional AUVs. This relationship is also used to keep everything consistent with the Dorado AUV for the cost analysis. The M_{excess} is then the difference between the mass of the vehicle calculated from the volume and the mass calculated based on the scaling relationship. This is combined with equation (5.6) to determine the volume of foam needed for each design and the results are given in table 6.6.

Figure 6.4 and table 6.6 shows that the conventional Hugin 4500 would be cheaper to manufacture than both the Cyberfish and Madeline with the latter platform being almost three times as expensive relative to the other two. Another consideration in the cost and design of these platforms is the number of actuators needed. In the case of the Hugin 4500, a single

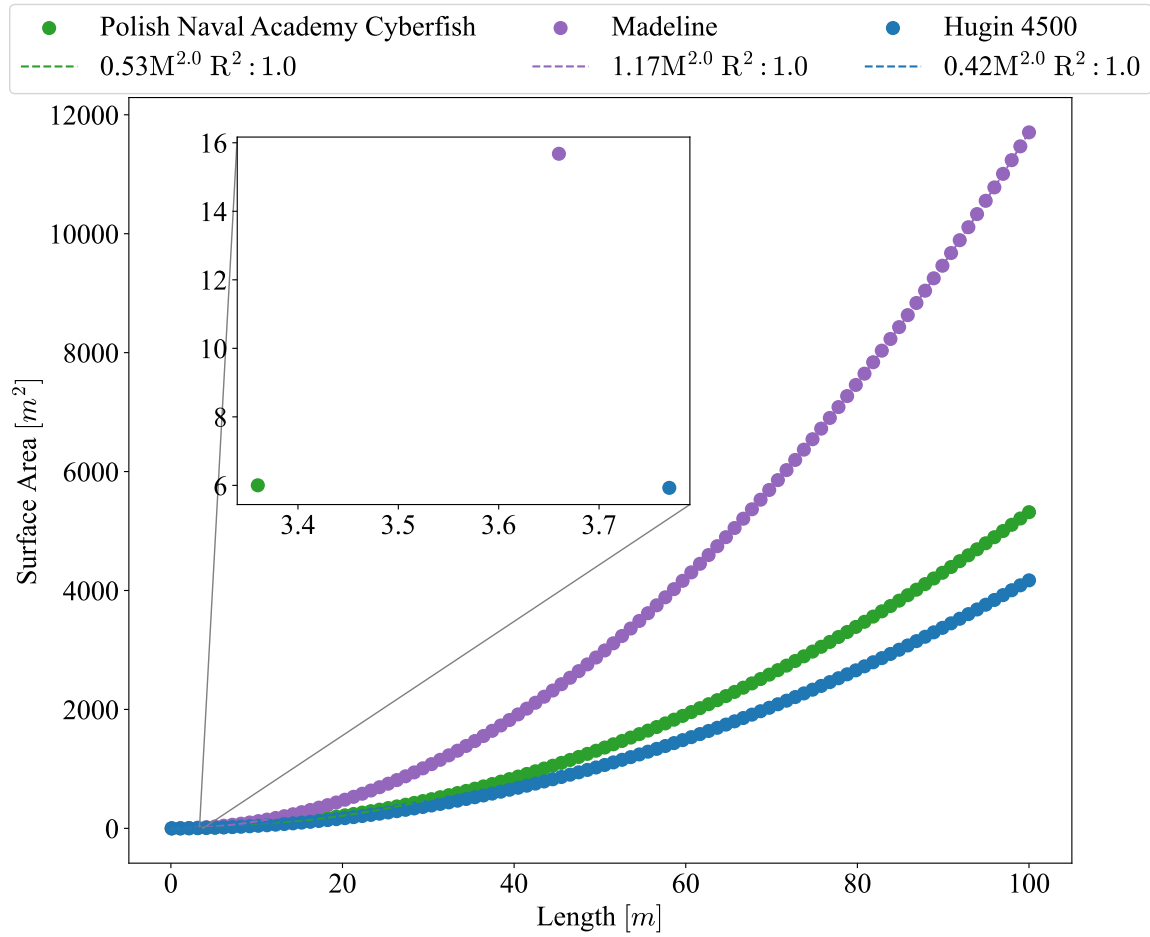


Fig. 6.4 Surface area scaling for the three designs given in table 6.1. Inset graph shows the surface area of each design at the correct scaling to match the volume of the MBARI AUV.

Table 6.6 Scaled parameters for MBARI AUV and three designs given in table 6.1.

Vehicle	Volume [m ³]	Surface Area [m ²]	Weight [kg]	M_{excess} [kg]	V_{foam} [m ³]
MBARI Dorado AUV	0.779	6.92	680.39	120.10	0.17
Cyberfish	0.784	6.00	334.32	471.28	0.65
Madeline	3.627	15.68	438.06	3286.70	4.52
Hugin 4500	0.782	5.93	481.03	322.54	0.44

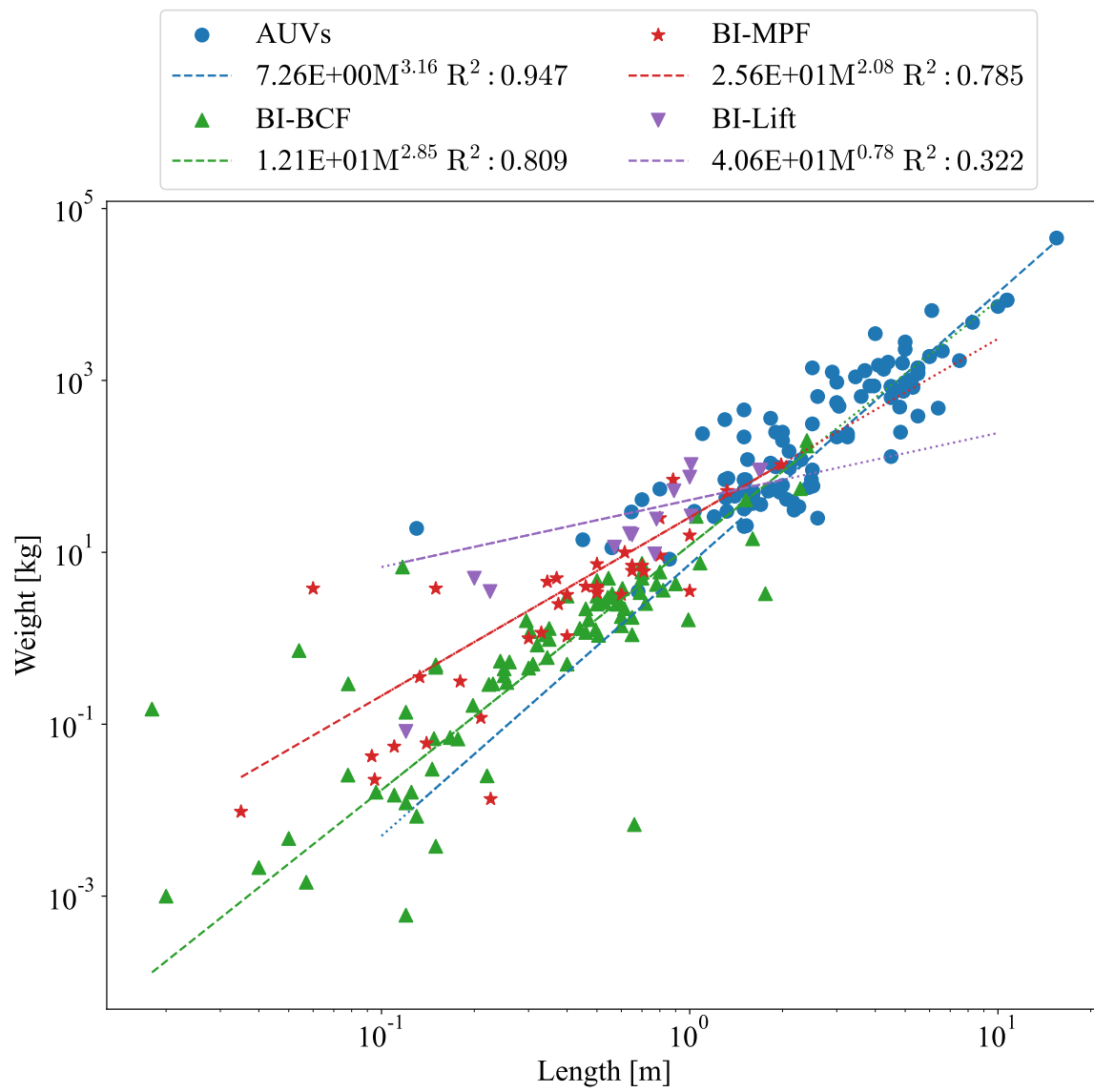


Fig. 6.5 Length versus Weight for AUVs with power scaling laws given.

motor and gear box is used to turn the screw propeller at the aft end of the vehicle. For both the Cyberfish and Madeline platforms, 4 servo motors are used for actuation. Servo motors are motors with specialized gear boxes, so this implies that the Cyberfish and Madeline platforms would be four times as much as the Hugin 4500 platform in terms of actuator price. The bio-inspired platforms also require a more complex actuator control scheme in order to operate optimally which would add to the cost of the platform. Overall, this analysis shows that conventional AUVs would cost drastically less than their bio-inspired counterparts when holding the same payload for a typical mission.

One aspect of cost that is left out of the discussion for cost scaling is the cost of designing a vehicle in terms of man-hours and prototyping. With respect to man-hours, a more simplistic design, including the structure and actuation scheme, is less costly. This would favor a torpedo style conventional AUV with a single propulsion thruster and heading thrusters when compared to something like a four linked carangiform swimming robot using a central pattern generator for each servo. The control scheme for the carangiform robot is much more complex and will require more design time and experimentation than the single motor of the torpedo style AUV.

6.8 Concluding Remarks

A framework and case study that utilizes all the work in this research is presented as a way to arrive at a suitable design for an AUV. The framework takes into consideration the payload, actuation scheme, kinematic performance, and COT. This is done to determine the most suitable design or if a custom design is needed for a particular application. This framework is meant to be a first step in choosing a specific platform and to determine if there is an existing platform that fits the application requirements. The case study presented in this chapter shows that conventional AUVs scale and perform better than their bio-inspired counterparts in terms of kinematic and energetic performance, as well as cost, when constrained to a specific application.

While conventional AUVs scale better than bio-based platforms in kinematic and COT, one area in which bio-inspired platforms excel over conventional AUVs is in maneuverability and near shore or bay applications that are detrimental to propeller systems. An example of this is surveying within a kelp bed in which the kelp might wrap around the propeller and cause an AUV to become immobile. Figure 2.27 shows that bio-inspired vehicles are about 3 times more maneuverable and twice as fast at turning than conventional AUVs. Besides maneuverability, bio-inspired vehicles are useful for studying biological motion, actuation, and mechanisms in a controlled way. As Fish [68] concludes, the study of bio-inspired

designs will only help to improve the capability of current AUV systems. Currently, there is a trend towards hybrid systems which combine bio-inspired mechanisms with conventional propulsion, as discussed in Section 2.7, which presents a promising avenue for future work.

One caveat to this framework is that it can be applied as a tool for comparison of different platforms. It can also be used as a first guess to design a platform that does not already exist. This is the basis for using scaling relationships over several Reynolds numbers. This framework and the scaling laws therein are not a substitute for numerical simulation and optimization of an AUV for a specific application.

Chapter 7

Conclusions and Recommendations for Future Work

7.1 Novelties and Conclusions

This work presents a departure from established thinking and seeks to answer the questions of when taking inspiration from biology is appropriate instead of applying a blanket "biology is always better" approach. The interim conclusions for each chapter have been provided at the end of each chapter of this thesis, therefore, the following will discuss the novelties and overall conclusions for this research.

7.1.1 Comparing Performance of Natural and Artificial Swimmers

The comparison of performance parameters performed in this research is an extension of the work done by Haroutunian [142] with the inclusion of bio-derived platforms. This research collected data on the design and performance of 229 biological animals, 163 bio-inspired platforms, and 109 conventional AUVs. Cost of Transport (COT) scaling laws for biological swimmers, conventional AUVs, and bio-inspired platforms are derived and compared to each other in Chapter 2 Section 2.3.

Hotel Power Model

As engineered systems hold all energy onboard and cannot replenish energy stores as biological animals, sources of energy consumption are identified as a combination of the hotel power and propulsion power. This research seeks to quantify the scaling of both hotel power and propulsion power. To that end, data was collected on the operating parameters and

prices for 110 motors, 41 servo motors, and 57 stepper motors of varying sizes, efficiencies, and output torques. Scaling laws were created relating motor and servo output power to torque, weight, and price and presented in Figs. 2.19 and 2.20.

It was found, in Chapter 2 Section 2.6, that hotel power is difficult to quantify as it is very application specific and there is no correlation between the types of actuators, the number of actuators, or the number of linkages used in any system/platform/design. There is a weak correlation in scaling with hotel power and the type of AUV vehicle in terms of displacement and length.

7.1.2 Computer Vision Estimation of Physical Parameters

In Chapter 6 Section 6.4, there needs to be a way to scale up or down a platform based on the amount of payload needed for a specific application. Furthermore, there needs to be a way to scale up smaller platforms to a sufficient size and their physical dimensions, surface area, displacement, and volume be accurate. Physical scaling is easy for torpedo shaped AUVs by using a prolate spheroid correlation, presented in Chapter 3 Section B.0.2, but is difficult for bio-inspired vehicles.

A methodology using computer vision and NACA airfoils was developed in Chapter 3 for biological swimmers and bio-inspired vehicles. This method is more accurate than existing methods, as discussed in Chapter 3 Section 3.2.2, and requires only two images of the platform. The developed computer vision method was validated for scaling platforms to lengths of 100 meters and accurately determined the volume, as described in Chapter 3 Section 3.2. Making the assumption that a platform is neutrally buoyant, displacement can be calculated at every length scale.

7.1.3 Simulation of BCF Locomotion Modes

Propulsive power for fish swimming has been extensively investigated for anguilliform and carangiform locomotion modes. Despite this, there are very few studies into how much power the swimmer exerts at self-propulsion. While these studies exist, there is no literature on fish swimming at Reynolds numbers above $5 \cdot 10^4$. Since conventional AUVs occupy Reynolds numbers up to $1 \cdot 10^7$, 2D computational fluid dynamics (CFD) simulations were carried out for 4 body caudal fin (BCF) locomotion modes from Reynolds numbers $5 \cdot 10^3$ to $4 \cdot 10^7$.

An overset mesh method was used in OpenFOAM®, which is an open-source CFD library. An overset mesh was used due to its stability and computational times when dealing with complex mesh movement. An entirely rewritten motion solver library was adapted from an

existing C++ library to accommodate the different complex mesh motions. The simulations were validated against existing literature and results are consistent with experimental and numerical CFD simulations.

Piece-wise power scaling laws were derived for the 4 BCF locomotion modes in Chapter 4 Section 4.10, and it was found that the carangiform mode is the consumes the least amount of energy for self propulsion through all Reynolds numbers. Second was the anguilliform locomotion mode followed by thunniform and then ostraciiform. Investigating chord-wise pressure coefficient plots in Figs. 4.34 and 4.35 of Section 4.10.1, it is found that the pitching tail present in the thunniform and ostraciiform modes causes a high pressure zone that increases the amount of power needed that is not present in the other two modes.

The locomotion mode is translated to the number of linkages needed for a bio-inspired robotic platform. Anguilliform requires the most amount of linkages, followed by carangiform, thunniform, and ostraciiform requiring only 1 linkage. The argument is made that ≈ 4 linkages is the optimal number of linkages to accurately capture the complex prescribed motion of carangiform swimming and that this is the most energy efficient locomotion mode out of the four modes tested, as presented in Chapter 4 Section 4.10.

7.1.4 Scaling Laws

To accurately compare platforms at different length, displacement, and Reynolds scales, this research opts to use allometric scaling correlations for kinematic parameters and power consumption. The Swim number, first introduced by Gazzola *et al.* [4], is adapted to bio-inspired platforms as a kinematic performance metric as shown in Fig. 5.3 in Section 5.3. There is no one-to-one relationship between the Swim number and conventional AUVs. For conventional AUVs, a completely new dimensionless number named the Propulsion number is derived that is functionally equivalent in Chapter 5 Section 5.4. The correlation between the Propulsion number and available data is presented in Fig. 5.5 of Section 5.4 and shows good agreement.

Kinematic Scaling

Using these two numbers, the kinematic performance of each type of AUV can be compared to other platforms in that category in Chapter 5 Section 5.3. It further allows for the comparison of different platforms in terms of ideal performance. More importantly, it allows for the estimation of kinematic parameters, actuation angular frequency (ω) in a straightforward manner. Using this metric, it is possible to determine the feasibility of a design in terms of actuation, velocity, and length before building a prototype. An incredibly crucial aspect of

determining the kinematic parameters is that, since both the Swim number and the Propeller number are functionally the same, the kinematic parameters are the same. As a result, the performance of the two types of AUVs can be compared as shown in Chapter 6 Section 6.5.

Cost of Transport Scaling

While kinematic performance is critical, energy consumption is often the limiting factor for a mission. This research derives novel COT scaling laws for several decades of Reynolds numbers and displacements for bio-inspired and conventional AUVs, shown in Fig. 5.2 of Section 5.2. When correlated over displacement rather than Reynolds numbers, the scaling laws indicate a better fit to the data. Similar to the kinematic scaling laws, these correlations allow for the estimation of COT for a platform in a specific category during the design phase, or for the relative comparison of different types of platforms as seen in the case study of Chapter 6.

Cost Scaling

Finally, cost scaling of AUVs was analyzed for the first time in Chapter 5 Section 5.5. Under the assumption that for a particular application, the payload would be the same which simplifies costs to structural materials and actuators. The argument is made that if tooling is the same, then the cost of structural materials is proportional to the surface area of the vehicle. Scaling laws for the surface area for displacement and length for all AUV categories were derived with a high correlation with the data when combining all vehicles. This is useful for designers as it provides a metric for the relative cost of different designs as shown in the case study of Chapter 6.

Besides structural materials, actuator cost is determined in terms of torque output and speed in Fig. 5.10 of Section 5.5. These relate to design and kinematic parameters given by the kinematic scaling laws. There is no apparent scaling law that can be applied to these data sets, but there are distinct clusters that arise. In general, a motor or servo that fits most application needs and designs will cost less than 250 USD. The cost of the actuation system is then highly dependent on the required torque and speed needed, as well as, the control scheme employed for the actuators. It is argued that conventional AUVs require a simpler actuator design and so will be more cost effective when considering the man-hours and money needed to develop the complex control schemes needed for bio-inspired platforms.

7.1.5 Framework for Design

The last part of this research combines all the previous sections into a framework for designing an efficient AUV based on application requirements. This novel framework utilizes the scaling laws in order to compare existing platforms to one another even if they are not part of the same category (i.e. bio-inspired versus conventional). In the case study involving two different types of bio-inspired platforms as well as a conventional platform, the Cyberfish, Madeline, and Hugin 4500 described in Table 6.5 of Section 6.7, the utility of scaling laws is explicitly demonstrated.

The developed framework, Fig. B.2 in Section 6.1, has the added benefit of providing a tool for the design of platforms that do not currently exist. It allows for the comparison of a platform in the design phase to current platforms at any length, displacement, or Reynolds scale. Physical scaling is enabled through the use of volume and the methodology developed in Chapter 3 to determine a scale based on the payload required, which is dependent on application. The use of scaling laws then allows for the comparison of this non-existing design to other designs at that scale, as well as providing a baseline for the designs performance.

7.2 Recommendations for Future Work

There is a lack of published data in terms of COT for both bio-inspired platforms and conventional AUVs. Biologists have been using this metric for a long time, but this concept is rather new to the AUV community. There has been more of a trend in this direction since 2018. To derive even better scaling laws, more data is needed for both types of platforms. Only a small amount of data was able to be found describing the frequency, velocity, and diameter or amplitude for AUVs. This was particularly true for conventional AUVs which is attributed to their more commercial nature and the protection of intellectual property. In the future, CFD simulations and experiments should be performed to increase the confidence in the Swim number and Propeller number for these platforms.

Concerning CFD simulations, the justification for using 2D models instead of 3D models was provided by Gazzola *et al.* [4]. Their findings were that the scaling of the Swim number with Reynolds numbers was the same for 2D and 3D simulations. While this is true, to obtain better data for the power scaling of the BCF locomotion modes, 3D simulations at different kinematic parameter sweeps should be performed. Similarly, optimization of the kinematic parameters for the thunniform and ostraciiform modes should also be undertaken. A parameter sweep should assist with this as well.

In terms of the surface area, volume and mass algorithm presented in Chapter 3, the current state-of-the-art in this field is the training of neural networks to quickly and accurately

determine a value from a source image. This requires a large data set of images for each platform but could potentially be trained using randomized 3D models. In the future, increasing the amount of data and training a neural network will greatly enhance the speed and accuracy of this model.

Finally, there has been some research into combining biological locomotion modes and conventional AUV thrusters into one robot. This approach provides the benefits of both worlds, where bio-inspired locomotion is used for maneuverability and thrusters are used for straight-line swimming. Early data suggests this to be a promising approach and this should be studied and appropriate scaling laws derived.

References

- [1] NOAA, “Ocean facts.” <https://oceanservice.noaa.gov/facts/>, 2022. Accessed: 2022-07-06.
- [2] W. G. E. Group, “World auv market forecast.” <https://www.westwoodenergy.com/reports/world-auv-market-forecast-2018-2022>, 2022.
- [3] A. B. Phillips, M. Haroutunian, S. Man, A. J. Murphy, S. W. Boyd, J. Blake, and G. Griffiths, “Nature in engineering for monitoring the oceans: comparison of the energetic costs of marine animals and AUVs,” in *Further Advances in Unmanned Marine Vehicles*, pp. 373–405, The Institution of Engineering and Technology, 2012.
- [4] M. Gazzola, M. Argentina, and L. Mahadevan, “Scaling macroscopic aquatic locomotion,” *Nature Physics*, vol. 10, no. 10, pp. 758–761, 2014.
- [5] M. Sfakiotakis, D. M. Lane, and J. B. C. Davies, “Review of fish swimming modes for aquatic locomotion,” *IEEE Journal of Oceanic Engineering*, vol. 24, no. 2, pp. 237–252, 1999.
- [6] C. C. Lindsey, “1 - form, function, and locomotory habits in fish,” *Fish Physiology*, vol. 7, pp. 1–100, 1978.
- [7] K. Schmidt-Nielsen, “Locomotion : Energy Cost of Swimming , Flying , and Running,” *American Association for the Advancement of Science*, vol. 177, no. 4045, pp. 222–228, 1972.
- [8] J. J. Videler and B. A. Nolet, “Costs of swimming measured at optimum speed: Scale effects, differences between swimming styles, taxonomic groups and submerged and surface swimming,” *Comparative Biochemistry and Physiology – Part A: Physiology*, vol. 97, no. 2, pp. 91–99, 1990.
- [9] G. Gabrielli and T. von Kármán, “What Price Speed?,” *Journal of the American Society for Naval Engineers*, vol. 63, no. 1, pp. 188–200, 1951.
- [10] AUVAC, “Auvac: Autonomous undersea vehicle applications center,” 2022.
- [11] P. Marty, “Alive: An autonomous light intervention vehicle,” *Scandinavian Oil-Gas Magazine*, vol. 32, 7 2004.
- [12] T. Copros and D. Scourzic, *Alister – Rapid Environment Assessment AUV (Autonomous Underwater Vehicle)*, pp. 233–238. Springer Netherlands, 2010.

- [13] H. Yoshida, T. Hyakudome, S. Ishibashi, T. Sawa, S. Tsukioka, T. Aoki, T. Tani, M. Iwata, and T. Moriga, "A compact high efficiency pefc system for underwater platforms," *ECS Transactions*, vol. 26, pp. 67–76, 5 2010.
- [14] B. H. Jun, J. Y. Park, F. Y. Lee, P. M. Lee, C. M. Lee, K. Kim, Y. K. Lim, and J. H. Oh, "Development of the auv 'isimi' and a free running test in an ocean engineering basin," *Ocean Engineering*, vol. 36, pp. 2–14, 1 2009.
- [15] Kongsberg, "Brochure - naval auv product range - the hugin and remus family," 2014.
- [16] S. Fujiwara and S. Yamaguchi, "Development of fishlike robot that imitates carangiform and subcarangiform swimming motions," *Journal of Aero Aqua Bio-mechanisms*, vol. 6, pp. 1–8, 2017.
- [17] K. H. Low, "Parametric study of modular and reconfigurable robotic fish with oscillating caudal fin mechanisms," in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, ICMA 2007*, pp. 123–128, IEEE, 8 2007.
- [18] B. P. Epps, P. V. Y. Alvarado, K. Youcef-Toumi, and A. H. Techet, "Swimming performance of a biomimetic compliant fish-like robot," *Experiments in Fluids*, vol. 47, pp. 927–939, 12 2009.
- [19] M. Morawski, M. Malec, and J. Zajac, "Development of cyberfish - polish biomimetic unmanned underwater vehicle buuv," in *Applied Mechanics and Materials*, vol. 613, pp. 76–82, Trans Tech Publications Ltd, 2014.
- [20] S. Jatsun, B. Lushnikov, E. Politov, and S. Knyazev, "Underwater floating robot-fish: A comparative analysis of the results of mathematical modelling and full-scale tests of the prototype," in *MATEC Web of Conferences*, vol. 113, 2017.
- [21] R. K. Katzschmann, A. D. Marchese, and D. Rus, *Hydraulic autonomous soft robotic fish for 3D swimming*, vol. 109, pp. 405–420. Springer Verlag, 2016.
- [22] R. K. Katzschmann, J. DelPreto, R. MacCurdy, and D. Rus, "Exploration of underwater life with an acoustically controlled soft robotic fish," *Science Robotics*, vol. 3, 3 2018.
- [23] J. Yu, C. Zhang, and L. Liu, "Design and control of a single-motor-actuated robotic fish capable of fast swimming and maneuverability," *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 1711–1719, 6 2016.
- [24] H. Herr and R. G. Dennis, "A swimming robot actuated by living muscle tissue," *Journal of NeuroEngineering and Rehabilitation*, vol. 1, 10 2004.
- [25] K. H. Low and C. W. Chong, "Parametric study of the swimming performance of a fish robot propelled by a flexible caudal fin," *Bioinspiration and Biomimetics*, vol. 5, pp. 90–95, 5 2010.
- [26] A. Inc., "Miro (marine intelligence robot)," 2018.
- [27] J. M. Kumph, *Maneuvering of a Robotic Pike*. PhD thesis, Massachussets Institute of Technology, 2000.

- [28] F. Berlinger, M. Saadat, H. Haj-Hariri, G. V. Lauder, and R. Nagpal, “Fish-like three-dimensional swimming with an autonomous, multi-fin, and biomimetic robot,” *Bioinspiration and Biomimetics*, vol. 16, 3 2021.
- [29] P. Hou, Z. Ye, and Z. Chen, “Bio-inspired robotic fish propelled by multiple artificial fins,” in *ASME 2016 Dynamic Systems and Control Conference, DSCC 2016*, vol. 1, pp. 1–8, 10 2016.
- [30] J. Zhu, C. White, D. K. Wainwright, V. D. Santo, G. V. Lauder, and H. Bart-Smith, “Tuna robotics: A high-frequency experimental platform exploring the performance space of swimming fishes,” *Science Robotics*, vol. 4, 9 2019.
- [31] J. Liang, T. Wang, and L. Wen, “Development of a two-joint robotic fish for real-world exploration,” *Journal of Field Robotics*, vol. 28, pp. 70–79, 1 2011.
- [32] L. Cen and A. Erturk, “Bio-inspired aquatic robotics by untethered piezohydroelastic actuation,” *Bioinspiration and Biomimetics*, vol. 8, 3 2013.
- [33] K. Struebig, B. Bayat, P. Eckert, A. Looijestijn, T. C. Lueth, and A. J. Ijspeert, “Design and development of the efficient anguilliform swimming robot-mar,” *Bioinspiration and Biomimetics*, vol. 15, 2020.
- [34] L. Manfredi, T. Assaf, S. Mintchev, S. Marrazza, L. Capantini, S. Orofino, L. Ascari, S. Grillner, P. Wallén, O. Ekeberg, C. Stefanini, and P. Dario, “A bioinspired autonomous swimming robot as a tool for studying goal-directed locomotion,” *Biological Cybernetics*, vol. 107, pp. 513–527, 10 2013.
- [35] E. Kelasidi, K. Y. Pettersen, P. Liljebäck, and J. T. Gravdahl, “Locomotion efficiency of underwater snake robots with thrusters,” in *SSRR 2016 - International Symposium on Safety, Security and Rescue Robotics*, pp. 174–181, Institute of Electrical and Electronics Engineers Inc., 12 2016.
- [36] A. N. A. Mazlan, *A Fully Actuated Tail Propulsion System for a Biomimetic Autonomous Underwater Vehicle*. Phd, University of Glasgow, 2015.
- [37] Z. Chen, T. I. Um, and H. Bart-Smith, “Bio-inspired robotic manta ray powered by ionic polymer-metal composite artificial muscles,” *International Journal of Smart and Nano Materials*, vol. 3, pp. 296–308, 2012.
- [38] V. A. Pham, T. T. Nguyen, B. R. Lee, and T. Q. Vo, “Dynamic analysis of a robotic fish propelled by flexible folding pectoral fins,” *Robotica*, vol. 38, pp. 699–718, 2019.
- [39] J. SHINTAKE, A. MING, and M. SHIMOJO, “Development of underwater robots using piezoelectric fiber composite : Diversification of propulsion movement by single fin structure,” in *The 5th International Conference on the Advanced Mechatronics(ICAM2010)*, pp. 124–128, 2010.
- [40] J. H. Long, T. J. Koob, K. Irving, K. Combie, V. Engel, N. Livingston, A. Lammert, and J. Schumacher, “Biomimetic evolutionary analysis: Testing the adaptive value of vertebrate tail stiffness in autonomous swimming robots,” *Journal of Experimental Biology*, vol. 209, pp. 4732–4746, 12 2006.

- [41] E. Zürich, “Naro-Nautical Robot,” 2022.
- [42] J. J. Videler, *Fish Swimming*. Springer Netherlands, 1993.
- [43] Y. L. Yu and K. J. Huang, “Scaling law of fish undulatory propulsion,” *Physics of Fluids*, vol. 33, no. 6, p. 1ENG, 2021.
- [44] Maxon, “Maxon motor.” <https://www.maxongroup.com/maxon/view/content/index>, 2021. Accessed: 2021-05-30.
- [45] I. Hitec RCD USA, “Hitec servos and servo accessories.” <https://hitecrcd.com/products/servos>, 2021. Accessed: 2021-05-30.
- [46] M. Malec, M. Morawski, and Z. Jerzy, “Fish-like swimming prototype of mobile underwater robot,” *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol. 4, pp. 25–30, 5 2010.
- [47] J. S. Palmisano, J. D. Geder, M. D. Pruessner, and R. Ramamurti, “Power and thrust comparison of bio-mimetic pectoral fins with traditional propeller-based thrusters,” *18th International Symposium on Unmanned Untethered Submersible Technology*, pp. 1–6, 2013.
- [48] X. Ye, Y. Su, and S. Guo, “A centimeter-scale autonomous robotic fish actuated by ipmc actuator,” in *2007 IEEE International Conference on Robotics and Biomimetics, ROBIO*, pp. 262–267, IEEE, 12 2007.
- [49] M. Tomie, A. Takiguchi, T. Honda, and J. Yamasaki, “Turning performance of fish-type microrobot driven by external magnetic field,” *IEEE Transactions on Magnetics*, vol. 41, pp. 4015–4017, 10 2005.
- [50] K. Hirata, T. Takimoto, and K. Tamura, “Study on turning performance of a fish robot,” in *First International Symposium on Aqua Bio-Mechanisms*, pp. 287–292, 1 2000.
- [51] Z. Wang, G. Hang, Y. Wang, J. Li, and W. Du, “Embedded sma wire actuated biomimetic fin: A module for biomimetic underwater propulsion,” *Smart Materials and Structures*, vol. 17, 4 2008.
- [52] S. McGovern, G. Alici, V. T. Truong, and G. Spinks, “Finding nemo (novel electromagnetic muscle oscillator): A polypyrrole powered robotic fish with real-time wireless speed and directional control,” *Smart Materials and Structures*, vol. 18, 2009.
- [53] C. Bal, G. O. Koca, D. Korkmaz, Z. H. Akpolat, and M. Ay, “Cpg-based autonomous swimming control for multi-tasks of a biomimetic robotic fish,” *Ocean Engineering*, vol. 189, 10 2019.
- [54] F. Xie, Z. Li, Y. Ding, Y. Zhong, and R. Du, “An experimental study on the fish body flapping patterns by using a biomimetic robot fish,” *IEEE Robotics and Automation Letters*, vol. 5, pp. 64–71, 1 2020.
- [55] Y. Hu, L. Wang, W. Zhao, Q. Wang, and L. Zhang, “Modular design and motion control of reconfigurable robotic fish,” in *Proceedings of the IEEE Conference on Decision and Control*, pp. 5156–5161, IEEE, 12 2007.

- [56] D. Shin, S. Y. Na, J. Y. Kim, and S.-J. Baek, "Fish robots for water pollution monitoring using ubiquitous sensor networks with sonar localization," in *2007 International Conference on Convergence Information Technology*, pp. 1298–1303, Institute of Electrical and Electronics Engineers (IEEE), 4 2007.
- [57] Z. Yu, Q. Dongxing, and S. Yan, "Dynamic modeling of robot fish based on experiment," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, 7 2019.
- [58] J. McColgan and E. W. McGookin, "Coordination of multiple biomimetic autonomous underwater vehicles using strategies based on the schooling behaviour of fish," *Robotics*, vol. 5, pp. 1–23, 3 2016.
- [59] X. Niu, J. Xu, Q. Ren, and Q. Wang, "Locomotion generation and motion library design for an anguilliform robotic fish," *Journal of Bionic Engineering*, vol. 10, pp. 251–264, 7 2013.
- [60] W. Chen, D. Xia, and J. Liu, "Modular design and realization of a torpedo-shape robot fish," in *Proceedings of 2008 IEEE International Conference on Mechatronics and Automation, ICMA 2008*, pp. 125–130, IEEE, 2008.
- [61] M. Conry, A. Keefe, W. Ober, M. Rufo, and D. Shane, "Bioswimmer: Enabling technology for port security," in *2013 IEEE International Conference on Technologies for Homeland Security, HST 2013*, pp. 364–368, 2013.
- [62] J. M. Anderson and N. K. Chhabra, "Maneuvering and stability performance of a robotic tuna," *Integrative and Comparative Biology*, vol. 42, pp. 118–126, 2002.
- [63] K. H. Low, C. Zhou, G. Seet, S. Bi, and Y. Cai, "Improvement and testing of a robotic manta ray (roman-iii)," in *2011 IEEE International Conference on Robotics and Biomimetics, ROBIO 2011*, pp. 1730–1735, IEEE, 12 2011.
- [64] T. Li, G. Li, Y. Liang, T. Cheng, J. Dai, X. Yang, B. Liu, Z. Zeng, Z. Huang, Y. Luo, T. Xie, and W. Yang, "Fast-moving soft electronic fish," *Science Advances*, vol. 3, pp. 1–7, 4 2017.
- [65] C. Zhou and K. H. Low, "Better endurance and load capacity: An improved design of manta ray robot (roman-ii)," *Journal of Bionic Engineering*, vol. 7, 9 2010.
- [66] Z. Wang, Y. Wang, J. Li, and G. Hang, "A micro biomimetic manta ray robot fish actuated by sma," in *2009 IEEE International Conference on Robotics and Biomimetics, ROBIO 2009*, pp. 1809–1813, IEEE, 2009.
- [67] J. H. Long, J. Schumacher, N. Livingston, and M. Kemp, "Four flippers or two? tetrapodal swimming with an aquatic robot," *Bioinspiration and Biomimetics*, vol. 1, pp. 20–29, 3 2006.
- [68] F. E. Fish, "Advantages of aquatic animals as models for bio-inspired drones over present AUV technology," *Bioinspiration and Biomimetics*, vol. 15, no. 2, 2020.

- [69] A. B. Phillips, M. Haroutunian, A. J. Murphy, S. W. Boyd, J. I. Blake, and G. Griffiths, "Understanding the power requirements of autonomous underwater systems, Part I: An analytical model for optimum swimming speeds and cost of transport," *Ocean Engineering*, vol. 133, pp. 271–279, 2017.
- [70] J. Rantung, M. T. Tran, H. Y. Jang, J. W. Lee, H. K. Kim, and S. B. Kim, "Determination of the Fish Surface Area and Volume Using Ellipsoid Approximation Method Applied for Image Processing," *Lecture Notes in Electrical Engineering*, vol. 465, pp. 334–347, 2018.
- [71] B. O'Shea, A. J. Mordue-Luntz, R. J. Fryer, C. C. Pert, and I. R. Bricknell, "Determination of the surface area of a fish," *Journal of Fish Diseases*, vol. 29, no. 7, pp. 437–440, 2006.
- [72] R. Williams and D. P. Noren, "Swimming speed, respiration rate, and estimated cost of transport in adult killer whales," *Marine Mammal Science*, vol. 25, no. 2, pp. 327–350, 2009.
- [73] A. P. Palstra and G. E. van den Thillart, "Swimming physiology of European silver eels (*Anguilla anguilla* L.): Energetic costs and effects on sexual maturation and reproduction," *Fish Physiology and Biochemistry*, vol. 36, no. 3, pp. 297–322, 2010.
- [74] J. R. Brett, "The Respiratory Metabolism and Swimming Performance of Young Sockeye Salmon," *Journal of the Fisheries Research Board of Canada*, vol. 21, no. 5, pp. 1183–1226, 1964.
- [75] V. Castro, B. Grisdale-Helland, S. J. Helland, T. Kristensen, S. M. Jørgensen, J. Helgerud, G. Claireaux, A. P. Farrell, A. Krasnov, and H. Takle, "Aerobic training stimulates growth and promotes disease resistance in Atlantic salmon (*Salmo salar*)," *Comparative Biochemistry and Physiology - A Molecular and Integrative Physiology*, vol. 160, no. 2, pp. 278–290, 2011.
- [76] H. Liu and O. Curet, "Swimming performance of a bio-inspired robotic vessel with undulating fin propulsion," *Bioinspiration and Biomimetics*, vol. 13, 7 2018.
- [77] J. Ahrens, B. Geveci, and C. Law, *ParaView: An End-User Tool for Large Data Visualization*. Elsevier, 2005.
- [78] J. Guerrero, "OpenFOAM advanced training. Moving meshes, rigid body motion, adaptive mesh refinement, and overset meshes," 2022.
- [79] G. J. Dong and X. Y. Lu, "Characteristics of flow over traveling wavy foils in a side-by-side arrangement," *Physics of Fluids*, vol. 19, no. 5, pp. 1–11, 2007.
- [80] W. L. Chan, T. Kang, Y. J. Lee, S. K. Sung, and K. J. Yoon, "Swimming study on an ostraciiform fish robot," in *ICCAS 2007 - International Conference on Control, Automation and Systems*, pp. 700–705, Institute of Control, Robotics, and Systems, 2007.
- [81] B. Chen and H. Jiang, "Swimming performance of a tensegrity robotic fish," *Soft Robotics*, vol. 6, pp. 520–531, 8 2019.

- [82] D. Chen, W. Shao, and C. Xu, "Development of a soft robotic fish with bcf propulsion using mfc smart materials," in *Chinese Control Conference, CCC*, vol. 2018-July, pp. 5358–5363, 7 2018.
- [83] Z. Chen, S. Shatara, and X. Tan, "Modeling of biomimetic robotic fish propelled by an ionic polymer-metal composite caudal fin," *IEEE/ASME Transactions on Mechatronics*, vol. 15, pp. 448–459, 6 2010.
- [84] R. J. Clapham and H. Hu, "isplash-micro: A 50mm robotic fish generating the maximum velocity of real fish," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 287–293, Institute of Electrical and Electronics Engineers Inc., 10 2014.
- [85] R. J. Clapham and H. Hu, "isplash-optimize: Optimized linear carangiform swimming motion," in *Advances in Intelligent Systems and Computing*, vol. 302, pp. 1257–1270, Springer Verlag, 2016.
- [86] H. E. Daou, T. Salumäe, L. D. Chambers, W. M. Megill, and M. Kruusmaa, "Modelling of a biologically inspired robotic fish driven by compliant parts," *Bioinspiration and Biomimetics*, vol. 9, 3 2014.
- [87] R. Ding, J. Yu, Q. Yang, M. Tan, and J. Zhang, "Cpg-based dynamics modeling and simulation for a biomimetic amphibious robot," in *2009 IEEE International Conference on Robotics and Biomimetics, ROBIO 2009*, pp. 1657–1662, 2009.
- [88] F. Gibouin, C. Raufaste, Y. Bouret, and M. Argentina, "Study of the thrust-drag balance with a swimming robotic fish," *Physics of Fluids*, vol. 30, 9 2018.
- [89] J. Guo, "Maneuvering and control of a biomimetic autonomous underwater vehicle," *Autonomous Robots*, vol. 26, pp. 241–249, 5 2009.
- [90] S. Guo, Y. Ge, L. Li, and S. Liu, "Underwater swimming micro robot using ipmc actuator," in *2006 IEEE International Conference on Mechatronics and Automation, ICMA 2006*, vol. 2006, pp. 249–254, IEEE, 6 2006.
- [91] E. Kelasidi, *Modeling , Control and Energy Efficiency of Underwater Snake Robots*. PhD thesis, Norwegian University of Science and Technology, 12 2015.
- [92] B. Kim, D. H. Kim, J. Jung, and J. O. Park, "A biomimetic undulatory tadpole robot using ionic polymer-metal composite actuators," *Smart Materials and Structures*, vol. 14, pp. 1579–1585, 12 2005.
- [93] S. H. Kim, K. Shin, S. Hashi, and K. Ishiyama, "Magnetic fish-robot based on multi-motion control of a flexible magnetic actuator," *Bioinspiration and Biomimetics*, vol. 7, 9 2012.
- [94] V. Kopman and M. Porfiri, "Design, modeling, and characterization of a miniature robotic fish for research and education in biomimetics and bioinspiration," *IEEE/ASME Transactions on Mechatronics*, vol. 18, pp. 471–483, 2013.

- [95] D. Lachat, A. Crespi, and A. J. Ijspeert, "Boxybot: A swimming and crawling fish robot controlled by a central pattern generator," in *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006, BioRob 2006*, vol. 2006, pp. 643–648, 2006.
- [96] K. Li, H. Jiang, S. Wang, and J. Yu, "A soft robotic fish with variable-stiffness decoupled mechanisms," *Journal of Bionic Engineering*, vol. 15, pp. 599–609, 7 2018.
- [97] W. Liu, X. Jia, F. Wang, and Z. Jia, "An in-pipe wireless swimming microrobot driven by giant magnetostrictive thin film," *Sensors and Actuators, A: Physical*, vol. 160, pp. 101–108, 2010.
- [98] S. F. Masoomi, S. Gutschmidt, X. Q. Chen, and M. Sellier, "The kinematics and dynamics of undulatory motion of a tuna-mimetic robot," *International Journal of Advanced Robotic Systems*, vol. 12, 7 2015.
- [99] A. Ming, S. Park, Y. Nagata, and M. Shimojo, "Development of underwater robots using piezoelectric fiber composite," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3821–3826, [IEEE], 2009.
- [100] K. A. Morgansen, B. I. Triplett, and D. J. Klein, "Geometric methods for modeling and control of free-swimming fin-actuated underwater vehicles," *IEEE Transactions on Robotics*, vol. 23, pp. 1184–1199, 2007.
- [101] D. Q. Nguyen and V. A. Ho, "Kinematic evaluation of a series of soft actuators in designing an eel-inspired robot," in *Proceedings of the 2020 IEEE/SICE International Symposium on System Integration*, pp. 1288–1293, 1 2020.
- [102] Q. S. Nguyen, S. Heo, H. C. Park, N. S. Goo, T. Kang, K. J. Yoon, and S. S. Lee, "A fish robot driven by piezoceramic actuators and a miniaturized power supply," *International Journal of Control, Automation and Systems*, vol. 7, pp. 267–272, 4 2009.
- [103] E. Papadopoulos, E. Apostolopoulos, and P. Tsigkourakos, "Design, control, and experimental performance of a teleoperated robotic fish," in *17th Mediterranean Conference on Control & Automation*, pp. 766–771, IEEE, 7 2009.
- [104] C. Rossi, W. Coral, J. Colorado, and A. Barrientos, "A motor-less and gear-less biomimetic robotic fish design," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3646–3651, IEEE, 5 2011.
- [105] M. Shibata and N. Sakagami, "Fabrication of a fish-like underwater robot with flexible plastic film body," *Advanced Robotics*, vol. 29, pp. 103–113, 1 2015.
- [106] A. Suleman and C. Crawford, "Design and testing of a biomimetic tuna using shape memory alloy induced propulsion," *Computers and Structures*, vol. 86, pp. 491–499, 2 2008.
- [107] Z. Wang, G. Hang, J. Li, Y. Wang, and K. Xiao, "A micro-robot fish with embedded sma wire actuated flexible biomimetic fin," *Sensors and Actuators, A: Physical*, vol. 144, pp. 354–360, 6 2008.

- [108] L. Wen, J. Liang, Q. Shen, L. Bao, and Q. Zhang, "Hydrodynamic performance of an undulatory robot: Functional roles of the body and caudal fin locomotion," *International Journal of Advanced Robotic Systems*, vol. 10, 1 2013.
- [109] Q. Yan, Z. Han, S. Wu Zhang, and J. Yang, "Parametric research of experiments on a carangiform robotic fish," *Journal of Bionic Engineering*, vol. 5, pp. 95–101, 2008.
- [110] Y. Zhong, Z. Li, and R. Du, "Robot fish with two-dof pectoral fins and a wire-driven caudal fin," *Advanced Robotics*, vol. 32, pp. 25–36, 11 2018.
- [111] P. E. Sitorus, Y. Y. Nazaruiddin, E. Leksono, and A. Budiyo, "Design and implementation of paired pectoral fins locomotion of labriform fish applied to a fish robot," *Journal of Bionic Engineering*, vol. 6, pp. 37–45, 3 2009.
- [112] D. G. Simons, M. M. Bergers, S. Henrion, J. I. Hulzenga, R. W. Jutte, W. M. Pas, M. V. Schravendijk, T. G. Vercruyssen, and A. P. Wilken, "A highly versatile autonomous underwater vehicle with biomechanical propulsion," in *OCEANS '09 IEEE Bremen: Balancing Technology with Future Needs*, pp. 1–6, 2009.
- [113] M. J. Hong and M. R. Arshad, "Modeling and motion control of a riverine autonomous surface vehicle (asv) with differential thrust," *Jurnal Teknologi*, vol. 74, pp. 137–143, 6 2015.
- [114] X. Yicun, Z. Guanghua, B. Shusheng, and G. Jun, "Initial development of a flapping propelled unmanned underwater vehicle (uuv)," in *2007 IEEE International Conference on Robotics and Biomimetics, ROBIO*, pp. 524–529, IEEE, 12 2007.
- [115] Y. Cai, S. Bi, K. H. Low, L. Zhang, and G. Zong, "Posture analysis and application of a bionic pectoral foil," in *2011 IEEE International Conference on Robotics and Biomimetics, ROBIO 2011*, pp. 1783–1788, IEEE, 12 2011.
- [116] G. Jun, B. Shusheng, X. Yicun, and L. Cong, "Development and design of a robotic manta ray featuring flexible pectoral fins," in *2007 IEEE International Conference on Robotics and Biomimetics, ROBIO*, pp. 519–523, IEEE, 12 2007.
- [117] Y. Cai, S. Bi, and L. Zheng, "Design and experiments of a robotic fish imitating cow-nosed ray," *Journal of Bionic Engineering*, vol. 7, pp. 120–126, 6 2010.
- [118] M. Wang, J. Yu, and M. Tan, "Modeling neural control of robotic fish with pectoral fins using a cpg-based network," in *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pp. 6502–6507, IEEE, 2009.
- [119] K. Takagi, M. Yamamura, Z. W. Luo, M. Onishi, S. Hirano, K. Asaka, and Y. Hayakawa, "Development of a rajiform swimming robot using ionic polymer artificial muscles," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 1861–1866, IEEE, 2006.
- [120] K. Suzumori, S. Endo, T. Kanda, N. Kato, and H. Suzuki, "A bending pneumatic rubber actuator realizing soft-bodied manta swimming robot," in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4975–4980, 2007.

- [121] A. Punning and M. Anton, "A biologically inspired ray-like underwater robot with electroactive polymer pectoral fins," *Proc. of IEEE/Int. Conf. on Mechatronics and Robotics*, vol. 2, p. 241–245, 1 2004.
- [122] W. Zhao, T. Osaka, A. Ming, and M. Shimojo, "Development of a soft underwater robot mimicking cow-nosed ray," in *2011 IEEE International Conference on Robotics and Biomimetics, ROBIO 2011*, pp. 1724–1729, 12 2011.
- [123] Z. Chen, T. I. Um, J. Zhu, and H. Bart-Smith, "Bio-inspired robotic cownose ray propelled by electroactive polymer pectoral fin," in *ASME 2011 International Mechanical Engineering Congress and Exposition, IMECE 2011*, vol. 2, pp. 817–824, 2011.
- [124] R. D. Love, A. A. Arroyo, and E. M. Schwartz, "Solar ray : An autonomous solar-powered bio-mimetic bio flapping wing underwater vehicle," *Florida Conference on Recent Advances in Robotics*, pp. 2–8, 2010.
- [125] S. C. Subramanian, T. Le, J. Olson, S. Bhat, and S. Redkar, "Design and development of an unmanned underwater vehicle (uuv) in the form of a cuttlefish," in *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*, vol. 4B-2018, pp. 1–8, 2018.
- [126] Q. Shen, Z. Olsen, T. Stalbaum, S. Trabia, J. Lee, R. Hunt, K. Kim, J. Kim, and I. K. Oh, "Basic design of a biomimetic underwater soft robot with switchable swimming modes and programmable artificial muscles," *Smart Materials and Structures*, vol. 29, 2020.
- [127] K. H. Low and A. Willy, "Biomimetic motion planning of an undulating robotic fish fin," *JVC/Journal of Vibration and Control*, vol. 12, pp. 1337–1359, 12 2006.
- [128] H. J. Kim, S. H. Song, and S. H. Ahn, "A turtle-like swimming robot using a smart soft composite (ssc) structure," *Smart Materials and Structures*, vol. 22, 1 2013.
- [129] G. H. Ang, S. Fan, Y. Jin, H. Lim, C. K. Chin, S. Chai, and N. Bose, "A comprehensive comparison of computational methods on propeller modelling of an auv," in *AUV 2018 - 2018 IEEE/OES Autonomous Underwater Vehicle Workshop, Proceedings*, 2018.
- [130] F. Azarsina and C. D. Williams, "Manoeuvring simulation of the mun explorer auv based on the empirical hydrodynamics of axi-symmetric bare hulls," *Applied Ocean Research*, vol. 32, pp. 443–453, 10 2010.
- [131] Ø. Hegrenæs, O. Hallingstad, and B. Jalving, "Comparison of mathematical models for the HUGIN 4500 AUV based on experimental data," *International Symposium on Underwater Technology, UT 2007 - International Workshop on Scientific Use of Submarine Cables and Related Technologies 2007*, no. 7491, pp. 558–567, 2007.
- [132] A. B. Phillips, S. R. Turnock, and M. Furlong, "The use of computational fluid dynamics to aid cost-effective hydrodynamic design of autonomous underwater vehicles," *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, vol. 224, pp. 239–254, 11 2010.
- [133] D. Sgarioto and C. Madden, "Full-scale manoeuvring trials for the wayamba unmanned underwater vehicle," *Underwater Technology*, vol. 32, pp. 67–79, 2014.

- [134] P. J. B. Sánchez, M. Papaelias, and F. P. G. Márquez, “Autonomous underwater vehicles: Instrumentation and measurements,” *IEEE Instrumentation and Measurement Magazine*, vol. 23, no. 2, pp. 105–114, 2020.
- [135] R. Henthorn, D. W. Caress, H. Thomas, R. McEwen, W. J. Kirkwood, C. K. Paull, and R. Keaten, “High-resolution multibeam and subbottom surveys of submarine canyons, deep-sea fan channels, and gas seeps using the MBARI mapping AUV,” *Oceans 2006*, pp. 1–6, 2006.
- [136] S. M. Mirvakili and I. W. Hunter, “Artificial Muscles: Mechanisms, Applications, and Challenges,” *Advanced Materials*, vol. 30, no. 6, pp. 1–28, 2018.
- [137] F. Nations and A. O. of the United, “Trophic Levels,” 2022.
- [138] F. J. Ballesteros, V. J. Martinez, B. Luque, L. Lacasa, E. Valor, and A. Moya, “On the thermodynamic origin of metabolic scaling,” *Scientific Reports*, vol. 8, no. 1, pp. 1–14, 2018.
- [139] A. Amory and E. Maehle, “Energy Efficiency of the Swarm-Capable Micro AUV SEMBIO,” *OCEANS 2019 - Marseille, OCEANS Marseille 2019*, vol. 2019-June, 2019.
- [140] W. J. McCroskey, “A Critical Assessment of Wind Tunnel Results for the NACA 0012 Airfoil,” *NASA technical Memorandum 100019 USAAVSCOM Technical Report 87-A-5*, 1987.
- [141] P. Stevenson and D. Graham, *Technology and Applications of Autonomous Underwater Vehicles*, ch. Advanced Materials and Their Influence on the Structural Design of AUVs. Taylor and Francis, 2003.
- [142] M. Haroutunian, *An holistic bio-inspired approach for improving the performance of Unmanned Underwater Vehicles*. PhD thesis, Newcastle University, 2014.
- [143] I. H. Abbott, *Theory of Wing Sections - Including a Summary of Airfoil Data*. Dover Publications, 1949; 1959.
- [144] OpenCFD Ltd (ESI Group), “OpenFOAM® v2012.”
- [145] J. Gray, “Studies in animal locomotion vi. the propulsive powers of the dolphin,” *Journal of Experimental Biology*, vol. 13, pp. 192–199, 1936.
- [146] H. Blasius, “Grenzschichten in Flüssigkeiten mit Kleiner Reibung,” *Zeitschrift für angewandte Mathematik und Physik*, vol. 56, no. 1, pp. 1–37, 1908.
- [147] R. Bale, M. Hao, A. P. S. Bhalla, N. Patel, and N. A. Patankar, “Gray’s paradox: A fluid mechanical perspective,” *Scientific Reports*, vol. 4, 2014.
- [148] D. S. Barrett, *Propulsive Efficiency of a Flexible Hull Underwater Vehicle*. PhD thesis, Massachusetts Institute of Technology, 5 1996.
- [149] R. N. Govardhan and J. H. Arakeri, “Fluid mechanics of aquatic locomotion at large Reynolds numbers,” *Journal of the Indian Institute of Science*, vol. 91, no. 3, pp. 329–353, 2011.

- [150] B. Liao, Z. Li, and R. Du, "Robot fish with a novel biomimetic wire-driven flapping propulsor," *Advanced Robotics*, vol. 28, pp. 339–349, 3 2014.
- [151] B. Liu, L. Hao, J. Deng, and X. Liu, "A remote operated robotic fish with temperature sensor based on ipmc actuator," in *2009 Chinese Control and Decision Conference, CCDC 2009*, pp. 5730–5734, 2009.
- [152] Y. Y. Watanabe, K. Sato, Y. Watanuki, A. Takahashi, Y. Mitani, M. Amano, K. Aoki, T. Narazaki, T. Iwata, S. Minamikawa, and N. Miyazaki, "Scaling of swim speed in breath-hold divers," *Journal of Animal Ecology*, vol. 80, no. 1, pp. 57–68, 2011.
- [153] P. B. Liao, "Water requirements of salmonids," *Progressive Fish-Culturist*, vol. 33, no. 4, pp. 209–215, 1971.
- [154] H. Dewar, J. B. Graham, and R. W. Brill, "Studies of tropical tuna swimming performance in a large water tunnel. II. Thermoregulation," *Journal of Experimental Biology*, vol. 192, pp. 33–44, 1994.
- [155] J. M. Elliott and W. Davison, "Energy Equivalents of Oxygen Consumption in Animal Energetics," *Oecologia*, vol. 19, no. 3, pp. 195–201, 1975.
- [156] J. J. Magnuson, "Locomotion By Scombrid Fishes: Hydrodynamics, Morphology, and Behavior," in *Fish Physiology*, ch. 4, pp. 239–313, Academic Press Inc., 1978.
- [157] ITTC, "Recommended procedures and guidelines - resistance test," tech. rep., International Towing Tank Committee, 2011.
- [158] S. F. Hoerner, *Fluid-Dynamic Drag*. P.O. Box 342, Brick Town, N.J. 08723: Hoerner Fluid Dynamics, 2 ed., 7 1965.
- [159] M. Coe and S. Gutschmidt, "Computer vision estimation of physical parameters and its application to power requirements of natural and artificial swimmers," *Designs*, vol. 5, no. 4, 2021.
- [160] N. A. Curtin and R. C. Woledge, "Efficiency of energy conversion during sinusoidal movement of red muscle fibres from the dogfish *Scyliorhinus canicula*," *Journal of Experimental Biology*, vol. 185, pp. 195–206, 1993.
- [161] S. Dharwada, A. Agarwal, and P. Rajagopal, "Comparison of bio-inspired flapping foil propulsion systems with rotary propulsion," *Proc. of SPIE*, vol. 10593, no. March 2018, p. 29, 2018.
- [162] G. Griffiths, *Technology and Applications of Autonomous Underwater Vehicles*. Taylor & Francis, 2003.
- [163] J. S. Palmisano, M. Pruessner, and J. D. Geder, "A comprehensive allometric analysis of bio-mimetic mpf-type uuv's," *18th International Symposium on Unmanned Untethered Submersible Technology*, 2013.
- [164] S. Prior and A. White, "Measurements and simulation of a pneumatic muscle actuator for a rehabilitation robot," *Simulation Practice and Theory*, vol. 3, pp. 81–117, sep 1995.

- [165] T. Ura and T. Obara, "Twelve hour operation of cruising type auv 'r-one robot' equipped with a closed cycle diesel engine system," in *Oceans Conference Record (IEEE)*, vol. 3, pp. 1188–1193, 1999.
- [166] E. A. Jones, K. S. Lucey, and D. J. Ellerby, "Efficiency of labriform swimming in the bluegill sunfish (*Lepomis macrochirus*)," *Journal of Experimental Biology*, vol. 210, no. 19, pp. 3422–3429, 2007.
- [167] D. J. Hartl and D. C. Lagoudas, "Aerospace applications of shape memory alloys," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, vol. 221, no. 4, pp. 535–552, 2007.
- [168] A. E. Aliev, J. Oh, M. E. Kozlov, A. A. Kuznetsov, S. Fang, A. F. Fonseca, R. Ovalle, M. D. Lima, M. H. Haque, Y. N. Gartstein, M. Zhang, A. A. Zakhidov, and R. H. Baughman, "Giant-stroke, superelastic carbon nanotube aerogel muscles," *Science*, vol. 323, no. 5921, pp. 1575–1578, 2009.
- [169] C. S. Haines, M. D. Lima, N. Li, G. M. Spinks, J. Foroughi, J. D. Madden, S. H. Kim, S. Fang, M. J. De Andrade, F. Göktepe, Ö. Göktepe, S. M. Mirvakili, S. Naficy, X. Lepró, J. Oh, M. E. Kozlov, S. J. Kim, X. Xu, B. J. Swedlove, G. G. Wallace, and R. H. Baughman, "Artificial muscles from fishing line and sewing thread," *Science*, vol. 343, no. 6173, pp. 868–872, 2014.
- [170] K. J. Kim and M. Shahinpoor, "A novel method of manufacturing three-dimensional ionic polymer-metal composites (IPMCs) biomimetic sensors, actuators and artificial muscles," *Polymer*, vol. 43, no. 3, pp. 797–802, 2001.
- [171] J. D. Nam, H. R. Choi, J. C. Koo, Y. K. Lee, and K. J. Kim, *Dielectric Elastomers for Artificial Muscles*, pp. 37–48. London: Springer London, 2007.
- [172] J. Ding, L. Liu, G. M. Spinks, D. Zhou, G. G. Wallace, and J. Gillespie, "High performance conducting polymer actuators utilising a tubular geometry and helical wire interconnects," *Synthetic Metals*, vol. 138, no. 3, pp. 391–398, 2003.
- [173] K. Uchino, *Advanced piezoelectric materials: Science and technology*. Woodhead Publishing, 2017.
- [174] M. E. Furlong, S. D. McPhail, and P. Stevenson, "A concept design for an ultra-long-range survey class AUV," *OCEANS 2007 - Europe*, 2007.
- [175] M. S. Triantafyllou and G. S. Triantafyllou, "An efficient swimming machine," *Scientific American*, vol. 272, no. 3, pp. 40–48, 1995.
- [176] D. S. Barrett, M. S. Triantafyllou, D. K. Yue, M. A. Grosenbaugh, and M. J. Wolfgang, "Drag reduction in fish-like locomotion," *Journal of Fluid Mechanics*, vol. 392, pp. 183–212, 8 1999.
- [177] K. Tiwari, X. Xiao, A. Malik, and N. Y. Chong, "A unified framework for operational range estimation of mobile robots operating on a single discharge to avoid complete immobilization," *Mechatronics*, vol. 57, no. October 2018, pp. 173–187, 2019.

- [178] M. Trancossi, “What price of speed? A critical revision through constructal optimization of transport modes,” *International Journal of Energy and Environmental Engineering*, vol. 7, no. 4, pp. 425–448, 2016.
- [179] A. Bejan, J. D. Charles, S. Lorente, and E. H. Dowell, “Evolution of airplanes, and what price speed?,” *AIAA Journal*, vol. 54, no. 3, pp. 1116–1119, 2016.
- [180] A. Bejan, U. Gunes, and B. Sahin, “The evolution of air and maritime transport,” *Applied Physics Reviews*, vol. 6, no. 2, 2019.
- [181] T. M. Williams, “The evolution of cost efficient swimming in marine mammals: Limits to energetic optimization,” *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 354, pp. 193–201, 1999.
- [182] M. J. Lighthill, “Note on the swimming of slender fish,” *Journal of Fluid Mechanics*, vol. 9, no. 2, pp. 305–317, 1960.
- [183] M. J. Lighthill, “Aquatic animal propulsion of high hydromechanical efficiency,” *Journal of Fluid Mechanics*, vol. 44, no. 2, pp. 265–301, 1970.
- [184] M. J. Lighthill, “Large-amplitude elongated-body theory of fish locomotion,” *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 179, no. 1055, pp. 125–138, 1971.
- [185] T. Y. Wu, “Fish swimming and bird/insect flight,” *Annual Review of Fluid Mechanics*, vol. 43, pp. 25–58, 2011.
- [186] E. D. Tytell, “The hydrodynamics of eel swimming II. Effect of swimming speed,” *Journal of Experimental Biology*, vol. 207, no. 19, pp. 3265–3279, 2004.
- [187] M. Kleiber, “Body size and metabolism,” *Hilgardia*, vol. 6, no. 11, pp. 315–353, 1932.
- [188] P. Eckert, “Part III: Integration of Physiological System,” in *Animal Physiology*, pp. 465–722, W. H. Freeman and Company, 1980.
- [189] P. Szymak, M. Morawski, and M. Malec, “Conception of research on bionic underwater vehicle with undulating propulsion,” in *Solid State Phenomena*, vol. 180, pp. 160–167, Trans Tech Publications Ltd, 2012.
- [190] M. S. Triantafyllou, G. S. Triantafyllou, and R. Gopalkrishnan, “Wake mechanics for thrust generation in oscillating foils,” *Physics of Fluids A*, vol. 3, no. 12, pp. 2835–2837, 1991.
- [191] J. M. Anderson, K. Streitlein, D. S. Barrett, and M. S. Triantafyllou, “Oscillating foils of high propulsive efficiency,” *Journal of Fluid Mechanics*, vol. 360, pp. 41–72, 1998.
- [192] A. Jaworski and J. Holm, “Distribution and structure of the population of sea lice, *Lepeophtheirus salmonis* Krøyer, on Atlantic salmon, *Salmo salar* L., under typical rearing conditions,” *Aquaculture and Fisheries Management*, vol. 23, no. 5, pp. 577–589, 1992.

- [193] C. S. Tucker, C. Sommerville, and R. Wootten, “Does size really matter? Effects of fish surface area on the settlement and initial survival of *Lepeophtheirus salmonis*, an ectoparasite of Atlantic salmon *Salmo salar*,” *Diseases of Aquatic Organisms*, vol. 49, no. 2, pp. 145–152, 2002.
- [194] F. Ling, J. G. Wang, A. H. Li, J. Y. Zhang, M. Li, Z. M. Gu, and X. N. Gong, “Determination of the surface area of common carp, *Cyprinus carpio* L.,” *Journal of Applied Ichthyology*, vol. 24, no. 6, pp. 690–693, 2008.
- [195] J. Siswantoro, A. S. Prabuwo, A. Abdullah, and B. Idrus, “Automatic image segmentation using sobel operator and k-means clustering: A case study in volume measurement system for food products,” *Proceedings - 2015 International Conference on Science in Information Technology: Big Data Spectrum for Future Information Economy, ICSITech 2015*, vol. M, pp. 13–18, 2016.
- [196] M. O. Balaban, M. Chombeau, B. Gümüş, and D. Cirban, “Determination of volume of alaska pollock (*Theragra chalcogramma*) by image analysis,” *Journal of Aquatic Food Product Technology*, vol. 20, no. 1, pp. 45–52, 2011.
- [197] X. Yang, S. Zhang, J. Liu, Q. Gao, S. Dong, and C. Zhou, “Deep learning for smart fish farming: applications, opportunities and challenges,” *Reviews in Aquaculture*, vol. 13, no. 1, pp. 66–90, 2021.
- [198] A. F. Fernandes, E. M. Turra, É. R. de Alvarenga, T. L. Passafaro, F. B. Lopes, G. F. Alves, V. Singh, and G. J. Rosa, “Deep Learning image segmentation for extraction of fish body measurements and prediction of body weight and carcass traits in Nile tilapia,” *Computers and Electronics in Agriculture*, vol. 170, no. January, pp. 105–274, 2020.
- [199] D. A. Konovalov, A. Saleh, J. A. Domingos, R. D. White, and D. R. Jerry, “Estimating Mass of Harvested Asian Seabass *Lateolabrax japonicus* from Images,” *World Journal of Engineering and Technology*, vol. 06, no. 03, pp. 15–23, 2018.
- [200] O. Ulucan, D. Karakaya, and M. Turkan, “A Large-Scale Dataset for Fish Segmentation and Classification,” *Proceedings - 2020 Innovations in Intelligent Systems and Applications Conference, ASYU 2020*, pp. 1–9, 2020.
- [201] A. J. Murphy and M. Haroutunian, “Using bio-inspiration to improve capabilities of underwater vehicles,” in *17th International Symposium on Unmanned Untethered Submersible Technology (UUST)*, Newcastle University, 2011.
- [202] R. Froese and D. Pauly, “Fishbase.” www.fishbase.org, 2019.
- [203] M. A. Bigg and A. A. Wolman, “Live-Capture Killer Whale (*Orcinus orca*) Fishery, British Columbia and Washington, 1962–73,” *Journal of the Fisheries Research Board of Canada*, vol. 32, no. 7, pp. 1213–1221, 1975.
- [204] L. Wen, Z. Ren, V. D. Santo, K. Hu, T. Yuan, T. Wang, and G. V. Lauder, “Understanding fish linear acceleration using an undulatory biorobotic model with soft fluidic elastomer actuated morphing median fins,” *Soft Robotics*, vol. 5, pp. 375–388, 8 2018.

- [205] P. Phamduy, *Robotic Fish to Aid Animal Behavior Studies and Informal Science Learning*. PhD thesis, New York University, 1 2017.
- [206] R. Ramamurti and J. D. Geder, “Development of a four-fin bio-inspired uuv : Cfd studies,” tech. rep., 2012.
- [207] T. Y.-T. Wu, “Hydromechanics of swimming propulsion. Part 1. Swimming of a two-dimensional flexible plate at variable forward speeds in an inviscid fluid,” *Journal of Fluid Mechanics*, vol. 46, no. 2, pp. 337–355, 1971.
- [208] T. Y.-T. Wu, “Hydromechanics of swimming propulsion . Part 2 . Some optimum shape problems,” *Journal of Fluid Mechanics*, vol. 46, pp. 521–544, 1971.
- [209] T. Y. T. Wu, “Hydromechanics of swimming propulsion. Part 3. Swimming and optimum movements of slender fish with side fins,” *Journal of Fluid Mechanics*, vol. 46, no. 3, pp. 545–568, 1971.
- [210] S. Jung, “Swimming, flying, and diving behaviors from a unified 2D potential model,” *Scientific Reports*, vol. 11, no. 1, pp. 1–11, 2021.
- [211] A. P. Maertens, A. Gao, and M. S. Triantafyllou, “Optimal undulatory swimming for a single fish-like body and for a pair of interacting swimmers,” *Journal of Fluid Mechanics*, vol. 813, pp. 301–345, 2017.
- [212] I. Borazjani and F. Sotiropoulos, “Numerical investigation of the hydrodynamics of carangiform swimming in the transitional and inertial flow regimes,” *Journal of Experimental Biology*, vol. 211, no. 10, pp. 1541–1558, 2008.
- [213] I. Borazjani and F. Sotiropoulos, “On the role of form and kinematics on the hydrodynamics of self-propelled body/caudal fin swimming,” *Journal of Experimental Biology*, vol. 213, no. 1, pp. 89–107, 2010.
- [214] M. Abbaspour and M. Ebrahimi, “Comparative numerical analysis of the flow pattern and performance of a foil in flapping and undulating oscillations,” *Journal of Marine Science and Technology (Japan)*, vol. 20, no. 2, pp. 257–277, 2015.
- [215] M. Ebrahimi and M. Abbaspour, “Numerical investigation of the forward and backward travelling waves through an undulating propulsor: performance and wake pattern,” *Ships and Offshore Structures*, vol. 11, no. 5, pp. 517–539, 2016.
- [216] S. Gupta, A. Sharma, A. Agrawal, M. C. Thompson, and K. Hourigan, “Hydrodynamics of a fish-like body undulation mechanism: Scaling laws and regimes for vortex wake modes,” *Physics of Fluids*, vol. 33, no. 10, 2021.
- [217] C. E. Lan, “The unsteady quasi-vortex-lattice method with applications to animal propulsion,” *Journal of Fluid Mechanics*, vol. 93, no. 4, pp. 747–765, 1979.
- [218] H. Kagemoto, M. J. Wolfgang, D. K. Yue, and M. S. Triantafyllou, “Force and power estimation in fish-like locomotion using a vortex-lattice method,” *Journal of Fluids Engineering, Transactions of the ASME*, vol. 122, no. 2, pp. 239–253, 2000.

- [219] G. Xue, Y. Liu, M. Zhang, and H. Ding, “Numerical Analysis of Hydrodynamics for Bionic Oscillating Hydrofoil Based on Panel Method,” *Applied Bionics and Biomechanics*, vol. 2016, 2016.
- [220] Y. Zheng and M. S. Liou, “A novel approach of three-dimensional hybrid grid methodology: Part 1. Grid generation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 192, no. 37-38, pp. 4147–4171, 2003.
- [221] J. Guerrero, “Overset composite grids for the simulation of complex moving geometries,” in *EUA4X. European Atelier for Engineering and Computational Sciences*, (Roma, Italy), pp. 1–14, 2006.
- [222] D. Chandar and H. Gopalan, “Comparative analysis of the arbitrary mesh interface(AMI), generalized grid interface (GGI) and overset methods for dynamic body motions in openFOAM,” *46th AIAA Fluid Dynamics Conference*, pp. 1–12, 2016.
- [223] D. D. Chandar and V. B. Boppana, “A Comparative Study of Different Overset Grid Solvers Between OpenFOAM, STAR-CCM+ and ANSYS-Fluent,” *AIAA Aerospace Sciences Meeting, 2018*, no. 210059, 2018.
- [224] P. Laws, J. S. Saini, A. Kumar, T. Dadri, and T. Dadri, “Performance Analysis of Two-Dimensional NACA 0018 Airfoil Using an Overset Grid Solver,” in *20th Annual CFD Symposium*, pp. 1–10, 2018.
- [225] D. D. Chandar, “On overset interpolation strategies and conservation on unstructured grids in OpenFOAM,” *Computer Physics Communications*, vol. 239, pp. 72–83, 2019.
- [226] S. Takahash, I. Monjugawa, and K. Nakahash, “Unsteady flow computations around moving airfoils by overset unstructured grid method,” *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 51, no. 172, pp. 78–85, 2008.
- [227] S. Deng, T. Xiao, B. van Oudheusden, and H. Bijl, “A dynamic mesh strategy applied to the simulation of flapping wings,” *International Journal for Numerical Methods in Engineering*, vol. 106, no. January, pp. 664–680, 2016.
- [228] G. Loubimov and M. P. Kinzel, “A cfd-based assessment of undulating propulsion,” *AIAA Aviation 2019 Forum*, no. June, pp. 1–15, 2019.
- [229] H. Dewar, *Studies of tropical tuna swimming performance: Thermoregulation, swimming mechanics and energetics*. PhD thesis, University of California, San Diego, 1993.
- [230] D. Xia, J. Liu, W. Chen, and L. Han, “Hydrodynamic analysis of fishlike robot swimming in the straight forward way,” *2009 IEEE International Conference on Mechatronics and Automation, ICMA 2009*, pp. 3342–3347, 2009.
- [231] J. Carling, T. L. Williams, and G. Bowtell, “Self-propelled anguilliform swimming: Simultaneous solution of the two-dimensional Navier-Stokes equations and Newton’s laws of motion,” *Journal of Experimental Biology*, vol. 201, no. 23, pp. 3143–3166, 1998.

- [232] N. Thekkethil, A. Sharma, and A. Agrawal, “Self-propulsion of fishes-like undulating hydrofoil: A unified kinematics based unsteady hydrodynamics study,” *Journal of Fluids and Structures*, vol. 93, p. 102875, 2020.
- [233] M. J. Lighthill, “Hydrodynamics of Aquatic Animal Propulsion,” *Annual Review of Fluid Mechanics*, vol. 1, no. 1, pp. 413–446, 1969.
- [234] B. Y. J. J. Videler and F. Hess, “Fast Continuous Swimming of Two Pelagic Predators, Saithe (*Pollachius Virens*) and Mackerel (*Scomber Scombrus*): a Kinematic Analysis,” *Journal of Experimental Biology*, vol. 109, no. MAR, pp. 209–228, 1984.
- [235] D. Scaradozzi, G. Palmieri, D. Costa, and A. Pinelli, “BCF swimming locomotion for autonomous underwater robots: a review and a novel solution to improve control and efficiency,” *Ocean Engineering*, vol. 130, no. October 2016, pp. 437–453, 2017.
- [236] T. Petra, “Description of the overset mesh approach in ESI version of OpenFOAM Learning outcomes,” in *Proceedings of CFD with OpenSource Software* (H. Nilsson, ed.), pp. 1–46, 2019.
- [237] Ansys Inc., “Ansys meshing.”
- [238] C. Geuzaine and J.-F. Remacle, “Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities,” *International Journal for Numerical Methods in Engineering*, vol. 0, no. 1, pp. 1–24, 2009.
- [239] W. M. Chan, R. J. Gomez III, S. E. Rogers, and P. G. Buning, “Best Practices in Overset Grid Generation,” *American Institute of Aeronautics and Astronautics*, vol. 3191, no. June, 2002.
- [240] F. Goetten, D. Felix, M. Marino, C. Bil, M. Havermann, and C. Braun, “A review of guidelines and best practices for subsonic aerodynamic simulations using RANS CFD,” *11th Asia-Pacific International Symposium of Aerospace Technology*, no. December, pp. 227–245, 2019.
- [241] P. M. Knupp, “Algebraic Mesh Quality Metrics,” *Journal of Scientific Computing*, vol. 23, no. 1, pp. 193–218, 2001.
- [242] J. Arlow and I. Neustadt, “Uml 2 and the unified process: Practical object-oriented analysis and design,” *Journal of Object Technology - JOT*, 01 2002.
- [243] F. R. Menter, M. Kuntz, and R. Langtry, “Ten Years of Industrial Experience with the SST Turbulence Model,” in *Proceedings of the fourth international symposium on turbulence, heat and mass transfer*, vol. 4, pp. 625–632, 2003.
- [244] A. Meana-Fernández, J. Fernández Oro, K. Argüelles Díaz, and S. Velarde-Suárez, “Turbulence-Model Comparison for Aerodynamic-Performance Prediction of a Typical Vertical-Axis Wind-Turbine Airfoil,” *Energies*, vol. 12, no. 3, p. 488, 2019.
- [245] J. Fürst, J. Příhoda, and P. Straka, “Numerical simulation of transitional flows,” *Computing*, vol. 95, no. SUPPL.1, 2013.

- [246] P. R. Spalart and S. R. Allmaras, “One-equation turbulence model for aerodynamic flows,” *Recherche aerospatiale*, no. 1, pp. 5–21, 1994.
- [247] M. Gokdepe, “Turbulence Models for the Numerical Prediction of Transitional Flows with RANSE,” *Turkish Naval Academy*, 2011.
- [248] R. Courant, K. Friedrichs, and H. Lewy, “On the Partial Difference Equations of Mathematical Physics,” *IBM Journal of Research and Development*, vol. 11, no. 2, pp. 215–234, 1967.
- [249] A. P. Maertens, M. S. Triantafyllou, and D. K. Yue, “Efficiency of fish propulsion,” *Bioinspiration and Biomimetics*, vol. 10, no. 4, 2015.
- [250] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.
- [251] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, Í. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . . Contributors, “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python,” *Nature Methods*, vol. 17, pp. 261–272, 2020.
- [252] J. Hunt, A. Wray, and P. Moin, “Eddies, Streams, and Convergence Zones in Turbulent Flows,” in *Proceedings of the Summer Program 1988*, pp. 193–208, 1988.
- [253] I. B. Celik, U. Ghia, P. J. Roache, C. J. Freitas, H. Coleman, and P. E. Raad, “Procedure for estimation and reporting of uncertainty due to discretization in CFD applications,” *Journal of Fluids Engineering, Transactions of the ASME*, vol. 130, no. 7, pp. 0780011–0780014, 2008.
- [254] C. Suvanjumrat, “Comparison of turbulence models for flow past NACA0015 airfoil using OpenFOAM,” *Engineering Journal*, vol. 21, no. 3, pp. 207–221, 2017.
- [255] P. A. Davidson, *Turbulence*. New York, New York: Oxford University Press, 2015.
- [256] H. Schlichting and K. Gersten, *Boundary-layer Theory*. Heidelberg: Springer Berlin,, 9 ed., 2017.
- [257] M. Drela, “XFOIL: an analysis and design system for low Reynolds number airfoils,” in *Low Reynolds Number Aerodynamics* (T. J. Mueller, ed.), pp. 1–12, Berlin, Heidelberg: Springer Berlin Heidelberg, 1989.
- [258] Y. Cha, J. Laut, P. Phamduy, and M. Porfiri, “Swimming Robots Have Scaling Laws, Too,” in *IEEE/ASME Transactions on Mechatronics*, vol. 21, 2016.

- [259] I. T. T. Committee, "Ittc quality system manual recommended procedures and guidelines: Freshwater and seawater properties." <https://www.ittc.info/media/7989/75-02-01-03.pdf>, 2011. Accessed: 2022-04-06.
- [260] N. Oceanic and A. A. N. C. for Environmental Information, "Hypsographic curve of the earth's surface from etopo1." https://ngdc.noaa.gov/mgg/global/etopo1_surface_histogram.html, 2012. Accessed: 2022-04-06.
- [261] S. I. F. SAE, "Cost catalogs." <https://www.fsaeonline.com/cdsweb/gen/DocumentResources.aspx>, 2022. Accessed: 2022-04-06.
- [262] B. Robotics, "Bluerov2." <https://bluerobotics.com/store/rov/bluerov2/>, 2022. Accessed: 2022-04-06.
- [263] K. H. Low, "Current and future trends of biologically inspired underwater vehicles," *2011 Defense Science Research Conference and Expo, DSR 2011*, pp. 1–8, 2011.
- [264] P. Ridao, M. Carreras, D. Ribas, P. J. Sanz, and G. Oliver, "Intervention auvs: The next challenge," *Annual Reviews in Control*, vol. 40, 2015.
- [265] A. B. Phillips, S. R. Turnock, and M. Furlong, "The use of computational fluid dynamics to aid cost-effective hydrodynamic design of autonomous underwater vehicles," *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, vol. 224, pp. 239–254, 11 2010.
- [266] L. L. V. Steenson, A. A. B. Phillips, E. Rogers, M. E. Furlong, S. R. Turnock, and N. O. Centre, "Control of an auv from thruster actuated hover to control surface actuated flight," *Specialists Meeting AVT-189/RSM-028, Assessment of Stability and Control Prediction, Methods for NATO Air & Sea Vehicles*, pp. 1–13, 2011.
- [267] A. B. Phillips, L. Steenson, E. Rogers, S. R. Turnock, C. Harris, and M. Furlong, "Delphin2: An over actuated autonomous underwater vehicle for manoeuvring research," *Transactions of the Royal Institution of Naval Architects Part A: International Journal of Maritime Engineering*, vol. 155, 2013.
- [268] L. V. Steenson, S. R. Turnock, A. B. Phillips, C. Harris, M. E. Furlong, E. Rogers, L. Wang, K. Bodles, and D. W. Evans, "Model predictive control of a hybrid autonomous underwater vehicle with experimental verification," *Proceedings of the Institution of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, vol. 228, pp. 166–179, 2014.
- [269] A. B. Phillips, G. Salavasidis, M. Kingsland, C. Harris, M. Pebody, D. R. R. Templeton, S. McPhail, T. Prampart, T. Wood, R. Taylor, and T. Jones, "Autonomous surface/-subsurface survey system field trials," in *AUV 2018 - 2018 IEEE/OES Autonomous Underwater Vehicle Workshop, Proceedings*, 2018.
- [270] A. D. Bowen, D. R. Yoerger, C. Taylor, R. McCabe, J. Howland, D. Gomez-Ibanez, J. C. Kinsey, M. Heintz, G. McDonald, D. B. Peters, B. Fletcher, C. Young, J. Buescher, L. L. Whitcomb, S. C. Martin, S. E. Webster, and M. V. Jakuba, "The nereus hybrid underwater robotic vehicle for global ocean science operations to 11,000m depth," in *OCEANS 2008*, 2008.

- [271] N. Kato, M. Choyekh, R. Dewantara, H. Senga, H. Chiba, E. Kobayashi, M. Yoshie, T. Tanaka, and T. Short, “An autonomous underwater robot for tracking and monitoring of subsea plumes after oil spills and gas leaks from seafloor,” *Journal of Loss Prevention in the Process Industries*, vol. 50, pp. 386–396, 11 2017.
- [272] S. Yu, S. Ma, B. Li, and Y. Wang, “An amphibious snake-like robot: Design and motion experiments on ground and in water,” in *Proceedings of the 2009 IEEE International Conference on Information and Automation*, pp. 500–505, IEEE, 2009.
- [273] B. Bayat, A. Crespi, and A. Ijspeert, “Envirobot: A bio-inspired environmental monitoring platform,” in *Autonomous Underwater Vehicles 2016, AUV 2016*, pp. 381–386, 11 2016.
- [274] A. Crespi and A. J. Ijspeert, “Online optimization of swimming and crawling in an amphibious snake robot,” *IEEE Transactions on Robotics*, vol. 24, pp. 75–87, 2 2008.
- [275] A. Crespi, K. Karakasiliotis, A. Guignard, and A. J. Ijspeert, “Salamandra robotica ii: An amphibious robot to study salamander-like swimming and walking gaits,” *IEEE Transactions on Robotics*, vol. 29, pp. 308–320, 2013.
- [276] A. Raj and A. Thakur, “Hydrodynamic parameter estimation for an anguilliform-inspired robot,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 99, pp. 837–857, 9 2020.
- [277] M. C. Leftwich and A. J. Smits, “Thrust production by a mechanical swimming lamprey,” *Experiments in Fluids*, vol. 50, pp. 1349–1355, 5 2011.
- [278] N. Kamamichi, M. Yamakita, K. Asaka, and Z. W. Luo, “A snake-like swimming robot using ipmc actuator/sensor,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, pp. 1812–1817, 5 2006.
- [279] C. Stefanini, S. Orofino, L. Manfredi, S. Mintchev, S. Marrazza, T. Assaf, L. Capantini, E. Sinibaldi, S. Grillner, P. Wallén, and P. Dario, “A novel autonomous, bioinspired swimming robot developed by neuroscientists and bioengineers,” *Bioinspiration and Biomimetics*, vol. 7, 6 2012.
- [280] J. Yu, M. Wang, W. Wang, M. Tan, and J. Zhang, “Design and control of a fish-inspired multimodal swimming robot,” in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3664–3669, IEEE, 5 2011.
- [281] J. Yu, S. Chen, Z. Wu, and W. Wang, “On a miniature free-swimming robotic fish with multiple sensors,” *International Journal of Advanced Robotic Systems*, vol. 13, 2016.
- [282] Z. X. Wu, J. Z. Yu, Z. S. Su, M. Tan, and Z. L. Li, “Towards an esox lucius inspired multimodal robotic fish,” *Science China Information Sciences*, vol. 58, pp. 1–13, 5 2015.
- [283] M. Marcin, S. Adam, Z. Jerzy, and M. Marcin, “Fish-like shaped robot for underwater surveillance and reconnaissance – hull design and study of drag and noise,” *Ocean Engineering*, vol. 217, 12 2020.

- [284] J. Shintake, H. Shea, and D. Floreano, "Biomimetic underwater robots based on dielectric elastomer actuators," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2016-Novem, pp. 4957–4962, IEEE, 10 2016.
- [285] D. Tan, Y. C. Wang, E. Kohtanen, and A. Erturk, "Trout-like multifunctional piezoelectric robotic fish and energy harvester," *Bioinspiration and Biomimetics*, vol. 16, 7 2021.
- [286] L. Wen, T. M. Wang, G. H. Wu, and J. H. Liang, "Hydrodynamic investigation of a self-propelled robotic fish based on a force-feedback control method," *Bioinspiration and Biomimetics*, vol. 7, pp. 1–17, 9 2012.
- [287] L. Wen, T. Wang, G. Wu, and J. Liang, "Quantitative thrust efficiency of a self-propulsive robotic fish: Experimental method and hydrodynamic investigation," *IEEE/ASME Transactions on Mechatronics*, vol. 18, pp. 1027–1038, 2013.
- [288] P. Romero, B. Sensale-Rodriguez, D. Astessiano, and R. Canetti, "FisHo: A cost-effective intelligent autonomous robot fish," in *2013 16th International Conference on Advanced Robotics, ICAR 2013*, 11 2013.
- [289] K. Hirata, "Development of experimental fish robot," *Sixth International Symposium On Marine Engineering*, pp. 711–714, 2000.
- [290] G. H. Yang and Y. Ryuh, "Design of high speed robotic fish 'ichthus v5.6'," in *International Conference on Control, Automation and Systems*, pp. 894–896, 10 2013.
- [291] Y. S. Ryuh, G. H. Yang, J. Liu, and H. Hu, "A school of robotic fish for mariculture monitoring in the sea coast," *Journal of Bionic Engineering*, vol. 12, pp. 37–46, 2015.
- [292] S. W. Tolkoff, *Robotics and Power Measurements of the RoboTuna*. PhD thesis, Massachusetts Institute of Technology, 2 1996.
- [293] X. Tant, D. Kim, N. Usher, D. Laboy, J. Jackson, A. Kapetanovic, J. Rapai, B. Sabadus, and X. Zhou, "An autonomous robotic fish for mobile sensing," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 5424–5429, 2006.
- [294] P. V. Alvarado, *Design of Biomimetic Compliant Devices for Locomotion in Liquid Environments*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [295] P. V. y Alvarado and K. Youcef-Toumi, "Modeling and design methodology of an efficient underwater propulsion system," in *Proceedings of the IASTED International Conference on Robotics and Applications*, pp. 161–166, 1 2003.
- [296] P. V. Y. Alvarado and K. Youcef-Toumi, "Performance of machines with flexible bodies designed for biomimetic locomotion in liquid environments," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, pp. 3324–3329, IEEE, 4 2005.
- [297] P. V. y Alvarado and K. Youcef-Toumi, "Design of machines with compliant bodies for biomimetic locomotion in liquid environments," *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME*, vol. 128, pp. 3–13, 3 2006.

- [298] S. Verma and J. X. Xu, "Analytic modeling for precise speed tracking of multilink robotic fish," *IEEE Transactions on Industrial Electronics*, vol. 65, pp. 5665–5672, 7 2018.
- [299] Q. Ren, J. Xu, L. Fan, and X. Niu, "A gim-based biomimetic learning approach for motion generation of a multi-joint robotic fish," *Journal of Bionic Engineering*, vol. 10, pp. 423–433, 10 2013.
- [300] J. Xu, Q. Ren, Z. Guo, and X. Niu, "Motion control for a multi-joint robotic fish with the pectoral fins assistance," in *IEEE International Symposium on Industrial Electronics*, pp. 1–6, IEEE, 2013.
- [301] P. Phamduy, R. Legrand, and M. Porfiri, "Robotic fish: Design and characterization of an interactive idevice-controlled robotic fish for informal science education," *IEEE Robotics and Automation Magazine*, vol. 22, pp. 86–96, 3 2015.
- [302] P. Phamduy, J. Cheong, and M. Porfiri, "An autonomous charging system for a robotic fish," *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 2953–2963, 12 2016.
- [303] J. Shao, L. Wang, and J. Yu, "Development of multiple robotic fish cooperation platform," *International Journal of Systems Science*, vol. 38, pp. 257–268, 1 2007.
- [304] Y. Hou, G. Du, X. Li, and D. Zhang, "A novel cpg controller of robotic fish: based on body wave function," *Computer Modeling & New Technologies*, vol. 19, pp. 16–19, 2015.
- [305] R. Tian, L. Li, W. Wang, X. Chang, S. Ravi, and G. Xie, "Cfd based parameter tuning for motion control of robotic fish," *Bioinspiration and Biomimetics*, vol. 15, 2020.
- [306] M. Malec and M. Morawski, "Analysis of thrust of underwater vehicle with undulating propulsion," *Advances in Intelligent Systems and Computing*, vol. 267, pp. 453–461, 2014.
- [307] P. Piskur, P. Szymak, Z. Kitowski, and L. Flis, "Influence of fin's material capabilities on the propulsion system of biomimetic underwater vehicle," *Polish Maritime Research*, vol. 27, pp. 179–185, 12 2020.
- [308] C. Christianson, N. N. Goldberg, D. D. Deheyne, S. Cai, and M. T. Tolley, "Translucent soft robots driven by frameless fluid electrode dielectric elastomer actuators," *Science Robotics*, vol. 3, 2018.
- [309] A. Ming and W. Zhao, *Design of biomimetic soft underwater robots*, pp. 91–111. Springer International Publishing, 1 2016.
- [310] J. Liu, I. Dukes, R. Knight, and H. Hu, "Development of fish-like swimming behaviours for an autonomous robotic fish," *Proceedings of the Control*, pp. 1–5, 9 2004.
- [311] J. Liu, I. Dukes, and H. Hu, "Novel mechatronics design for a robotic fish," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 807–812, IEEE Computer Society, 2005.

- [312] H. Hu, "Biologically inspired design of autonomous robotic fish at essex," in *Proceedings of the IEEE SMC UK-RI Chapter Conference on Advances in Cybernetic Systems*, pp. 1–8, IEEE, 10 2006.
- [313] J.-D. Liu and H. Hu, "Biologically inspired behaviour design for autonomous robotic fish," *International Journal of Automation and Computing*, vol. 3, pp. 336–347, 2006.
- [314] J. Liu and H. Hu, "A methodology of modelling fish-like swim patterns for robotic fish," in *Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, ICMA 2007*, pp. 1316–1321, IEEE, 8 2007.
- [315] A. I. Majeed and A. A. Ali, "Design and implementation of swimming robot based on carp fish biomimetic," in *Al-Sadiq International Conference on Multidisciplinary in IT and Communication Techniques Science and Applications, AIC-MITCSA 2016*, pp. 66–71, Institute of Electrical and Electronics Engineers Inc., 11 2016.
- [316] J. J. Hubbard, M. Fleming, V. Palmre, D. Pugal, K. J. Kim, and K. K. Leang, "Monolithic ipmc fins for propulsion and maneuvering in bioinspired underwater robotics," *IEEE Journal of Oceanic Engineering*, vol. 39, pp. 540–551, 2014.
- [317] D. Mohammadshahi, A. Yousefi-koma, S. Bahmanyar, and H. Maleki, "Design, fabrication and hydrodynamic analysis of a biomimetic robot fish," *International Journal of Mechanics*, vol. 2, pp. 59–66, 2008.
- [318] T. Q. Vo, H. S. Kim, and B. R. Lee, "Propulsive velocity optimization of 3-joint fish robot using genetic-hill climbing algorithm," *Journal of Bionic Engineering*, vol. 6, pp. 415–429, 12 2009.
- [319] K. A. Morgansen, T. M. L. Fond, and J. X. Zhang, "Agile maneuvering for fin-actuated underwater vehicles," in *Proceedings of the ISCCSP*, 2006.
- [320] Z. Qian, H. Liu, and Z. Bi, "Design and experiment of miniaturized and low-cost robotic fish with customerized electromagnetic actuation," in *International Conference on Theoretical Mechanics and Applied Mechanics*, pp. 9–15, s. n., 2015.
- [321] S. B. Behbahani, J. Wang, and X. Tan, "A dynamic model for robotic fish with flexible pectoral fins," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing, AIM 2013*, pp. 1552–1557, 2013.
- [322] W. Wang and G. Xie, "Cpg-based locomotion controller design for a boxfish-like robot," *International Journal of Advanced Robotic Systems*, vol. 11, 6 2014.
- [323] W. Wang, J. Zhao, X. Wei, F. Cao, and G. Xie, "Underwater electric current communication of robotic fish: Design and experimental results," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, pp. 1166–1171, IEEE, 5 2015.
- [324] M. Aureli, V. Kopman, and M. Porfiri, "Free-locomotion of underwater vehicles actuated by ionic polymer metal composites," *IEEE/ASME Transactions on Mechatronics*, vol. 15, pp. 603–614, 8 2010.

- [325] X. Deng and S. Avadhanula, "Biomimetic micro underwater vehicle with oscillating fin propulsion: System design and force measurement," in *2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, pp. 3312–3317, IEEE, 2005.
- [326] P. Kodati, J. Hinkle, A. Winn, and X. Deng, "Microautonomous robotic ostraciiform (marco): Hydrodynamics, design, and fabrication," *IEEE Transactions on Robotics*, vol. 24, pp. 105–117, 2 2008.
- [327] P. Kodati and X. Deng, *Bio-inspired Robotic Fish with Multiple Fins*, pp. 96–109. InTech, 2009.
- [328] S. Zhang, Y. Qian, P. Liao, F. Qin, and J. Yang, "Design and control of an agile robotic fish with integrative biomimetic mechanisms," *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 1846–1857, 8 2016.
- [329] Y. Zhong, Z. Li, and R. Du, "A novel robot fish with wire-driven active body and compliant tail," *IEEE/ASME Transactions on Mechatronics*, vol. 22, pp. 1633–1643, 8 2017.
- [330] T. Salumäe and M. Kruusmaa, "Flow-relative control of an underwater robot," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 469, 5 2013.
- [331] C. Rossi, J. Colorado, W. Coral, and A. Barrientos, "Bending continuous structures with smas: A novel robotic fish design," *Bioinspiration and Biomimetics*, vol. 6, pp. 1–15, 12 2011.
- [332] T. Wang, L. Wen, J. Liang, and G. Wu, "Fuzzy vorticity control of a biomimetic robotic fish using a flapping lunata tail," *Journal of Bionic Engineering*, vol. 7, pp. 56–65, 3 2010.
- [333] F. Shen, C. Wei, Z. Cao, D. Xu, J. Yu, and C. Zhou, "Implementation of a multi-link robotic dolphin with two 3-dof flippers," *Journal of Computational Information Systems*, vol. 7, pp. 2601–2607, 2011.
- [334] C. Wei and J. Yu, "Mechanical design of a slider-crank centered robotic dolphin," in *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*, pp. 3741–3746, IEEE, 2012.
- [335] J. Yu, Z. Su, M. Wang, M. Tan, and J. Zhang, "Control of yaw and pitch maneuvers of a multilink dolphin robot," *IEEE Transactions on Robotics*, vol. 28, pp. 318–329, 4 2012.
- [336] J. Yu, Z. Su, Z. Wu, and M. Tan, "Development of a fast-swimming dolphin robot capable of leaping," *IEEE/ASME Transactions on Mechatronics*, vol. 21, pp. 2307–2316, 10 2016.
- [337] W. P. Lau, Y. Zhong, R. Du, and Z. Li, "Bladderless swaying wire-driven robot shark," in *Proceedings of the 2015 7th IEEE International Conference on Cybernetics and Intelligent Systems, CIS 2015 and Robotics, Automation and Mechatronics, RAM 2015*, pp. 155–160, IEEE, 2015.

- [338] H. Wang, S. S. Tjahyono, B. MacDonald, P. A. Kilmartin, J. Travas-Sejdic, and R. Kiefer, "Robotic fish based on a polymer actuator," in *Proceedings of the 2007 Australasian Conference on Robotics and Automation, ACRA 2007*, 2007.
- [339] S. F. Masoomi, *An Efficient Biomimetic Swimming Robot Capable of Multiple Gaits of Locomotion Design, Modelling and Fabrication*. PhD thesis, University of Canterbury, 2014.
- [340] C. H. White, G. V. Lauder, and H. Bart-Smith, "Tunabot flex: A tuna-inspired robot with body flexibility improves high-performance swimming," *Bioinspiration and Biomimetics*, vol. 16, 3 2021.
- [341] M. I. Uddin and O. M. Curet, "Modeling and control of a bio-inspired underwater vessel with undulating-fin propulsion," in *OCEANS 2018 MTS/IEEE Charleston, OCEAN 2018*, pp. 1–7, 2019.
- [342] F. Veenstra, J. Jørgensen, and S. Risi, "Evolution of fin undulation on a physical knifefish-inspired soft robot," in *GECCO 2018 - Proceedings of the 2018 Genetic and Evolutionary Computation Conference*, pp. 157–164, 2018.
- [343] J. D. Geder, R. Ramamurti, J. S. Palmisano, M. Pruessner, B. Ratna, and W. C. Sandberg, "Sensor data fusion and submerged test results of a pectoral fin propelled uuv," in *International Symposium on Unmanned Untethered Submersible Technology, 2009*, 2009.
- [344] N. Kato, "Locomotion by mechanical pectoral fins," *Journal of Marine Science and Technology*, vol. 3, pp. 113–121, 1998.
- [345] N. Kato and T. Inaba, "Control performance of fish robot with pectoral fins in horizontal plane," in *UT 1998 - Proceedings of the 1998 International Symposium on Underwater Technology*, pp. 357–362, 1998.
- [346] N. Kato, B. W. Wicaksono, and Y. Suzuki, "Development of biology-inspired autonomous underwater vehicle "bass iii" with high maneuverability," in *Proceedings of the 2000 International Symposium on Underwater Technology, UT 2000*, pp. 84–89, 2000.
- [347] B. Kwak and J. Bae, "Toward fast and efficient mobility in aquatic environment: A robot with compliant swimming appendages inspired by a water beetle," *Journal of Bionic Engineering*, vol. 14, pp. 260–271, 4 2017.
- [348] C. Tang, R. Wang, Y. Wang, S. Wang, U. F. V. Lukas, and M. Tan, "Robcutt: A framework of underwater biomimetic vehicle-manipulator system for autonomous interventions," in *IEEE International Conference on Automation Science and Engineering*, vol. 2018-Augus, pp. 477–482, IEEE, 8 2018.
- [349] R. Wang, S. Wang, Y. Wang, M. Cai, and M. Tan, "Vision-based autonomous hovering for the biomimetic underwater robot - robcutt-ii," *IEEE Transactions on Industrial Electronics*, vol. 66, pp. 8578–8588, 11 2019.

- [350] L. Shi, S. Guo, and K. Asaka, "A novel jellyfish-and butterfly-inspired underwater microrobot with pectoral fins," in *Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation*, pp. 853–858, IEEE, 8 2011.
- [351] M. M. Rahman, W. W. Nik, and Y. Toda, "History of development of Squid-like biomimetic underwater robots with undulating side fins," *Jurnal Teknologi*, vol. 74, no. 9, pp. 129–136, 2015.
- [352] C. Niu, L. Zhang, S. Bi, and Y. Cai, "Development and depth control of a robotic fish mimicking cownose ray," in *2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012 - Conference Digest*, pp. 814–818, IEEE, 12 2012.
- [353] C. Niu, L. Zhang, S. Bi, and Y. Cai, "Development and depth control of a robotic fish mimicking cownose ray," in *2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012 - Conference Digest*, pp. 814–818, IEEE, 12 2012.
- [354] Y. Cai, S. Bi, and L. Zhang, "Design and implication of a bionic pectoral fin imitating cow-nosed ray," in *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings*, pp. 3525–3529, IEEE, 10 2010.
- [355] J. Gao, S. Bi, J. Li, and C. Liu, "Design and experiments of robot fish propelled by pectoral fins," in *2009 IEEE International Conference on Robotics and Biomimetics, ROBIO 2009*, pp. 445–450, IEEE, 12 2009.
- [356] L. Zheng, S. Bi, Y. Cai, and C. Niu, "Design and optimization of a robotic fish mimicking cow-nosed ray," in *2010 IEEE International Conference on Robotics and Biomimetics, ROBIO 2010*, pp. 1075–1080, IEEE, 2010.
- [357] F. Inc., "Festo bionic."
- [358] A. Cloitre, V. Subramaniam, N. Patrikalakis, and P. V. Y. Alvarado, "Design and control of a field deployable batoid robot," in *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 707–712, 2012.
- [359] C. Zhou and K. H. Low, "Design and locomotion control of a biomimetic underwater vehicle with fin propulsion," *IEEE/ASME Transactions on Mechatronics*, vol. 17, pp. 25–35, 2 2012.
- [360] S. bo Yang, J. Qiu, and X. yun Han, "Kinematics modeling and experiments of pectoral oscillation propulsion robotic fish," *Journal of Bionic Engineering*, vol. 6, pp. 174–179, 6 2009.
- [361] M. Anton, A. Punning, and A. Aabloo, "Towards a biomimetic eap robot," in *Proc. of TAROS*, pp. 1–7, 2004.
- [362] D. N. Beal, H. A. Leinhos, A. R. Fredette, and R. Berube, "Unified scaling for flapping fins," *IEEE Journal of Oceanic Engineering*, vol. 38, pp. 1–11, 2013.
- [363] J. D. Geder, R. Ramamurti, J. Palmisano, M. Pruessner, B. Ratna, and W. C. Sandberg, "Four-fin bio-inspired uuv: Modeling and control solutions," in *ASME 2011 International Mechanical Engineering Congress and Exposition, IMECE 2011*, vol. 2, pp. 799–808, 2011.

- [364] J. D. Geder, R. Ramamurti, M. Pruessner, and J. Palmisano, "Maneuvering performance of a four-fin bio-inspired uuv," in *OCEANS 2013 MTS/IEEE - San Diego: An Ocean in Common*, IEEE, 2013.
- [365] J. S. Palmisano, J. D. Geder, R. Ramamurti, W. C. Sandberg, and B. Ratna, "Robotic pectoral fin thrust vectoring using weighted gait combinations," *Applied Bionics and Biomechanics*, vol. 9, pp. 333–345, 2012.
- [366] C. Georgiades, A. German, A. Hogue, H. Liu, C. Prahacs, A. Ripsman, R. Sim, L. A. Torres, P. Zhang, M. Buehler, G. Dudek, M. Jenkin, and E. Milios, "Aqua: An aquatic walking robot," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 4, pp. 3525–3531, 2004.
- [367] B. Hobson, M. Murray, and C. Pell, "Pilotfish: maximizing agility in an unmanned underwater vehicle," in *Unmanned Untethered Submersible Technology*, pp. 41–51, 1999.
- [368] M. Kemp, B. Hobson, J. Janet, C. A. Pell, and E. D. Tytell, "Assessing the performance of oscillating fin thruster vehicles," in *Int. Symp. Unmanned Untethered Submersible Tech.*, vol. 12th, 2001.
- [369] S. Hsu, C. Mailey, E. Eade, and J. Janét, "Autonomous control of a horizontally configured undulatory flap propelled vehicle," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2194–2199, IEEE, 10 2003.
- [370] K. Seo, S. J. Chung, and J. J. E. Slotine, "Cpg-based control of a turtle-like underwater vehicle," *Autonomous Robots*, vol. 28, pp. 247–269, 4 2010.
- [371] N. Kato, "Swimming and walking of an amphibious robot with fin actuators," *Marine Technology Society Journal*, vol. 45, pp. 181–197, 2011.
- [372] A. KONNO, T. FURUYA, A. MIZUNO, K. HISHINUMA, K. HIRATA, and M. KAWADA, "Development of turtle-like submersible vehicle," *Journal of The Japan Institute of Marine Engineering*, vol. 41, pp. 158–163, 2006.
- [373] R. Altendorfer, N. Moore, H. Komsuoglu, M. Buehler, H. B. Brown, D. McMordie, U. Saranlı, R. Full, and D. E. Koditschek, "Rhex: A biologically inspired hexapod runner," *Autonomous Robots*, vol. 11, pp. 207–213, 11 2001.
- [374] S. Ravichandran, A. B. Arackal, A. S. Mazumdar, J. S. P. Kiran, and P. Rajagopal, "Bio-inspired underwater robot with reconfigurable and detachable swimming modules," in *ACM International Conference Proceeding Series*, vol. Part F1320, Association for Computing Machinery, 6 2017.
- [375] S. Licht, V. Polidoro, M. Flores, F. S. Hover, and M. S. Triantafyllou, "Design and projected performance of a flapping foil auv," *IEEE Journal of Oceanic Engineering*, vol. 29, pp. 786–794, 7 2004.
- [376] H. Xing, S. Guo, L. Shi, X. Hou, Y. Liu, H. Liu, Y. Hu, D. Xia, and Z. Li, "A novel small-scale turtle-inspired amphibious spherical robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1702–1707, IEEE, 11 2019.

- [377] W. Zhao, Y. Hu, L. Wang, and Y. Jia, "Development of a flipper propelled turtle-like underwater robot and its cpg-based control algorithm," in *Proceedings of the IEEE Conference on Decision and Control*, pp. 5226–5231, Institute of Electrical and Electronics Engineers, 2008.
- [378] C. Christianson, C. Bayag, G. Li, S. Jadhav, A. Giri, C. Agba, T. Li, and M. T. Tolley, "Jellyfish-inspired soft robot driven by fluid electrode dielectric organic robotic actuators," *Frontiers in Robotics and AI*, vol. 6, 11 2019.
- [379] M. Cianchetti, M. Calisti, L. Margheri, M. Kuba, and C. Laschi, "Bioinspired locomotion and grasping in water: The soft eight-arm octopus robot," *Bioinspiration and Biomimetics*, vol. 10, 6 2015.
- [380] T. Paschal, J. Shintake, S. Mintchev, and D. Floreano, "Development of bio-inspired underwater robot with adaptive morphology capable of multiple swimming modes," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, pp. 4197–4202, IEEE, 9 2017.
- [381] M. Sfakiotakis, A. Kazakidi, and D. P. Tsakiris, "Octopus-inspired multi-arm robotic swimming," *Bioinspiration and Biomimetics*, vol. 10, 6 2015.
- [382] A. Villanueva, C. Smith, and S. Priya, "A biomimetic robotic jellyfish (robojelly) actuated by shape memory alloy composite actuators," *Bioinspiration and Biomimetics*, vol. 6, 2011.
- [383] A. A. Villanueva, K. J. Marut, T. Michael, and S. Priya, "Biomimetic autonomous robot inspired by the cyanea capillata (cyro)," *Bioinspiration and Biomimetics*, vol. 8, 12 2013.
- [384] R. Berquist, G. K.M., M. Peterson, A. Doan, G. Baxter, K. Yopak, N. Kang, H. Walker, P. Hastings, and L. Frank, "The Digital Fish Library: Using MRI to Digitize, Database, and Document the Morphological Diversity of Fish," *PLoS ONE*, vol. 7, no. 4, p. e34499, 2012.
- [385] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.
- [386] S. Suzuki and K. A. Be, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics and Image Processing*, vol. 30, no. 1, pp. 32–46, 1985.
- [387] L. Charles L., J. Cuyler W. Brooks, H. Acquilla S., and S. Darrell W., "Computer program to obtain ordinates for naca airfoils," tech. rep., National Aeronautic and Space Association, 1996.
- [388] F. W. White, *Fluid Mechanics*. McGraw-Hill Companies, Inc., 2011.
- [389] D. K. Walters and D. Cokljat, "A three-equation eddy-viscosity model for reynolds-averaged navier-stokes simulations of transitional flow," *Journal of Fluids Engineering, Transactions of the ASME*, vol. 130, no. 12, pp. 1214011–12140114, 2008.

Appendix A

Data For models

A.1 Biological Kinematic Data

Table A.1 Table of kinematic parameters for various fish taken from Videler [42].

Species	Length [m]	A [m]	A [BL]	Freq [Hz]	T [s]	Lambda_b [m]	Lambda_b [BL]	Lambda_s [m]	Lambda_s [BL]	U [m/s]	U [BL/s]	V [m/s]	V [BL/s]	u/v	Re	Sw
Abramis brama	0.19	0.02	0.12	3.80	0.26	0.16	0.82	0.12	0.63	0.45	2.40	0.59	3.10	0.76	85500	206862.57
Leuciscus leuciscus	0.25	0.03	0.13	2.80	0.36	0.20	0.82	0.15	0.60	0.42	1.70	0.57	2.30	0.74	105000	285884.93
Carassius auratus	0.16	0.02	0.12	4.00	0.25	0.16	1.02	0.11	0.71	0.46	2.90	0.65	4.10	0.71	73600	154415.56
Esox sp.	0.18	0.01	0.06	6.00	0.17	0.15	0.82	0.08	0.41	0.46	2.50	0.91	5.00	0.50	83814	151500.67
Oncorhynchus mykiss	0.20	0.02	0.08	4.00	0.25	0.18	0.89	0.13	0.63	0.50	2.50	0.72	3.60	0.70	101103	162462.06
Oncorhynchus mykiss	0.06	0.01	0.10	9.90	0.10	0.05	0.96	0.04	0.65	0.35	6.50	0.53	9.60	0.67	19525	37633.14
Oncorhynchus mykiss	0.12	0.01	0.09	7.10	0.14	0.11	0.91	0.08	0.70	0.58	5.00	0.76	6.50	0.77	67280	108050.48
Oncorhynchus mykiss	0.25	0.02	0.10	3.60	0.28	0.20	0.82	0.16	0.64	0.57	2.30	0.73	2.90	0.78	141930	280485.92
Oncorhynchus mykiss	0.43	0.03	0.07	2.40	0.42	0.39	0.89	0.28	0.64	0.67	1.50	0.92	2.10	0.72	287945	395817.45
Salmo Salar	0.67	0.06	0.09	1.30	0.76	0.63	0.95	0.39	0.59	0.51	0.80	0.83	1.20	0.62	340480	650188.10
Salmo Salar	0.67	0.05	0.08	1.70	0.57	0.69	1.03	0.44	0.66	0.77	1.10	1.21	1.80	0.63	510720	755774.20
Salmo Salar	0.67	0.03	0.05	5.30	0.19	0.61	0.92	0.46	0.68	2.37	3.60	3.22	4.80	0.74	1576050	1472648.26
Liza ramada	0.36	0.03	0.08	4.60	0.22	0.34	0.95	0.26	0.72	1.20	3.30	1.58	4.40	0.76	432000	599325.40
Gadus morhua	0.42	0.03	0.08	2.80	0.36	0.35	0.83	0.26	0.62	0.73	1.70	0.96	2.30	0.76	306600	496542.54
Gadus morhua	0.42	0.03	0.07	3.30	0.30	0.34	0.81	0.27	0.64	0.91	2.20	1.14	2.70	0.80	382200	512059.30
Gadus morhua	0.42	0.03	0.08	3.60	0.28	0.36	0.85	0.26	0.61	0.92	2.20	1.27	3.00	0.72	386400	638411.84
Gadus morhua	0.69	0.06	0.09	1.90	0.52	0.63	0.91	0.45	0.64	0.87	1.20	1.23	1.80	0.71	603955	1037947.97
Pollachius virens	0.35	0.02	0.07	2.20	0.45	0.32	0.92	0.28	0.80	0.63	1.80	0.72	2.10	0.87	220500	237064.58
Pollachius virens	0.35	0.03	0.09	2.70	0.37	0.35	1.01	0.28	0.79	0.73	2.10	0.94	2.70	0.78	257250	374069.44
Pollachius virens	0.35	0.03	0.09	3.30	0.31	0.37	1.05	0.29	0.82	0.94	2.70	1.21	3.40	0.78	330750	457195.98
Pollachius virens	0.35	0.04	0.10	3.70	0.27	0.36	1.02	0.28	0.79	1.01	2.90	1.31	3.80	0.77	355250	569570.75
Pollachius virens	0.35	0.04	0.10	3.50	0.29	0.36	1.03	0.30	0.86	1.05	3.00	1.26	3.60	0.84	367500	538783.14
Scomber scombrus	0.35	0.04	0.10	3.80	0.26	0.35	1.00	0.28	0.81	1.08	3.10	1.34	3.80	0.81	379750	584964.55
Scomber scombrus	0.35	0.04	0.10	5.30	0.19	0.34	0.98	0.23	0.66	1.23	3.50	1.82	5.20	0.67	428750	815871.61
Scomber scombrus	0.40	0.04	0.09	4.50	0.22	0.37	0.92	0.29	0.73	1.32	3.30	1.66	4.10	0.80	528000	814300.82
Scomber scombrus	0.40	0.03	0.08	4.90	0.21	0.36	0.91	0.28	0.70	1.36	3.40	1.77	4.40	0.77	544000	783162.76
Scomber scombrus	0.30	0.03	0.10	5.60	0.18	0.34	1.15	0.24	0.81	1.35	4.50	1.93	6.40	0.70	405000	633345.08
Scomber scombrus	0.31	0.03	0.11	5.50	0.18	0.31	1.01	0.28	0.89	1.52	4.90	1.73	5.60	0.88	470890	730615.07
Scomber scombrus	0.31	0.03	0.11	5.00	0.20	0.34	1.10	0.32	1.03	1.58	5.10	1.70	5.50	0.93	490110	664195.52
Scomber scombrus	0.31	0.04	0.12	12.00	0.08	0.29	0.95	0.22	0.71	2.67	8.60	3.55	11.40	0.75	826460	1738984.63
Scomber scombrus	0.33	0.03	0.10	5.10	0.20	0.33	1.01	0.26	0.78	1.32	4.00	1.70	5.20	0.78	435600	697925.69
Scomber scombrus	0.33	0.04	0.11	5.40	0.19	0.36	1.08	0.31	0.93	1.65	5.00	1.92	5.80	0.86	544500	812875.76
Scomber scombrus	0.34	0.04	0.11	12.20	0.08	0.32	0.94	0.26	0.75	3.13	9.20	3.90	11.50	0.80	1063520	1949486.42
Scomber scombrus	0.34	0.04	0.11	13.90	0.07	0.34	1.00	0.25	0.73	3.47	10.20	4.72	13.90	0.73	1179120	2221136.17
Scomber scombrus	0.34	0.03	0.10	14.90	0.07	0.32	0.94	0.26	0.75	3.81	11.20	4.77	14.00	0.80	1294720	2164481.94
Anguilla anguilla	0.14	0.01	0.10	3.60	0.28	0.11	0.79	0.08	0.55	0.28	2.00	0.40	2.90	0.70	39060	86668.31
Hyperoplus lanceolatus	0.29	0.02	0.08	3.60	0.28	0.21	0.75	0.15	0.51	0.52	1.80	0.77	2.70	0.68	149240	298102.86
Hyperoplus lanceolatus	0.29	0.02	0.07	3.00	0.33	0.21	0.74	0.15	0.54	0.47	1.60	0.64	2.20	0.72	135455	217366.67
Hyperoplus lanceolatus	0.29	0.03	0.09	2.70	0.37	0.23	0.79	0.14	0.51	0.59	1.40	0.61	2.10	0.64	112504	251524.29
Hyperoplus lanceolatus	0.29	0.02	0.06	2.90	0.34	0.23	0.79	0.15	0.52	0.44	1.50	0.67	2.30	0.65	125706	180103.81
Hyperoplus lanceolatus	0.33	0.04	0.11	2.60	0.39	0.29	0.87	0.18	0.56	0.47	1.40	0.74	2.20	0.64	154770	391384.64
Ammodytes marinus	0.09	0.01	0.09	6.00	0.17	0.07	0.79	0.05	0.55	0.28	3.20	0.41	4.70	0.68	24273	51362.02
Ammodytes marinus	0.08	0.01	0.09	7.70	0.13	0.07	0.81	0.05	0.59	0.37	4.50	0.51	6.20	0.73	30504	58555.92
Ammodytes marinus	0.10	0.01	0.11	5.00	0.20	0.10	0.92	0.07	0.64	0.54	3.20	0.48	4.60	0.69	35175	76190.33
Ammodytes marinus	0.09	0.01	0.10	3.50	0.29	0.07	0.79	0.03	0.36	0.11	1.20	0.24	2.70	0.45	9309	33290.20

Table A.2 Table of kinematic parameters for various fish taken from Videler and Nolet [8].

Common Name	Species	Mode	Type	Trophic Level	T	Weight [kg]	Length [m]	Upl [m/s]	Re	CGT0p [J/m]	CG0p [J/m]	CG0T [J/m]	CGT0p [J]	GC0T0W	GC00W	NC00
Baltic Fraon	Paderion adespas	Packing	Slump	2	20	0.00	0.03	0.04	1200.00	8.21	80.54	0.01	0.00	0.00	0.00	3.27
Baltic Fraon	Paderion adespas	Packing	Slump	2	20	0.00	0.04	0.04	1600.00	5.83	57.10	0.02	0.00	0.00	0.00	2.26
Baltic Fraon	Paderion adespas	Packing	Slump	2	20	0.00	0.04	0.05	2000.00	5.31	52.09	0.05	0.00	0.00	0.00	2.09
Baltic Fraon	Paderion adespas	Packing	Slump	2	20	0.00	0.05	0.04	2100.00	4.77	46.97	0.00	0.00	0.00	0.00	1.87
Baltic Fraon	Paderion adespas	Packing	Slump	2	20	0.00	0.06	0.05	2300.00	4.00	40.22	0.06	0.00	0.00	0.00	1.54
Baltic Fraon	Paderion adespas	Packing	Slump	2	20	0.00	0.07	0.04	2300.00	3.65	35.81	0.08	0.01	0.00	0.00	1.42
Cordill	Larusus auratus	Slump	Slump	1	15	0.10	0.15	0.21	63000.00	0.97	6.57	0.02	0.02	0.07	0.01	0.15
Largecast Mullet	Larusus auratus	Undulation	Fish	3	29	0.01	0.10	0.48	27000.00	0.14	0.05	0.01	0.01	0.01	0.01	0.21
Nile Tilapia	Tilapia nilotica	Undulation	Fish	3	25	0.08	0.14	0.21	95000.00	0.21	6.57	0.05	0.05	0.05	0.05	0.07
Atlantic Salmon	Salmo salar	Undulation	Fish	3	25	0.05	0.14	0.21	29486.05	0.45	4.43	0.02	0.00	0.03	0.02	0.05
European silver eel	Anguilla anguilla	Undulation	Fish	3	12	0.05	0.14	0.21	14080.00	0.06	0.05	0.02	0.00	0.00	0.00	0.08
Green Sea Turtle	Caretta caretta	Undulation	Turtle	3	20	0.71	0.68	0.50	50523.46	0.16	0.37	0.05	0.02	0.12	0.12	0.02
Green Sea Turtle	Caretta caretta	Undulation	Turtle	3	25	0.30	1.5000.00	2.25	250000.00	0.04	0.30	0.05	0.00	5886.74	67689.00	13243.15
Green Sea Turtle	Caretta caretta	Undulation	Turtle	3	29	1.15	0.30	0.40	170000.00	0.31	3.04	0.50	0.66	1.06	0.71	0.18
Green Sea Turtle	Caretta caretta	Undulation	Turtle	3	17	0.38	0.34	0.46	153000.00	0.23	2.56	0.11	1.28	0.28	0.38	0.16
Las Whitefish	Coregonus clupeaformis	Undulation	Fish	3	15	0.13	0.12	0.42	153000.00	0.21	2.26	0.02	0.07	0.07	0.13	0.12
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	20	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.46	21000.00	0.34	3.14	0.16	0.01	0.01	0.02	0.16
Largefin Mullet	Chelonula mydas	Undulation	Fish	3	15	0.13	0.12	0.4								

A.2 Conventional AUV Data

Table A.3 Table of kinematic and COT data for conventional AUV platforms.

Manufacturer	Name	Locomotion	Length [m]	Width [m]	Height [m]	Weight [kg]	Umax [m/s]	Uopt [m/s]	COTopt [J/m]	Re	Reference	
Boeing-Liquid Robotics	Wave Glider	Glider	3.050	0.810	8.000	500.000	1.300	0.670		2043500.000	[10]	
Hydroid	Seaglider	Glider	2.000	1.300	1.300	60.000	0.500	0.260	1.923	520000.000	[10]	
iRobot	Seaglider	Glider	1.800	0.300	0.400	52.000		0.250	3.640	450000.000	[10]	
Teledyne Marine	Slocum Electric - 1km	Glider	1.500	0.210	0.210	52.000		0.350	10.400	525000.000	[10]	
Teledyne Marine	Slocum Electric - Coastal	Glider	1.500	0.210	0.210	52.000		0.350	6.240	525000.000	[10]	
Teledyne Marine	Slocum G3 Glider	Glider	1.500	0.220	0.220	70.000		0.000			[12]	
Atlas Elektronik	Seacat MKII	Propeller	3.450	0.980	0.480	1100.000	4.120	2.060	726.000	7107000.000	[12]	
Atlas Maridan	SeaCat	Propeller	3.250	0.580	0.670	220.000	6.000	3.080		10010000.000	[11, 264]	
Atlas Maridan	SeaFox	Propeller	1.310	0.390	0.390	43.000		0.000		4034000.000	[10]	
Australian National University	Serafina	Propeller	0.455	0.100	0.140		1.000	2.200		1001000.000	[10]	
BAE Systems	IMP	Propeller	0.647	0.191	0.191	29.500	7.000	3.600		2329200.000	[10]	
BAE Systems	2MP	Propeller	0.800	23.800	23.800	54.500	6.000	3.080		2464000.000	[10]	
BAE Systems	microUUV	Propeller	0.559	0.124	0.124	11.300	10.000	5.140		2873560.000	[13]	
Bluefin Robotics	Bluefin 21	Propeller	4.930	0.530	0.530	750.000	2.300	1.540	352.500	7592200.000	[10]	
Bluefin Robotics	BPAUV	Propeller	1.830	0.530	0.530	362.870	2.060	1.540	163.292	2818200.000	[10]	
Boeing	Echo Voyager	Propeller	15.500	2.600	2.600	45360.000	8.000	4.110		63705000.000	[10]	
Central Mechanical Engineering Research Institute	AUV-150	Propeller	4.800	0.500	0.500	490.000	4.000	2.060		9888000.000	[10]	
Commonwealth Scientific and Industrial Research	Starbug	Propeller	1.200	0.150	0.150	26.000	2.910	1.500		1800000.000	[10]	
Cybernetix	ALIVE	Propeller	4.000	2.200	1.600	3500.000	2.570	1.540	4095.000	6160000.000	[10]	
Daewoo	OKPO-300	Propeller	1.830	0.260	0.260	55.000	6.000	3.080		5636000.000	[10]	
ECA SA	Alistar	Propeller	5.000	1.680	1.140	2300.000	2.060	1.030	1058.000	5150000.000	[10]	
ECA SA	Alister Daurade	Propeller	5.000	0.700		950.000	4.110	2.050	1073.500	10250000.000	[132, 265–269]	
Evo Logics	Sonobot	Propeller	1.320	0.920	0.450	30.000	6.480	3.330		4395600.000	[10]	
Exocetus	Coastal Glider	Propeller	1.829	1.062	0.324	109.000	2.000	1.030		1883870.000	[10]	
GD-Bluefin Robotics	Bluefin-12	Propeller	2.500	0.320	0.320	250.000	3.087	1.029		4049633.870	[10]	
GD-Bluefin Robotics	Bluefin-21	Propeller	4.930	0.530	0.530	750.000	2.315	1.543		7608469.000	[10]	
GD-Bluefin Robotics	Bluefin-9	Propeller	2.481	0.238	0.264	70.000	3.087	1.029		2552673.609	[10]	
GD-Bluefin Robotics	HALUV	Propeller	1.330	93.000	38.000	72.600	1.500	0.770		1024100.000	[10]	
Grail Tech	Poluga	Propeller	2.180	0.150	0.150	3.000	1.060			4490800.000	[10]	
Heriot-Watt University Ocean Systems Laboratory	Micro AUV	Propeller	0.680	0.110	0.110	3.500		0.000			[10]	
Heriot-Watt University Ocean Systems Laboratory	Nessie 2012	Propeller	1.600	0.280	0.280	45.000	5.000	2.570		4112000.000	[10]	
Heriot-Watt University Ocean Systems Laboratory	Nessie IV (2009)	Propeller	0.700	0.700	0.600	41.000	5.000	2.570		1799000.000	[10]	
Heriot-Watt University Ocean Systems Laboratory	PADV AUV	Propeller	1.300	1.200		350.000		0.000			[10]	
Hydroid	Remus 100	Propeller	1.700	0.190	0.190	36.000	5.000	2.570	17.510	4369000.000	[10]	
Hydroid	Remus 100	Propeller	1.600	0.190	0.190	37.000	2.600	1.500	30.340	2400000.000	[10]	
Hydroid	Remus 300	Propeller	2.510	0.190	0.190	59.000	5.000	2.570		6450700.000	[15, 10]	
Hydroid	Remus 600	Propeller	3.250	0.320	0.320	240.000	2.600	1.500	50.400	4875000.000	[15, 10]	
Hydroid	Remus 600	Propeller	5.500	0.324	0.324	385.000	4.000	2.060		11330000.000	[15, 10]	
Hydroid	Remus 6000	Propeller	3.840	0.710	0.710	862.000	2.600	1.543	327.560	5925120.000	[15, 10]	
Hydroid	Remus 6000	Propeller	3.960	0.710	0.710	862.000	4.500	2.310		9147600.000	[15, 10]	
Hydroid	Remus MAV	Propeller	0.910	0.120	0.120	10.000	5.140			4677400.000	[15, 10]	
International Submarine Engineering Ltd.	Explorer	Propeller	5.500	0.740	0.740	1250.000	2.500	1.500	487.500	8250000.000	[10]	
International Submarine Engineering Ltd.	Explorer	Propeller	4.500	0.690	0.690	630.000	2.570	1.540	390.600	6930000.000	[14]	
International Submarine Engineering Ltd.	Explorer	Propeller	7.500	0.740	0.740	1700.000	3.900	2.000		15000000.000	[10]	
International Submarine Engineering Ltd.	Thesius	Propeller	10.700	0.130	0.130	8600.000	2.060		4816.000	22842000.000	[10]	
International Submarine Engineering Ltd.	Thesius	Propeller	10.700	1.270	1.270	8600.000	4.000	2.060		22042000.000	[10]	
Japan Agency for Marine-Earth Science and Technology	Urashima	Propeller	10.000	1.300	1.500	7257.480	3.000	1.540		15400000.000	[10]	
Kongsberg	Hugin 1000	Propeller	4.500	0.750	0.750	850.000	6.000	3.080		1386000.000	[10]	
Kongsberg	Hugin 3000	Propeller	5.500	1.000	1.000	1400.000	4.000	2.060		11330000.000	[10]	
Kongsberg	Hugin 4500	Propeller	6.000	1.000	1.000	1900.000	4.000	2.060		12360000.000	[10]	
Kongsberg	Hugin Superior	Propeller	6.600	0.875	0.875	2200.000	5.200	2.670		17622000.000	[10]	
Kongsberg Maritime	HUGIN 1000	Propeller	4.500	0.750	0.750	850.000	3.080	2.050	306.000	9225000.000	[10]	
Kongsberg Maritime	HUGIN 3000	Propeller	5.500	1.000	1.000	1400.000	2.050	2.050	364.000	11275000.000	[10]	
Kongsberg Maritime	HUGIN 4500	Propeller	6.000	1.000	1.000	1900.000	2.050	2.050	589.000	12300000.000	[10]	
L3 Harris OceanServer	Iver3 Standard	Propeller	2.160	0.150	0.150	38.640	4.000	2.060		4449600.000	[10]	
L3 Harris OceanServer	Iver4 900	Propeller	2.500	0.230	0.230	90.910	5.000	2.570		6425000.000	[10]	
Lockheed Martin	Marlin Mk1	Propeller	1.500	0.800	0.800	454.000	8.000	4.110		6165000.000	[270]	
Lockheed Martin	Marlin Mk2	Propeller	3.000	1.500	1.300	954.000	6.000	3.080		9240000.000	[10]	
Lockheed Martin	Marlin Mk3	Propeller	4.900	1.500	1.300	1590.000	6.000	3.080		15092000.000	[10]	
Marine Autonomous Systems Engineering	SAUVIM	Propeller	6.100	1.800	2.100	6500.000	3.000	1.540		9394000.000	[160]	
Maritime and Ocean Engineering Research Institute	USM-H	Propeller	1.500	0.200	0.200	20.000	0.700		74.000	1087000.000	[10]	
Memorial University	MUN Explorer	Propeller	5.300	0.690	0.690	830.000	2.500	1.500	729.166	7950000.000	[10]	
MIT AUV Lab	Odyssey IV	Propeller	2.600	1.500	1.300	25.000	3.900	2.000		5200000.000	[10]	
MIT AUV Laboratory	Odyssey IV	Propeller	2.600	0.700	1.400	650.000	2.060	1.540	526.500	4004000.000	[10]	
Monterey Bay Aquarium Research Institute	Benthic Rover	Propeller	2.500	1.500	1.200	1400.000	0.390			800000.000	[10]	
Monterey Bay Aquarium Research Institute	Dorado	Propeller	5.240	0.530	0.530	1018.000	2.060	1.540	488.640	8069600.000	[10]	
Monterey Bay Aquarium Research Institute	Dorado	Propeller	6.400	0.533	0.533	476.000		0.000			[10]	
Monterey Bay Aquarium Research Institute	Tethys	Propeller	2.300	0.305	0.305	120.000	1.950	1.000		23000.000	[10]	
National Deep Submergence Facility	Sentry	Propeller	2.900	2.200	1.800	1250.000	2.000	1.030		2987000.000	[15, 10]	
National Oceanography Center	Autosub 3	Propeller	3.700	0.800	0.800	1300.000	3.900	2.000	127.500	7400000.000	[15, 10]	
National Oceanography Center	Autosub6000	Propeller	3.700	0.900	0.900	1300.000	2.000	1.000	195.000	3700000.000	[15, 10]	
National University of Singapore	Starfish	Propeller	1.600	0.200	0.200	45.000	4.480	2.300		3680000.000	[15, 10]	
Naval Postgraduate School	Aries	Propeller	3.000	0.400	0.250	220.000	3.500	1.800		5400000.000	[15, 10]	
Newcastle University	Delphin 2	Propeller	1.960	0.254	0.254	50.000	1.000	0.510		999600.000	[15, 10]	
Ocean Aero	SubMaran UUSV	Propeller	2.430	0.480	2.130	56.700	5.830	3.000		7290000.000	[15, 10]	
OceanServer Technology	Iver2	Propeller	0.130	0.010	0.010	19.000	2.060	1.290		33.250	167700.000	[10]
Office of Naval Research	Odyssey	Propeller	2.100	0.600	0.600	150.000	3.890	0.420	121.428	882000.000	[10]	
Osaka University	SOTAB-I	Propeller	2.500	0.667	0.667	312.000		0.000			[10]	
QinetiQ	SEAScouter	Propeller	1.220	0.120	0.120		15.000	7.710		9406200.000	[10]	
Saab Seacye	Sabertooth Double Hull	Propeller	4.004	1.350	0.670	1500.000	4.000	2.060		8433640.000	[10]	
Saab Seacye	Sabertooth Single Hull	Propeller	3.600	0.660	0.450	650.000	5.000	2.570		9252000.000	[10]	
Scripps Institution of Oceanography	Spray	Propeller	1.800	0.300	0.300	51.800	0.350	0.250	2.072	450000.000	[10]	
Sone Aerospace	DepthX	Propeller	4.260	3.040	3.040	1350.000	0.390	0.200		852000.000	[10]	
Sone Aerospace	Sunfish	Propeller	1.610	0.470	0.200	50.000		0.000			[10]	
Tecnico Lisboa Dynamical Systems and Ocean Robotics Laboratory	Melissa	Propeller	1.035	0.350	0.875	30.000	2.910	1.500		1552500.000	[10]	
Teledyne Marine	Gavia	Propeller	4.500	0.200	0.200	130.000	5.500	2.830		12735000.000	[10]	
Teledyne Marine	Sea Raptor	Propeller	5.500	0.630	0.630	1200.000	3.000	1.540		8470000.000	[10]	
University of Delaware Robotic Discovery Laboratories	Fech 3	Propeller	2.110	0.340	0.340	97.000	5.830	3.000		633000.000	[10]	
University of Porto Ocean Systems Group	MARES	Propeller	1.500	0.200	0.200	32.000	3.900	2.000		3000000.000	[10]	
University of Porto Ocean Systems Group	TriMARES	Propeller	1.300	0.800	0.500	70.000	3.900	2.000		2600000.000	[10]	
University of Tokyo Institute of Industrial Science	ALBAC	Propeller	1.400	1.200	0.236	45.000	2.000	1.030		1442000.000	[10]	
University of Tokyo Institute of Industrial Science	Maria-ceresia	Propeller	0.450	0.630	0.135	14.000	1.950	1.000		1450000.000	[10]	
University of Tokyo Institute of Industrial Science	PIERON150	Propeller	1.500	0.750	0.450	220.000	2.000	1.030		1545000.000	[10]	
University of Tokyo Institute of Industrial Science	R-one	Propeller	8.270	1.150	1.150	4740.000	3.000	1.540		12735800.000	[10]	
University of Tokyo Institute of Industrial Science	r2DA	Propeller	4.400	1.080	0.810	1630.000	3.000	1.540		6776000.000	[271]	
University of Tokyo Institute of Industrial Science	Tri-Dog 1	Propeller	2.000	0.900	0.600	200.000	1.370	0.700		1400000.000	[10]	
University of Tokyo Institute of Industrial Science	Tuna-Sand	Propeller	1.100	0.700	0.710	240.000	2.500	1.290		1419000.000	[10]	
University of Tokyo Institute of Industrial Science	Twin Burger	Propeller	1.540	0.860	0.540	120.000	1.000	0.510		785400.000	[10]	
University of Victoria	Maco	Propeller	1.520	0.170	0.440	70.000	2.910	1.500		2280000.000	[10]	
Virginia Institute of Marine Science	Fech	Propeller	1.900	0.290	0.290	99.000	5.840	3.000		57000.000	[10]	

Table A.4 Table of kinematic and COT data for conventional AUV platforms (Continued).

Manufacturer	Name	Max Depth [m]	Endurance [hr]	Battery Rating [kWh]	Battery Type	Hotel Power [W]	Propulsion Power [W]	Reference
Boeing-Liquid Robotics	Wave Glider	15 000	8766 000					[10]
Hydroid	Seaglider	1000 000	7305 000			0 500		[10]
Robot	Seaglider	1000 000	5111 000	4 720	Lithium			[10]
Telodyne Marine	Slocum Electric – 1km	1000 000	528 000	1 901	Alkaline C cell Or Li			[10]
Telodyne Marine	Slocum Electric – Coastal	200 000	840 000	1 901	Alkaline C cell Or Li			[10]
Telodyne Marine	Slocum G3 Glider	1000 000	13149 000					[12]
Atlas Elektronik	Seascout MKII	600 000	24 000	36 000				[12]
Atlas Maridan	SeaCat	600 000	20 000					[11, 264]
Atlas Maridan	SeaFox	300 000	0 110					[10]
Australian National University	Serafina		2 000					[10]
BAE Systems	1MP	300 000						[10]
BAE Systems	2MP	300 000						[10]
BAE Systems	microUUV	300 000	30 000					[13]
Bluefin Robotics	Bluefin 21	4500 000	25 000	13 500				[10]
Bluefin Robotics	BPALUV	6000 000	18 000	4 500	Li-Ion			[10]
Boeing	Echo Voyager	300 000	18 780					[10]
Central Mechanical Engineering Research Institute	AUV-150	150 000						[10]
Commonwealth Scientific and Industrial Research	Starbug	100 000	4 000					[10]
Cybernetix	ALIVE		7 000	44 000	Lead acid			[10]
Daewoo	OKPO-300	300 000	10 000					[10]
ECA SA	Alistar	3000 000	20 000	22 000	Li-Ion			[10]
ECA SA	Alistar Daurade	300 000	10 000	22 000	Li-Ion			[132, 265–269]
EvoLogics	Sonobot		10 000					[10]
Excetus	Coastal Glider	200 000	720 000					[10]
GD-Bluefin Robotics	Bluefin-12	200 000	36 000	7 600	Li-Ion			[10]
GD-Bluefin Robotics	Bluefin-21	200 000	25 000	13 500	Li-Polymer			[10]
GD-Bluefin Robotics	Bluefin-9	200 000	12 000	1 900	Li-Ion			[10]
GD-Bluefin Robotics	HALUV	60 000	3 500					[10]
Grad Tech	Pelaga	80 000	14 000					[10]
Heriot-Watt University Ocean Systems Laboratory	Micro AUV	5 000	3 000					[10]
Heriot-Watt University Ocean Systems Laboratory	Nessie 2012	100 000	22 000					[10]
Heriot-Watt University Ocean Systems Laboratory	Nessie IV (2009)	100 000	3 000					[10]
Heriot-Watt University Ocean Systems Laboratory	PAUV AUV	1000 000						[10]
Hydroid	Remus 100	100 000	12 000			45 000		[10]
Hydroid	Remus 100	100 000	22 000	1 000	Li-Ion			[10]
Hydroid	Remus 300	305 000	29 000					[15, 10]
Hydroid	Remus 600	600 000	70 000	5 200	Li-Ion			[15, 10]
Hydroid	Remus 600	1500 000	24 000					[15, 10]
Hydroid	Remus 6000	6000 000	22 000	11 000	Li-Ion			[15, 10]
Hydroid	Remus 6000	6000 000	22 000					[15, 10]
Hydroid	Remus M3V	300 000						[15, 10]
International Submarine Engineering Ltd.	Explorer	5000 000	19 000	14 000	Li-Ion			[10]
International Submarine Engineering Ltd.	Explorer	3000 000	22 000	13 200	Li-Ion			[14]
International Submarine Engineering Ltd.	Explorer	6000 000	85 000					[10]
International Submarine Engineering Ltd.	Theseus	2000 000	60 000	600 000	Li-Ion			[10]
International Submarine Engineering Ltd.	Theseus	1000 000	183 400					[10]
Japan Agency for Marine-Earth Science and Technology	Urashima	3500 000	18 000					[10]
Kongsberg	Hugin 1000	1000 000	24 000					[10]
Kongsberg	Hugin 3000	3000 000	60 000					[10]
Kongsberg	Hugin 4500	4500 000	60 000					[10]
Kongsberg	Hugin Superior	6000 000	72 000					[10]
Kongsberg Maritime	HUGIN 1000	1000 000	24 000	15 000	Li-Polymer			[10]
Kongsberg Maritime	HUGIN 3000	3000 000	60 000	45 000	AI/HP semi Fuelcell			[10]
Kongsberg Maritime	HUGIN 4500	4500 000	50 000	60 000	AI/HP semi Fuelcell			[10]
L3 Harris OceanServer	Iver3 Standard	100 000	14 000					[10]
L3 Harris OceanServer	Iver4 900	300 000	26 700					[10]
Lockheed Martin	Marlin MK1	304 000	16 000					[270]
Lockheed Martin	Marlin MK2	4000 000	24 000					[10]
Lockheed Martin	Marlin MK3	4000 000	80 000					[10]
Marine Autonomous Systems Engineering	SAUVIM	6000 000						[160]
Maritime and Ocean Engineering Research Institute	ISIMI	20 000	4 000	0 207	Li-Polymer			[10]
Memorial University	MUN Explorer	3000 000	16 000	17 500	Moli-Lithium-Ion Cobalt			[10]
MIT AUV Lab	Odyssey IV	6000 000	8 000					[10]
MIT AUV Laboratory	Odyssey IV	6000 000	5 560	4 500	Li-Ion			[10]
Monterey Bay Aquarium Research Institute	Benthic Rover	6000 000	8766 000					[10]
Monterey Bay Aquarium Research Institute	Dorado	1500 000	8 000	6 000	Li-Ion			[10]
Monterey Bay Aquarium Research Institute	Dorado	6000 000	20 000					[10]
Monterey Bay Aquarium Research Institute	Tethys	740 000						[10]
National Deep Submergence Facility	Sentry	6000 000	60 000					[15, 10]
National Oceanography Center	Autosub 3	6000 000	72 000			255 000		[15, 10]
National Oceanography Center	Autosub6000	6000 000	103 000	42 000	Li-Ion Polymer			[15, 10]
National University of Singapore	Starfish	100 000	6 000					[15, 10]
Naval Postgraduate School	Aries	50 000	8 000					[15, 10]
Newcastle University	Delphin 2	50 000	8 000	0 030		30 000		[15, 10]
Ocean Aero	SubMaran UUSV	22 860	4000 000					[15, 10]
OceanServer Technology	Iver2	100 000	24 000	0 600				[10]
Office of Naval Research	Odyssey	6700 000	4000 000			51 000		[10]
Osaka University	SOTAB-1	2000 000						[10]
QinetiQ	SEAScout	243 840						[10]
Saab Seacye	Sabertooth Double Hull	3000 000	14 000					[10]
Saab Seacye	Sabertooth Single Hull	1200 000	8 000					[10]
Scripps Institution of Oceanography	Spray	1500 000	6666 000	3 611				[10]
Stone Aerospace	DepthX	1000 000						[10]
Stone Aerospace	Sunfish							[10]
Techno Labou Dynamical Systems and Ocean Robotics Laboratory	Medusa	1 000	11 000					[10]
Telodyne Marine	Gavia	1000 000	8 000					[10]
Telodyne Marine	Sea Raptor	6000 000	24 000					[10]
University of Delaware Robotic Discovery Laboratories	Fetch 3	200 000	10 000					[10]
University of Porto Ocean Systems Group	MARES	100 000	10 000					[10]
University of Porto Ocean Systems Group	TriMARES	100 000	10 000					[10]
University of Tokyo Institute of Industrial Science	ALBAC	300 000	0 500					[10]
University of Tokyo Institute of Industrial Science	Manta-cesaria	10 000	2 500					[10]
University of Tokyo Institute of Industrial Science	PTEROA150	2000 000	1 000					[10]
University of Tokyo Institute of Industrial Science	R-one	400 000	18 040					[10]
University of Tokyo Institute of Industrial Science	rDM	4000 000	10 820					[271]
University of Tokyo Institute of Industrial Science	Tri-Dog 1	110 000	4 000					[10]
University of Tokyo Institute of Industrial Science	Tuna-Sand	1500 000	5 000					[10]
University of Tokyo Institute of Industrial Science	Twin Burger	50 000	2 000					[10]
University of Victoria	Mako	60 000	2 500					[10]
Virginia Institute of Marine Science	Fetch	150 000	18 000					[10]
Virginia Tech Autonomous Systems and Controls Laboratory	475 AUV		8 000					[10]
Virginia Tech Autonomous Systems and Controls Laboratory	690 AUV	500 000	24 000					[10]
Virginia Tech Autonomous Systems and Controls Laboratory	Self-Mooring AUV	500 000	25 000					[10]
Woods Hole Oceanographic Institution	Autonomous Benthic Explorer (ABE)	6000 000	20 000	5 000				[10]
Woods Hole Oceanographic Institution	Autonomous Benthic Explorer (ABE)	6000 000	20 000	5 000				[10]
Woods Hole Oceanographic Institution	Nereus	11000 000	20 000	18 000	Li-Ion			[10]
Woods Hole Oceanographic Institution	SeaBED	2000 000	8 000	2 000	Li-Ion			[10]
Woods Hole Oceanographic Institution	seaBED	6000 000	24 000					[10]
YSI	IXHO EcoMapper	1000 000	14 000					[10]

A.3 Bio-inspired BCF AUV Data

Table A.5 Table of kinematic and COT data for BCF Bio-inspired Robotic Platforms.

Author	Affiliation	Locomotion	Actuator	No Actuators	No Linkages	Compliant Tail	Length [m]	Width [m]	Height [m]	Weight [kg]	Reference
Ding	Chinese Academy of Science	Anguilliform	Servo	7	4	0	0.700	0.320	0.150	4.995	[87]
Yu	Chinese Academy of Science	Anguilliform	Motor	18	9	0	0.117	0.075	0.075	6.750	[272]
Bayat	Ecole Polytechnique Federale de Lausanne	Anguilliform	Motor	6	6	0	1.300	0.050	0.090		[273]
Crespi	Ecole Polytechnique Federale de Lausanne	Anguilliform	Motor	7	7	0	0.770				[274]
Crespi	Ecole Polytechnique Federale de Lausanne	Anguilliform	Motor	9	8	0	1.100				[275]
Raj	Indian Institute of Technology Patna	Anguilliform	Servo	5	5	0	0.650	0.074	0.074	1.750	[276]
Nguyen	Japan Advanced Institute of Science and Technology	Anguilliform	Hydraulic	4	2	1	0.255	0.045	0.045		[101]
Niu	National University of Singapore	Anguilliform	Servo	3	3	0	0.720	0.056	0.063	2.540	[59]
Kelassidi	NTNU	Anguilliform	Servo	9	9	0	1.600	0.038	0.097	14.400	[91]
Leftwich	Princeton University	Anguilliform	Servo	13	8	0	1.140	0.038	0.097		[277]
Stuehlig	Technical University of Munich	Anguilliform	Motor	1	15	0	1.080	0.055	0.250	7.500	[33]
Kamamichi	Tokyo Institute of Technology	Anguilliform	IPMC	2	2	1	0.120	0.000	0.000	0.001	[278]
Manfredi	University of Dundee	Anguilliform	Servo	10	10	0	0.990	0.054	0.054	1.640	[279]
Joo	Airo Inc.	Carangiform	Servo	2	2	0	0.530	0.110	0.250	2.600	[26]
Joo	Airo Inc.	Carangiform	Servo	2	2	0	0.350	0.070	0.190	1.300	[26]
Joo	Airo Inc.	Carangiform	Servo	2	2	0	0.800	0.300	0.390	5.900	[26]
Yu	Chinese Academy of Science	Carangiform	Servo	3	2	0	0.350	0.610	0.830	0.970	[150]
Liao	Chinese Academy of Science	Carangiform	Servo	2	13	0	0.495			1.256	[280]
Yu	Chinese Academy of Science	Carangiform	Servo	6	4	0	0.400	0.040	0.078	0.500	[281]
Yu	Chinese Academy of Science	Carangiform	Motor	1	2	0	0.370	0.028	0.075		[23]
ZhengXing	Chinese Academy of Science	Carangiform	Servo	6	4	0	0.614	0.083	0.081	2.210	[282]
Shin	Chonnam National University	Carangiform	Motor	4	4	0	0.780	0.210	0.190	4.230	[56]
Morawski	Cracow University of Technology	Carangiform	Servo	6	2	0	2.285	0.242	0.509	55.000	[283]
Anderson	Draper Laboratory	Carangiform	Hydraulic	5	4	0	2.400	0.400	0.500	173.000	[284]
Shintake	Ecole Polytechnique Federale de Lausanne	Carangiform	DEA	1	1	1	0.150			0.004	[53]
Bal	Firat University	Carangiform	Servo	2	2	0	0.500	0.076	0.215	3.100	[285]
Tan	Georgia Institute of Technology	Carangiform	MFC	2	2	0	0.305				[92]
Kim	Hankuk Aviation University	Carangiform	IPMC	1	1	0	0.096	0.024	0.025	0.016	[81]
Chen	Harbin Institute of Technology	Carangiform	Servo	1	7	0	0.420				[107, 51]
Wang	Harbin Institute of Technology	Carangiform	SMA	1	1	0	0.146	0.017	0.034	0.030	[286, 108, 287]
Wen	Harvard and Beihang University	Carangiform	Servo	4	4	0	0.588	0.080	0.950	2.790	[88]
Gibouin	Institute de Physique de Nice	Carangiform	Servo	1	1	1	0.170		0.060		[288]
Romero	Instituto de Matematica y Estadística	Carangiform	Servo	2	2	0					[289]
Hirata	Japan Marine Science and Technology Center	Carangiform	Servo	2	3	0	0.650	0.148	0.148	1.100	[50]
Hirata	Japan Marine Science and Technology Center	Carangiform	Servo	2	2	0	0.340			1.100	[90]
Guo	Kagawa University	Carangiform	IPMC	2	2	1	0.057	0.010	0.007	0.001	[102]
Nguyen	Konkuk University	Carangiform	Piezoelectric	2	2	0	0.400	0.150	0.040		[290, 291]
Ryuh	Korea Institute of Industrial Technology	Carangiform	Servo	5	3	0	0.500	0.146	0.170	4.700	[16]
Fujisawa	Kyushu University	Carangiform	Motor	1	1	1	0.345			4.597	[148, 176, 292, 62]
Kumph	Massachusetts Institute of Technology	Carangiform	Servo	3	6	0	0.820			3.630	[27]
Tan	Michigan State University	Carangiform	IPMC	1	1	1	0.230	0.065	0.130	0.295	[293]
ValdiviaAlvarado	MIT	Carangiform	Motor	1	1	1	0.470	0.180	0.230	1.650	[294–297, 18]
Herr	MIT	Carangiform	Muscle	2	1	0	0.120	0.070	0.050	0.012	[24]
Epps	MIT	Carangiform	Servo	1	1	1	0.148	0.025	0.043	0.068	[21, 22]
Low	Nanyang Technological University	Carangiform	Motor	2	4	0	0.500	0.065	0.150	2.500	[17]
Low	Nanyang Technological University	Carangiform	Motor	2	3	0	0.661	0.100	0.260	0.077	[25]
Guo	National Taiwan University	Carangiform	Motor	2	2	1	2.400	0.400	0.700	200.000	[89]
Verna	National University of Singapore	Carangiform	Servo	2	2	0	0.580			2.490	[298]
Xu	National University of Singapore	Carangiform	Servo	4	2	0	0.600				[299, 300]
Phamduy	New York University	Carangiform	Servo	3	4	0	0.460	0.100	0.190	1.170	[301, 302, 205]
Liu	Northeastern University	Carangiform	IPMC	1	1	1	0.198	0.060	0.065	0.166	[151]
Hu	Peking University	Carangiform	Servo	4	3	0	0.460	0.270	0.280	2.200	[55]
Shao	Peking University	Carangiform	Servo	3	3	0	0.450				[303–305]
Szymak	Polish Naval Academy	Carangiform	Servo	4	3	0	0.690	0.220	0.230	3.400	[46, 189, 306, 19, 307]
Jatsun	South West State University, Kursk, Russia	Carangiform	Servo	3	1	0	0.700	0.280	0.230	5.900	[20]
Christianson	University of California, San Diego	Carangiform	DEA	1	1	1	0.220			0.025	[308]
Ming	University of Electro-Communications	Carangiform	MFC	2	1	1	0.158		0.064		[309]
Clapham	University of Essex	Carangiform	Motor	1	1	0	0.050	0.006	0.046	0.005	[85]
Clapham	University of Essex	Carangiform	Motor	1	2	0	0.250	0.050	0.062	0.367	[84]
Clapham	University of Essex	Carangiform	Motor	1	2	0	0.320	0.048	0.112	0.835	[85]
Hu	University of Essex	Carangiform	Servo	4	3	0	0.520				[310–314]
Majeed	University of Mustangsiyah	Carangiform	Servo	4	4	0	0.545			5.000	[315]
Hubbard	University of Nevada, Reno	Carangiform	IPMC	5	2	0	0.177			0.067	[316]
Yan	University of Science and Technology of China	Carangiform	Servo	4	4	0	0.605	0.080	0.200	3.800	[109]
Mohammadshahi	University of Tehran	Carangiform	Servo	3	2	0	0.600	0.120	0.200	1.400	[317]
Vo	University of Ulsan	Carangiform	Servo	2	3	0	1.200				[318]
Suleman	University of Victoria	Carangiform	SMA	8	3	0	1.600				[106]
Morgansen	University of Washington	Carangiform	Servo	4	2	0	0.543	0.114	0.127	3.000	[319, 100]
Yu	Chengdu University of Information	Ostraciiform	Motor	1	1	0	0.295	0.120	0.170	1.604	[57]
Liu	Dalian University of Technology	Ostraciiform	Magnetic	1	1	1	0.054	0.008	0.008	0.720	[97]
Qian	East China University of Science and Technology	Ostraciiform	Magnetic	4	4	0	0.078	0.049	0.036	0.026	[320]
Berlinger	Harvard University	Ostraciiform	Magnetic	1	1	0	0.120			0.138	[28]
Shibata	Kinokuni University	Ostraciiform	Servo	1	1	0	0.260	0.065	0.115	0.530	[105]
Chan	Konkuk University	Ostraciiform	Servo	1	1	0	0.304	0.074	0.088	1.196	[80]
Behbahani	Michigan State University	Ostraciiform	Servo	2	3	0	0.150	0.066	0.120	0.460	[321]
Papadopoulos	National Technical University of Athens	Ostraciiform	Motor	1	1	0	0.337				[103]
Wang	Peking University	Ostraciiform	Servo	3	3	0	0.400	0.140	0.142	3.100	[322, 323]
Auereli	Polytechnic Institute of New York University	Ostraciiform	IPMC	1	1	1	0.130	0.035	0.045	0.009	[324]
Kopman	Polytechnic Institute of New York University	Ostraciiform	Servo	1	1	0	0.167	0.026	0.048	0.070	[94]
Lachat	Swiss Federal Institute of Technology	Ostraciiform	Motor	3	1	0	0.250			0.440	[95]
Kim	Tohoku University	Ostraciiform	Magnetic	3	3	1	0.040	0.012	0.011	0.002	[93]
Deng	University of California, Berkeley	Ostraciiform	Piezoelectric	1	4	0	0.020			0.001	[325]
Kodati	University of Delaware	Ostraciiform	Servo	3	3	0	0.150	0.092	0.085	0.490	[326, 327]
Zhang	University of Science and Technology of China	Ostraciiform	Servo	4	4	0	0.440	0.316	0.080	1.300	[328]
McGovern	University of Wollongong	Ostraciiform	Polypropylene	2	1	1	0.125	0.020	0.020	0.016	[52]
Hou	Wichita State University	Ostraciiform	IPMC	3	3	1	0.018	0.008	0.008	0.150	[29]
Xie	Chinese University of Hong Kong	Sub-Carangiform	Servo	3	7	0	0.506	0.248	0.128	1.080	[54]
Zhong	Chinese University of Hong Kong	Sub-Carangiform	Servo	4	8	1	0.310			0.500	[329, 110]
Daou	Tallinn University of Technology	Sub-Carangiform	Servo	1	1	1	0.220	0.080	0.150		[330, 86]
Rossi	Universidad Politécnica de Madrid	Sub-Carangiform	SMA	6	6	0	0.300				[331, 104]
McColgan	University of Glasgow	Sub-Carangiform	Servo	8	8	0	0.900	0.250	0.160	4.300	[36, 58]
Liang	Beihang University	Thunniform	Servo	1	4	0	1.760	0.220	0.220	3.300	[332, 31]
Rufo	Boston Engineering	Thunniform	Hydraulic	3	1	1	1.520	0.360	0.460	40.800	[61]
Shen	Chinese Academy of Science	Thunniform	Servo	5	4	0	0.560	0.240	0.160	3.290	[333–336]
Pk	Chinese University of Hong Kong	Thunniform	Servo	2	7	0	0.600	0.075	0.100	1.790	[337]
Erak	Georgia Institute of Technology	Thunniform	MFC	1	1	1	0.243			0.542	[32]
Chen	Harbin Institute of Technology	Thunniform	Motor	1	3	0	1.050	0.220	0.220	26.200	[60]
Li	Harbin Institute of Technology	Thunniform	Servo	1	1	1	0.320	0.064	0.100		[96]
Ye	Harbin Institute of Technology	Thunniform	IPMC	1	1	1	0.078	0.022	0.030	0.295	[48]
Tomic	Kyushu Institute of Technology	Thunniform	Magnetic	1	1	1	0.063	0.034	0.010		[49]
Chen	Michigan State University	Thunniform	IPMC	1	1	1	0.223	0.006		0.290	[83]
Chen	Tongji University	Thunniform	MFC	2	1	1	0.300	0.046	0.086	0.450	[82]
Wang	University of Auckland	Thunniform	Polypropylene	1	1	1	0.090	0.035	0.010		[338]
Masoomi	University of Canterbury	Thunniform	Motor	3	3	0	0.700			7.426	[339, 98]
Ming	University of Electro-Communications	Thunniform	Piezoelectric	2	2	1	0.110	0.020	0.045	0.015	[99]
Zhu	University of Virginia	Thunniform	Motor	1	1	1	0.255	0.049	0.068	0.306	[30, 340]

Table A.6 Table of kinematic and COT data for BCF Bio-inspired Robotic Platforms (Continued).

Author	Affiliation	U _{max} [m/s]	U _{opt} [m/s]	COT _{opt} [J/m]	Freq [Hz]	A [BL]	Re	Sw	Yaw Speed [m/s]	Yaw Radius [m]	Reference
Ding	Chinese Academy of Science	0.450	0.450		1.650	0.082	15000.00833112.673			0.230	[87]
Yu	Chinese Academy of Science	0.070	0.070				8190.000				[272]
Bayat	Ecole Polytechnique Federale de Lausanne	0.870	0.870		1.250		1131000.000				[273]
Crespi	Ecole Polytechnique Federale de Lausanne	0.249	0.222		1.000	0.66	1170940.00924847.352				[274]
Crespi	Ecole Polytechnique Federale de Lausanne	0.510	0.510	19.600	1.000		561000.000				[275]
Raj	Indian Institute of Technology Patna	0.019	0.019		3.500		12675.000				[276]
Nguyen	Japan Advanced Institute of Science and Technology				1.700	0.110	152802.983				[101]
Niu	National University of Singapore	0.115	0.115		1.510		82800.000			1.000	[59]
Kelasidi	NTNU	0.100	0.075	933.330	2.093	0.129	20000.00816452.383				[91]
Leftwich	Princeton University	0.115	0.115		0.550		131100.000				[277]
Struwig	Technical University of Munich	0.470	0.297	69.940	1.350		321192.000				[33]
Kamanichi	Tokyo Institute of Technology	8.000	8.000		0.400		960000.000				[278]
Manfredi	University of Dundee	0.247	0.247	40.486	0.600		245025.000			1.000	[279]
Joo	Airo Inc.	0.500	0.500	15.960			265000.000			0.500	[26]
Joo	Airo Inc.	0.500	0.500	11.400			175000.000			0.300	[26]
Joo	Airo Inc.	0.600	0.200				160000.000				[26]
Yu	Chinese Academy of Science	0.500	0.500	50.320			175000.000				[150]
Liao	Chinese Academy of Science	0.335	0.335		1.000		165825.000		51.400		[280]
Yu	Chinese Academy of Science	0.320	0.320		2.000		128000.000		57.300	0.200	[281]
Yu	Chinese Academy of Science	1.200	1.200		8.000		444000.000		63.800		[23]
ZhengXing	Chinese Academy of Science	0.460	0.460		6.200		282440.000				[282]
Shin	Chonnam National University	0.720	0.720				561600.000			0.370	[56]
Morawski	Cracow University of Technology	1.500	1.500			0.078	27500.000		30.000	1.300	[283]
Anderson	Draper Laboratory	1.250	1.250		1.000		300000.000		75.000	7.680	[284]
Shintake	Ecole Polytechnique Federale de Lausanne	0.037	0.037	24.731	0.750	0.047	5580.000				[53]
Bal	Firat University	0.680	0.680		2.500	0.182	40000.00829267.578			0.260	[285]
Tan	Georgia Institute of Technology	0.256	0.256		5.400		78141.000				[92]
Kim	Hankuk Aviation University	0.024	0.024		4.000	0.013	2265.600	6022.207			[81]
Chen	Harbin Institute of Technology	0.298	0.298		1.500	0.220	25244.000731513.566				[107, 51]
Wang	Harbin Institute of Technology	0.112	0.112		2.500	0.013	16352.000	8698.908	0.057	0.136	[286, 108, 287]
Wen	Harvard and Beihang University	0.355	0.355		1.500	0.100	208740.00851712.086				[88]
Gibouin	Institute de Physique de Nice	0.104	0.104		1.600	0.250	17680.00045267.244				[288]
Romero	Intituto de Matematica y Estadistica	0.300	0.300		1.200		260000.000				[289]
Hirata	Japan Marine Science and Technology Center	0.400	0.400		3.000		68000.000			0.087	[50]
Hirata	Georgia Institute of Technology	0.200	0.200		2.300		1485000.000				[90]
Guo	Kagawa University	7.000	45.000		9.000	877.000	2565000.00256494.229				[102]
Nguyen	Konkuk University	0.032	0.032		1.400	1.000	12800.00814867.018				[290, 291]
Ryuh	Korea Institute of Industrial Technology	1.500	1.500				750000.000				[16]
Fujiwara	Kyushu University	0.580	0.580	34.483	16.000	0.030	200100.000727514.444				[148, 176, 292, 62]
Kumph	Massachusetts Institute of Technology	0.090	0.090	94.000	1.000	0.366	73800.00890873.776	15.000		0.580	[27]
Tan	Michigan State University	63.000	63.000		2.000		14490000.000				[293]
ValdiviaAlvarado	MIT	0.235	0.235	69.360	1.400		110450.000				[294–297, 18]
Herr	MIT	0.045	0.045	1.500	3.100		5400.000			0.400	[24]
Epps	MIT	0.125	0.100	32.710	3.500		14800.000				[21, 22]
Low	Nanyang Technological University	0.050	0.050	121.500	2.600	0.100	25000.00816814.090				[17]
Low	Nanyang Technological University	0.330	0.330	16.670	2.000		218130.000			0.100	[25]
Guo	National Taiwan University	0.900	0.900		0.800	0.100	60000.00817703.781	1.900			[89]
Verma	National University of Singapore	0.150	0.150		1.000		87000.000				[298]
Xu	National University of Singapore	0.500	0.500		2.000		300000.000				[299, 300]
Phamduy	New York University	0.137	0.137		2.000		63020.000				[301, 302, 205]
Liu	Northeastern University	75.000	75.000		0.270		1485000.000				[151]
Hu	Peking University	0.380	0.380		2.000		174800.000			0.360	[55]
Shao	Peking University	0.378	0.378				170100.000				[303–305]
Szymak	Polish Naval Academy	0.660	0.660	13.450	1.400	0.080	55400.00870079.094				[46, 189, 306, 19, 307]
Jatsun	SouthWest State University, Kursk, Russia	0.420	0.420	130.952	8.000		294000.000			0.450	[20]
Christianson	University of California, San Diego	19.000	19.000	10.500	0.330	0.050	80000.00010035.504				[308]
Ming	University of Electro-Communications	0.792	0.792		25.000		125136.000				[309]
Clapham	University of Essex	0.520	0.520	1.538	19.000	0.120	26000.00071628.313				[85]
Clapham	University of Essex	0.700	0.700		6.600	0.170	75000.00812318.507				[84]
Clapham	University of Essex	3.700	3.700		20.000	0.190	84000.00866760.632				[85]
Hu	University of Essex	0.530	0.530		1.900		275600.000	75.000			[310–314]
Majeed	University of Mustansiriyah										[315]
Hubbard	University of Nevada, Reno	0.028	0.028		2.000						[316]
Yan	University of Science and Technology of China	0.280	0.280		1.670	0.090	69400.000760452.828				[109]
Mohammadshahi	University of Tehran	0.700	0.700		3.500		420000.000				[317]
Vo	University of Ulsan	0.620	0.620	19.355	1.000		744000.000				[318]
Suleman	University of Victoria				0.500	0.056	351858.377				[106]
Morgansen	University of Washington	0.597	0.597		1.900	0.110	24051.60019326.909				[319, 100]
Yu	Chengdu University of Information	2.000	2.000		1.000	0.070	90000.00080925.542			0.600	[57]
Liu	Dalian University of Technology	467.000	467.000		4.700	1.296	20218000.00203110.817				[97]
Qian	East China University of Science and Technology	0.026	0.026					40.000			[320]
Berlinger	Harvard University	0.122	0.690	11.095	1.750	0.100	82800.00031667.254				[28]
Shibata	Kindai University	0.032	0.032		0.320	0.346	8320.00094055.162				[105]
Chan	Konkuk University	0.190	0.190		4.500	0.078	57760.000409718.532				[80]
Behbahani	Michigan State University	1.380	1.380		2.000		207000.000				[321]
Papadopoulos	National Technical University of Athens	0.157	0.157		3.125	0.107	52909.000475151.801				[103]
Wang	Peking University	0.400	0.400				160000.000				[322, 323]
Auerli	Polytechnic Institute of New York University	78.000	78.000		1.000		10140000.000				[324]
Kopman	Polytechnic Institute of New York University	0.080	0.080		2.000	0.150	13360.00004928.775				[94]
Lachat	Swiss Federal Institute of Technology	0.370	0.370		8.000	0.210	92500.00018023.782				[95]
Kim	Tohoku University	0.065	0.065		5.000	0.156	15662.724				[93]
Deng	University of California, Berkeley				5.000						[325]
Kodati	University of Delaware	0.041	0.041		0.800		6165.000				[326, 327]
Zhang	University of Science and Technology of China	0.538	0.538		2.600		236720.000				[328]
McGovern	University of Wollongong	0.033	0.033		0.750			0.150			[52]
Hou	Wichita State University	935.000	7.000	534.759	0.400		126000.000	2.000			[29]
Xie	Chinese University of Hong Kong	0.425	0.043		2.000	0.050	21999.86821744.327	91.000			[54]
Zhong	Chinese University of Hong Kong	0.670	0.670		3.000	0.160	207700.000579661.544				[329, 110]
Daou	Tallinn University of Technology	0.161	0.161		1.000	0.105	35420.00063862.295				[330, 86]
Rossi	Universidad Politecnica de Madrid	0.052	0.052		1.570	43.000	15600.00052011.216				[331, 104]
McColgan	University of Glasgow	0.146	0.024	162.884	1.000	0.150	21600.00026814.030			0.600	[36, 58]
Liang	Beihang University	1.360	1.100	81.818	2.000		1936000.000			1.750	[332, 31]
Rufo	Boston Engineering	2.570	2.570		1.000		3906400.000			1.000	[61]
Shen	Chinese Academy of Science	0.616	0.616		1.400		344960.000			0.224	[333–336]
Pik	Chinese University of Hong Kong	0.130	0.130		2.000		77760.000				[337]
Ertuk	Georgia Institute of Technology	0.075	0.075	18.670	5.000		18225.000				[32]
Chen	Harbin Institute of Technology	0.650	0.650		2.000		682500.000			1.000	[60]
Li	Harbin Institute of Technology	0.173	0.173		2.400	0.141	55360.00035451.885				[96]
Ye	Harbin Institute of Technology	0.030	0.030		4.000					0.078	[48]
Tomie	Kyushu Institute of Technology	0.020	0.020		3.000					0.040	[49]
Chen	Michigan State University	0.020	0.020		1.000	0.090	4460.00056045.951				[83]
Chen	Tongji University	0.174	0.174		30.000	0.047	52200.00098065.351	7.500			[82]
Wang	University of Auckland	15.000	15.000		1.000		1350000.000	0.000			[338]
Masoomi	University of Canterbury	0.290	0.290		1.500	0.080	403000.00072153.209				[339, 98]
Ming	University of Electro-Communications	0.325	0.225		23.000	0.145	24750.00008495.925				[99]
Zhu	University of Virginia	1.020	0.408	3.900	5.800	0.170	04040.000805688.454				[30, 340]

Table A.7 Table of kinematic and COT data for BCF Bio-inspired Robotic Platforms (Continued).

Author	Affiliation	Max Depth [m]	Endurance [hr]	Operating Voltage [V]	Battery Rating [Ah]	Hotel Power [W]	Mechanical Power [W]	Total Power [W]	Reference
Ding	Chinese Academy of Science		1.000						[87]
Yu	Chinese Academy of Science								[272]
Bayat	Ecole Polytechnique Federale de Lausanne								[273]
Crespi	Ecole Polytechnique Federale de Lausanne								[274]
Crespi	Ecole Polytechnique Federale de Lausanne				0.600	3.500	6.500	10.000	[275]
Raj	Indian Institute of Technology Patna								[276]
Nguyen	Japan Advanced Institute of Science and Technology								[101]
Niu	National University of Singapore								[59]
Kelasidi	NTNU								[91]
Leftwich	Princeton University								[277]
Struebzig	Technical University of Munich				48.000				[33]
Kanumichi	Tokyo Institute of Technology								[278]
Manfredi	University of Dundee		5.000	5.000		8.000	4.000	12.000	[279]
Joo	Airo Inc.	30.000	20.000	14.000	11.400				[26]
Joo	Airo Inc.	20.000	14.000	14.000	5.700				[26]
Joo	Airo Inc.	50.000	12.000						[26]
Yu	Chinese Academy of Science		1.000	7.400	3.400				[150]
Liao	Chinese Academy of Science								[280]
Yu	Chinese Academy of Science								[281]
Yu	Chinese Academy of Science								[23]
ZhengXing	Chinese Academy of Science								[282]
Shin	Chonnam National University								[56]
Morawski	Cracow University of Technology	30.000	6.000						[283]
Anderson	Draper Laboratory	10.000							[284]
Shintake	Ecole Polytechnique Federale de Lausanne						0.920	0.920	[53]
Bal	Firat University		0.500						[285]
Tan	Georgia Institute of Technology								[92]
Kim	Hankuk Aviation University								[81]
Chen	Harbin Institute of Technology								[107, 51]
Wang	Harbin Institute of Technology				0.620				[286, 108, 287]
Wen	Harvard and Beihang University								[88]
Giboudin	Institute de Physique de Nice								[288]
Romero	Instituto de Matematica y Estadística								[289]
Hirata	Japan Marine Science and Technology Center								[50]
Hirata	Japan Marine Science and Technology Center								[90]
Gao	Kagawa University								[102]
Nguyen	Konkuk University								[290, 291]
Ryuh	Korea Institute of Industrial Technology								[16]
Fujiwara	Kyushu University						20.400	20.400	[148, 176, 292, 62]
Kumph	Massachusetts Institute of Technology						8.500	8.500	[27]
Tan	Michigan State University								[293]
ValdiviaAlvarado	MIT	0.670		8.400	1.300				[294–297, 18]
Heir	MIT	4.000		2.800	0.096				[24]
Epps	MIT								[21, 22]
Low	Nanyang Technological University		8.000	18.000	2.700				[17]
Low	Nanyang Technological University		4.000	15.000	4.000				[25]
Gao	National Taiwan University								[89]
Verma	National University of Singapore								[298]
Xu	National University of Singapore								[299, 300]
Phanduy	New York University			7.400	2.200				[301, 302, 205]
Liu	Northeastern University								[151]
Hu	Peking University			4.800	2.500				[55]
Shao	Peking University								[305–305]
Szymak	Polish Naval Academy		3.000	11.100	2.400				[46, 189, 306, 19, 307]
Jatsun	SouthWest State University, Kursk, Russia				3.500				[20]
Christianson	University of California, San Diego								[308]
Ming	University of Electro-Communications								[309]
Clapham	University of Essex								[85]
Clapham	University of Essex								[84]
Clapham	University of Essex								[85]
Hu	University of Essex								[310–314]
Majeed	University of Mustansiriyah			7.400	2.100				[315]
Hubbard	University of Nevada, Reno								[316]
Yan	University of Science and Technology of China			6.000	2.500				[109]
Mohammadshahi	University of Tehran		0.250						[317]
Vo	University of Ulsan						10.000	10.000	[318]
Suleman	University of Victoria								[106]
Morgansen	University of Washington								[319, 100]
Yu	Chengdu University of Information								[57]
Liu	Dalian University of Technology								[97]
Qian	East China University of Science and Technology								[320]
Berlinger	Harvard University								[28]
Shibata	Kindai University								[105]
Chan	Konkuk University								[80]
Behbahani	Michigan State University								[321]
Papadopoulos	National Technical University of Athens			5.000	2.750				[103]
Wang	Peking University		5.000						[322, 323]
Auerdi	Polytechnic Institute of New York University		0.250						[324]
Kopman	Polytechnic Institute of New York University								[94]
Lachat	Swiss Federal Institute of Technology								[95]
Kim	Tohoku University								[93]
Deng	University of California, Berkeley								[325]
Kodati	University of Delaware								[326, 327]
Zhang	University of Science and Technology of China			11.100	2.000				[328]
McGovern	University of Wollongong								[52]
Hou	Wichita State University								[29]
Xie	Chinese University of Hong Kong			7.400	1.500				[54]
Zhong	Chinese University of Hong Kong						5.600	5.600	[329, 110]
Daou	Tallinn University of Technology								[330, 86]
Rossi	Universidad Politécnica de Madrid								[331, 104]
McColgan	University of Glasgow			12.000	2.600	3.900	0.010	3.910	[36, 58]
Liang	Beihang University		20.000			7.000	73.000	80.000	[332, 31]
Rufo	Boston Engineering	91.400	8.000						[61]
Shen	Chinese Academy of Science		2.000	7.400					[333–336]
Pik	Chinese University of Hong Kong								[337]
Ertuk	Georgia Institute of Technology						1.400	1.400	[32]
Chen	Harbin Institute of Technology								[60]
Li	Harbin Institute of Technology								[96]
Ye	Harbin Institute of Technology		0.250						[48]
Tomic	Kyushu Institute of Technology								[49]
Chen	Michigan State University								[83]
Chen	Tongji University								[82]
Wang	University of Auckland		11.000						[338]
Masoomi	University of Canterbury								[339, 98]
Ming	University of Electro-Communications								[99]
Zhu	University of Virginia		1.160						[30, 340]

A.4 Bio-inspired MPF AUV Data

Table A.8 Table of kinematic and COT data for MPF Bio-inspired Robotic Platforms.

Author	Affiliation	Locomotion	Actuator	No Actuators	No Linkages	Length [m]	Width [m]	Height [m]	Weight [kg]	Reference
Liu	Florida Atlantic University	Gymnotiform	Motor	16	16	0.462	0.077	0.125		[76, 341]
Veenstra	IT University of Copenhagen	Gymnotiform	Servo	6	6	0.272	0.060	0.136		[342]
Low	Nanyang Technological University	Gymnotiform	Servo	8	10	0.800	0.560	0.110	9.200	[127]
Shen	University of Nevada Las Vegas	Gymnotiform	IPMC	12	12	0.050	0.038	0.005		[126]
Sitorus	Bandung Institute of Technology	Labriform	Servo	4	2	0.375	0.164	0.110	2.500	[111]
Wang	Chinese Academy of Sciences	Labriform	Servo	7	7	0.600			3.220	[118]
Ahn Pham	Ho Chi Min City University of Technology	Labriform	Motor	2	3	0.400			1.059	[38]
Geder	Naval Research Laboratory	Labriform	Servo	12	10	0.330	0.178	0.033	1.160	[343]
Kato	Tokai University	Labriform	Servo	2	2	1.000	0.180	0.350	3.550	[344]
Kato	Tokai University	Labriform	Servo	2	2	1.000	0.180	0.350	15.740	[345]
Kato	Tokai University	Labriform	Motor	3	3	1.980	0.440	0.440	104.100	[346]
Kwak and Bae	UNIST	Labriform	Motor	1	2	0.095			0.023	[347]
Tang	Chinese Academy of Science	Oscillatory	Servo	3	3	1.323	0.884	0.381	51.900	[348, 349]
Simmons	Delft University of Technology	Oscillatory	Servo	34	34	0.883	0.386	0.100	70.000	[112]
Shi	Kogawa University	Oscillatory	SMA	4	4	0.060	0.065	0.043	3.800	[350]
Rahman	University Malaysia	Oscillatory	Servo	16	16	0.800	0.400	0.720	25.000	[351]
Susheelkumar	Arizona State University	Rajiform	Servo	4	12	0.345	0.762		4.550	[125]
Cai	Beihang University	Rajiform	Motor	3	3	0.500	0.670	0.060	4.000	[352, 353]
Cai	Beihang University	Rajiform	Motor	3	3	0.460	0.830	0.060	4.000	[352, 353]
Niu	Beihang University	Rajiform	Motor	2	2	0.400	0.620	0.055	3.200	[354, 117]
Xu	Beihang University	Rajiform	Motor	1	1	0.500	0.700	0.140	3.800	[354, 117]
Cai	Beijing University	Rajiform	Hydraulic	1	1	0.500	0.600	0.065	3.400	[114]
Cai	Beijing University	Rajiform	Hydraulic	1	1	0.700	0.980	0.065	7.000	[354, 117]
Gao	Beijing University	Rajiform	Servo	2	2	0.500	0.600	0.140	3.400	[116]
Gao	Beijing University	Rajiform	Servo	2	2	0.650	0.900	0.170	7.000	[355, 356]
Festo	Festo	Rajiform	Motor	3	3	0.615	0.960	0.145	10.000	[357]
Wang	Harbin Institute of Technology	Rajiform	SMA	4	4	0.133	0.220	0.066	0.354	[66]
Cloitre	MIT	Rajiform	Servo	2	2	0.650	0.610	0.140	6.100	[358]
Takagi	Nagoya University	Rajiform	IPMC	16	16	0.180	0.140	0.600	0.315	[119]
Low	Nanyang Technological University	Rajiform	Servo	6	6	0.370	0.800	0.170	5.000	[359, 63]
Zhou	Nanyang Technological University	Rajiform	Servo	6	6	0.500	1.000	0.200	7.300	[65]
Yang	National University of Defense Technology	Rajiform	Servo	8	8	0.300	0.500	0.020	1.000	[360]
Suzumori	Okayama University	Rajiform	Hydraulic	2	2	0.150	0.170	0.010	3.800	[120]
Anton	Tartu University	Rajiform	EAP	4	4	0.035	0.027	0.005	0.010	[361]
Punning	Tartu University	Rajiform	EAP	16	16	0.140	0.108	0.040	0.060	[121]
Shintake	University of Electro-communications	Rajiform	Piezoelectric	2	2	0.226	0.226	0.000	0.013	[39]
Zhao	University of Electro-communications	Rajiform	MFC	2	2	0.110	0.230	0.001		[122]
Love	University of Florida	Rajiform	Servo	2	2	0.710	0.910	0.150	6.000	[124]
Chen	University of Virginia	Rajiform	EAP	2	2	0.210	0.330	0.050	0.119	[123]
Chen	University of Virginia	Rajiform	IPMC	2	2	0.110	0.210	0.025	0.055	[37]
Li	Zhejiang University	Rajiform	DEA	2	2	0.093			0.043	[64]

Table A.9 Table of kinematic and COT data for MPF Bio-inspired Robotic Platforms (Continued).

Author	Affiliation	U _{max} [m/s]	U _{opt} [m/s]	COT _{opt} [J/m]	Freq [Hz]	A [BL]	Re	Sw	Yaw Speed [m/s]	Yaw Radius [m]	Reference
Liu	Florida Atlantic University	0.325	0.170	3.500	1.000		78540.000				[76, 341]
Veenstra	IT University of Copenhagen	0.080	0.030			0.137	8160.000				[342]
Low	Nanyang Technological University	0.200	0.200		0.880	0.144	60000.000	40598.569			[127]
Shen	University of Nevada Las Vegas	0.053	0.032		0.500	0.040	1600.000	18594.207			[126]
Sitorus	Bandung Institute of Technology	0.035	0.035		1.000	0.416	13162.500	43177.344			[111]
Wang	Chinese Academy of Sciences	0.115	0.115		6.400		69000.000				[118]
Ahn Pham	Ho Chi Min City University of Technology	0.231	0.150	0.614	2.000		60000.000		0.150	0.250	[38]
Geder	Naval Research Laboratory	0.240	0.240		1.800		79200.000		260.000		[343]
Kato	Tokai University	0.149	0.149		1.570		149000.000		2.200		[344]
Kato	Tokai University	0.130	0.130				130000.000				[345]
Kato	Tokai University	0.040	0.040		3.000		79200.000		0.160		[346]
Kwak and Bae	UNIST	0.117	0.117	5.641	1.300		11115.000				[347]
Tang	Chinese Academy of Science	0.330	0.330		2.000		436590.000		55.000		[348, 349]
Simmons	Delft University of Technology	0.330	0.330		1.000	0.095	91456.000	10774.092			[112]
Shi	Kogawa University	0.057	0.057		0.400		3432.000				[350]
Rahman	University Malaysia	0.193	0.193		1.500	0.127	54400.000	63515.231			[351]
Susheelkumar	Arizona State University	0.250	0.250			0.330	86250.000				[125]
Cai	Beihang University	0.400	0.400				200000.000				[352, 353]
Cai	Beihang University	0.300	0.300		0.500	0.380	38000.000	84185.330			[352, 353]
Niu	Beihang University	0.600	0.600		1.000		240000.000				[354, 117]
Xu	Beihang University	0.700	0.700		2.000	0.336	50000.000	784750.348		3.000	[354, 117]
Cai	Beijing University	0.700	0.700		1.000	0.110	85000.000	221168.123			[114]
Cai	Beijing University	0.900	0.900		1.200	0.189	30000.000	710892.152			[354, 117]
Gao	Beijing University	0.700	0.700		0.800	0.450	85000.000	30973.355			[116]
Gao	Beijing University	0.620	0.620		2.000		403000.000				[355, 356]
Festo	Festo	0.500	0.500				0.550	807500.000			[357]
Wang	Harbin Institute of Technology	0.057	0.057		1.280	0.429	7581.000	912515.961		0.118	[66]
Cloitre	MIT	0.180	0.180		1.400		117000.000				[358]
Takagi	Nagoya University	0.018	0.018		1.250	0.015	3240.000	4242.894			[119]
Low	Nanyang Technological University	0.300	0.300		1.500		111000.000			0.010	[359, 63]
Zhou	Nanyang Technological University	0.400	0.400		1.000		200000.000			0.100	[65]
Yang	National University of Defense Technology	0.130	0.130		1.000		39000.000				[360]
Suzumori	Okayama University	0.100	0.100		10.000	0.153	15000.000	16061.538			[120]
Anton	Tartu University	0.009	0.009		0.417		315.000				[361]
Punning	Tartu University	0.005	0.005		0.400	0.107	700.000	12101.415			[121]
Shintake	University of Electro-communications	0.250	0.240	3.097	15.000		54240.000				[39]
Zhao	University of Electro-communications	0.200	0.200		10.000	0.227	22000.000	559595.563			[122]
Love	University of Florida	0.500	0.500		0.500	0.352	55000.000	12999.132			[124]
Chen	University of Virginia	0.070	0.070	280.000	0.157	0.010	14700.000	246.649			[123]
Chen	University of Virginia	0.007	0.007	339.210	0.167	0.202	810.700	14097.021			[37]
Li	Zhejiang University	0.135	0.135	0.180	5.000		12555.000			0.085	[64]

Table A.10 Table of kinematic and COT data for MPF Bio-inspired Robotic Platforms (Continued).

Author	Affiliation	Max Depth [m]	Endurance [hr]	Operating Voltage [V]	Battery Rating [Ah]	Hotel Power [W]	Mechanical Power [W]	Total Power [W]	Reference
Liu	Florida Atlantic University								[76, 341]
Veenstra	IT University of Copenhagen								[342]
Low	Nanyang Technological University			7.500	3.000				[127]
Shen	University of Nevada Las Vegas								[126]
Sitorus	Bandung Institute of Technology								[111]
Wang	Chinese Academy of Sciences								[118]
Ahn Pham	Ho Chi Min City University of Technology						0.100	0.100	[38]
Geder	Naval Research Laboratory								[343]
Kato	Tokai University								[344]
Kato	Tokai University								[345]
Kato	Tokai University	10.000							[346]
Kwak and Bae	UNIST						0.660	0.660	[347]
Tang	Chinese Academy of Science								[348, 349]
Simmons	Delft University of Technology								[112]
Shi	Kogawa University								[350]
Rahman	University Malaysia								[351]
Susheelkumar	Arizona State University								[125]
Cai	Beihang University								[352, 353]
Cai	Beihang University								[352, 353]
Niu	Beihang University								[354, 117]
Xu	Beihang University								[354, 117]
Cai	Beijing University								[114]
Cai	Beijing University								[354, 117]
Gao	Beijing University								[116]
Gao	Beijing University								[355, 356]
Festo	Festo	0.500	0.500						[357]
Wang	Harbin Institute of Technology								[66]
Cloitre	MIT								[358]
Takagi	Nagoya University								[119]
Low	Nanyang Technological University								[359, 63]
Zhou	Nanyang Technological University								[65]
Yang	National University of Defense Technology								[360]
Suzumori	Okayama University								[120]
Anton	Tartu University								[361]
Puntning	Tartu University								[121]
Shintake	University of Electro-communications								[39]
Zhao	University of Electro-communications								[122]
Love	University of Florida								[124]
Chen	University of Virginia					2.000		2.000	[123]
Chen	University of Virginia								[37]
Li	Zhejiang University		3.250				0.024	0.024	[64]

A.5 Bio-inspired Lift Based AUV Data

Table A.11 Table of kinematic and COT data for Lift-Based Bio-inspired Robotic Platforms.

Author	Affiliation	Locomotion	Actuator	No Actuators	No Linkages	Length [m]	Width [m]	Height [m]	Weight [kg]	Reference	
Xing	Beijing Institute of Technology	LiftBased	Servo	12	12	0.300	0.300			[362]	
Hsu	Duke University	LiftBased	Motor	4	4	0.889	1.067	0.508	52	[357]	
Festo	Festo	LiftBased	Motor	2	2	0.770	0.190	0.190	9.600	[363–365]	
Independent Robotics	Independent Robotics	LiftBased	Motor	6	6	0.638	0.440	0.130	16.500	[366]	
Ravichandran	Indian Institute of Technology Madras	LiftBased	Servo	8	8	0.225	0.140	0.140	3.500	[367, 368]	
Konno	Kogakuin University	LiftBased	Servo	6	6	0.570	0	0.202	11.500	[369]	
Seo	Massachusetts Institute of Technology	LiftBased	Servo	8	8	2	0.500	0.500		[370]	
Kato	McGill University	LiftBased	Motor	6	6	0.650	0.450	0.130	16	[371]	
Naro	Naro	LiftBased	Motor	8	8	1	0		75	[128]	
Geder	Naval Research Laboratory	LiftBased	Servo	20	20	1.010	0		26.500	[372]	
Beal	Naval Undersea Warfare Center	LiftBased	Motor	4	4	1.700	0.760	0.280	90.720	[373, 67]	
Hobson	Nekton Research, LLC	LiftBased	Servo	4	4	1.010	0.530	0.530	105	[10]	
Kato	Osaka University	LiftBased	Motor	12	12	1.680	0.730	0.550	90	[374]	
Zhao	Peking University	LiftBased	Servo	8	8	0.200	0.200	0.175	5	[375, 370]	
Kim	Seoul National University	LiftBased	SMA	8	8	0.120	0.162	0.030	0.083	[376]	
Long	Vassar College	LiftBased	Motor	4	4	0.780	0.440	0.130	24.400	[377]	
Author	Affiliation	Umax [m/s]	Uopt [m/s]	COTopt [J/m]	Freq [Hz]	A [BL]	Re	Sw	Yaw Speed [m/s]	Yaw Radius [m]	Reference
Xing	Beijing Institute of Technology	0.035	0.035		0.500		10500		11.500		[362]
Hsu	Duke University	0.200	0.200		3.500		177800				[357]
Festo	Festo	1.389	1.389				1069530				[363–365]
Independent Robotics	Independent Robotics	1	1				638000			30.480	[366]
Ravichandran	Indian Institute of Technology Madras	0.050	0.050				11250				[367, 368]
Konno	Kogakuin University	0.060	0.048		0.350		27291.600				[369]
Seo	Massachusetts Institute of Technology	2	2				4000000				[370]
Kato	McGill University	1	1				650000			30	[371]
Naro	Naro	2	2	115.200	2		2000000				[128]
Geder	Naval Research Laboratory	1.200	1.200		2.500		1212000			37	[372]
Beal	Naval Undersea Warfare Center	1.200	1.200		1		2040000				[373, 67]
Hobson	Nekton Research, LLC	0.600	0.600		6.400		606000			100	[10]
Kato	Osaka University	0.168	0.168				282240				[374]
Zhao	Peking University	0.198	0.198				39600				[375, 370]
Kim	Seoul National University	0.022	0.022		1.110	0.458	2700	92060.562			[376]
Long	Vassar College	0.740	0.740	83.777	6	0.029	577200	1321121.644			[377]
Author	Affiliation	Max Depth [m]	Endurance [hr]	Operating Voltage [V]	Battery Rating [Ah]	Hotel Power [W]	Mechanical Power [W]	Reference			
Xing	Beijing Institute of Technology							[362]			
Hsu	Duke University	12.190						[357]			
Festo	Festo		7					[363–365]			
Independent Robotics	Independent Robotics	5						[366]			
Ravichandran	Indian Institute of Technology Madras							[367, 368]			
Konno	Kogakuin University							[369]			
Seo	Massachusetts Institute of Technology							[370]			
Kato	McGill University	5						[371]			
Naro	Naro	100						[128]			
Geder	Naval Research Laboratory							[372]			
Beal	Naval Undersea Warfare Center		12					[373, 67]			
Hobson	Nekton Research, LLC	100						[10]			
Kato	Osaka University							[374]			
Zhao	Peking University							[375, 370]			
Kim	Seoul National University							[376]			
Long	Vassar College						58.300	[377]			

A.6 Bio-inspired Jet Based AUV Data

Table A.12 Table of kinematic and COT data for Jet Bio-inspired Robotic Platforms.

Author	Affiliation	Locomotion	Actuator	No Actuators	No Linkages	Compliant	Length [m]	Width [m]	Height [m]	Weight [kg]	Reference
Christianson	University of California, San Diego	Jet	DEA	8	8	1	0.163			0.230	[378]
Cianchetti	The BioRobotics Institute	Jet	SMA	8	8	1	0.300			3.000	[379]
Paschal	Swiss national Center	Jet	Servo	9	8	1	0.600	0.500		2.100	[380]
Sfakiotakis	Foundation for Research and Technology-Hellas	Jet	Servo	8	8	1	0.380			2.680	[381]
Villanueva	Virginia Tech	Jet	BISMAC	8	8	1	0.600			0.242	[382]
Villanueva	Virginia Tech	Jet	Linear	8	8	0	0.516			76.000	[383]
Author	Affiliation	Umax [m/s]	Uopt [m/s]	COTopt [J/m]	Freq [Hz]	A [BL]	Re	Sw	Yaw Speed [m/s]	Yaw Radius [m]	Reference
Christianson	University of California, San Diego	0.003	0.003	78.000	0.200		521.600				[378]
Cianchetti	The BioRobotics Institute	0.050	0.050	53.000	0.750		15000.000				[379]
Paschal	Swiss national Center	0.100	0.100	45.000	2.000		60000.000				[380]
Sfakiotakis	Foundation for Research and Technology-Hellas	0.099	0.099	38.844	0.900		37468.000				[381]
Villanueva	Virginia Tech	0.054	0.054	313.650	0.300	0.050	32520.000				[382]
Villanueva	Virginia Tech	0.085	0.085	826.450	0.120		26765.200				[383]
Author	Affiliation	Max Depth [m]	Endurance [hr]	Operating Voltage [V]	Battery Rating [Ah]	Hotel Power [W]	Mechanical Power [W]	Reference			
Christianson	University of California, San Diego						0.250				[378]
Cianchetti	The BioRobotics Institute						2.600				[379]
Paschal	Swiss national Center						4.510				[380]
Sfakiotakis	Foundation for Research and Technology-Hellas						3.830				[381]
Villanueva	Virginia Tech						17.000				[382]
Villanueva	Virginia Tech						70.000				[383]

A.7 Conventional AUV Propeller Number Data

Manufacturer	Model	Length	Diameter	Number Thrusters	Number Blades	Velocity [m/s]	Shaft RPM [rpm]	Omega [rps]	Prop Diameter [mm]	Prop Diameter [m]	Re	J	Jw
Australian Maritime College	AMC Explorer	6.60	0.74	1	2	1.00	123.60	2.06	650	0.65	6610375.20	0.75	8837400.00
Australian Maritime College	AMC Explorer	6.60	0.74	1	2	1.02	127.08	2.12	650	0.65	6714716.58	0.74	9086220.00
Australian Maritime College	AMC Explorer	6.60	0.74	1	2	1.26	156.00	2.60	650	0.65	8320884.00	0.75	11154000.00
Australian Maritime College	AMC Explorer	6.60	0.74	1	2	1.48	186.36	3.11	650	0.65	9753709.68	0.73	13324740.00
Australian Maritime College	AMC Explorer	6.60	0.74	1	2	1.86	229.98	3.83	650	0.65	1229790.36	0.75	16443570.00
National Oceanography Center	Autosub3	6.80	0.90	1	2	1.00	149.88	2.50	700	0.70	6800000.00	0.57	11890190.59
National Oceanography Center	Autosub3	6.80	0.90	1	2	1.50	223.56	3.73	700	0.70	10200000.00	0.58	17736045.91
National Oceanography Center	Autosub3	6.80	0.90	1	2	2.00	294.00	4.90	700	0.70	13600000.00	0.58	23323615.16
Kongsberg Maritime	HUGIN 4500	6.50	1.00	1	3	1.73	155.14	2.59	700	0.70	11245000.00	0.96	1765061.95
Kongsberg Maritime	HUGIN 4500	6.50	1.00	1	3	1.87	165.12	2.75	700	0.70	12155000.00	0.97	12521806.26
Kongsberg Maritime	HUGIN 4500	6.50	1.00	1	3	2.00	175.06	2.92	700	0.70	13000000.00	0.98	13275662.19
Maritime and Ocean Engineering Research Institute	ISI-MI	1.50	0.20	1	5	0.41	323.68	5.39	93	0.09	614449.06	0.82	752550.38
Maritime and Ocean Engineering Research Institute	ISI-MI	1.50	0.20	1	5	0.80	648.61	10.81	93	0.09	1203242.05	0.80	1508028.97
Maritime and Ocean Engineering Research Institute	ISI-MI	1.50	0.20	1	5	0.98	812.34	13.54	93	0.09	1477479.20	0.78	1888696.47
Maritime and Ocean Engineering Research Institute	ISI-MI	1.50	0.20	1	5	1.18	973.55	16.23	93	0.09	1766383.56	0.78	2263507.56
International Submarine Engineering	MUN Explorer	4.50	0.70	1	2	1.00	115.00	1.92	650	0.65	4500000.00	0.80	5606250.00
International Submarine Engineering	MUN Explorer	4.50	0.70	1	2	1.50	173.75	2.90	650	0.65	6750000.00	0.80	8470312.50
International Submarine Engineering	MUN Explorer	4.50	0.70	1	2	2.00	217.00	3.62	650	0.65	9000000.00	0.85	10578750.00
Defence Science and Technology Organization	Wayamba	4.30	1.50	2	4	2.50	275.00	4.58	650	0.65	11250000.00	0.84	13406250.00
Defence Science and Technology Organization	Wayamba	4.30	1.50	2	4	0.49	394.83	6.58	560	0.56	2099233.78	0.13	15845726.04
Defence Science and Technology Organization	Wayamba	4.30	1.50	2	4	0.77	601.06	10.02	560	0.56	3308881.94	0.14	24122429.60
Defence Science and Technology Organization	Wayamba	4.30	1.50	2	4	0.81	598.67	9.98	560	0.56	3468913.21	0.14	24026456.95
Defence Science and Technology Organization	Wayamba	4.30	1.50	2	4	1.06	804.81	13.41	560	0.56	4546770.91	0.14	32299878.06
Defence Science and Technology Organization	Wayamba	4.30	1.50	2	4	1.07	799.88	13.33	560	0.56	4617372.95	0.14	32101719.54
Defence Science and Technology Organization	Wayamba	4.30	1.50	2	4	1.31	994.77	16.58	560	0.56	5624628.62	0.14	39923569.00

A.8 AUV Applications Data

Table A.13 Legend for AUV applications. Application classification and Data for conventional AUVs adapted from Haroutunian [142].

AUV Mission	Abbreviation
Anti-Submarine Warfare	ASW
Beach Survey	BS
Cable Deployment	CD
Cable Route Survey	CRS
Coastal Mapping	CM
Controls Research	CR
Entertainment	ET
Environmental Monitoring	EM
Explosive Ordnance Disposal	EOD
Force Protection	FP
Freshwater Mapping	FM
Geophysical Survey	GS
Harbor and Port Security	HPS
Hull Inspection	HI
Hydro-acoustic Research	HAR
Inspectioning Maintenance and Repair	IMR
Intelligence, Surveillance, Reconnaissance	ISR
Marine Science Survey	MSS
Mine Countermeasures	MC
Mineral Field Survey	MFS
Oceanographic Survey	OS
Oil and Gas Survey	OGS
Pipeline Route Survey	PRS
Rapid Environmental Assessment	REA
Scientific Research	SR
Seabed Mapping	SM
Search and Recovery	SR
Search, Classify, and Map	SCM
Sensor Developement	SD
Surf Zone Surveillance	SZS
Vehicle Research	VR
Wind Park Construction Survey	WCS
Mobility	MOB
Efficient Propulsion	EP
Performance	PER

Table A.14 Applications for conventional AUVs. Data from Haroutunian [142] and AU-VAC.org [10].

Manufacturer	Name	ASW	BS	CD	CM	CRS	EM	EOD	FM	FP	GS	HAR	HI	HPS	IMR	ISR	MC	MFS	MSS	OGS	OS	PRS	REA	SR	SCM	SD	SM	SR.1	SZS	VR	WCS
	Number AUVs performing mission	2	1	0.00	9	5	10	2	4	1	16	3	3	4	6	4	3	1	9	7	14	6	3	8	5	0.00	11	2	0.00	5	0.00
	Percent AUVs performing mission	6.67	3.33	0.00	30	16.67	33.33	6.67	13.33	3.33	53.33	10	10	13.33	20	13.33	10	3.33	30	23.33	46.67	20	10	26.67	16.67	0.00	36.67	6.67	0.00	16.67	0.00
ECA SA	Alistar					CRS					GS				IMR					OGS		PRS									
ECA SA	Alister Daurade					CRS					GS				IMR					OGS		PRS									
Cybernetix	ALIVE														IMR																
Woods Hole Oceanographic Institution	Autonomous Benthic Explorer (ABE)										GS																SM				
Woods Hole Oceanographic Institution	Autonomous Benthic Explorer (ABE)										GS																SM				
National Oceanography Center	Autosub 3										GS	EN	MSS	IS	VR	SM															
National Oceanography Center	Autosub6000										GS	MSS	OS	SM	VR																
Bluefin Robotics	Bluefin 21	ASW					EM	EOD			GS					ISR	MC		MSS		OS					SCM	SM			VR	
Bluefin Robotics	BPAUV																														
Monterey Bay Aquarium Research Institute	Dorado				CM						GS	OS	SCM	SM	MSS	SR															
International Submarine Engineering Ltd.	Explorer																														
Kongsberg Maritime	HUGIN 1000				CM	CRS	EM				GS									MSS	OGS	OC	PRS		SR	SCM					
Kongsberg Maritime	HUGIN 3000				CM	CRS	EM				GS									MSS	OGS	OC	PRS		SR	SCM					
Kongsberg Maritime	HUGIN 4500				CM	CRS	EM				GS									MSS	OGS	OC	PRS		SR	SCM					
Maritime and Ocean Engineering Research Institute	ISIMI																														VR
OceanServer Technology	Iver2				CM		EM		FM														REA								
Memorial University	MUN Explorer				CM			MSS	SM	OS	VR	GS																			
Woods Hole Oceanographic Institution	Nereus																		MSS		OS			SR			SM			VR	
Office of Naval Research	Odyssey																														VR
MIT AUV Laboratory	Odyssey IV																														VR
Hydroid	Remus 100				CM				FM											MSS		OS								SM	
Hydroid	Remus 600		BS		CM		EM		FM		GS													SR					SM	SCM	
Hydroid	Remus 5000				CM		EM				GS									MFS	MSS	OGS	OS	PRS	SR			SM	SCM		
Woods Hole Oceanographic Institution	SeaBED										GS																				SM
Hydroid	Seaglider													HPS		ISR															
iRobot	Seaglider																														
Adas Elektronik	Seastar M&H	ASW																		MC											
Teledyne Marine	Slocum Electric – 1km						EM																								SM
Teledyne Marine	Slocum Electric – Coastal						EM																								
Scripps Institution of Oceanography	Spray						EM																								
International Submarine Engineering Ltd.	Theseus																			MSS		OS									SM

Table A.15 Applications for BCF AUVs.

Name	Author	Affiliation	Locomotion	CR	EM	EP	ET	ISR	MOB	MSS	PER	SM	SR	VR
		Number AUVs performing mission		4	1	2	2	1	2	2	1	1	6	18
		Percent AUVs performing mission		20	5	10	10	5	10	10	5	5	30	90
Polish Naval Academy Cyberfish	Szymak	Polish Naval Academy	Carangiform				EP	ISR	MOB					VR
Miro-9	Joo	Airo Inc.	Carangiform											
Miro-7	Joo	Airo Inc.	Carangiform					ET						
MAR	Struebig	Technical University of Munich	Anguilliform											VR
UV Tunabot	Zhu	University of Virginia	Thunniform				EP							SR VR
SouthWest State Carangiform	Jatsun	SouthWest State University, Kursk, Russia	Carangiform											SR VR
IMSat Artefact	Manfredi	University of Dundee	Carangiform											SR VR
Beihang SPC-III	Liang	Beihang University	Thunniform											VR
MIT RoboPike	Kumph	Massachusetts Institute of Technology	Carangiform											VR
University of Glasgow RoboSalmon	McColgan	University of Glasgow	Sub-Carangiform	CR										SR VR
MIT Pneumatic SoFi	Katzschmann	MIT	Carangiform		EM					MSS				VR
Nanyang Arowana	Low	Nanyang Technological University	Carangiform											VR
CAS Robotic Shark	Yu	Chinese Academy of Science	Carangiform	CR										VR
NTNU Mamba	Kelasidi	NTNU	Anguilliform											VR
Harvard Finbot	Berlinger	Harvard University	Ostraciiform	CR					MOB					SR VR
Kyushu University Carangiform	Fujiwara	Kyushu University	Carangiform							MSS		SM		VR
Harvard Live Muscle Fish	Herr	MIT	Carangiform											VR
MIT Carangiform	Epps	MIT	Carangiform										SR	
Nanyang NAF-1	Low	Nanyang Technological University	Carangiform	CR										VR
Wichita State IPMC	Hou	Wichita State University	Ostraciiform											VR
University of Essex iSplash Micro	Clapham	University of Essex	Carangiform								PER			VR

Table A.16 Applications for MPF AUVs.

Name	Author	Affiliation	Locomotion	CR	VR
		Number AUVs performing mission		1	3
		Percent AUVs performing mission		33.33	100
Ho Chi Minh Labriform	Ahn Pham	Ho Chi Min City University of Technology	Labriform	CR	VR
University of EC PFC	Shintake	University of Electro-communications	Rajiform		VR
UV Robotic Mantaray	Chen	University of Virginia	Rajiform		VR

Table A.17 Applications for Lift Based AUVs.

	Author	Affiliation	Locomotion	VR
		Number AUVs performing mission		2
		Percent AUVs performing mission		100
Naro-Tartaruga	Naro	Naro	LiftBased	VR
Madeline	Long	Vassar College	LiftBased	VR

Appendix B

The Ika-Fit Method

B.0.1 Ika-Fit Method

This research differs from previous techniques, described in Section 3.1, in that the curvature of the specimen's body is fit to a 6 degree polynomial, a National Advisory Committee for Aeronautics (NACA) airfoil, and the top and bottom of the animal is treated as separate ellipse partition discs. This formulation allows for a more accurate estimation of the physical parameters, discussed in detail in Section 3.2. This is in contrast to existing methods that treat the fish-like body as a prolate spheroid where the surface area can be readily calculated or as an ellipse disc based on the height and width of each partition.

Side and top images of fish were obtained from the digital fish library project [384]. Image artifacts were manually removed from the black background and two copies of the side view, one with fins and one with fins removed, provided only body geometry, as shown in Fig. B.1. For validation purposes, a king salmon (*Oncorhynchus tshawytscha*) was laser scanned into a mesh using a Kreon Ace-7-30 laser measurement arm (KREON Technologies, Limoges Cedex, France). This scanned data was imported into the open-source software Blender where it was scaled in the interval [0.1:100], and the surface area and volume were readily evaluated. Figure B.1 shows the images used for the scanned salmon, but all images used in this research follow the same format.

Image Segmentation and Contours

The Python OpenCV package [385] was used to process the images and determine body geometry contours. The image segmentation algorithm used was an adapted version of the automatic image segmentation algorithm given in Siswanto *et al.* [195], and an overview of the adapted algorithm is shown in Fig. B.2.

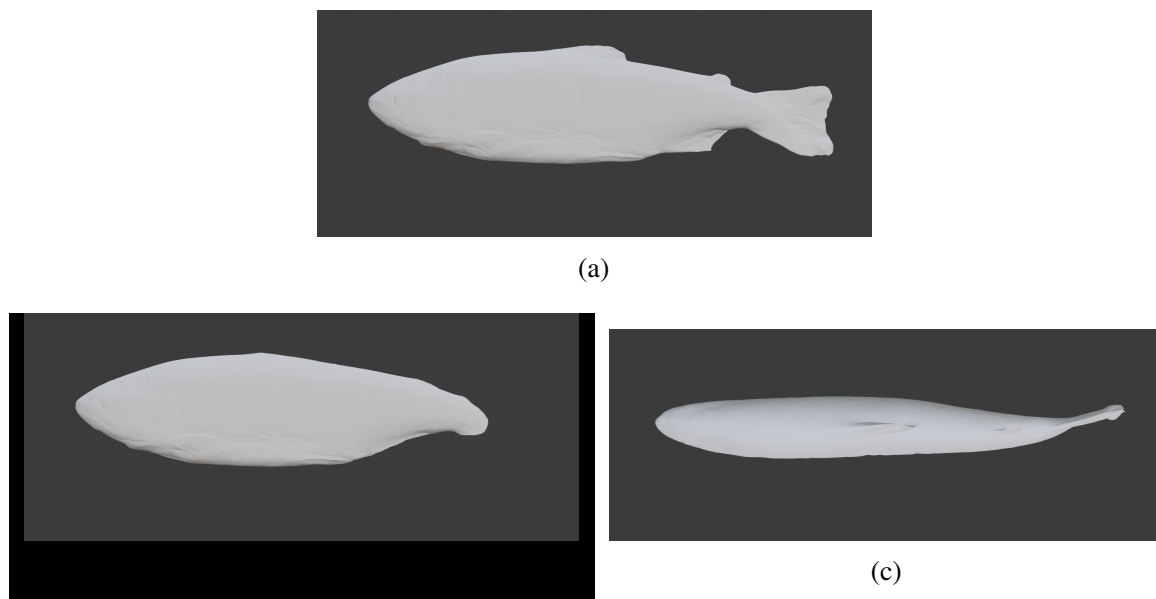


Fig. B.1 Images used in the validation of the Ika-Fit method with salmon modeled in Blender. (a) Salmon with caudal fin, (b) Salmon with no caudal fin, and (c) Salmon top with fins.

When an image is read into the program, it is read as a matrix with dimensions (H, W, C) , where H and W are the height and width of the image and C represents a vector of three color channels (blue, green, red) as in Figure B.3. The image is converted from RGB (red, green, blue) to HSV (hue, saturation, value), where a Gaussian blur is applied to the S and V components before merging the channels back to an image shown in Fig. B.3b and B.3c. K-means color quantization is performed to threshold the image into two colors, as in Figure B.3d. Morphological opening and closing on the image is used to fill any gaps in the fish shown in Figure B.3e. Canny edge detection is used to determine the edges where the fish foreground meets the background as in Fig. B.3f.

Shape extraction is used on the canny edge detection image using the find contours method in [385], which implements the algorithm presented in Suzuki *et al.* [386]. The algorithm works on a binary image with the background being black pixels (255) and the foreground being white pixels (0). The first foreground border pixel is found and the surrounding pixels are checked for being black or white. In the case of multiple white pixels, the outside pixel is chosen and the window is moved to that pixel. The contours represent (x, y) coordinates of continuously connected white pixels. The final segmented image is shown in Figure B.3g, where the contour is shown over the image after morphological operations had been performed.

The centroid and extremum points of the contour were used as a dividing line to divide the specimen into a top half and bottom half, as shown in Fig. B.4. The figure shows the

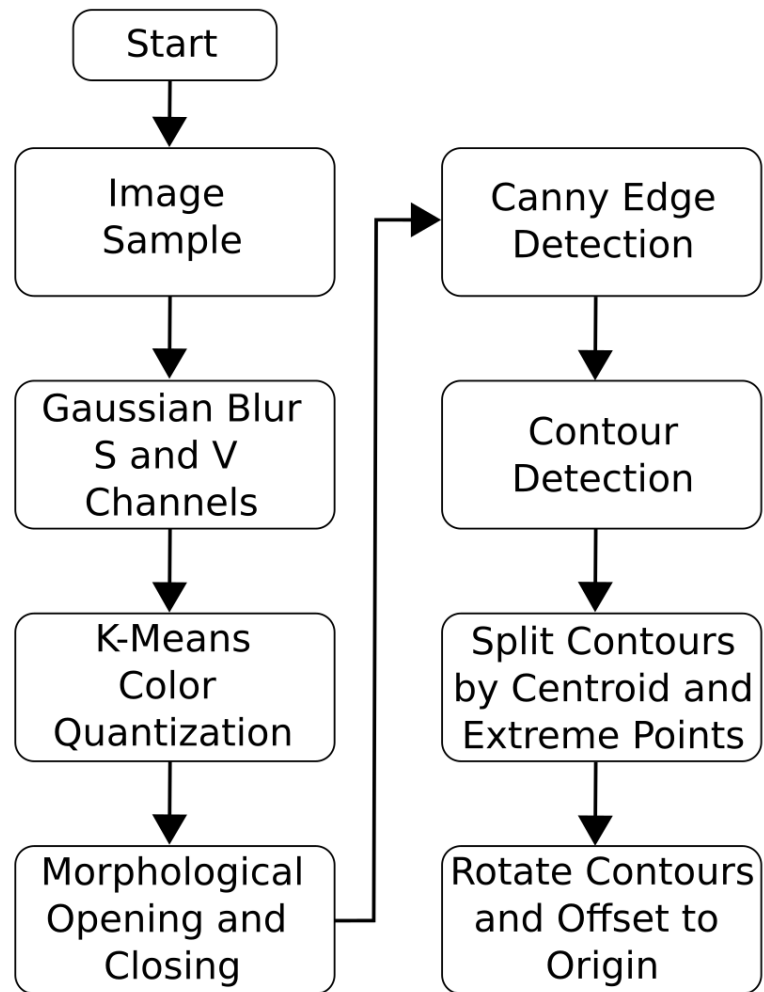


Fig. B.2 The steps taken for image segmentation.



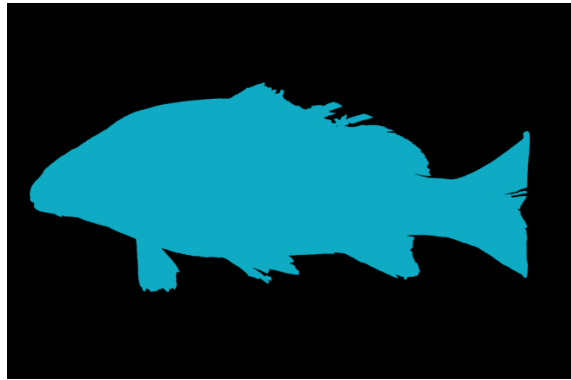
(a)



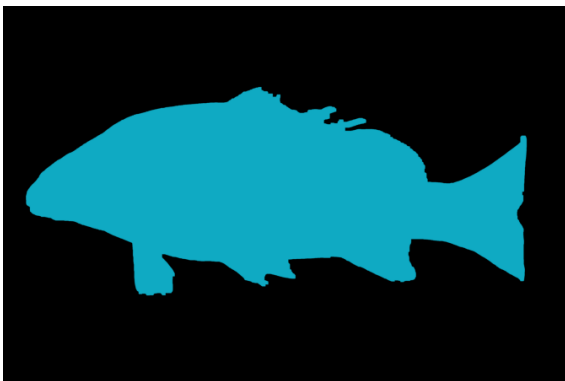
(b)



(c)



(d)

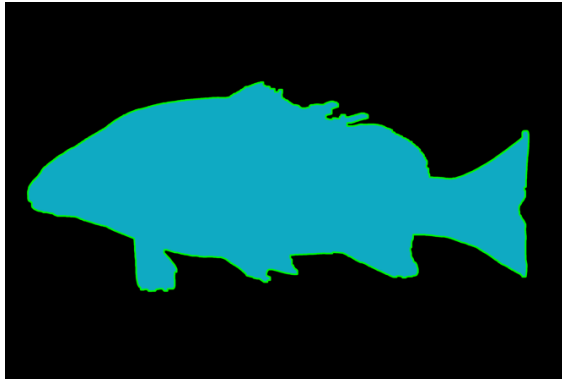


(e)



(f)

Fig. B.3 *Cont.*



(g)

Fig. B.3 Images showing the algorithm on a complex fish specimen. (a) Initial image, (b) S and (c) V components, (d) K-means color quantized image, (e) image after morphological opening and closing, (f) canny edge-detected Image, and (g) contours overlayed over image (e).

resulting contour in white around the original image. The blue line represents a line that passes through the centroid of the specimen to the x-coordinates of the extreme left and right points called the mid plane. The red line represents the line used to divide the specimen along its extreme left and right points. This shows the difference between partitioning the specimen using the midline versus the dividing line used in this research. The contours above and below the divide line are separated into the top and bottom hull, respectively. This separation of the contour into two hulls allows the algorithm to treat the top and bottom of the specimen as different ellipsoid discs.

Figure B.3 shows how the proposed method of image segmentation can successfully resolve the complex fin features of a fish, which is crucial when determining the surface area contribution of the fins. The outlined method above was used to normalize data acquisition for specimens with widely varied geometry. In all, 50 different specimens were used to assess the robustness of the image segmentation method.

NACA Airfoil Fit

This image segmentation process described in Section B.0.1 was repeated for the top view to get the complete shape of the specimen. The top view of the fish and bio-robotic platforms resembles a symmetric NACA airfoil about the center line, as shown in Figure B.5b.

The approach of fitting the top to a NACA airfoil provides three useful benefits over fitting the top view to a polynomial. The first is that any asymmetry in the image is converted into a symmetrical shape about the center line of the animal. The second benefit is that

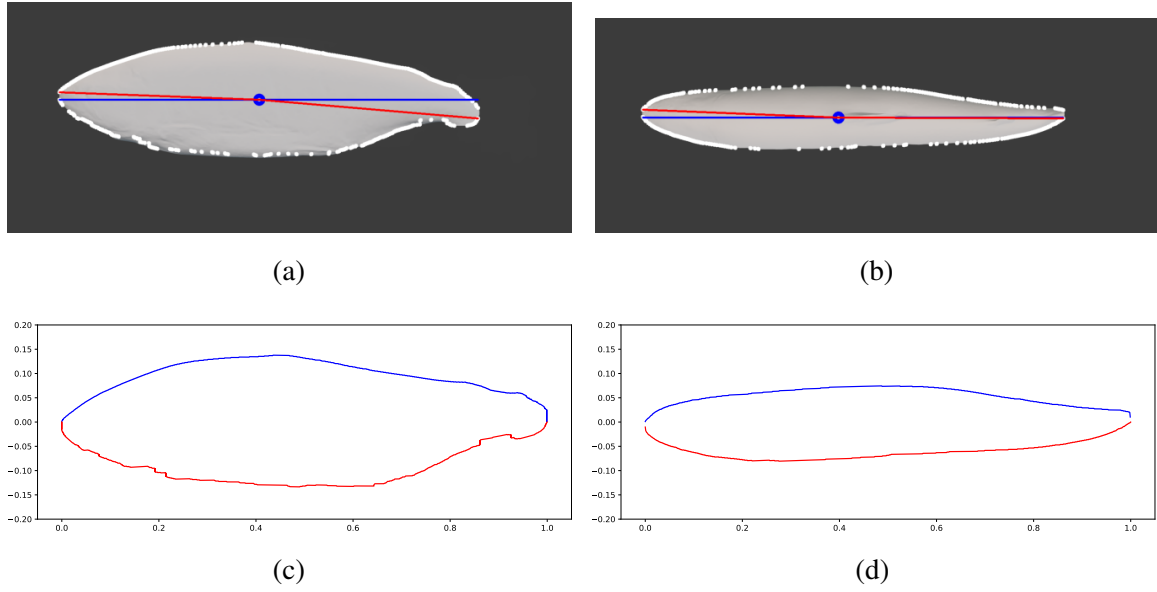


Fig. B.4 **(a,c)** Side and top view with the overlaid centroid (blue dot), line slicing through the middle of the specimen (blue line), dividing line (red line) to extrema points, and white line representing the contours. **(b,d)** Plotted contour after dividing the top and bottom of the specimen and zeroing the resulting contours. Image obtained from photographs of salmon purchased from local fishery.

the computational requirements for these calculations are less because calculations are only required for one half of the animal. Finally, if a certain animal or artificial swimmer is desired for computational fluid dynamic simulations, a 2D symmetric profile is readily available.

NACA airfoils are given by an equation that is a function of chord length (the length of the foil from leading edge to trailing edge) and a percentage of the thickness of the chord. The default position of maximum thickness for a symmetric airfoil is at 30% of the chord length, but fish have varying locations of maximum thicknesses; therefore, a modified symmetrical NACA foil was used where the location of maximum thickness could be varied.

Normally, a symmetric NACA foil is represented by one equation, but the modified version is broken into a leading edge and trailing edge equation. For reference, a schematic is provided in Fig. B.6 and the equations used for the modified NACA airfoil are given by Charles *et al.* and Abbot in [143, 387]:

$$\bar{y}_{leading} = a_0\sqrt{\bar{x}_L} + a_1\bar{x}_L + a_2\bar{x}_L^2 + a_3\bar{x}_L^3, \quad (\text{B.1})$$

$$\bar{y}_{trailing} = d_0 + d_1(1 - \bar{x}_T) + d_2(1 - \bar{x}_T)^2 + d_3(1 - \bar{x}_T)^3, \quad (\text{B.2})$$

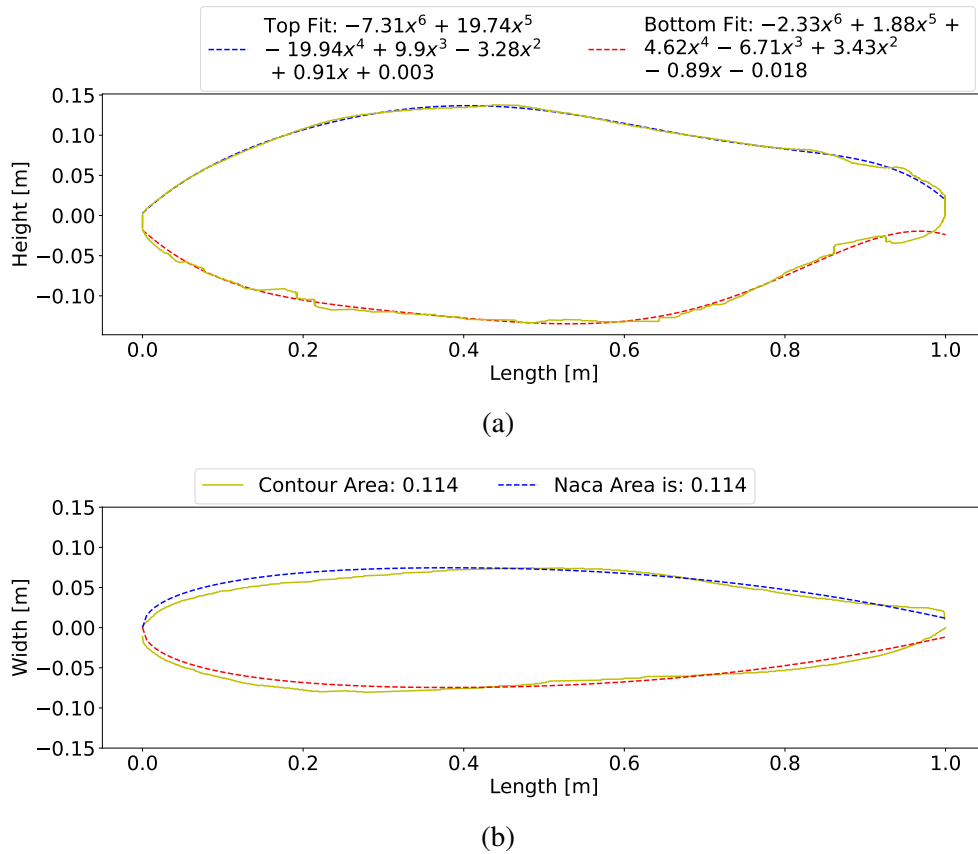


Fig. B.5 Contour lines (side and top views) of scanned salmon; the yellow line is the real contour line, solid (blue and red) lines are computed lines for side contour, and the blue dashed line is the computed line for top views. **(a)** King salmon side contour with fit equations shown in the legend. **(b)** King salmon top view contour with NACA fit overlaid. Area output from the area matching algorithm is shown.

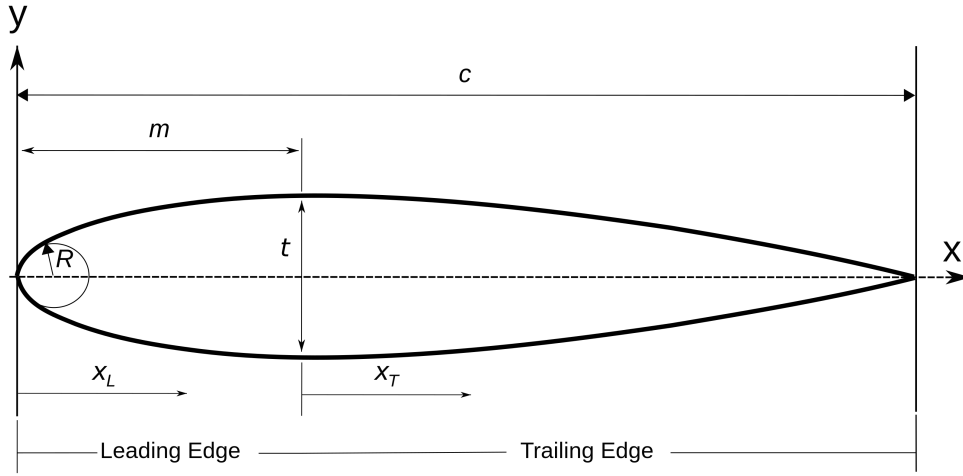


Fig. B.6 Schematic view of a symmetric airfoil. c is the chord length, t is the maximum thickness as a percentage of chord length, and m is the x -position where maximum thickness occurs. x_L represents the x -coordinate of the Leading Edge and x_T is the coordinates of the Trailing Edge.

where \bar{y} is the thickness over chord length $\left(\frac{y}{c}\right)$ and \bar{x}_L and \bar{x}_T is the x -position over chord length $\left(\frac{x}{c}\right)$ for the leading and trailing edge, respectively. The foil is a piece-wise combination of these two equations with the break point being the x -position of maximum thickness, m . In practice, the position is kept in tenths of chord length between 0.2 and 0.6. A survey of 33 fish contours was performed, and all fit within this interval, with the minimum value of 0.21 belonging to the ocean sunfish (*Mola mola*) and the maximum value of 0.48 belonging to the bowfin (*Amia calva*). In order to solve for the coefficients, the equations are subject to the following boundary conditions provided by Charles *et al.* and Abbot [143, 387]:

For a maximum ordinate at $x = m$:

$$\bar{y} = \frac{t}{2c}, \quad \left. \frac{d\bar{y}}{dx} \right|_{\bar{x}_L} = \left. \frac{d\bar{y}}{dx} \right|_{\bar{x}_T} = 0, \quad \left. \frac{d^2\bar{y}}{dx^2} \right|_{\bar{x}_L} = \left. \frac{d^2\bar{y}}{dx^2} \right|_{\bar{x}_T} \quad (\text{B.3})$$

For a trailing edge ordinate at $(\bar{x}_T = 1.0)$:

$$\bar{y} = k, \quad (\text{B.4})$$

where k is half the size of the tail width given by the top view contour of the fish. This value defines the coefficient d_0 .

The leading edge radius at $\bar{x}_L = 0$ is provided by Abbot [143]:

$$R = \frac{a_0^2}{2} = 1.1019 \left(\frac{tI}{6} \right)^2, \quad (\text{B.5})$$

where I is an index number that determines the sharpness of the airfoil nose, 0 being sharp and 6 being a default value where the leading edge is round. This was kept to 6 for the algorithm in this research. The values of d_1 are chosen such that reversals in curvature are avoided using Table B.1 [143, 387]:

Table B.1 Experimental values to define the coefficient d_1 as a function of m from Abbot [143].

m	d_1
0.2	$1.000t$
0.3	$1.170t$
0.4	$1.575t$
0.5	$3.325t$
0.6	$3.500t$

Table B.1 values are fit with a 3rd order polynomial so that a smooth interpolated value can be obtained for any max thickness, m , position. Coefficient a_3 is dependent on d_2 and d_3 ; therefore, $\bar{y}_{trailing}$ is solved before $\bar{y}_{leading}$. The coefficients are formed as a system of equations. d_0 and d_1 are given by the boundary conditions above, and d_2, d_3 are solved by:

$$\begin{bmatrix} (1-m)^2 & (1-m)^3 \\ -2(1-m) & -3(1-m)^2 \end{bmatrix} \begin{bmatrix} d_2 \\ d_3 \end{bmatrix} = \begin{bmatrix} \frac{t}{2} - d_1(1-m) - d_0 \\ d_1 \end{bmatrix} \quad (\text{B.6})$$

Coefficients a_0 are given by the boundary condition, and a_1, a_2, a_3 are solved by:

$$\begin{bmatrix} m & m^2 & m^3 \\ 1 & 2m & 3m^2 \\ 0 & 2 & 6m \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \frac{t}{2} - a_0 m^{\frac{1}{2}} \\ -\frac{a_0 m^{-\frac{1}{2}}}{2} \\ 2d_2 + 6d_3(1-m) + \frac{1}{4}a_0 m^{-\frac{3}{2}} \end{bmatrix} \quad (\text{B.7})$$

Due to the irregular shape of the contour, a recursive function that minimizes the difference in area between the NACA foil and the contour is used to fit the thickness parameter. The function changes the airfoil thickness until a difference between the areas is within 0.001 of each other. Area is calculated using the Simpson integration procedure available through the SciPy package [251]. Figure B.5 shows both the side and top contour and associated curve fits.

B.0.2 Ellipsoid Approximation

Three methods from the literature and the Ika-Fit method use different ellipsoid approximations to determine the surface area and volume of a specimen. The key difference between all these methods is in how the ellipse parameters are determined. Table B.2 lists and briefly describes the methods and their respective references. Murphy and Haroutunian and Phillips *et al.* use the same formulation of an equivalent diameter for a prolate ellipsoid as a function of length and mass [142, 201, 3]:

Table B.2 Abbreviation of methods, description of how the methods are implemented, and references.

Method	Description	Reference
Ika-Fit (IF)	Developed method using computer vision.	This paper
PDR	Ellipsoid partition disc method using computer vision	[70]
PSR	Prolate spheroid method method using computer vision	[70]
PSM	Prolate spheroid method using equivalent diameter	[142, 201, 3]

$$D_e = \sqrt{\frac{6M}{\rho\pi L}}, \quad (\text{B.8})$$

where M is the mass of the fish in kilograms and L is the total length of the subject in meters. This method is referred to as the prolate spheroid Murphy (PSM) method in this text. In contrast to the PSM method, Rantung *et al.* uses the measured width of the fish from their computer vision operations as the equivalent diameter, referred to as the prolate spheroid Rantung (PSR) method. The surface area can then be directly calculated using the equation for a prolate spheroid [142, 201, 3, 70]:

$$SA = 2\pi \left(\frac{D_e}{2}\right)^2 + 2\pi \left(\frac{D_e L}{4e}\right) \sin e_p \quad (\text{B.9})$$

e_p is the prolate spheroid eccentricity defined as:

$$e_p = \sqrt{1 - \frac{D_e^2}{L^2}}. \quad (\text{B.10})$$

Rantung *et al.* proposed a second method using an ellipse partition disc, which this text refers to as the partition disc Rantung (PDR) method. The fish geometry is composed as a series of partition discs as shown in Fig. B.7. The dimensions of two adjacent discs are defined as [70]:

$$\begin{aligned} a_{1i} &= \frac{W_{1i}}{2}, & a_{2i} &= \frac{W_{2i}}{2}, \\ b_{1i} &= \frac{T_{1i}}{2}, & b_{2i} &= \frac{T_{2i}}{2}, \end{aligned} \quad (\text{B.11})$$

where W_{1i} and T_{1i} are the body width and height at point i , and dx_i is the thickness between discs 1 and 2, as shown in Figure B.7. This is referred to as the partition disc Rantung method in this text. The surface area is calculated as [70]:

$$\begin{aligned} S_i &= \pi dx_i \left[\frac{(a_{1i} + b_{1i}) + (a_{2i} + b_{2i})}{2} \right] \text{ and} \\ S &= \sum_i^{n/2} S_i \text{ for } i = 2, 3, \dots, n/2. \end{aligned} \quad (\text{B.12})$$

In contrast with the existing methods discussed above, the Ika-Fit method uses the partition disc concept, but treats the top and bottom of the fish separately. The Ika-Fit method also uses the standard length of the subject versus the total length. The cross section of the specimen is assumed to be an ellipse. This is true for fusiform fish, but not the case for some other species. For elliptical cross-sections, an ellipse is fit as shown in Fig. B.7, where the top and bottom axis values at a certain z-axis point are given by the 6th degree poly fit of the side contour, b_T and b_B . The top fit given by the NACA airfoil gives the semi-minor axis a and the thickness of each disc is dt , as shown in Fig. B.7.

Consider a length along the perimeter of an ellipse:

$$ds = \sqrt{dx^2 + dy^2} = \sqrt{a^2 \cos^2 \phi + b^2 \sin^2 \phi} d\phi; \quad (\text{B.13})$$

using the polar coordinate transform:

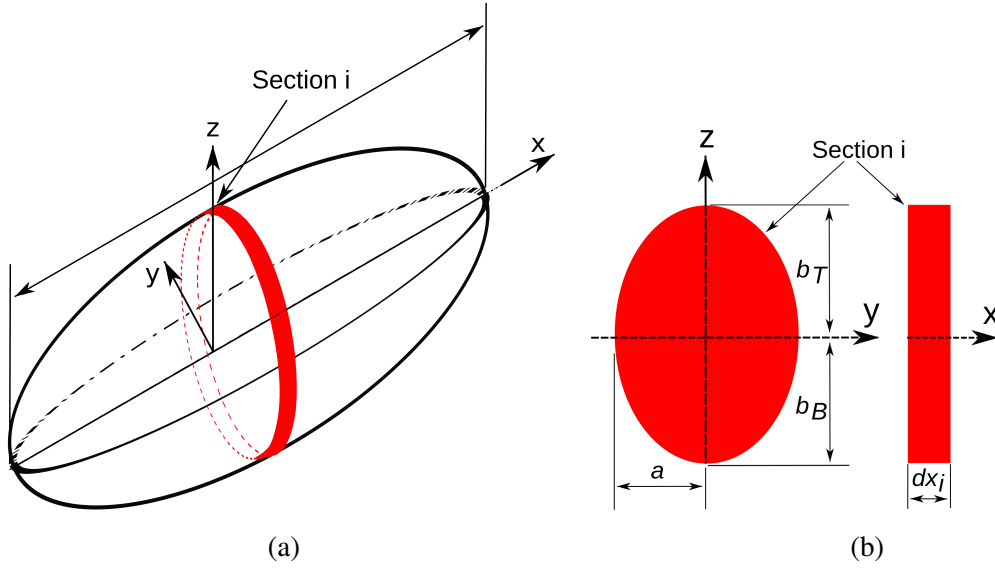


Fig. B.7 (a) Prolate spheroid with elliptical section shown. Spheroid is partitioned into discs shown in red. (b) Schematic of elliptical section showing parameters used in surface area approximation: b_T is the top contour, b_B is the bottom contour, a is the thickness of the NACA airfoil fit, and dx is the thickness of the elliptical section.

$$\begin{aligned}
 x &= a \sin \phi \\
 y_T &= b_T \cos \phi \\
 y_B &= b_B \cos \phi
 \end{aligned}
 \tag{B.14}$$

and defining eccentricity as:

$$e^2 = 1 - \frac{b_i^2}{a^2}, \tag{B.15}$$

where b_i is b_T or b_B as described in Figure B.7. Note that b_i must be less than a or e^2 is negative. Thus, the ellipse is oriented such that e^2 is a positive value. Substituting this into ds gives:

$$ds = a \sqrt{1 - e^2 \sin^2 \phi} d\phi \tag{B.16}$$

Natural and artificial fish subjects are assumed to be planarly symmetric, and so only 1/4 of the ellipse is needed for the top section and bottom section. Integrating ds from 0 to $\pi/2$ gives:

$$\int ds = a \int_0^{\pi/2} \sqrt{1 - e^2 \sin^2 \phi} d\phi = a \left[E\left(e, \frac{\pi}{2}\right) - E(e, 0) \right] \quad (\text{B.17})$$

The SciPy package is used to solve this integral and it approximates the solution as follows [251]:

$$E(e) \approx P(1 - e) - (1 - e) \log(1 - e) Q(1 - e), \quad (\text{B.18})$$

where P and Q are 10th degree polynomials. To convert the perimeter to a surface area, the x -components are evenly spaced and the difference in each x value is taken as the thickness. For $(N-1)$ points, the perimeters of the ellipse are multiplied by the thickness and the sum is taken. This differs from the PDR approach in that the solution they use for the perimeter is an infinite binomial series with higher order terms vanishing to zero and a_i and b_i defined in (B.11) [70]:

$$p = \pi(a_i + b_i) \quad (\text{B.19})$$

A further difference is that the PDR method only applies to half the subject. The Ika-Fit method uses the NACA airfoil as the top view contour and divides the ellipse into a sum of two different ellipses, with the top view contour determining the a , and b_T and b_B being the difference in height from the x -axis to the side view contour.

To include the fins in the surface area, the contour area is calculated on images with and without fins intact using the OpenCV package and Green's functions [385]. The two areas are divided to get an area ratio of fins to no fins. The area ratio is added to the calculated surface area as a percentage of the total surface area. Output from the algorithm is provided for biological animals in Table S1: Biological Fit Data and for robotic platforms in Table S2: Robot Fit and COT Data.

Appendix C

Implementation of Turbulence Models in OpenFOAM[®]

The model used to solve the Reynolds Averaged Navier-Stokes (RANS) formulation in this thesis is the 2 equation k-omega Shear Stress Transport (SST) model proposed by Menter *et al.* [243].

C.1 General Reynolds Averaged Navier-Stokes (RANS) Formulation

The start of this implementation is the classical Navier Stokes equations for incompressible flow:

$$\frac{D}{Dt} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \frac{\partial}{\partial t} \begin{pmatrix} u \\ v \\ w \end{pmatrix} + (\mathbf{u} \cdot \nabla) \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \frac{-\nabla \mathbf{P}}{\rho} + \mathbf{g} + \nu \nabla^2 \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (\text{C.1})$$

where the operators $\frac{\partial}{\partial t}$, $(\mathbf{u} \cdot \nabla)$ and $(\nu \nabla^2)$ are applied to each component of the velocity vector. Averaging the equation assumes that the variables velocity and pressure can be decomposed into a mean and fluctuating turbulent components. The mean part is denoted by a bar over the variable (\bar{u}) and the fluctuating part is denoted with an apostrophe (u'). This what is known as Reynolds decomposition [388].

The time mean \bar{u} of the turbulent function $u(x, y, z, t)$ is given by:

$$\bar{u} = \frac{1}{T} \int_0^T u dt \quad (\text{C.2})$$

where T is the period that is averaged over. This is chosen such that it is longer than any period of the fluctuations. The fluctuation u' is defined as a deviation from the mean value:

$$u' = u - \bar{u} \quad (\text{C.3})$$

which allows for the fluctuation to have zero mean value [388]

$$\overline{u'} = \frac{1}{T} \int_0^T (u - \bar{u}) dt = \bar{u} - \bar{u} = 0. \quad (\text{C.4})$$

Of important note is that the mean square of the fluctuation or measure of the intensity of the turbulence is not zero:

$$\overline{u'^2} = \frac{1}{T} \int_0^T u'^2 dt \neq 0 \quad (\text{C.5})$$

and, in general, the mean fluctuating products of the components.

Performing Reynolds averaging on the u-momentum equation is done by inserting $u = \bar{u} + u'$ and $p = \bar{P} + P'$ and then averaging. This gives the rise to the following equation [388]:

$$\frac{d\bar{u}}{dt} = -\frac{\partial \bar{p}}{\partial x} + g_x + \frac{\partial}{\partial x} \left(\bar{v} \frac{\partial \bar{u}}{\partial x} - \overline{u'^2} \right) + \frac{\partial}{\partial y} \left(\bar{v} \frac{\partial \bar{u}}{\partial y} - \overline{u'v'} \right) + \frac{\partial}{\partial z} \left(\bar{v} \frac{\partial \bar{u}}{\partial z} - \overline{u'w'} \right) \quad (\text{C.6})$$

which can then be generalized into the full 3D Navier-Stokes equation C.1. The full 3D RANS equation is as given by:

$$\frac{D\bar{\mathbf{u}}}{Dt} = \frac{\partial}{\partial t} \begin{pmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{pmatrix} + (\bar{\mathbf{u}} \cdot \nabla) \begin{pmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{pmatrix} = \frac{-\nabla \bar{P}}{\rho} + \bar{\mathbf{g}} + \nu \nabla^2 \begin{pmatrix} \bar{u} \\ \bar{v} \\ \bar{w} \end{pmatrix} + \nabla \cdot \begin{pmatrix} -\overline{u'^2} & -\overline{u'v'} & -\overline{u'w'} \\ -\overline{v'u'} & -\overline{v'^2} & -\overline{v'w'} \\ -\overline{w'u'} & -\overline{w'v'} & -\overline{w'^2} \end{pmatrix} \quad (\text{C.7})$$

where the last term is known as the Reynolds stress tensor.

Since the equations are quite long, stress tensor notation based on Einstein summation is generally utilized. This means that if a term has a variable with a repeated index, then the term is summed over that index. The following is a continuity example:

$$\nabla \cdot \mathbf{u} = \frac{\partial u_i}{\partial x_i} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \quad (\text{C.8})$$

where the velocity vector has an i appear in both the u_i and x_i so that the terms are summed over $i = 1, 2, 3$. This gives rise to the partial of all the components with their respective

direction. Another important note about Einstein notation is that if both i and j are summed over, it gives a total of 9 terms as in $i, j = 1, 2, 3$. Using this notation, the RANS equation in tensor notation becomes:

$$\left(\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} \right) = -\frac{\partial \bar{P}}{\rho \partial x_i} + \bar{g}_i + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial}{\rho \partial x_j} \overline{u'_i u'_j} \quad (\text{C.9})$$

and it is from here that various RANS turbulence models try to relate the unknown components of the stress tensor to the mean flow quantities.

Eddy viscosity models attempt to relate the Reynolds stress components to the mean rate of strain tensor. The Reynolds Averaged Navier-Stokes (RANS) equation in tensor notation is

$$\left(\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} \right) = -\frac{\partial \bar{P}}{\rho \partial x_i} + \bar{g}_i + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial}{\rho \partial x_j} \overline{u'_i u'_j}, \quad (\text{C.10})$$

The quantity $\tau_{ij} = -\overline{u'_i u'_j}$ is known as the Reynolds stress tensor. Eddy viscosity models relate the Reynolds stress components to the mean rate of the strain tensor. This is performed through the Boussinesq hypothesis [255] that relates the Reynolds stresses to the mean velocity gradients

$$-\overline{u'_i u'_j} = \nu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} k \delta_{ij} \quad (\text{C.11})$$

where δ_{ij} is the Kronecker delta function. This is a special function where $\delta_{ij} = 1$ if $i = j$ and 0 if $i \neq j$. ν_t is the kinetic eddy viscosity and k is the kinetic energy defined as $k = \frac{\overline{u'_i u'_i}}{2}$.

The difference in each eddy viscosity model is the number of additional transport equations added. For instance, the k- ω Shear stress Transport (SST) equation adds 2 additional transport equations: one for turbulence specific dissipation rate, ω , and one for turbulent kinetic energy, k .

C.2 k- ω Shear stress Transport (SST)

The 2 equation SST model was first introduced by Menter in 1994 [243]. The model uses two equations for the turbulence specific dissipation rate and turbulent kinetic energy. OpenFOAM's implementation of this model uses corrections presented by Menter *et al.* in 2003 [243].

First define the turbulence specific dissipation rate (ω) as:

$$\begin{aligned} \frac{D}{Dt}(\rho\omega) = \nabla \cdot (\rho D_\omega \nabla \omega) + \frac{\rho\gamma G}{\nu} - \frac{2}{3}\rho\gamma\omega(\nabla \cdot \mathbf{u}) - \rho\beta\omega^2 \\ - \rho(F_1 - 1)CD_{k\omega} + S_\omega, \end{aligned} \quad (C.12)$$

and the turbulence kinetic energy (k) as:

$$\frac{D}{Dt}(\rho k) = \nabla \cdot (\rho D_k \nabla k) + \rho G - \frac{2}{3}\rho k(\nabla \cdot \mathbf{u}) - \rho\beta^*\omega k + S_k, \quad (C.13)$$

where F_1 is a blending function defined as:

$$F_1 = \tanh \left\{ \left\{ \min \left[\max \left(\frac{\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega} \right), \frac{4\rho\sigma_{\omega 2}k}{CD_{k\omega}y^2} \right] \right\}^4 \right\}. \quad (C.14)$$

$CD_{k\omega} = \max \left(2\rho\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_i} \frac{\partial \omega}{\partial x_i}, 10^{-10} \right)$ and y is the distance to the nearest wall. Finally, the turbulent eddy viscosity is defined as:

$$\nu_t = \frac{a_1 k}{\max(a_1\omega, SF_2)} \quad (C.15)$$

where S is the measure of the strain rate defined as:

$$S = \sqrt{S^2} = \left| \frac{1}{2} (\partial_j u_i + \partial_i u_j) \right| \quad (C.16)$$

and F_2 is a blending function defined as:

$$F_2 = \tanh \left\{ \left\{ \min \left(\max \left(\frac{2\sqrt{k}}{\beta^*\omega y}, \frac{500\nu}{y^2\omega} \right), 100 \right) \right\}^2 \right\}. \quad (C.17)$$

Menter specifies a production limiter that is used to prevent the build-up of turbulence in stagnation regions [243]:

$$P_k = \nu_t \frac{\partial U_i}{\partial x_j} \left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \rightarrow \tilde{P}_k = \min(P_k, 10\beta^*k\omega). \quad (C.18)$$

The default coefficients implemented in OpenFOAM is shown in table C.1

C.3 k-k_L- ω

OpenFOAM®'s implementation of the k-k_L- ω are the equations from Walter and Cokljat [389] with corrections given by Fürst *et al.* [245]. It is a transition model coupled with

Table C.1 Default SST Model coefficients implemented in OpenFOAM® [144].

Coefficient	Value
α_{k1}	0.850
α_{k2}	1.000
$\alpha_{\omega1}$	0.500
$\alpha_{\omega2}$	0.856
β_1	0.075
β_2	0.0828
γ_1	5/9
γ_2	0.44
β^*	0.09
a_1	0.31
b_2	1.0
c_1	10.0

the two equation k- ω model with a detailed derivation given in Fürst *et al.* [245] with the following derivation being adapted from that work. The specific dissipation rate is defined as:

$$\begin{aligned} \frac{D}{Dt}(\omega) = \nabla \cdot (D_\omega \nabla \omega) + C_{w1} P_{kt} \frac{\omega}{k_t} - \left(1.0 - \frac{C_{wR}}{f_w}\right) k_l (R_{bp} + R_{nat}) \frac{\omega}{k_t} \\ - C_{w2} f_w^2 \omega^2 + C_{w3} f_\omega \alpha_t f_w^2 \frac{k_t^{0.5}}{y^3} \end{aligned} \quad (C.19)$$

The laminar turbulent kinetic energy is given as:

$$\frac{D}{Dt}(k_l) = \nabla \cdot (v \nabla k_l) + P_{kl} - R_{bp} + R_{nat} + D_l \quad (C.20)$$

The turbulent kinetic energy is given as:

$$\frac{D}{Dt}(k_t) = \nabla \cdot (D_k \nabla k_t) + P_{kt} + (R_{bp} + R_{nat}) k_l - \omega + D_t. \quad (C.21)$$

The production of turbulent and laminar kinetic energy is given by the following [245]:

$$P_{kt} = \nu_{T,s} S^2, \quad (C.22)$$

$$P_{kl} = \nu_{T,l} S^2, \quad (C.23)$$

where $S = \sqrt{2S_{ij}S_{ij}}$ and S_{ij} is the strain rate.

$$\nu_{T,s} = f_v F_{INT} C_\mu \sqrt{k_{T,s} \lambda_{eff}} \quad (C.24)$$

is the small-scale eddy viscosity and

$$k_{T,s} = f_{SS} f_W k_T \quad (C.25)$$

is the effective small-scale turbulence. The wall-limited turbulence length scale, λ_{eff} and damping function f_W is given by [245]:

$$\lambda_{eff} = \min(C_\lambda d, \lambda_T), \quad (C.26)$$

$$\lambda_T = \sqrt{k_T} / \omega, \quad (C.27)$$

$$f_W = \left(\frac{\lambda_{eff}}{\lambda_T} \right)^{\frac{2}{3}}, \quad (C.28)$$

where d is the distance to the wall. The rest of the terms are given as follows [389, 245]:

$$f_v = 1 - \exp - \frac{\sqrt{Re_T}}{A_v}, \quad (C.29)$$

$$Re_T = \frac{f_W^2 k_T}{\nu \omega}, \quad (C.30)$$

$$f_{SS} = \exp \left[- \left(\frac{C_{ss} \nu \Omega}{k_T} \right)^2 \right], \quad (C.31)$$

$$C_\mu = \frac{1}{A_0 + A_S \left(\frac{S}{\omega} \right)}, \quad (C.32)$$

$$f_{INT} = \min \left(\frac{k_T}{C_{INT} (k_T + k_L)}, 1 \right). \quad (C.33)$$

The laminar kinetic energy, k_L , is given by the large-scale near wall turbulence which is described by:

$$k_{T,l} = k_T - k_{T,s}, \quad (C.34)$$

with the production term given as:

$$\nu_{T,l} = \min \left(f_{\tau,l} C_{l1} \frac{\Omega \lambda_{eff}^2}{\nu} \sqrt{k_{T,l}} \lambda_{eff} + \beta_{TS} C_{l2} Re_{\Omega} d^2 \Omega, \frac{k_L + k_{T,l}}{2S} \right). \quad (C.35)$$

The coefficients are given as:

$$Re_{\omega} = \frac{d^2 \Omega}{\nu}, \quad (C.36)$$

$$\beta_{TS} = 1 - \exp \left(- \frac{\max (Re_{\Omega} - C_{TS,crit}, 0)^2}{A_{TS}} \right), \quad (C.37)$$

$$f_{\tau,l} = 1 - \exp \left(- C_{\tau,l} \frac{k_{\tau l}}{\lambda_{eff}^2 \Omega^2} \right). \quad (C.38)$$

The turbulent diffusivity term, α_t is given as:

$$\alpha_t = f_{\nu} C_{\mu,std} \sqrt{k_{T,s}} \lambda_{eff}, \quad (C.39)$$

and the damping function, f_{ω} , is given as:

$$f_{\omega} = 1 - \exp \left[-0.41 \left(\frac{\lambda_{eff}}{\lambda_T} \right)^4 \right]. \quad (C.40)$$

The terms R_{bp} and R_{nat} are the laminar/turbulent transition terms of the energy transfer from k_L to k_T given by [245]:

$$R_{bp} = \frac{C_R \beta_{bp} k_L \omega}{f_W}, \quad (\text{C.41})$$

$$R_{nat} = C_{R,nat} \beta_{nat} k_L \Omega, \quad (\text{C.42})$$

$$\beta_{bp} = 1 - \exp\left(-\frac{\phi_{bp}}{A_{bp}}\right), \quad (\text{C.43})$$

$$\phi_{bp} = \max\left(\frac{k_T}{\nu \Omega} - C_{bp,crit}, 0\right), \quad (\text{C.44})$$

$$\beta_{nat} = 1 - \exp\left(-\frac{\phi_{nat}}{A_{nat}}\right), \quad (\text{C.45})$$

$$\phi_{nat} = \max\left(Re_\Omega - \frac{C_{nat,crit}}{f_{nat,crit}}, 0\right), \quad (\text{C.46})$$

$$f_{nat,crit} = 1 - \exp\left(-C_{NC} \frac{\sqrt{k_L} d}{\nu}\right). \quad (\text{C.47})$$

The turbulent viscosity for the momentum equations is given by:

$$\nu_T = \nu_{T,s} + \nu_{T,l}, \quad (\text{C.48})$$

and the turbulent thermal diffusivity, α_θ , is defined as:

$$\alpha_\theta = f_w \frac{k_T}{k_T + k_L} \frac{\nu_{T,s}}{P_r} + (1 - f_w) C_{\alpha,\theta} \sqrt{k_T} \lambda_{eff}, \quad (\text{C.49})$$

and the coefficient, $C_{\omega 2}$ is:

$$C_{\omega 2} = 0.92 f_W^2. \quad (\text{C.50})$$

The default coefficients implemented in OpenFOAM is shown in table C.2

C.4 Spalart-Allmaras

The Spalart-Allmaras model is given by Spalart and Allmaras [246] and consists of one equation based on a modified turbulence viscosity, $\tilde{\nu}$. This specific dissipation rate is given as:

Table C.2 Default k - k_L - ω coefficients implemented in OpenFOAM[®] [144].

Coefficient	Value
A_0	4.04
A_s	2.12
A_v	6.75
A_{bp}	0.6
A_{nat}	200.00
A_{ts}	200.00
C_{bpCrit}	1.2
C_{nc}	0.1
$C_{natCrit}$	1250.00
C_{int}	0.75
C_{tsCrit}	1000.00
C_{rNat}	0.02
C_{11}	$3.4e-6$
C_{12}	$1.0e-10$
C_R	0.12
$C_{alphaTheta}$	0.035
C_{ss}	1.5
C_{tauL}	4360.00
C_{w1}	0.44
C_{w2}	0.92
C_{w3}	0.3
C_{wR}	1.5
C_{lambda}	2.495
C_{muStd}	0.09
P_{rtheta}	0.85
$Sigma_k$	1.00
$Sigma_w$	1.17

$$\begin{aligned} \frac{D}{Dt}(\rho \tilde{v}) = \nabla \cdot (\rho D_{\tilde{v}} \tilde{v}) + \frac{C_{b2}}{\sigma_{v_t}} \rho |\nabla \tilde{v}|^2 + C_{b1} \rho \tilde{S} \tilde{v} (1 - f_{t2}) \\ - \left(C_{w1} f_w - \frac{C_{b1}}{\kappa^2} f_{t2} \right) \rho \frac{\tilde{v}^2}{\tilde{d}^2} + S_{\tilde{v}}, \end{aligned} \quad (\text{C.51})$$

with the turbulent viscosity is given by:

$$\nu_t = \tilde{\nu} f_{v1}, \quad (\text{C.52})$$

f_{v1} is given by:

$$f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \quad (\text{C.53})$$

and χ is defined as:

$$\chi = \frac{\tilde{\nu}}{\nu}. \quad (\text{C.54})$$

\tilde{S} is given by:

$$\tilde{S} = S_{ij} + \frac{\tilde{\nu}}{(\kappa y)^2} \chi, \quad (\text{C.55})$$

where S_{ij} is the mean vorticity tensor. The default model coefficients given in OpenFOAM® [144] are given in table C.3.

C.5 Turbulent Boundary Conditions

OpenFOAM® implementation of the turbulent boundary conditions starts with turbulent kinetic energy being estimated as [144]:

$$k = \frac{3}{2} (I |\mathbf{u}_{ref}|)^2, \quad (\text{C.56})$$

Table C.3 Default Spalart-Allmaras Model coefficients implemented in OpenFOAM® [144].

Coefficient	Value
σ_{v_t}	2/3
C_{b1}	0.1355
C_{b2}	0.622
C_{w1}	$\frac{C_{b1}}{\kappa^2} + \frac{1 + C_{b2}}{\sigma_{v_t}}$
C_{w2}	0.3
C_{w3}	2
C_{v1}	7.1
C_s	0.3

where I is the intensity, and \mathbf{u}_{ref} is the reference velocity. The specific turbulent dissipation rate, ω , is then given as:

$$\omega = \frac{k^{0.5}}{C_\mu^{0.25} L}, \quad (C.57)$$

where C_μ is equal to 0.09, and L is a reference length scale. The turbulent dissipation rate, ε , is given by:

$$\varepsilon = C_\mu \frac{k^{1.5}}{l}, \quad (C.58)$$

where l is the turbulent length scale set as 0.07 meters in the case of this research. Finally, the kinematic viscosity is defined as

$$\nu_t = \frac{k}{\omega}. \quad (C.59)$$

C.6 First Layer Thickness

The first layer thickness starts with calculating the Reynolds number, Re , as:

$$Re = \frac{\rho UL}{\mu}, \quad (C.60)$$

where ρ is the fluid density, U is the freestream velocity, L is the characteristic length, and μ is the dynamic viscosity. Skin friction, C_f is estimated using the empirical correlation for fully developed turbulent flow, $Re < 10^9$, over a flat plate [256]

$$C_f = [2 \log_{10} Re - 0.65]^{-2.3}. \quad (C.61)$$

The skin friction coefficient is used to compute the wall shear stress, τ_w , is given as

$$\tau_w = \frac{1}{2} \rho U^2 C_f, \quad (C.62)$$

and the friction velocity, μ_t is then given by

$$\mu_t = \sqrt{\frac{\tau_w}{\rho}}. \quad (C.63)$$

The y^+ equation can be rearranged to give the height of the wall adjacent cell centroid from the wall, y_p , as [256]:

$$y^+ = \frac{\rho y_p \mu_t}{\mu} \Rightarrow y_p = \frac{y^+ \mu}{\mu_t \rho}. \quad (C.64)$$

y_p is the distance of the cell centroid from the wall which makes the total height of the first cell to be:

$$y_H = 2y_p. \quad (C.65)$$

Once the number of inflation layers, n , and growth ratio, r , is defined, the final layer thickness is given by:

$$y_{final} = y_H r^{N-1}, \quad (C.66)$$

the total height of inflation layers is given as:

$$y_T = y_H \left(\frac{1 - r^N}{1 - r} \right), \quad (\text{C.67})$$

and the totally boundary layer thickness, δ_b is given by the Blasius solution [256]:

$$\delta_b = \begin{cases} \frac{4.91L}{Re_L^{1/2}}, & Re < 5 \times 10^5 \\ \frac{0.38L}{Re_L^{1/5}}, & Re \geq 5 \times 10^5. \end{cases} \quad (\text{C.68})$$

Appendix D

Ika-Flow Code

The following is the code used in OpenFOAM® to achieve the fish midline motion. Listing D.1 - D.11 is code implemented for overset motion, listing D.12 - D.15 is code implemented for non-overset patch movement. Finally, listing D.16 is a user defined function (UDF) to implement the same motion in Ansys Fluent.

D.1 Ika-Flow Overset motion code

Listing D.1 fishBodyMotionFunction header file

```
/*-----*\
=====
\\      /  F i e l d      |  OpenFOAM: The Open Source CFD Toolbox
\\      /  O p e r a t i o n      |
\\      /  A n d      |  www.openfoam.com
\\      /  M a n i p u l a t i o n      |
-----*/

Copyright (C) 2011–2016 OpenFOAM Foundation
Copyright (C) 2020 OpenCFD Ltd.

License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Namespace
Foam::fishBodyMotionFunctions

Description
Namespace for solid-body motions
```

```

Class
    Foam::fishBodyMotionFunction

Description
    Base class for defining solid-body motions

SourceFiles
    fishBodyMotionFunction.C
    dynamicFvMeshNew.C

/*-----*/

#ifndef fishBodyMotionFunction_H
#define fishBodyMotionFunction_H

#include "Time.H"
#include "dictionary.H"
#include "septrnion.H"
#include "pointFields.H"
#include "autoPtr.H"
#include "runTimeSelectionTables.H"

// * * * * *

namespace Foam
{
/*-----*/
    Class fishBodyMotionFunction Declaration
/*-----*/

class fishBodyMotionFunction
{
protected:

    // Protected Data

    dictionary SBMFCoeffs_;

    const Time& time_;

    pointField p0_;

    // Protected Member Functions

    //- No copy construct
    fishBodyMotionFunction(const fishBodyMotionFunction&) = delete;

    //- No copy assignment
    void operator=(const fishBodyMotionFunction&) = delete;

public:

    //- Runtime type information
    TypeName("fishBodyMotionFunction");

    // Declare run-time constructor selection table

    declareRunTimeSelectionTable
    (
        autoPtr,
        fishBodyMotionFunction,
        dictionary,
        (const dictionary& SBMFCoeffs, const Time& runTime),
        (SBMFCoeffs, runTime)
    );

    // Constructors

    //- Construct from the SBMFCoeffs dictionary and Time
    fishBodyMotionFunction
    (

```



```

        const dictionary& SBMFCoeffs,
        const Time& runTime
    );

    // Construct and return a clone
    virtual autoPtr<fishBodyMotionFunction> clone() const = 0;

// Selectors

    // Select constructed from the SBMFCoeffs dictionary and Time
    static autoPtr<fishBodyMotionFunction> New
    (
        const dictionary& SBMFCoeffs,
        const Time& runTime
    );

// Destructor
    virtual ~fishBodyMotionFunction() = default;

// Member Functions

    // Return the transformed pointField
    virtual tmp<vectorField> transformationPoints(pointField& p0_) const = 0;

    // Update properties from given dictionary
    virtual bool read(const dictionary& SBMFCoeffs) = 0;

    // Write in dictionary format
    virtual void writeData(Ostream&) const;
};

// ***** //

} // End namespace Foam

// ***** //

#endif

// ***** //

```

Listing D.2 fishBodyMotionFunction code file

```

/*-----\
=====|
\\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\      /  O p e r a t i o n |
\\      /  A n d           | www.openfoam.com
\\      /  M a n i p u l a t i o n |

-----
Copyright (C) 2011–2017 OpenFOAM Foundation

-----
License

This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

-----*/

```

```

#include "fishBodyMotionFunction.H"

// ***** Static Data Members ***** //

namespace Foam
{
    defineTypeNameAndDebug(fishBodyMotionFunction, 0);
    defineRunTimeSelectionTable(fishBodyMotionFunction, dictionary);
}

// ***** Constructors ***** //

Foam::fishBodyMotionFunction::fishBodyMotionFunction
(
    const dictionary& SBMFCoeffs,
    const Time& runTime
)
:
    SBMFCoeffs_
    (
        SBMFCoeffs.optionalSubDict
        (
            SBMFCoeffs.get<word>("fishBodyMotionFunction") + "Coeffs"
        )
    ),
    time_(runTime)
{}

// ***** Member Functions ***** //

bool Foam::fishBodyMotionFunction::read(const dictionary& SBMFCoeffs)
{
    SBMFCoeffs_ = SBMFCoeffs.optionalSubDict(type() + "Coeffs");

    return true;
}

void Foam::fishBodyMotionFunction::writeData(Ostream& os) const
{
    os << SBMFCoeffs_;
}

// ***** //

```

Listing D.3 fishBodyMotionFunctionNew code file

```

/*-----*/
===== |
\\      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\      / O p e r a t i o n |
\\      / A n d           | www.openfoam.com
\\      / M a n i p u l a t i o n |
-----|

Copyright (C) 2011–2016 OpenFOAM Foundation
Copyright (C) 2019 OpenCFD Ltd.
-----|

License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

```

```

    You should have received a copy of the GNU General Public License
    along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

\<-----*/

#include "fishBodyMotionFunction.H"

// ***** Selectors ***** //

Foam::autoPtr<Foam::fishBodyMotionFunction> Foam::fishBodyMotionFunction::New
(
    const dictionary& dict,
    const Time& runTime
)
{
    const word motionType(dict.get<word>("fishBodyMotionFunction"));

    Info<< "Selecting_fish-body_motion_function_" << motionType << endl;

    auto cstrIter = dictionaryConstructorTablePtr_>cfind(motionType);

    if (!cstrIter.found())
    {
        FatalIOErrorInLookup
        (
            dict,
            "fishBodyMotionFunction",
            motionType,
            *dictionaryConstructorTablePtr_
        ) << exit(FatalIOError);
    }

    return autoPtr<fishBodyMotionFunction>(cstrIter()(dict, runTime));
}

// ***** //

```

Listing D.4 Anguilliform header file

```

/*-----*/
=====
\\    /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\    /  O p e r a t i o n      |
\\    /  A n d      | www.openfoam.com
\\    /  M a n i p u l a t i o n      |

-----

Copyright (C) 2011-2016 OpenFOAM Foundation
Copyright (C) 2020 OpenCFD Ltd.

-----

License

This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Class
Foam::fishBodyMotionFunctions::anguilliformMotion

Description
SolidBodyMotionFvMesh 6DoF motion function. Oscillating rotation.

```

```

SourceFiles
    anguilliformMotion.C

\*-----*/

#ifndef anguilliformMotion_H
#define anguilliformMotion_H

#include "fishBodyMotionFunction.H"
#include "primitiveFields.H"
#include "pointFields.H"
#include "point.H"

// ***** //

namespace Foam
{
    namespace fishBodyMotionFunctions
    {

\*-----*/
        Class anguilliformMotion Declaration
\*-----*/

    class anguilliformMotion
    :
    {
        public fishBodyMotionFunction

        // Private Data

        //- Centre of gravity
        point origin_;

        //- Amplitude
        scalar amplitude_;

        //- wave number growth
        scalar waveNumber_;

        //- Length
        scalar length_;

        //- Ramp
        scalar ramp_;

        //- body wave speed
        scalar omega_;

        //- amplitude growth speed
        scalar alpha_;

        //- time delay
        scalar delay_;

        //- Define a pointField
        pointField p0_;

        // Private Member Functions

        //- No copy construct
        anguilliformMotion(const anguilliformMotion&) = delete;

        //- No copy assignment
        void operator=(const anguilliformMotion&) = delete;

    public:

        //- Runtime type information
        TypeName("anguilliformMotion");

        // Constructors

        //- Construct from components

```

```

anguilliformMotion
(
    const dictionary& SBMFCoeffs,
    const Time& runTime
);

// Construct and return a clone
virtual autoPtr<fishBodyMotionFunction> clone() const
{
    return autoPtr<fishBodyMotionFunction>
    (
        new anguilliformMotion
        (
            SBMFCoeffs_,
            time_
        )
    );
}

// Destructor
virtual ~anguilliformMotion() = default;

// Member Functions

// Return the transformed pointField
virtual tmp<vectorField> transformationPoints(pointField& p0_) const;

// Update properties from given dictionary
virtual bool read(const dictionary& SBMFCoeffs);
};

// *****

} // End namespace fishBodyMotionFunctions
} // End namespace Foam

// *****

#endif

// *****

```

Listing D.5 Anguilliform code file

```

/*-----\
=====
\\      / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\      / O p e r a t i o n |
\\      / A n d           | www.openfoam.com
\\      / M a n i p u l a t i o n |

-----
Copyright (C) 2011-2016 OpenFOAM Foundation
Copyright (C) 2019 OpenCFD Ltd.

-----
License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

```

```

\<----- */

#include "anguilliformMotion.H"
#include "addToRunTimeSelectionTable.H"
#include "unitConversion.H"

// ***** Static Data Members ***** //

namespace Foam
{
namespace fishBodyMotionFunctions
{
    defineTypeNameAndDebug(anguilliformMotion, 0);
    addToRunTimeSelectionTable
    (
        fishBodyMotionFunction,
        anguilliformMotion,
        dictionary
    );
}
}

// ***** Constructors ***** //

Foam::fishBodyMotionFunctions::anguilliformMotion::
anguilliformMotion
(
    const dictionary& SBMFCoeffs,
    const Time& runTime
)
:
    fishBodyMotionFunction(SBMFCoeffs, runTime)
{
    read(SBMFCoeffs);
}

// ***** Member Functions ***** //

Foam::tmp<Foam::pointField>
Foam::fishBodyMotionFunctions::anguilliformMotion::
transformationPoints(pointField& p0) const
{
    scalar tm = time_.value();

    if (delay_ <= tm)
    {
        scalar t = tm - delay_;

        forAll(p0, pointI)
        {
            const scalar x = (p0[pointI].component(0) - origin_[0]) / length_;
            const scalar y = p0[pointI].component(1) - origin_[1];
            const scalar z = p0[pointI].component(2) - origin_[2];

            // new value by equation
            const scalar localAmplitude = amplitude_ * exp(alpha_ * (x - 1));
            const scalar dampFactor = 0.5 - 0.5 * tanh(ramp_ * x - (ramp_ + 9));

            const scalar yr = y + localAmplitude * dampFactor * sin(waveNumber_ * x - omega_ * t) * length_;

            p0[pointI] = vector(x, yr, z);
        }
        return p0;
    }
    else
    {
        return p0;
    }
}

bool Foam::fishBodyMotionFunctions::anguilliformMotion::read
(

```

```

    const dictionary& SBMFCoeffs
)
{
    fishBodyMotionFunction::read(SBMFCoeffs);

    SBMFCoeffs_.readEntry("origin", origin_);
    SBMFCoeffs_.readEntry("amplitude", amplitude_);
    SBMFCoeffs_.readEntry("waveNumber", waveNumber_);
    SBMFCoeffs_.readEntry("length", length_);
    SBMFCoeffs_.readEntry("ramp", ramp_);
    SBMFCoeffs_.readEntry("omega", omega_);
    SBMFCoeffs_.readEntry("alpha", alpha_);
    SBMFCoeffs_.readEntry("delay", delay_);

    return true;
}

// ***** //

```

Listing D.6 Carangiform header file

```

/*-----*\
=====
\\ / Field | OpenFOAM: The Open Source CFD Toolbox
\\ / Operation |
\\ / And | www.openfoam.com
\\ / Manipulation |

-----
Copyright (C) 2011-2016 OpenFOAM Foundation
Copyright (C) 2020 OpenCFD Ltd.

-----
License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Class
Foam::fishBodyMotionFunctions::carangiformMotion

Description
Fish Body Motion. Carangiform.

SourceFiles
carangiformMotion.C

\*-----*/

#ifndef carangiformMotion_H
#define carangiformMotion_H

#include "fishBodyMotionFunction.H"
#include "primitiveFields.H"
#include "pointFields.H"
#include "point.H"

// ***** //

namespace Foam
{
namespace fishBodyMotionFunctions
{

```

```

/*-----\
Class carangiformMotion Declaration
-----*/

class carangiformMotion
:
public fishBodyMotionFunction
{
    // Private Data

    // - Centre of gravity
    point origin_;

    // - Amplitude
    scalar amplitude_;

    // - Coefficients
    vector coefficients_;

    // - Wave number
    scalar waveNumber_;

    // - Length
    scalar length_;

    // - Ramp
    scalar ramp_;

    // - Radial velocity
    scalar omega_;

    // - time delay
    scalar delay_;

    // - Define a pointField
    pointField p0_;

    // Private Member Functions

    // - No copy construct
    carangiformMotion(const carangiformMotion&) = delete;

    // - No copy assignment
    void operator=(const carangiformMotion&) = delete;

public:

    // - Runtime type information
    TypeName("carangiformMotion");

    // Constructors

    // - Construct from components
    carangiformMotion
    (
        const dictionary& SBMFCoeffs,
        const Time& runTime
    );

    // - Construct and return a clone
    virtual autoPtr<fishBodyMotionFunction> clone() const
    {
        return autoPtr<fishBodyMotionFunction>
        (
            new carangiformMotion
            (
                SBMFCoeffs_,
                time_
            )
        );
    }
}

```



```

    //~ Destructor
    virtual ~carangiformMotion() = default;

    // Member Functions

    //~ Return the transformed pointField
    virtual tmp<vectorField> transformationPoints(pointField& p0_) const;

    //~ Update properties from given dictionary
    virtual bool read(const dictionary& SBMFCoeffs);
};

// *****

} // End namespace fishBodyMotionFunctions
} // End namespace Foam

// *****

#endif

// *****

```

Listing D.7 Carangiform code file

```

/*-----*/
=====
\\    /  F i e l d          | OpenFOAM: The Open Source CFD Toolbox
\\    /  O p e r a t i o n   |
\\    /  A n d               | www.openfoam.com
\\ \  /  M a n i p u l a t i o n |

-----
Copyright (C) 2011-2016 OpenFOAM Foundation
Copyright (C) 2019 OpenCFD Ltd.

-----
License

This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

/*-----*/

#include "carangiformMotion.H"
#include "addToRunTimeSelectionTable.H"
#include "unitConversion.H"

// ***** Static Data Members *****

namespace Foam
{
    namespace fishBodyMotionFunctions
    {
        defineTypeNameAndDebug(carangiformMotion, 0);
        addToRunTimeSelectionTable
        (
            fishBodyMotionFunction,
            carangiformMotion,
            dictionary
        );
    }
}

```

```

}
}

// ***** Constructors ***** //

Foam::fishBodyMotionFunctions::carangiformMotion::
carangiformMotion
(
    const dictionary& SBMFCoeffs,
    const Time& runTime
)
:
    fishBodyMotionFunction(SBMFCoeffs, runTime)
{
    read(SBMFCoeffs);
}

// ***** Member Functions ***** //

Foam::tmp<Foam::pointField>
Foam::fishBodyMotionFunctions::carangiformMotion::
transformationPoints(pointField& p0) const
{
    scalar tm = time_.value();

    if (delay_ <= tm)
    {
        scalar t = tm - delay_;

        forAll(p0, pointI)
        {
            const scalar x = (p0[pointI].component(0)-origin_[0])/length_;
            const scalar y = p0[pointI].component(1)-origin_[1];
            const scalar z = p0[pointI].component(2)-origin_[2];

            // new value by equation
            const scalar localAmplitude = amplitude_ * (1 + (coefficients_[0] * (x-1)) + (coefficients_[1] * (x*x-1)));
            const scalar dampFactor = 0.5-0.5*tanh(ramp_*x-(ramp_+9));

            const scalar yr = y + localAmplitude * dampFactor * sin(waveNumber_*x - omega_*t) * length_;

            p0[pointI] = vector(x, yr, z);
        }
        return p0;
    }
    else
    {
        return p0;
    }
}

bool Foam::fishBodyMotionFunctions::carangiformMotion::read
(
    const dictionary& SBMFCoeffs
)
{
    fishBodyMotionFunction::read(SBMFCoeffs);

    SBMFCoeffs_.readEntry("origin", origin_);
    SBMFCoeffs_.readEntry("amplitude", amplitude_);
    SBMFCoeffs_.readEntry("coefficients", coefficients_);
    SBMFCoeffs_.readEntry("waveNumber", waveNumber_);
    SBMFCoeffs_.readEntry("length", length_);
    SBMFCoeffs_.readEntry("ramp", ramp_);
    SBMFCoeffs_.readEntry("omega", omega_);
    SBMFCoeffs_.readEntry("delay", delay_);

    return true;
}

// ***** //

```

Listing D.8 Ostraciiform header file

```

/*-----*/
=====
\\      /  F i e l d      |  OpenFOAM: The Open Source CFD Toolbox
\\      /  O p e r a t i o n  |
\\      /  A n d           |  www.openfoam.com
\\      /  M a n i p u l a t i o n  |

-----
Copyright (C) 2011–2016 OpenFOAM Foundation
Copyright (C) 2020 OpenCFD Ltd.

-----
License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Class
Foam::fishBodyMotionFunctions::ostraciiformMotion

Description
SolidBodyMotionFvMesh 6DoF motion function. Oscillating rotation.

SourceFiles
ostraciiformMotion.C

/*-----*/

#ifndef ostraciiformMotion_H
#define ostraciiformMotion_H

#include "fishBodyMotionFunction.H"
#include "primitiveFields.H"
#include "pointFields.H"
#include "point.H"

// *****

namespace Foam
{
namespace fishBodyMotionFunctions
{

/*-----*/
Class ostraciiformMotion Declaration
/*-----*/

class ostraciiformMotion
:
public fishBodyMotionFunction
{
// Private Data

// Centre of gravity
point origin_;

// Amplitude
scalar amplitude_;

// Coefficients

```

```

    vector coefficients_;

    // - Wave number
    scalar waveNumber_;

    // - Length
    scalar length_;

    // - Ramp
    scalar ramp_;

    // - Radial velocity
    scalar omega_;

    // - Pivot point
    scalar pivot_;

    // - Max Angle
    scalar maxAngle_;

    // - Phase angle
    scalar phase_;

    // - time delay
    scalar delay_;

    // - Define a pointField
    pointField p0_;

    // Private Member Functions

    // - No copy construct
    ostraciiiformMotion(const ostraciiiformMotion&) = delete;

    // - No copy assignment
    void operator=(const ostraciiiformMotion&) = delete;

public:

    // - Runtime type information
    TypeName("ostraciiiformMotion");

    // Constructors

    // - Construct from components
    ostraciiiformMotion
    (
        const dictionary& SBMFCoeffs,
        const Time& runTime
    );

    // - Construct and return a clone
    virtual autoPtr<fishBodyMotionFunction> clone() const
    {
        return autoPtr<fishBodyMotionFunction>
        (
            new ostraciiiformMotion
            (
                SBMFCoeffs_,
                time_
            )
        );
    }

    // - Destructor
    virtual ~ostraciiiformMotion() = default;

    // Member Functions

    // - Return the transformed pointField
    virtual tmp<vectorField> transformationPoints(pointField& p0_) const;

```

```

        // - Update properties from given dictionary
        virtual bool read(const dictionary& SBMFCoeffs);
};

// *****

} // End namespace fishBodyMotionFunctions
} // End namespace Foam

// *****

#endif

// *****

```

Listing D.9 Ostraciiform code file

```

/*-----*/
=====
\\      /  F ield          | OpenFOAM: The Open Source CFD Toolbox
\\      /  O peration      |
\\      /  A nd             | www.openfoam.com
\\      /  M anipulation    |
-----

Copyright (C) 2011–2016 OpenFOAM Foundation
Copyright (C) 2019 OpenCFD Ltd.
-----

License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

\*-----*/

#include "ostraciiformMotion.H"
#include "mathematicalConstants.H"
#include "addToRunTimeSelectionTable.H"
#include "unitConversion.H"

// ***** Static Data Members ***** //

namespace Foam
{
    namespace fishBodyMotionFunctions
    {
        defineTypeNameAndDebug(ostraciiformMotion, 0);
        addToRunTimeSelectionTable
        (
            fishBodyMotionFunction,
            ostraciiformMotion,
            dictionary
        );
    }
}

// ***** Constructors ***** //

Foam::fishBodyMotionFunctions::ostraciiformMotion::
ostraciiformMotion

```

```

(
    const dictionary& SBMFCoeffs,
    const Time& runTime
)
:
    fishBodyMotionFunction(SBMFCoeffs, runTime)
{
    read(SBMFCoeffs);
}

// ***** Member Functions ***** //
Foam::tmp<Foam::pointField>
Foam::fishBodyMotionFunctions::ostraciiiformMotion::
transformationPoints(pointField& p0) const
{
    const scalar tm = time_.value();

    if (delay_ <= tm)
    {
        scalar t = tm - delay_;

        scalar maxAngle = maxAngle_ * 2 * M_PI / 180;
        scalar phase = phase_ * 2 * M_PI / 180;

        forAll(p0, pointI)
        {
            const scalar x = (p0[pointI].component(0)-origin_[0])/length_;
            const scalar y = p0[pointI].component(1)-origin_[1];
            const scalar z = p0[pointI].component(2)-origin_[2];

            scalar yr = 0;

            // new value by equation
            const scalar xPivot = x - pivot_;

            const scalar localAmplitude = (amplitude_ * (1 + (coefficients_[0]*(pivot_-1)) +
                (coefficients_[1]*(pivot_*pivot_-1))));
            const scalar yEnd = localAmplitude * sin(waveNumber_*pivot_ - omega_*t) * length_;

            if (x >= pivot_)
            {
                const scalar thetaT = maxAngle * sin(waveNumber_*pivot_ - omega_*t + phase);

                yr = y + yEnd + xPivot * tan(thetaT) * length_;

            }
            else
            {
                const scalar localAmplitude = (amplitude_ * (1 + (coefficients_[0]*(pivot_-1)) +
                    (coefficients_[1]*(pivot_*pivot_-1))));
                const scalar yEnd = localAmplitude * sin(waveNumber_*pivot_ - omega_*t) * length_;
                const scalar yHead = (amplitude_ * (1 + (coefficients_[0] * (0-1) +
                    (coefficients_[1] * (0-1)))) * sin(waveNumber_*0 - omega_*t)) * length_;

                const scalar m = (yEnd - yHead) / (pivot_);

                yr = y + (m * x) + yHead;

            }

            p0[pointI] = vector(x, yr, z);
        }

        return p0;
    }
    else
    {
        return p0;
    }
}

bool Foam::fishBodyMotionFunctions::ostraciiiformMotion::read
(

```

```

    const dictionary& SBMFCoeffs
)
{
    fishBodyMotionFunction::read(SBMFCoeffs);

    SBMFCoeffs_.readEntry("origin", origin_);
    SBMFCoeffs_.readEntry("amplitude", amplitude_);
    SBMFCoeffs_.readEntry("coefficients", coefficients_);
    SBMFCoeffs_.readEntry("waveNumber", waveNumber_);
    SBMFCoeffs_.readEntry("length", length_);
    SBMFCoeffs_.readEntry("ramp", ramp_);
    SBMFCoeffs_.readEntry("omega", omega_);
    SBMFCoeffs_.readEntry("pivot", pivot_);
    SBMFCoeffs_.readEntry("maxAngle", maxAngle_);
    SBMFCoeffs_.readEntry("phaseAngle", phase_);
    SBMFCoeffs_.readEntry("delay", delay_);

    return true;
}

// ***** //

```

Listing D.10 Thunniform header file

```

/*-----\
=====|
\\ / F i e l d | OpenFOAM: The Open Source CFD Toolbox
\\ / O p e r a t i o n |
\\ / A n d | www.openfoam.com
\\ / M a n i p u l a t i o n |

-----
Copyright (C) 2011-2016 OpenFOAM Foundation
Copyright (C) 2020 OpenCFD Ltd.

-----
License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Class
Foam::fishBodyMotionFunctions::thunniformMotion

Description
SolidBodyMotionFvMesh 6DoF motion function. Oscillating rotation.

SourceFiles
thunniformMotion.C

\*-----*/

#ifndef thunniformMotion_H
#define thunniformMotion_H

#include "fishBodyMotionFunction.H"
#include "primitiveFields.H"
#include "pointFields.H"
#include "point.H"

// ***** //

namespace Foam

```

```

{
namespace fishBodyMotionFunctions
{

/*-----*/
Class thunniformMotion Declaration
/*-----*/

class thunniformMotion
:
{
    public fishBodyMotionFunction

    {
        // Private Data

        // - Centre of gravity
        point origin_;

        // - Amplitude
        scalar amplitude_;

        // - Coefficients
        vector coefficients_;

        // - Wave number
        scalar waveNumber_;

        // - Length
        scalar length_;

        // - Ramp
        scalar ramp_;

        // - Radial velocity
        scalar omega_;

        // - Pivot point
        scalar pivot_;

        // - Max Angle
        scalar maxAngle_;

        // - Phase angle
        scalar phase_;

        // - time delay
        scalar delay_;

        // - Define a pointField
        pointField p0_;

        // Private Member Functions

        // - No copy construct
        thunniformMotion(const thunniformMotion&) = delete;

        // - No copy assignment
        void operator=(const thunniformMotion&) = delete;

public:

        // - Runtime type information
        TypeName("thunniformMotion");

        // Constructors

        // - Construct from components
        thunniformMotion
        (
            const dictionary& SBMFCoeffs,
            const Time& runTime
        );

        // - Construct and return a clone

```



```

virtual autoPtr<fishBodyMotionFunction> clone() const
{
    return autoPtr<fishBodyMotionFunction>
    (
        new thunniformMotion
        (
            SBMFCoeffs_,
            time_
        )
    );
}

//~ Destructor
virtual ~thunniformMotion() = default;

// Member Functions

//~ Return the transformed pointField
virtual tmp<vectorField> transformationPoints(pointField& p0_) const;

//~ Update properties from given dictionary
virtual bool read(const dictionary& SBMFCoeffs);
};

// *****

} // End namespace fishBodyMotionFunctions
} // End namespace Foam

// *****

#endif

// *****

```

Listing D.11 Thunniform code file

```

/*-----*\
=====
\\ / Field | OpenFOAM: The Open Source CFD Toolbox
\\ / Operation |
\\ / And | www.openfoam.com
\\ / Manipulation |

-----
Copyright (C) 2011-2016 OpenFOAM Foundation
Copyright (C) 2019 OpenCFD Ltd.

-----
License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

-----*/

#include "thunniformMotion.H"
#include "mathematicalConstants.H"
#include "addToRunTimeSelectionTable.H"
#include "unitConversion.H"

```

```

// * * * * * Static Data Members * * * * * //

namespace Foam
{
namespace fishBodyMotionFunctions
{
    defineTypeNameAndDebug(thunniformMotion, 0);
    addToRunTimeSelectionTable
    (
        fishBodyMotionFunction,
        thunniformMotion,
        dictionary
    );
}
}

// * * * * * Constructors * * * * * //

Foam::fishBodyMotionFunctions::thunniformMotion::
thunniformMotion
(
    const dictionary& SBMFCoeffs,
    const Time& runTime
)
:
    fishBodyMotionFunction(SBMFCoeffs, runTime)
{
    read(SBMFCoeffs);
}

// * * * * * Member Functions * * * * * //

Foam::tmp<Foam::pointField>
Foam::fishBodyMotionFunctions::thunniformMotion::
transformationPoints(pointField& p0) const
{
    const scalar tm = time_.value();

    if (delay_ <= tm)
    {
        scalar t = tm - delay_;

        scalar maxAngle = maxAngle_ * 2 * M_PI / 180;
        scalar phase = phase_ * 2 * M_PI / 180;

        forAll(p0, pointI)
        {
            const scalar x = (p0[pointI].component(0) - origin_[0]) / length_;
            const scalar y = p0[pointI].component(1) - origin_[1];
            const scalar z = p0[pointI].component(2) - origin_[2];

            scalar yr = 0;

            if (x >= pivot_)
            {
                // new value by equation
                const scalar xPivot = x - pivot_;

                const scalar localAmplitude = (amplitude_ * (1 + (coefficients_[0]*(pivot_-1)) +
                    (coefficients_[1]*(pivot_*pivot_-1))));
                const scalar yEnd = localAmplitude * sin(waveNumber_*pivot_ - omega*t) * length_;

                const scalar thetaT = maxAngle * sin(waveNumber_*pivot_ - omega*t + phase);

                yr = y + yEnd + xPivot * tan(thetaT) * length_;
            }
            else
            {
                // new value by equation
                const scalar localAmplitude = (amplitude_ * (1 + (coefficients_[0]*(x-1)) +
                    (coefficients_[1]*(x*x-1))));

                yr = y + localAmplitude * sin(waveNumber_*x - omega*t) * length_;
            }
        }
    }
}

```

```

    }

    p0[pointI] = vector(x, yr, z);
}
return p0;
}
else
{
    return p0;
}
}

bool Foam::fishBodyMotionFunctions::thunniformMotion::read
(
    const dictionary& SBMFCoeffs
)
{
    fishBodyMotionFunction::read(SBMFCoeffs);

    SBMFCoeffs_.readEntry("origin", origin_);
    SBMFCoeffs_.readEntry("amplitude", amplitude_);
    SBMFCoeffs_.readEntry("coefficients", coefficients_);
    SBMFCoeffs_.readEntry("waveNumber", waveNumber_);
    SBMFCoeffs_.readEntry("length", length_);
    SBMFCoeffs_.readEntry("ramp", ramp_);
    SBMFCoeffs_.readEntry("omega", omega_);
    SBMFCoeffs_.readEntry("pivot", pivot_);
    SBMFCoeffs_.readEntry("maxAngle", maxAngle_);
    SBMFCoeffs_.readEntry("phaseAngle", phase_);
    SBMFCoeffs_.readEntry("delay", delay_);

    return true;
}

// *****

```

D.2 Ika-Flow patch motion code

Listing D.12 Carangiform header file for use with the displacement laplacian patch movement method

```

/*-----*/
=====
\\      /   F i e l d           | OpenFOAM: The Open Source CFD Toolbox
\\      /   O p e r a t i o n   |
\\      /   A n d                | www.openfoam.com
\\      /   M a n i p u l a t i o n |
-----

Copyright (C) 2011–2016 OpenFOAM Foundation
-----

License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Class
Foam::carangiformDisplacement

Description
Foam::carangiformDisplacement

SourceFiles
carangiformDisplacement.C

/*-----*/

#ifndef carangiformDisplacement_H
#define carangiformDisplacement_H

#include "fixedValuePointPatchField.H"

// *****

namespace Foam
{
/*-----*/
Class carangiformDisplacement Declaration
/*-----*/

class carangiformDisplacement
:
public fixedValuePointPatchField<vector>
{
// Private Data

//-- origin at tip of head
vector origin_;

//-- Amplitude
scalar amplitude_;

//-- Coefficients
vector coefficients_;

//-- Wave number
scalar waveNumber_;

```

```

    // - Length
    scalar length_;

    // - Radial velocity
    scalar omega_;

    // - Time delay
    scalar delay_;

    // - Axis of motion
    vector axis_;

    // - Define a pointField
    pointField p0_;

public:

    // - Runtime type information
    TypeName("carangiformDisplacement");

    // Constructors

    // Construct from patch and internal field
    carangiformDisplacement
    (
        const pointPatch&,
        const DimensionedField<vector, pointMesh>&
    );

    // Construct from patch, internal field and dictionary
    carangiformDisplacement
    (
        const pointPatch&,
        const DimensionedField<vector, pointMesh>&,
        const dictionary&
    );

    // Construct by mapping given patchField<vector> onto a new patch
    carangiformDisplacement
    (
        const carangiformDisplacement&,
        const pointPatch&,
        const DimensionedField<vector, pointMesh>&,
        const pointPatchFieldMapper&
    );

    // Construct and return a clone
    virtual autoPtr<pointPatchField<vector>> clone() const
    {
        return autoPtr<pointPatchField<vector>>
        (
            new carangiformDisplacement
            (
                *this
            )
        );
    }

    // Construct as copy setting internal field reference
    carangiformDisplacement
    (
        const carangiformDisplacement&,
        const DimensionedField<vector, pointMesh>&
    );

    // Construct and return a clone setting internal field reference
    virtual autoPtr<pointPatchField<vector>> clone
    (
        const DimensionedField<vector, pointMesh>& iF
    ) const
    {
        return autoPtr<pointPatchField<vector>>

```

```

    (
        new carangiformDisplacement
        (
            *this ,
            iF
        )
    );
}

// Member functions

// Mapping functions

//-- Map (and resize as needed) from self given a mapping object
virtual void autoMap
(
    const pointPatchFieldMapper&
);

//-- Reverse map the given pointPatchField onto this pointPatchField
virtual void rmap
(
    const pointPatchField<vector>&,
    const labelList&
);

// Evaluation functions

//-- Update the coefficients associated with the patch field
virtual void updateCoeffs();

//-- Write
virtual void write(Ostream&) const;
};

// ***** //

} // End namespace Foam

// ***** //

#endif

// ***** //
```

Listing D.13 Carangiform code file for use with the displacement laplacian patch movement method

```

/*-----*/
===== |
\\      / F ield      | OpenFOAM: The Open Source CFD Toolbox
\\      / O peration   |
\\      / A nd         | www.openfoam.com
\\      / M anipulation |

-----
Copyright (C) 2011–2016 OpenFOAM Foundation

-----
License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
```

```

    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

\<-----*/

#include "carangiformDisplacement.H"
#include "pointPatchFields.H"
#include "addToRunTimeSelectionTable.H"
#include "Time.H"
#include "polyMesh.H"

// *****

namespace Foam
{
// ***** Constructors *****

carangiformDisplacement::
carangiformDisplacement
(
    const pointPatch& p,
    const DimensionedField<vector, pointMesh>& iF
)
:
    fixedValuePointPatchField<vector>(p, iF),
    origin_(Zero),
    amplitude_(0.0),
    coefficients_(Zero),
    waveNumber_(0.0),
    length_(0.0),
    omega_(0.0),
    delay_(0.0),
    axis_(Zero),
    p0_(p.localPoints())
{}

carangiformDisplacement::
carangiformDisplacement
(
    const pointPatch& p,
    const DimensionedField<vector, pointMesh>& iF,
    const dictionary& dict
)
:
    fixedValuePointPatchField<vector>(p, iF, dict),
    origin_(dict.getOrDefault<vector>("origin", Zero)),
    amplitude_(dict.get<scalar>("amplitude")),
    coefficients_(dict.lookup("coefficients")),
    waveNumber_(dict.get<scalar>("waveNumber")),
    length_(dict.get<scalar>("length")),
    omega_(dict.get<scalar>("omega")),
    delay_(dict.get<scalar>("delay")),
    axis_(dict.lookup("axis"))
{
    if (!dict.found("value"))
    {
        updateCoeffs();
    }

    if (dict.found("p0"))
    {
        p0_ = vectorField("p0", dict, p.size());
    }
    else
    {
        p0_ = p.localPoints();
    }
}

```

```

carangiformDisplacement::
carangiformDisplacement
(
    const carangiformDisplacement& ptf ,
    const pointPatch& p,
    const DimensionedField<vector , pointMesh>& iF ,
    const pointPatchFieldMapper& mapper
)
:
    fixedValuePointPatchField<vector>(ptf , p, iF , mapper),
    origin_( ptf.origin_ ),
    amplitude_( ptf.amplitude_ ),
    coefficients_( ptf.coefficients_ ),
    waveNumber_( ptf.waveNumber_ ),
    length_( ptf.length_ ),
    omega_( ptf.omega_ ),
    delay_( ptf.delay_ ),
    axis_( ptf.axis_ ),
    p0_( ptf.p0_ , mapper)
{}

carangiformDisplacement::
carangiformDisplacement
(
    const carangiformDisplacement& ptf ,
    const DimensionedField<vector , pointMesh>& iF
)
:
    fixedValuePointPatchField<vector>(ptf , iF),
    origin_( ptf.origin_ ),
    amplitude_( ptf.amplitude_ ),
    coefficients_( ptf.coefficients_ ),
    waveNumber_( ptf.waveNumber_ ),
    length_( ptf.length_ ),
    omega_( ptf.omega_ ),
    delay_( ptf.delay_ ),
    axis_( ptf.axis_ ),
    p0_( ptf.p0_ )
{}

// * * * * * Member Functions * * * * *

void carangiformDisplacement::autoMap
(
    const pointPatchFieldMapper& m
)
{
    fixedValuePointPatchField<vector>::autoMap(m);
    p0_.autoMap(m);
}

void carangiformDisplacement::rmap
(
    const pointPatchField<vector>& ptf ,
    const labelList& addr
)
{
    const carangiformDisplacement& aOVptf =
        refCast<const carangiformDisplacement>(ptf);

    fixedValuePointPatchField<vector>::rmap(aOVptf , addr);

    p0_.rmap(aOVptf.p0_ , addr);
}

void carangiformDisplacement::updateCoeffs()
{
    if ( this ->updated() )
    {
        return;
    }
}

```



```

    }

    const polyMesh& mesh = this->internalField().mesh();
    const Time& t = mesh.time();

    vectorField p0Rel_ = (p0_ - origin_)/length_;
    //const scalarField points( waveNumber_ & p0Rel_);
    scalarField xCoord = p0Rel_.component(vector::X);

    if (delay_ <= t.value())
    {
        scalar tm = t.value() - delay_;

        vectorField::operator=
        (
            (
                axis_ * amplitude_*(1 + coefficients_[0]*(xCoord-1) + coefficients_[1]*(xCoord*xCoord-1)) *
                sin((waveNumber_ * xCoord) - (omega_ * tm)) * length_
            )
        );

        fixedValuePointPatchField<vector>::updateCoeffs();
    }
}

void carangiformDisplacement::write
(
    Ostream& os
) const
{
    pointPatchField<vector>::write(os);
    os.writeEntry("origin", origin_);
    os.writeEntry("amplitude", amplitude_);
    os.writeEntry("coefficients", coefficients_);
    os.writeEntry("waveNumber", waveNumber_);
    os.writeEntry("omega", omega_);
    os.writeEntry("length", length_);
    os.writeEntry("delay", delay_);
    os.writeEntry("axis", axis_);
    p0_.writeEntry("p0", os);
    writeEntry("value", os);
}

// *****

makePointPatchTypeField
(
    pointPatchVectorField,
    carangiformDisplacement
);

// *****

} // End namespace Foam

// *****

```

Listing D.14 Carangiform header file for use with the velocity laplacian patch movement method

```

/*-----*
=====|
\\      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\      /  O p e r a t i o n |
\\      /  A n d           | www.openfoam.com
\\      /  M a n i p u l a t i o n |
-----|
      Copyright (C) 2011-2016 OpenFOAM Foundation
-----|
License

```

```

    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by
    the Free Software Foundation, either version 3 of the License, or
    (at your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Class
    Foam::carangiformVelocity

Description
    Foam::carangiformVelocity

SourceFiles
    carangiformVelocity.C

/*-----*/

#ifndef carangiformVelocity_H
#define carangiformVelocity_H

#include "fixedValuePointPatchField.H"

// ***** //

namespace Foam
{
/*-----*/
    Class carangiformVelocity Declaration
/*-----*/

class carangiformVelocity
:
    public fixedValuePointPatchField<vector>
{
    // Private Data

    //- origin at tip of head
    vector origin_;

    //- Amplitude
    scalar amplitude_;

    //- Coefficients
    vector coefficients_;

    //- Wave number
    scalar waveNumber_;

    //- Length
    scalar length_;

    //- Radial velocity
    scalar omega_;

    //- Time delay
    scalar delay_;

    //- Axis of motion
    vector axis_;

    //- Define a pointField
    pointField p0_;

public:

```

```

//~ Runtime type information
TypeName( "carangiformVelocity" );

// Constructors

//~ Construct from patch and internal field
carangiformVelocity
(
    const pointPatch&,
    const DimensionedField<vector , pointMesh>&
);

//~ Construct from patch, internal field and dictionary
carangiformVelocity
(
    const pointPatch&,
    const DimensionedField<vector , pointMesh>&,
    const dictionary&
);

//~ Construct by mapping given patchField<vector> onto a new patch
carangiformVelocity
(
    const carangiformVelocity&,
    const pointPatch&,
    const DimensionedField<vector , pointMesh>&,
    const pointPatchFieldMapper&
);

//~ Construct and return a clone
virtual autoPtr<pointPatchField<vector>> clone() const
{
    return autoPtr<pointPatchField<vector>>
    (
        new carangiformVelocity
        (
            *this
        )
    );
}

//~ Construct as copy setting internal field reference
carangiformVelocity
(
    const carangiformVelocity&,
    const DimensionedField<vector , pointMesh>&
);

//~ Construct and return a clone setting internal field reference
virtual autoPtr<pointPatchField<vector>> clone
(
    const DimensionedField<vector , pointMesh>& iF
) const
{
    return autoPtr<pointPatchField<vector>>
    (
        new carangiformVelocity
        (
            *this ,
            iF
        )
    );
}

// Member functions

// Mapping functions

//~ Map (and resize as needed) from self given a mapping object
virtual void autoMap
(
    const pointPatchFieldMapper&

```

```

    );

    //~ Reverse map the given pointPatchField onto this pointPatchField
    virtual void rmap
    (
        const pointPatchField<vector>&,
        const labelList&
    );

    // Evaluation functions

    //~ Update the coefficients associated with the patch field
    virtual void updateCoeffs();

    //~ Write
    virtual void write(Ostream&) const;
};

// ***** //

} // End namespace Foam

// ***** //

#endif

// ***** //

```

Listing D.15 Carangiform code file for use with the velocity laplacian patch movement method

```

/*-----*/
===== |
\\      /  F ield      | OpenFOAM: The Open Source CFD Toolbox
\\      /  O peration   |
\\      /  A nd         | www.openfoam.com
\\      /  M anipulation |

-----
Copyright (C) 2011–2016 OpenFOAM Foundation

-----
License
This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
for more details.

You should have received a copy of the GNU General Public License
along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

/*-----*/

#include "carangiformVelocity.H"
#include "pointPatchFields.H"
#include "addToRunTimeSelectionTable.H"
#include "Time.H"
#include "polyMesh.H"

// ***** //

namespace Foam
{

```

```
// * * * * * Constructors * * * * * //

carangiformVelocity::
carangiformVelocity
(
    const pointPatch& p,
    const DimensionedField<vector, pointMesh>& iF
)
:
    fixedValuePointPatchField<vector>(p, iF),
    origin_(Zero),
    amplitude_(0.0),
    coefficients_(Zero),
    waveNumber_(0.0),
    length_(0.0),
    omega_(0.0),
    delay_(0.0),
    axis_(Zero),
    p0_(p.localPoints())
{}

carangiformVelocity::
carangiformVelocity
(
    const pointPatch& p,
    const DimensionedField<vector, pointMesh>& iF,
    const dictionary& dict
)
:
    fixedValuePointPatchField<vector>(p, iF, dict),
    origin_(dict.getDefault<vector>("origin", Zero)),
    amplitude_(dict.get<scalar>("amplitude")),
    coefficients_(dict.lookup("coefficients")),
    waveNumber_(dict.get<scalar>("waveNumber")),
    length_(dict.get<scalar>("length")),
    omega_(dict.get<scalar>("omega")),
    delay_(dict.get<scalar>("delay")),
    axis_(dict.lookup("axis"))
{
    if (!dict.found("value"))
    {
        updateCoeffs();
    }

    if (dict.found("p0"))
    {
        p0_ = vectorField("p0", dict, p.size());
    }
    else
    {
        p0_ = p.localPoints();
    }
}

carangiformVelocity::
carangiformVelocity
(
    const carangiformVelocity& ptf,
    const pointPatch& p,
    const DimensionedField<vector, pointMesh>& iF,
    const pointPatchFieldMapper& mapper
)
:
    fixedValuePointPatchField<vector>(ptf, p, iF, mapper),
    origin_(ptf.origin_),
    amplitude_(ptf.amplitude_),
    coefficients_(ptf.coefficients_),
    waveNumber_(ptf.waveNumber_),
    length_(ptf.length_),
    omega_(ptf.omega_),
    delay_(ptf.delay_),
    axis_(ptf.axis_),

```

```

    p0_(ptf.p0_, mapper)
{}

carangiformVelocity::
carangiformVelocity
(
    const carangiformVelocity& ptf,
    const DimensionedField<vector, pointMesh>& iF
)
:
    fixedValuePointPatchField<vector>(ptf, iF),
    origin_(ptf.origin_),
    amplitude_(ptf.amplitude_),
    coefficients_(ptf.coefficients_),
    waveNumber_(ptf.waveNumber_),
    length_(ptf.length_),
    omega_(ptf.omega_),
    delay_(ptf.delay_),
    axis_(ptf.axis_),
    p0_(ptf.p0_)
{}

// * * * * * Member Functions * * * * *

void carangiformVelocity::autoMap
(
    const pointPatchFieldMapper& m
)
{
    fixedValuePointPatchField<vector>::autoMap(m);

    p0_.autoMap(m);
}

void carangiformVelocity::rmap
(
    const pointPatchField<vector>& ptf,
    const labelList& addr
)
{
    const carangiformVelocity& aOVptf =
        refCast<const carangiformVelocity>(ptf);

    fixedValuePointPatchField<vector>::rmap(aOVptf, addr);

    p0_.rmap(aOVptf.p0_, addr);
}

void carangiformVelocity::updateCoeffs()
{
    if (this->updated())
    {
        return;
    }

    const polyMesh& mesh = this->internalField().mesh();
    const Time& t = mesh.time();

    vectorField p0Rel_ = (p0_ - origin_)/length_;
    //const scalarField points( waveNumber_ & p0Rel_);
    scalarField xCoord = p0Rel_.component(vector::X);

    if (delay_ <= t.value())
    {
        scalar tm = t.value() - delay_;

        vectorField::operator=
        (
            (
                axis_ * amplitude_*(1 + coefficients_[0]*(xCoord-1) + coefficients_[1]*(xCoord*xCoord-1)) *
                omega_ * cos((waveNumber_ * xCoord) - (omega_ * tm)) * length_

```

```

        )
    );

    fixedValuePointPatchField<vector>::updateCoeffs();
}

}

void carangiformVelocity::write
(
    Ostream& os
) const
{
    pointPatchField<vector>::write(os);
    os.writeEntry("origin", origin_);
    os.writeEntry("amplitude", amplitude_);
    os.writeEntry("coefficients", coefficients_);
    os.writeEntry("axis", axis_);
    os.writeEntry("omega", omega_);
    os.writeEntry("length", length_);
    os.writeEntry("waveNumber", waveNumber_);
    os.writeEntry("delay", delay_);
    p0_.writeEntry("p0", os);
    writeEntry("value", os);
}

// *****

makePointPatchTypeField
(
    pointPatchVectorField,
    carangiformVelocity
);

// *****

} // End namespace Foam

// *****

```

D.3 Ansys Fluent UDF code

Listing D.16 Thunniform UDF for Ansys Fluent

```

DEFINE_GRID_MOTION(Thunniform, domain, dt, time, dtime)
{
    /*This function defines the grid motion

    Parameters:
        Thunniform: name.
        Name of the UDF.
        domain: fluent object.
        Domains store data associated with a collection of nodes
        dt: fluent object.
        something to do with threads.
        time: double.
        time.
        dtime: double.
        change in time.

    */
    Thread* tf = DT_THREAD(dt);
    face_t f;
    Node* v;
    real A, xbar, hDot, thetaDot, endY, hDotTail; //Dynamic parameters
    real k, w, theta, thetaMax, phi, len, lenPed; //static parameters
    real c[3]; // static array
    real NV_VEC(v), NV_VEC(rvec), NV_VEC(origin); // dynamic arrays
    int n;

    c[0] = 0.012765;
    c[1] = -0.4003876;
    c[2] = 2.35321267;
    w = 34.852351;
    k = 0.43332312;

    thetaMax = 16.00 * M_PI / 180;
    phi = 90.0 * M_PI / 180;
    len = 1.0
    lenPed = 0.81 * len;

    NV_D (origin, =, 0.0, 0.00, 0.0);

    begin_f_loop(f, tf)
    {
        f_node_loop(f, tf, n)
        {
            // define the face node
            v = F_NODE(f, tf, n);

            if (NODE_POS_NEED_UPDATE(v))
            {
                NODE_POS_UPDATED(v);

                if (time > 0)
                {
                    if(NODE_X(v) <= lenPed)
                    {
                        //normalize the x-coordinate with body length
                        xbar = NODE_X(v) - origin[0];

                        // Amplitude Envelope
                        A = c[0] + (c[1] * xbar) + (c[2] * pow (xbar, 2));

                        // Calculates the new midline velocity
                        hDot = A * w * cos((w * time) - (k * xbar));

                        // update the velocity vector
                        NV_D (vel, =, 0.0, hDot, 0.0);

                        // update velocity to give a position
                        NV_S (vel, +=, dtime);

                        NV_V(NODE_COORD(v), +=, vel);
                    }
                }
            }
        }
    }
}

```



```

else
{
    //normalize the x-coordinate with body length
    xbar = NODE_X(v) - origin[0] - lenPed;

    // Amplitude Envelope
    A = c[0] + (c[1] * lenPed) + (c[2] * pow(lenPed, 2));

    // Calculates the new midline velocity
    hDot = A * w * cos((w * time) - (k * lenPed));

    // Find the y-coordinate for the preduncle
    NV_D(rvec, =, 0.0, hDot, 0.0);

    NV_S(rvec, *=, dtime);

    NV_V(rvec, +=, NODE_COORD(v));

    // calculate the lag of the tail
    theta = sin(thetaMax) * sin((w * time) - (k * lenPed) - phi);
    thetaDot = sin(thetaMax) * w * cos((w * time) - (k * lenPed) - phi);

    hDotTail = xbar * (1 / pow(cos(theta), 2)) * thetaDot;

    // update the velocity vector
    NV_D(v, =, 0.0, hDotTail, 0.0);

    // update velocity to give a position
    NV_S(v, *=, dtime);

    // add in the y-location of the tail
    NV_V(v, +=, rvec);

    NV_V(NODE_COORD(v), =, v);
}

}

}

}

end_f_loop(f, tf);
}

```

