

Wire-driven Mechanism and Highly Efficient Propulsion in Water

LI, Zheng

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
Doctor of Philosophy
in
Mechanical and Automation Engineering

The Chinese University of Hong Kong
July 2013

Thesis/Assessment Committee

Professor Yangsheng Xu (Chair)

Professor Ruxu Du (Thesis Supervisor)

Professor Yunhui Liu (Committee Member)

Professor Max Qinghu Meng (Committee Member)

Professor Kamal Youcef Toumi (External Examiner)

Abstract

Attracted by the outstanding performance of natural creatures, researchers have been mimicking animals and plants to develop their robots. Inspired by animals' musculoskeletal system, especially the skeletal structure of snakes and octopus arm muscle arrangement, in this thesis, a novel wire-driven mechanism (WDM) is designed. It is composed of a flexible backbone and a number of controlling wire groups. The flexible backbone provides support, while the wire groups transmit motion and force from the actuators, mimicking the muscles. According to its backbone structure, the WDM is categorized as serpentine WDM and continuum WDM. Depending on the backbone segmentation, WDM is divided into single segment WDM and multi-segment WDM. Each segment is controlled by one or two wire groups. Features of WDM include: flexible, highly under-actuated, leverage effect, and long range force and motion transmission. The flexibility enables the WDM making large deformation, while the under-actuation greatly reduces the number of actuators, simplifying the system. With the leverage effect, WDM distal end velocity and acceleration is greatly amplified from that of wire. Also, in the WDM, the actuators and the backbone are separated. Actuator's motion is transmitted by the wires. This makes the WDM very compact. With these features, the WDM is not only well suited to confined space, but also flapping propulsion, especially in water.

In the thesis, the design, kinematics, workspace, static and dynamic models of the WDM are explored systematically. Under the constant curvature assumption, the kinematic model of serpentine WDM and continuum WDM are established. A generalized model is also developed. Workspace model is built from the forward kinematic model. Rather than avoiding obstacles, a novel idea of employing obstacles or actively deploying constraints to expand workspace is also discussed for WDM-based flexible manipulators. The static model and dynamic model of serpentine WDM is developed using the Newton-Euler method and the Lagrange Equation, while that of continuum WDM is built under the non-linear Euler-Bernoulli Beam theory and the extended Hamilton's principle.

In the thesis, a number of novel WDM based underwater propulsors are developed. Compared with existing fish-like propulsor designs, including single

joint design, multi-joint design, and smart material based continuum design, the proposed WDM-based propulsors have advantages in several aspects, such as employing less actuators, better resembling the fish swimming body curve, ease of control, and more importantly, being highly efficient. Also, brand new propulsors can be easily developed using the WDM. To demonstrate the features as well as the advantages of WDM propulsors, four robot fish prototypes are developed. Experiments show that the serpentine WDM-based propulsor could provide large flapping force while the continuum WDM-based propulsor is less affected by joint friction. On the other hand, single segment WDM propulsor can make oscillatory swim while multi- segment WDM propulsor can make both oscillatory and undulatory swims. The undulatory swimming outperforms the oscillatory swimming in stability and speed, but is inferior in turning around. In addition, a novel robot fish with vector propulsion capability is also developed. It can provide thrust in arbitrary directions, hence, improving the maneuverability of the robot fish. In the experiments, with the power limit of two watts, the maximum forward speed of the WDM robot fishes can reach 0.67 BL (Body Length)/s. The minimum turning radius is 0.24 BL, and the turning speed is 51.4°/s. The maximum Froude efficiency of the WDM robot fishes is 92.85%. Finally, the WDM-based propulsor is used to build an indoor Lighter-than-Air-Vehicle (LTAV), named Flying Octopus. It is suspended in the air by a helium balloon and actuated by four independently controlled wire-driven flapping wings. With the wing propulsion, it can move in 3D space effectively.

摘要

自然生物的杰出表现往往令人们叹为观止。正因为如此，在机器人研究中对自然界动植物的模仿从未间断。本文受动物肌肉骨骼系统（尤其是蛇的脊柱以及章鱼手臂的肌肉分布）的启发，设计了一种新型的仿生拉线机构。该机构由柔性骨架以及成对拉线组成。柔性骨架提供支撑，拉线模拟肌肉将驱动器的运动和力传递给骨架，并控制骨架运动。从骨架结构分，拉线机构可分为蛇形拉线机构以及连续型拉线机构；从骨架分段来看，拉线机构可分为单段式拉线机构以及多段式拉线机构，其中每段由一或两对拉线控制。拉线机构的主要性能特征包括：大柔性，高度欠驱动，杠杆效应，以及远程传力。机构的柔性使得它可以产生很大的弯曲变形；欠驱动设计极大地减少了驱动器的数目，简化了系统结构；在杠杆效应下，骨架末端速度、加速度与拉线的速度、加速度相比得到数十倍放大；通过拉线将驱动器的运动和力远程传递给执行机构，使得拉线机构结构简单紧凑。基于以上特征，拉线机构不仅适合工作于狭窄空间，同时也适合于摆动推进，尤其是水下推进。

论文系统地介绍了拉线机构的设计，运动学，工作空间，静力学以及动力学模型。在常曲率假设下分别建立了蛇形拉线机构以及连续型拉线机构的运动学模型，在此基础上建立了一个通用运动学模型，以及工作空间模型。与传统避障相反，本文提出了一种利用现有障碍或主动布置约束来拓展工作空间的新方法。通过牛顿-欧拉法以及拉格朗日方程建立了蛇形拉线机构的静力学模型以及动力学模型。在非线性欧拉-伯努利梁理论下结合哈密顿原理建立了连续型拉线机构的静力学模型以及动力学模型。

论文中利用拉线机构设计了一系列新型水下推进器。与传统机器鱼推进器设计方法（单关节，多关节以及基于智能材料的连续型设计）相比，基于拉线机构的水下推进器的优点在于：所需驱动器少，能更好地模拟鱼的游动，易于控制，推进效率高，以及容易衍生新型推进器。设计制作了四条拉线驱动机器鱼，以此为平台验证了拉线推进器的性能以及优点。实验结果表明，基于蛇形拉线机构的推进器可以提供较大推力；基于连续型

拉线机构设计的推进器受摩擦影响较小；基于单段式拉线机构的推进器可以模仿鱼类摆动式推进，具有很好的转弯性能；基于多段式拉线机构的推进器可以同时模仿摆动式推进和波动式推进，具有更好的稳定性以及游速。此外，基于拉线机构制造了一种新型矢量推进器。该推进器可以提供任意方向的推力，从而提高机器鱼的机动性能。实验中，在两个额定功率为 1 瓦的电机驱动下，机器鱼的最大游速为 0.67 体长/秒；最小转弯半径为 0.24 倍体长；转弯速度为 51.4 度/秒；最高推进效率为 92.85%。最后，采用拉线推进器制作了一个室内空中移动机器人，取名为 Flying Octopus。它由一个氦气球提供浮力悬停在空中，通过四个独立控制的拉线扑翼驱动可在三维空间自由运动。

Acknowledgement

First, I would like to express the most sincere appreciation to my supervisor, Prof. Ruxu Du. Without him, I won't have the opportunity to study in CUHK. Without his never-ending support and invaluable suggestions, I won't be able to finish the dissertation in smooth. His guidance is conducive not only to my research but more importantly to my future life. The life philosophy he showed to me will help me conquer any difficulties in the future.

Second, I want to say thank you to all the people who helped me fulfilling the PhD degree. Special thanks to my thesis committee members, Prof. Kamal, Prof. Yangsheng Xu, Prof. Yunhui Liu and Prof. Max, Meng. Their insightful comments and valuable suggestions helped boosting the thesis grade to another level. Thanks to the department staff members, Allan, Joyce, Maggie, Thomas, etc. Thanks to all my colleagues, Dr. Longhan Xie, Dr. Xianshuai Chen, Mr. Mancheong Lei, Mr. Yupei Yao, Dr. Yuhua Guo, and Mr. Yingnan Wang, Baofeng Liao, Yong Zhong, etc. They helped me a lot in the PhD study, directly or indirectly. I would like to thank all my friends, Jiayun Shen, Xin Zhang, Chirs Zhang, Chuanhao Li, Fei Fang, Yapin Dou, Zhiyong He, Molei Zhan, Ting Gu, Maobin Lv, Jianwei Chen, etc. They helped in enriching my campus life, which is the lubricant to my research.

At last, I want to express my deepest gratitude to my warmest family members. My dear parents gave me life and the best enlightenment. Their unselfish devotion and forever encouragement is my solidest support. My beloved wife went through with me every happiness and depression. Her ardent love and innermost understanding is the sweetest thing I've ever had. My grandparents, parents in law, sisters and other family members all gave me the ubiquitous care. Thank you so much!

Table of Contents

Abstract	i
摘要	iii
Acknowledgement	v
List of Figures	xi
List of Tables	xvii
Chapter 1 Introduction	1
1.1 Background	1
1.2 Related Research	2
1.2.1 Flexible Manipulator	2
1.2.2 Robot Fish	10
1.3 Motivation of the Dissertation.....	13
1.4 Organization of the Dissertation	14
Chapter 2 Biomimetic Wire-Driven Mechanism	16
2.1 Inspiration from Nature.....	16
2.1.1 Snake Skeleton	18
2.1.2 Octopus Arm	19
2.2 Wire-Driven Mechanism Design	20
2.2.1 Flexible Backbone.....	20
2.2.2 Backbone Segmentation	26
2.2.3 Wire Configuration	28
2.3 Wire-Driven Mechanism Categorization	31
2.4 Summary	32
Chapter 3 Kinematics and Workspace of the Wire-Driven Mechanism	33
3.1 Kinematic Model of Single Segment WDM.....	33
3.1.1 Kinematic Model of the Serpentine WDM	34
3.1.2 Kinematic Model of the Continuum WDM	39
3.1.3 A Generalized Kinematic Model.....	43
3.2 Kinematic Model of Multi-Segment WDM.....	47
3.2.1 Forward Kinematics	47
3.2.2 Inverse Kinematics	51
3.3 Workspace.....	52

3.3.1	Workspace of Single Segment WDM	52
3.3.2	Workspace of Multi-Segment WDM	53
3.4	Employing Obstacles to Expand WDM Workspace.....	55
3.4.1	Constrained Kinematics Model of WDM	55
3.4.2	WDM Workspace with Constraints	61
3.5	Model Validation via Experiment.....	64
3.5.1	Single Segment WDM Kinematic Model Validation	64
3.5.2	Multi-Segment WDM Kinematic Model Validation	66
3.5.3	Constrained Kinematic Model Validation.....	70
3.6	Summary	73
Chapter 4	Statics and Dynamics of the Wire-Driven Mechanism	75
4.1	Static Model of the Wire-Driven Mechanism.....	75
4.1.1	Static Model of SPSP WDM.....	75
4.1.2	Static Model of SPCP WDM	81
4.2	Dynamic Model of the Wire-Driven Mechanism	88
4.2.1	Dynamic Model of SPSP WDM	88
4.2.2	Dynamic Model of SPCP WDM.....	92
4.3	Summary	94
Chapter 5	Application I - Wire-Driven Robot Fish.....	95
5.1	Fish Swimming Introduction	95
5.1.1	Fish Swimming Categories	95
5.1.2	Body Curve Function	96
5.1.3	Fish Swimming Hydrodynamics.....	101
5.1.4	Fish Swimming Data.....	103
5.2	Oscillatory Wire-Driven Robot Fish.....	104
5.2.1	Serpentine Oscillatory Wire-Driven Robot Fish Design.....	105
5.2.2	Continuum Oscillatory Wire-Driven Robot Fish Design.....	110
5.2.3	Oscillatory Robot Fish Propulsion Model.....	114
5.2.4	Robot Fish Swimming Control	116
5.2.5	Swimming Experiments	118
5.3	Undulatory Wire-Driven Robot Fish	125
5.3.1	Undulatory Wire-Driven Robot Fish Design	125
5.3.2	Undulatory Wire-Driven Robot Fish Propulsion Model.....	130
5.3.3	Swimming Experiments	131

5.4 Vector Propelled Wire-Driven Robot Fish	136
5.4.1 Vector Propelled Wire-Driven Robot Fish Design	136
5.4.2 Tail Motion Analysis.....	140
5.4.3 Swimming Experiments	142
5.5 Wire-Driven Robot Fish Performance and Discussion.....	144
5.5.1 Performance	144
5.5.2 Discussion	147
5.6 Summary	149
Chapter 6 Application II - Wire-Driven LTAV – Flying Octopus.....	151
6.1 Introduction.....	151
6.2 Flying Octopus Design	152
6.2.1 Flying Octopus Body Design	152
6.2.2 Wire-Driven Flapping Wing Design	153
6.3 Flying Octopus Motion Control.....	156
6.3.1 Propulsion Model	156
6.3.2 Motion Control Strategy.....	157
6.3.3 Motion Simulation.....	159
6.4 Prototype and Indoor Experiments	161
6.4.1 Flying Octopus Prototype.....	161
6.4.2 Indoor Experiments	163
6.4.3 Discussion	165
6.5 Summary	166
Chapter 7 Conclusions and Future Work	167
Appendix A - Publication Record	170
Appendix B - Derivation	172
Appendix C – Matlab Programs	176
References	205

List of Figures

Figure 1-1 Tensor Arm Manipulator: Side View of the Tensor Arm Manipulator (left) and Partial Cross-Section View of the Plate (right) [8].....	3
Figure 1-2 Amadeus: Grasping Objects (left); Bellow Configuration (right) [13]	3
Figure 1-3 The Elephant Trunk Robot: Hook Configuration (left); Schematic of the Manipulator Section (right) [15].....	4
Figure 1-4 Tentacle Manipulator: Spatial (left); Planar (right) [31, 32]	4
Figure 1-5 OctArm: OctArm V Prototype (left); Muscle Configuration (right) [37]	5
Figure 1-6 Air-Octor: Air-Octor Prototype (left); Cable Configuration (right) [38]	5
Figure 1-7 Octopus Arm: Muscle Configuration (left); Prototype (right) [47].....	6
Figure 1-8 Octopus Arm Inspired Robots in Caldwell's group [48] [49]	7
Figure 1-9 Snake Arm Robots and Simulator Developed by OC Robotics [50]..	7
Figure 1-10 Bionic Handling Assistant [51]	8
Figure 1-11 Continuum Robots Introduced by Nabil Simaan's Group [52]	8
Figure 1-12 Continuum Robots Introduced by Pierre E. Dupont's Group [65].....	9
Figure 1-13 Robert Webster's Active Cannulas [75]	9
Figure 1-14 Compliant Robot Fish Developed by MIT [113]	13
Figure 2-1 Examples of Flexible Parts in Nature	17
Figure 2-2 Skeleton of a Rattle Snake [122, 123]: (left) Sliding; (right) Coiling	18
Figure 2-3 Octopus Arm and Its Muscle Arrangement[124]	19
Figure 2-4 Two Types of Vertebra Articulations	20
Figure 2-5 Vertebra Design - Planar Serpentine Backbone	21
Figure 2-6 Planar Serpentine Backbone	22
Figure 2-7 Vertebra Design - Spatial Serpentine Backbone	23
Figure 2-8 Joint Motion - Spatial Serpentine Backbone	23
Figure 2-9 Planar Continuum Backbone	24
Figure 2-10 Spatial Continuum Backbone	25
Figure 2-11 Single Segment WDM with Continuum Backbone.....	27
Figure 2-12 Multi-Segment WDM with Continuum Backbone.....	28
Figure 2-13 Parallel Wire Configuration.....	29

Figure 2-14 Tapered Wire Configuration	30
Figure 2-15 Spiral Wire Configuration	30
Figure 2-16 Wire-Driven Mechanism Categories	31
Figure 3-1 Kinematics Defined by Mappings between the Spaces	33
Figure 3-2 Joint of the SXSP WDM.....	34
Figure 3-3 WDM Bending in Arbitrary Direction	36
Figure 3-4 Mapping between Configuration Space and Task Space – Serpentine	38
Figure 3-5 Joint of the SXCP WDM	40
Figure 3-6 Mapping between Configuration Space and Task Space – Continuum	42
Figure 3-7 Comparison of Serpentine WDM and Continuum WDM	44
Figure 3-8 WDM Distal End Velocity and Leverage Effect	47
Figure 3-9 Single Segment WDM Coordinate Transform	48
Figure 3-10 Multi-Segment WDM Coordinate Transform	50
Figure 3-11 Wire Pair Shift Angle	50
Figure 3-12 Workspace of a Single Segment Spatial WDM.....	53
Figure 3-13 Trajectories of the WDM Distal End with Increased Joint Number	53
Figure 3-14 Workspace of a Three-Segment WDM in the X - Z Plane. (a) $N=10$, Simulation Interval is 2.5° ; (b) $N=10$, Simulation Interval is 10° ; (c) $N=8$, Simulation Interval is 2.5° ; (d) $N=12$, Simulation Interval is 2.5°	54
Figure 3-15 Three Types of WDM Motion: (a) without Constraint; (b) with Bilateral Constraint; (c) with Unilateral Constraint	56
Figure 3-16 Process of Kinematics under Constraint.....	56
Figure 3-17 WDM with Constraint	57
Figure 3-18 Inverse Kinematics: (a) Solution with Exact Position; (b) Solution with Exact Orientation and Exact X position; (c) Solution with Exact Orientation and Exact Z position	60
Figure 3-19 Workspace with Single Bilateral Constraint.....	62
Figure 3-20 Workspace with Single Unilateral Constraint	63
Figure 3-21 Manipulator with SPSP WDM	64
Figure 3-22 Trajectory of the WDM Manipulator End Effector	65

Figure 3-23 Relative Positioning Error of the Manipulator	66
Figure 3-24 Three-Segment WDM Manipulator.....	66
Figure 3-25 Three-Segment WDM Manipulator Bending Cases.....	67
Figure 3-26 Trajectories Comparison of the Distal End	68
Figure 3-27 Relative Positioning Error of the Distal End	69
Figure 3-28 WDM Manipulator Moving with External Bilateral Constraint.....	70
Figure 3-29 End Effector Trajectory Comparison - Internal Bilateral Constraint	71
Figure 3-30 End Effector Trajectory Comparison - External Bilateral Constraint	72
Figure 3-31 End Effector Trajectory Comparison - Unilateral Constraint.....	73
Figure 4-1 SPSP WDM Static Analysis	76
Figure 4-2 SPSP WDM Static Analysis: (a) Deformed Backbone Curve; (b) Joint Rotations.....	78
Figure 4-3 SPSP WDM Backbone Reaches a Desired Position: (a) Deformed Backbone Curve; (b) Joint Rotations	81
Figure 4-4 SPCP WDM Static Analysis.....	82
Figure 4-5 Validity of Constant Curvature Assumption at Pure Moment Loading Condition	86
Figure 4-6 SPCP WDM Backbone Deformation under Different Loading Conditions.....	87
Figure 4-7 Dynamic Modeling of Serpentine WDM	88
Figure 5-1 Fish Swimming Categories (a) BCF Propulsion and (b) MPF Propulsion[131].....	96
Figure 5-2 Simplified Fish Swimming Categorization	96
Figure 5-3 Fish Swimming Body Curve - Oscillatory	97
Figure 5-4 Oscillatory Body Curve Comparison: (a) One Straight Line and One Circular Arc Fitting; (b) Two Straight Lines and One Circular Arc Fitting	98
Figure 5-5 Fish Swimming Body Curve – Undulatory	99
Figure 5-6 Undulatory Body Curve Comparison: (a) Three Straight Lines and Three Circular Arcs Fitting; (b) Six Straight Lines and Three Circular Arcs Fitting	99
Figure 5-7 Two Circular Arcs Fit the Undulatory Fish Swimming Body Curve	100

Figure 5-8 Spinal Column Configuration at Two Successive Instants [136].....	101
Figure 5-9 Fish Swimming Velocity Scaled to Body Length (data from [144])	104
Figure 5-10 Fish Swimming Strouhal Number (data from [144])	104
Figure 5-11 Serpentine Oscillatory Wire-Driven Flapping Tail Design	105
Figure 5-12 Serpentine Tail Joint Kinematics.....	106
Figure 5-13 Wire Length Change Approximation Error - Serpentine	107
Figure 5-14 Serpentine Oscillatory Wire-Driven Tail Flapping Cycle.....	108
Figure 5-15 Oscillatory Wire-Driven Robot Fish Body Design	108
Figure 5-16 Wire Connection.....	109
Figure 5-17 Designed Serpentine Oscillatory Robot Fish.....	110
Figure 5-18 Serpentine Oscillatory Wire-Driven Robot Fish Prototype.....	110
Figure 5-19 Continuum Oscillatory Wire-Driven Flapping Tail Design.....	111
Figure 5-20 Eyelet Connection.....	111
Figure 5-21 Continuum Backbone with Tapered Wire Configuration.....	112
Figure 5-22 Wire Length Change Approximation Error - Continuum.....	113
Figure 5-23 Continuum Oscillatory Wire-Driven Tail Flapping Cycle	114
Figure 5-24 Continuum Oscillatory Wire-Driven Robot Fish Prototype.....	114
Figure 5-25 Oscillatory Flapping Tail Coordinate Frame Setting.....	116
Figure 5-26 Oscillatory Flapping Cycle	116
Figure 5-27 Robot Fish Swimming Control Scheme	117
Figure 5-28 Robot Fish Swimming Experiment Setup	118
Figure 5-29 Serpentine Oscillatory Wire-Driven Robot Fish Swimming Forward	120
Figure 5-30 Flapping Cycle in the Turning Mode.....	121
Figure 5-31 Oscillatory Flapping Wire-Driven Robot Fish Turning Trajectory	122
Figure 5-32 Continuum Oscillatory Wire-Driven Robot Fish Cruising Example	123
Figure 5-33 Continuum Oscillatory Wire-Driven Robot Fish Turning Example	124
Figure 5-34 Undulatory Wire-Driven Tail Design.....	126
Figure 5-35 Vertebra Design and Joint Rotation.....	127
Figure 5-36 S-Shape and C-Shape Bending of the Wire-Driven Tail.....	127
Figure 5-37 Undulatory Swimming Curve Comparison	128
Figure 5-38 Undulatory Wire-Driven Robot Fish Body Design	129

Figure 5-39 Undulatory Wire-Driven Robot Fish Prototype	130
Figure 5-40 Undulatory Wire-Driven Tail Coordinate Frame Setting	130
Figure 5-41 Oscillatory and Undulatory Swimming	132
Figure 5-42 Undulatory Wire-Driven Robot Fish Turning	135
Figure 5-43 Vector Propulsor Design: (a) Vector Propulsor Isometric View; (b) Vertebra Top View	136
Figure 5-44 Joint Rotation: (a) Joint at Rest; (b) Joint Rotating to the Right	137
Figure 5-45 Fish Body Design	138
Figure 5-46 Vector Propelled Robot Fish Prototype: (a) Robot Fish in the Rest Position; (b) Propulsor Bending Horizontally; (c) Propulsor Bending Vertically; (d) Propulsor Bending in an Arbitrary Direction	139
Figure 5-47 Tail Bending and Wire Configuration	140
Figure 5-48 Vector Propulsor Planar Flapping	141
Figure 5-49 Vector Propulsor Spatial Flapping.....	142
Figure 5-50 Experiment Results: (a)-(e) is the Flapping Cycle of Shark Form Swimming; (f)-(j) is the Flapping Cycle of Dolphin Form Swimming.....	143
Figure 5-51 Wire-Driven Robot Fish Swimming Speed Scaled to Body Length	145
Figure 5-52 Strouhal Number of the Wire-Driven Robot Fish in Experiments .	146
Figure 6-1 Flying Octopus Design	153
Figure 6-2 Wire-Driven Flapping Wing Design	154
Figure 6-3 Flying Octopus Force Analysis	156
Figure 6-4 Thrust Force Illustration	157
Figure 6-5 Flying Octopus Motion Simulation	160
Figure 6-6 Simulation Results in 30 Flapping Cycles.....	161
Figure 6-7 Flying Octopus Prototype	162
Figure 6-8 Flapping Cycle of Flying Upward	163
Figure 6-9 Flying Octopus Velocity in the First Six Flapping Cycles	164
Figure 6-10 Flapping Cycle of Flying in <i>X</i> Direction	165

List of Tables

Table 1-1 Typical Macro-Scale Robot Fish	11
Table 1-2 Typical Meso-Scale Robot Fish	12
Table 3-1 Segment Bending Angles in the Experiment	68
Table 5-1 Tail Vertebra Parameters	105
Table 5-2 Serpentine Oscillatory Wire-Driven Tail Motion Parameters	115
Table 5-3 Continuum Oscillatory Wire-Driven Tail Motion Parameters	115
Table 5- 4 PWM Duty Cycle for Swimming (1/20).....	118
Table 5-5 Influence of Flapping Amplitudes on Velocity - Serpentine	119
Table 5-6 Influence of Flapping Frequency on Velocity - Serpentine	120
Table 5-7 Influence of Flapping Amplitude on Velocity - Continuum	123
Table 5-8 Influence of Flapping Frequency on Velocity - Continuum	124
Table 5-9 Vertebra Parameters	127
Table 5-10 Undulatory Wire-Driven Tail Motion Parameters	131
Table 5-11 Big-C-Motion Swimming Experiment Result	133
Table 5-12 Small-C-Motion Swimming Experiment Result.....	133
Table 5-13 S-Motion Swimming Experiment Result.....	134
Table 5-14 Vertebra Dimensions.....	137
Table 6-1 Eyelet Parameters.....	155
Table 6-2 Forces on Flying Octopus	156
Table 6-3 Five Basic Motions	159
Table 6-4 Flying Octopus Bill of Materials	162

Chapter 1 Introduction

In this chapter, the background of this research is presented. Related work and the status of current research are reviewed, which is followed by the motivations of this research. At the end, the organization of the thesis is given.

1.1 Background

Robots are considered as the solution to many of the world's hot issues, such as labor shortage, medical care, nuclear safety, extreme environment exploration, etc. The United States sees robotics as one of its key strategies to promote the industry [1]. In 2011, the manufacturing tycoon Foxconn announced its robot plan – building one million robots to perform the repetitive tasks and to address the labor issue [2]. In 2011, NASA launched the Curiosity to Mars [3]. In 2012 China sent the deep-sea submersible Jiao Long down to the Mariana Trench, reaching a depth of 7,062 meters [4]. All these show that robotics research is entering into a new era.

From the structural point of view, robots are categorized as discrete robots, serpentine robots and continuum robots [5]. Both serpentine robots and continuum robots are flexible. Compared with discrete rigid robots, the flexible robots have more degrees of freedom (DOF) and are well suited to confined spaces. They have wide applications, such as nuclear reactor inspection, minimally invasive surgery, disaster relief, etc. The research of flexible robots, including mechanism design, modeling and control has grown rapidly recently, especially after the Fukushima disaster in 2011. In flexible robot development, researchers tend to seek inspiration from nature, such as snakes, elephant trunks, octopus arms, mammal tongues, etc. On the other hand, with the growing desire for ocean exploration and water quality monitoring, highly efficient underwater robots, e.g. robot fish, are of great interest. Compared with traditional screw propellers, fish-like flapping propulsion is advantageous in aspects such as, efficiency, maneuverability, noise, etc. Europe launched the SHOAL project in 2009 to develop autonomous robot fish to inspect the harbors around the

Mediterranean [6]. The Mexico gulf oil spill in 2010 further promoted the research of robot fish to perform water quality monitoring and treatment [7].

Mechanism design is fundamental in robot development. It determines the motion, as well as the task a robot can perform. Robot fish's flapping tail deforms itself to interact with water. From the mechanism point of view, it is the same as that of a flexible robot. However, this seems to be ignored by most researchers. In the past, the researches of these two types of robots have little overlap.

1.2 Related Research

Research related to this work includes that focusing on the flexible manipulator, especially underactuated flexible manipulator (UFM), and robot fish.

1.2.1 Flexible Manipulator

The flexible manipulator usually has tens of DOFs. Its body is flexible and capable of large deformation. It was generally regarded as firstly introduced by Anderson and Horn in 1967 [8]. Their robot is named "tensor arm manipulator", and the patent was authorized in 1970 [9]. When the number of actuators is less than the flexible robot's DOFs, it is underactuated and is called an underactuated flexible manipulator (UFM). The structure of the UFM can be serpentine or continuum. The actuation method of a UFM is diverse. Some typical actuation methods are: Pneumatic Artificial Muscle (PAM); Shape Memory Alloy (SMA); Electro Active Polymer (EAP), especially Iron Polymer Metal Composite (IPMC); Cable/Tendon/Wire driven [10-12] and concentric tubes. The following shows some well-known flexible manipulators.

The Tensor Arm Manipulator [8, 9] was developed in 1967 by V. C. Anderson and R. C. Horn. It comprises a series of plates which are interconnected by universal joints. The plates have a number of apertures through which the tendons can pass by. The tendons are connected to the plates. The plates can be pivoted to various positions by pulling the tendons. In this design, the manipulator is fully actuated, as shown in Figure 1-1. As a result, the number of tendons needed is enormous. The structure of the manipulator is complicated and the control of this manipulator is very difficult.

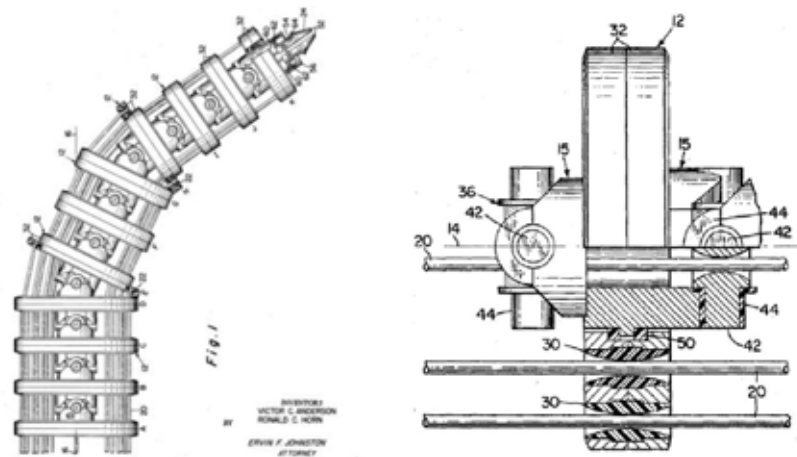


Figure 1-1 Tensor Arm Manipulator: Side View of the Tensor Arm Manipulator (left) and Partial Cross-Section View of the Plate (right) [8]

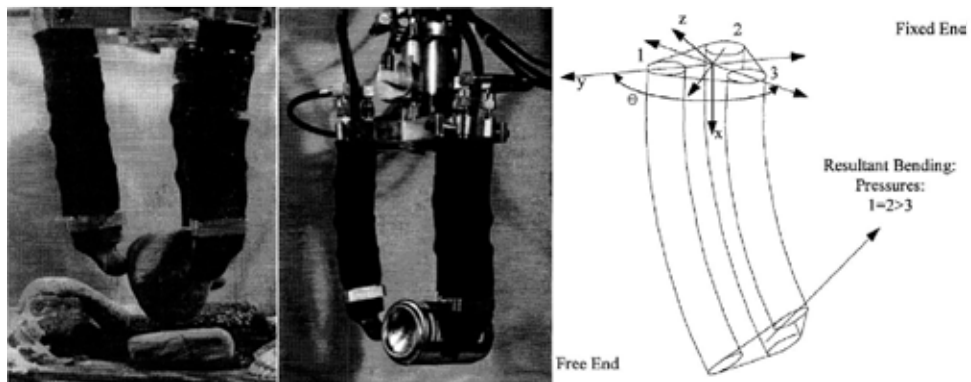


Figure 1-2 Amadeus: Grasping Objects (left); Bellow Configuration (right) [13]

The Amadeus [13, 14] (advanced manipulator for deep underwater sampling) is an international subsea manipulator project, funded by the European Union Marine Science and Technology Research Program. The first Amadeus prototype was completed in 1996 by G. Robinson, J. Davies and J. Jones. It has a three fingered hydraulic end-effector. The fingers have a continuum structure without moving parts. The finger motion is generated by the elastic deformation of the bellows. The compliant continuum fingers can grasp irregularly shaped objects. Also, they passively react to disturbances within the environment. This increases the grasping stability and reduces the risk of contact damage. In this project, applications are studied for the continuum manipulator, such as the “dual-trunk inspection device” and the “FLAPS fin-ray”. Figure 1-2 shows the Amadeus robot. The left-hand figure shows it grasping an apple, the middle figure shows it grasping a can, and the right-hand figure is the bellow configuration of the

manipulator. In this configuration, the three actuators are evenly displaced. The angular spacing is 120° . This actuator spatial configuration is adopted by most of the subsequent continuum robots. In this configuration, the number of actuators needed for 3D bending is three, which is the minimum. However, in the bending actuators' motions are coupled.

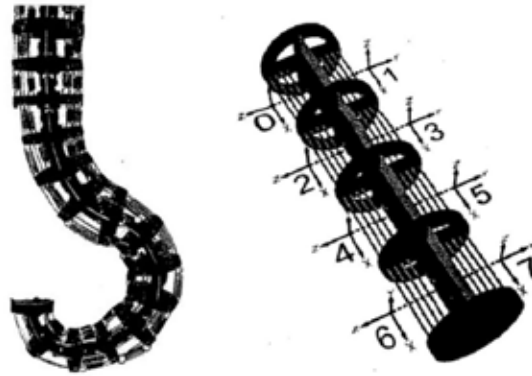


Figure 1-3 The Elephant Trunk Robot: Hook Configuration (left); Schematic of the Manipulator Section (right) [15]

The Clemson Elephant Trunk Robot [15-30] is a 4-section, 8-DOF manipulator developed by I. D. Walker and M. W. Hannan in 1999. Figure 1-3 shows the manipulator. The overall motion of the manipulator is controlled by a tendon servo system. There are in total 16 joints and each joint has two DOFs. Among all the 32 DOFs, only eight DOFs are controlled by the tendons. The remaining DOFs are constrained by the springs. The total length of the manipulator is 83.32 cm and the total mass is 4.0 Kg. Diameters of the four sections are 10.16 cm, 8.89 cm, 7.62 cm and 6.35 cm respectively. It is underactuated and the structure is simpler compared with the Tensor Arm Manipulator. However, the system is still very complicated and the positioning error is large.

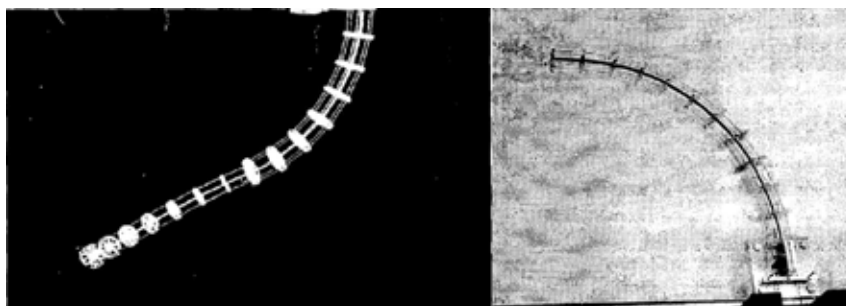


Figure 1-4 Tentacle Manipulator: Spatial (left); Planar (right) [31, 32]

The Clemson tentacle manipulator [31, 32] is a multi-segment cable-driven manipulator with a continuum backbone. It was first introduced by I. A. Gravagne, C. D. Rahn and I. D. Walker in 2000. Figure 1-4 shows the robot. With this manipulator, the kinematics, vibration and manipulability of the continuum robot are studied [16, 31-36]. Compared with the elephant trunk robot, the number of moving parts is greatly reduced. However, the deformation is limited by the backbone's elasticity. When the deformation is large, the kinematics model is highly nonlinear. Also, the payload capability is small.



Figure 1-5 OctArm: OctArm V Prototype (left); Muscle Configuration (right) [37]

The OctArm is a continuum robot developed by M. D. Grissom, I. D. Walker et al. in the early 2000s [18, 37]. Several OctArms were developed, such as OctArm IV has four sections and OctArm V has three sections. Figure 1-5 shows the OctArm V. It is actuated by the pneumatic artificial muscle. Each segment has three muscles. The muscle configuration is similar to that of the Amadeus robot. The load capacity of the OctArm is large compared with other continuum robots. For OctArm IV the vertical load capacity is 90 N and the transverse load capacity is 16 N; for OctArm V these are 220 N and 70 N respectively. However, the response time is long. For OctArm V the extension time is 2.5 s.

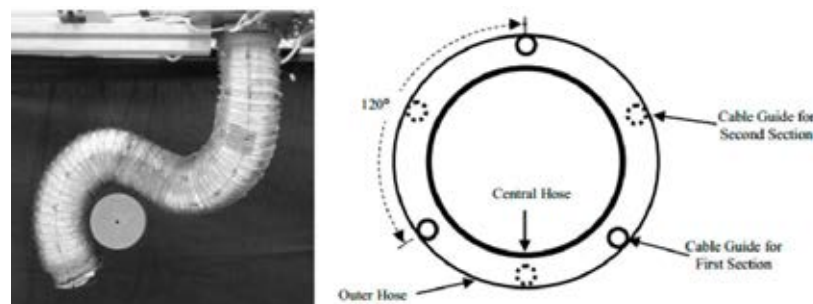


Figure 1-6 Air-Octor: Air-Octor Prototype (left); Cable Configuration (right) [38]

Air-Octor is a two-section continuum robot developed by W. McMahan, B. A. Jones and I. D. Walker in 2005 [36, 38-46]. Figure 1-6 shows the Air-Octor prototype. Its backbone is a pneumatically pressurized chamber. It uses a combination of motors and pneumatic pressure regulators for actuation. Each section has three DOFs: two bending DOFs and one extension DOF. The bending is controlled by three cables, and the extension is controlled by the air pressure. The outer diameter of the Air-Octor is 9 cm and the length can be controlled between 31 cm and 95 cm. For this robot, compliance is also controllable. However, precision motion control is difficult. Also, air leakage is a concern.



Figure 1-7 Octopus Arm: Muscle Configuration (left); Prototype (right) [47]

Inspired by the octopus arm, in 2009, Laschi, C. and his group proposed a robot arm [47]. It could bend in all directions, elongate, contract and control the stiffness. The robot uses the muscular hydrostat phenomenon, in which the volume does not change during the muscle contraction, to achieve these motions. Figure 1-7 shows the robot. The left-hand figure shows the muscle arrangement and the right-hand figure shows the robot prototype. There are two types of muscles: the longitudinal muscle and the transversal muscle. The bending is controlled by the longitudinal muscle and the elongation is controlled by the transversal muscle. When the transversal muscle contracts, the diameter of the arm reduces. Due to the hydrostat, the length of the robot increases. In this design, the muscle is a stack of EAP.

Guglielmino, Emanuele, Caldwell et al. introduced a continuum manipulator in 2010 [48]. The robot has 15 actuated degrees of motion (DOM) and eight DOFs as shown on the left of Figure 1-8. The actuation is achieved by pneumatic artificial muscles, which imitate the octopus arm's longitudinal muscle and transversal muscle. The robot has two sections. Different from the previous robots,

this robot not only has two bending DOFs and one translation DOF but is also capable of controlling the radial diameter. The diameter is controlled by the three transversal muscles. The problem with this robot is that it is soft with very limited payload capacity. Also, positioning and control is problematic. Another octopus arm-like robot was introduced in 2012 [49] as shown in the right-hand part of Figure 1-8. The bending motion of this robot is controlled by the cables and the diameter is controlled by the SMA.

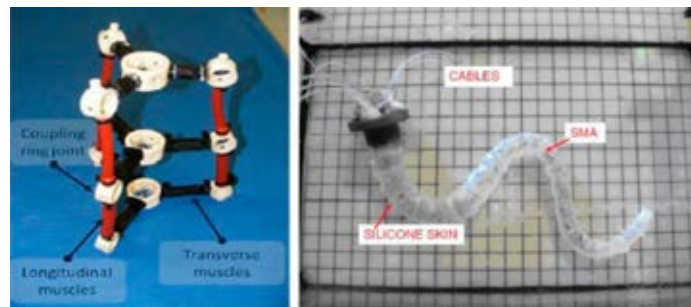


Figure 1-8 Octopus Arm Inspired Robots in Caldwell's group [48] [49]

In the industry, OC robotics developed a snake arm robot [50]. It has as a similar structure as in the tensor arm manipulator and the elephant trunk robot. The connections between the links are universal joints. The bending motion is controlled by the tendons. The snake arm is placed on a moving platform, which provides the translation motion. It is used to inspect the engineering parts, such as engines, aircraft, etc. A simulator was also developed for better operation. The snake arm won the Queen's Award for Enterprise: Innovation in 2009. This is the most successful UFM at that time.



Figure 1-9 Snake Arm Robots and Simulator Developed by OC Robotics [50]

Another successful industrial example is the Bionic Handling Assistant made by Festo [51]. It is a compliant, pneumatically actuated continuum manipulator for

cooperative manipulation. It has three sections and each section has two bending DOFs, as shown in Figure 1-10. The configuration is similar to that of the OctArm. It won the 2010 German Future Award. This again shows the world's recognition of UFM's.



Figure 1-10 Bionic Handling Assistant [51]

In recent years, UFM's have been of great interest especially in medical device development. Compared with the previous UFM's, the size of medical UFM's is small. Hence, the continuum structure is more frequently chosen. Also, the structure material and actuation method need to be bio-compatible. Therefore, tendon, cable or wire driven and concentric tube designs are the most frequently selected. Researchers such as Nabil Simaan, Pierre E. Dupont and Robert J. Webster III have been frequent contributors in this field.

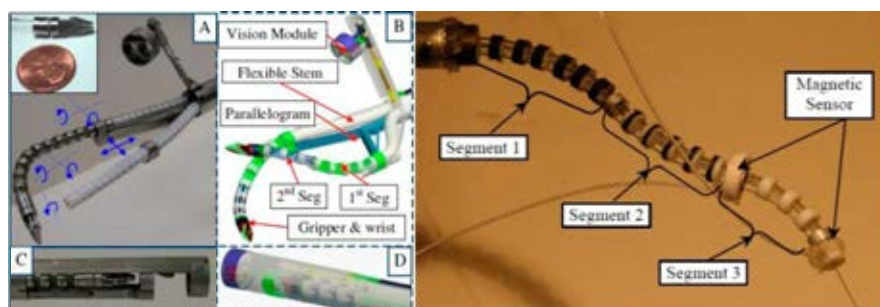


Figure 1-11 Continuum Robots Introduced by Nabil Simaan's Group [52]

Nabil Simaan and his team developed several insertable continuum robotic end-effectors for single port access surgery [52-64], as shown in Figure 1-11. These robots have a continuum backbone. Along the backbone are a couple of discs to guide the tendons. The backbone is partitioned into several segments. Each segment has two bending DOFs. The bending motions of each segment are controlled by pulling and pushing the three tendons. The tendon configuration is

the same as that of the tensor arm manipulator. These tendons also serve as the secondary backbone. With these robotic end-effectors, configuration estimation, kinematics, statics, force sensing capabilities, etc. are studied.

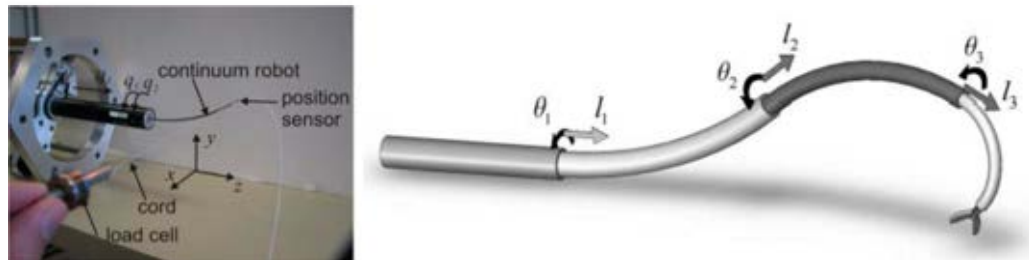


Figure 1-12 Continuum Robots Introduced by Pierre E. Dupont's Group [65]

Pierre E. Dupont and his team's continuum robots employ the concentric tube design [65-74]. As shown in Figure 1-12, the robot backbone is composed of three concentric elastic tubes. Each tube is pre-curved. The shape of the backbone is a combination of the three tubes' deformation. The end effector position and orientation are controlled by the rotation and insertion of the tubes with respect to each other. With this novel design, the robot is very compact and the size can be very small, which is well suited for medical applications. However, precise positioning and trajectory control is not easy. Also, material fatigue is a concern.

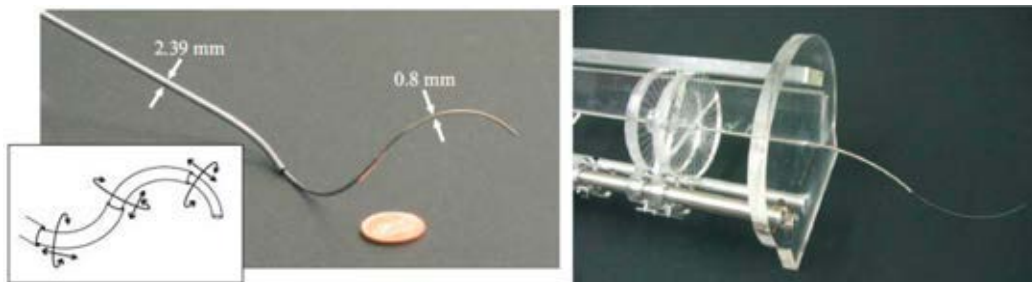


Figure 1-13 Robert Webster's Active Cannulas [75]

Robert J. Webster III and his team also worked on concentric tube robots [5, 75-88]. This type of robot is also named active cannulas. As shown in Figure 1-13, the outer diameter of the robot is less than 2.5 mm. With this robot, they worked on the kinematics, mechanics, calibration, visual sensing, path planning, Jacobian and compliance matrices, etc.

1.2.2 Robot Fish

Inspired by fish and other aquatic species, people have been building robot fishes for two decades since the robot tuna [89]. People are surprised by the efficiency and agility of their models. It is believed that fish's propulsion efficiency could reach 90% [89]. The speed of a sailfish could exceed 110 km/h [90], and the maximum recorded acceleration of a pike is 249 m/s^2 , which is over 25 g [91]. The outstanding performance is achieved by multiple aspects, such as flapping tail, streamlined body, mucous surface, etc. Among all these factors, the most important one is their flapping tail, which is their actuation system, and also is the main target that people have been imitating.

In fish, there are two types of propulsions, i.e. body and/or caudal fin (BCF) propulsion and medium and/or paired fin (MPF) propulsion [14]. BCF accounts for 85% of fish species. Within BCF, the motion modes are further divided into four categories, i.e. thunniform, carangiform, subcarangiform and anguilliform. Roughly speaking, the thunniform is viewed as oscillatory form and the other three are viewed as undulatory form. BCF is adopted by most of the fast swimmers, such as sailfish, tuna, pike, etc. Examples for MPF are manta ray and box fish. Compared with BCF swimmers, fish swimming by MPF is agile. In robot fish development, high speed and high efficiency are the main targets. Therefore, most current robot fishes adopt BCF, especially the undulatory form as the way to generate propulsion.

The fundamental motion of BCF propulsion is flapping. There are a couple of ways to generate the flapping motion. One method is by using traditional mechanisms, e.g. crank, four-bar mechanism, etc. It transforms rotation to back and forth motion, and then to the tail's flapping motion. In this method, the control is simple and the flapping is powerful. The drawback is that the tail is rigid and the flapping motion is far unlike the fish's swimming motion and lowers the propulsion efficiency. To better approximate the fish's swimming body curve, passive flexible fins can be used for this method. But the improvement is limited. Another way is by controlling the motion of serially linked motors, such as the robot fish introduced by [89, 92-97]. This is a widely used method in robot fish development. In this method, the fish body curve is fitted by a polyline, which contains several straight line segments. Compared with the single joint method,

the motion in this method is closer to the fish's swimming body curve. However, motor synchronization is important. Also, the system becomes complicated with the increasing number of motors, and the energy consumption is high. Smart materials, such as Ionic Polymer Metal Composite (IPMC) [98-100], Shape Memory Alloy (SMA) [101-105] and Piezoelectric Material (PZT) [106-109] could also be used to generate the flapping motion. These materials can bend into a circular arc under control. The deformed shape can be very close to that of a swimming fish segment. However, deficiencies also exist. For IPMC flapping fins, they are soft and the force generated is limited. For SMA flapping fins, due to the heating and cooling process, the frequency is limited. Also, it is difficult to control the flapping amplitude and motion. For PZT material, high voltage is needed and the strain is small. As a result, an additional mechanism to amplify the motion is needed. Another common drawback of the smart materials is that the material itself is highly energy inefficient.

Table 1-1 Typical Macro-Scale Robot Fish




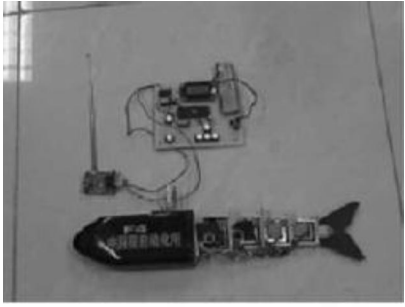
	
<p>MIT - Robot Tuna [110]</p>	<p>Essex - G8 [111]</p>
	
<p>BUAA-SPC II [112]</p>	<p>CAS Robot Fish [92]</p>

Table 1-1 shows some famous macro scale robot fish. They all have multiple links and are actuated by motors. For example, MIT's robot tuna has seven links and is driven by six actuators. It was the world's first robot fish and was tested with a strut in the water tank. The propulsion efficiency is reported close to 90%. The Essex G8 robot fish has four links and is driven by four motors. It was famous for its resembling a real fish, and was exhibited in the London aquarium. The maximum instantaneous turning speed of this robot is 450°/s. The SPC II developed by Beihang University (BUAA) has two joints and is driven by two motors. The maximum speed of this robot fish is 1.5 m/s, or 1.22 BL/s. It was used to inspect a sunk warship in Taiwan Strait in 2004. During the task, it patrolled a 4000 m² water area in 6 hours. The robot fish developed by the Chinese Academy of Science (CAS) also has four links and is driven by four motors. The maximum speed is 0.32 m/s, or 0.8 BL/s, when the flapping frequency is 2 Hz. The maximum turning speed is 68.8°/s.

Table 1-2 Typical Meso-Scale Robot Fish

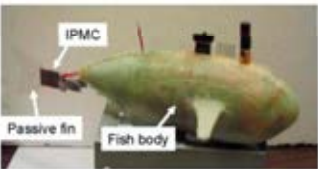
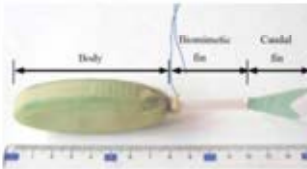
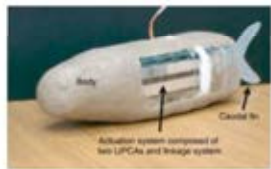
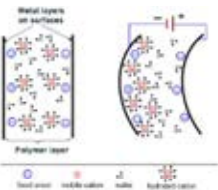
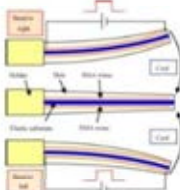
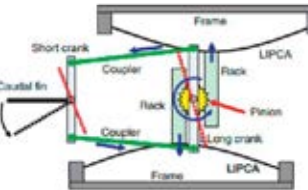
		
		
<p>IPMC Actuated Robot Fish – MSU [99]</p>	<p>SMA Actuated Robot Fish – HIT [101]</p>	<p>PZT Actuated Robot Fish – KU [106]</p>

Table 1-2 shows some representative meso-scale robot fishes driven by smart materials. Their working principle is also shown in the table, such as the robot fish developed by Michigan State University employs IPMC as the flapping tail. The maximum speed of this robot fish is 2.2 cm/s, or 0.096 BL/s. The robot fish developed in Harbin Institute of Technology employs SMA to drive the caudal fin. The maximum speed of this robot fish is 11.2 cm/s or 0.767 BL/s and the minimum turning radius is 0.952 BL. The Strouhal number at maximum speed is

0.58. The robot fish built by Konkuk University uses PZT to actuate the caudal fin. The maximum speed of this robot fish is 2.5 cm/s or 0.093 BL/s, and the Strouhal number is between 0.8 and 1.6, which means the swimming is inefficient.

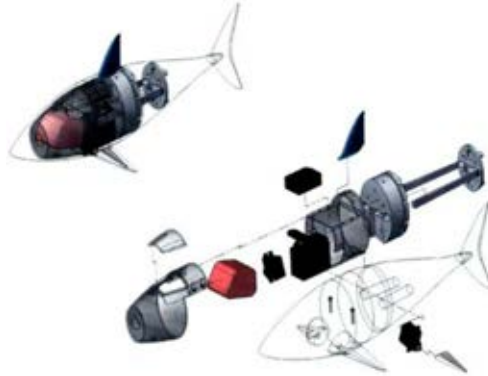


Figure 1-14 Compliant Robot Fish Developed by MIT [113]

In recent years, robot fish with compliant body is of interest. One example is the robot fish proposed by Pablo and Kamal in 2007, as shown in Figure 1-14 [113]. The robot fish's body is soft and is actuated by one motor. By tuning the body rigidity and placing lumped masses, the mode shape of the body is changed. As a result, the tail flapping can be controlled by the body excitation. Compared with the previous designs, this underactuated design reduces the number of actuators needed and simplifies the robot fish's structure. The drawback is that the flapping curve is related to the flapping frequency. This constrains the control of flapping curve. Besides, when the flapping frequency is high, the power consumption increases sharply. The maximum cruising speed of this robot fish is 1 BL/s, and the maximum hydrodynamic efficiency is less than 60%. However, due to the low motion transmission efficiency, the total propulsive efficiency is less than 0.15%.

1.3 Motivation of the Dissertation

Flexible robots have many advantages over traditional rigid discrete robots, such as being well-suited to confined spaces. However, the mechanisms suitable for flexible robots are limited. Also, the theoretical modeling of UFMs is not fully studied. On the other hand, fish-like flapping propellers are typical flexible structures. However, there is little overlap between these two researches. These motivated the research herein. The research contents include:

- Designing a mechanism capable of large deformation for UFM.
- Studying the kinematics, workspace, statics and dynamics model of the mechanism.
- Designing fish-like flapping propellers using the proposed mechanism.
- Developing the propulsion model of the flapping propellers.
- Evaluating the performance of the flapping propellers.

1.4 Organization of the Dissertation

The remainder of the dissertation is organized as follows:

Chapter 2 presents the biomimetic Wire-Driven Mechanism (WDM) design. The nature inspiration is presented at first. Then the design is described from three aspects: backbone structure, backbone segmentation and wire configuration. At last, WDM categorization and coding is given. One important categorization is by the backbone type, from which the WDM is categorized as serpentine WDM and continuum WDM.

Chapter 3 presents the forward kinematics, inverse kinematics and workspace models of the serpentine WDM and continuum WDM. A generalized kinematics model suitable for the two types of WDM is proposed subsequently. Constrained kinematics model is proposed as well. Furthermore, a novel idea of expanding the workspace by actively deploying constraints is proposed. Finally, the models and ideas are validated using two wire-driven robot arm prototypes.

Chapter 4 develops the static model and dynamic model of serpentine WDM and continuum WDM. For serpentine WDM, the static model is built using the Newton-Euler method and the dynamic model is developed using the Lagrange method. For continuum WDM, the nonlinear Euler-Bernoulli Beam theory is used to develop both the static model and the extended Hamiltonian principle is used in developing the dynamic model.

Chapter 5 presents the underactuated wire-driven robot fishes, including design, propulsion modeling and experiments. Four robot fishes with different wire-driven flapping propulsors are designed and prototyped. A simplified propulsion model is developed based on Lighthill's Elongated Body Theory. Cruise speed,

Froude efficiency and Strouhal number are used to evaluate the performance of the wire-driven robot fishes. At last, the performance of the robot fishes is compared with that of real fish.

Chapter 6 shows another application of the wire-driven flapping propulsor. An indoor Lighter-Than-Air-Vehicle (LTAV), named Flying Octopus, propelled by four independently controlled wire-driven flapping wings is designed and prototyped. Motion simulation and indoor experiments are also carried out.

Chapter 7 summarizes the contributions of this dissertation and suggests a few future research topics.

Chapter 2 Biomimetic Wire-Driven Mechanism

In this chapter, the biomimetic Wire-Driven Mechanism (WDM) is introduced. It is a highly underactuated flexible mechanism. The design is inspired by the skeleton of snakes and octopus arm muscle arrangement. It is composed of a flexible backbone and controlling wire pairs. The design of the WDM is presented in three aspects: backbone structure, backbone segmentation and wire configuration. Finally, the categorization and coding of the WDM is presented.

2.1 Inspiration from Nature

Nature is full of flexibility. Living creatures, whether animals or plants, are hardly rigid. As an example, snakes, octopus, caterpillars, plant tentacles, etc. are fully flexible. For some creatures, take the monkey as an example, even though some parts of the body, such as elbow and thigh, are somewhat rigid, some appendages are flexible, like the fingers, spinal column, tongue and tail. It is noted that, in the above examples, the flexibility falls into two categories. Snakes, monkey fingers, spinal columns and tails are composed of a series of short bony links, with adjacent links forming a joint. Although the rotation of each joint is small, the deformation of the structure can be very large due to the large number of joints. It can deform into complex shapes, such as an S shape or the like, but cannot extend or change the cross-sectional size. Octopus arms, caterpillars, plant tentacles and mammal tongues are continuous without any apparent joints. The flexibility comes from the material itself, such as octopus arm muscle, caterpillar tissue and plant tentacle fiber are all soft. They can deform largely under external load or internal actuation.

Figure 2-1 shows some examples of flexible living creatures or their appendages. Figure 2-1 (a) shows the white lined sphinx moth caterpillar [114]. Its body is soft and it can move around by deforming itself actively. Figure 2-1 (b) shows the octopus [115]. Its body, especially its arms, is highly flexible. By using the arms, the octopus can even open a jar and squeeze into it. Figure 2-1 (c) shows a giraffe grabbing tree leaves using its tongue [116]. Figure 2-1 (d) shows a climbing

morning glory [117]. It climbs along the tree branches by coiling its body. The caterpillar and octopus are soft animals, while the giraffe tongue and the glory tentacle are soft appendages. In these examples, there are no apparent joints. The flexibility of the whole body or the appendages comes from the compliance of materials.



(a) White Lined Sphinx Moth [114]



(b) Octopus [115]



(c) Giraffe Tongue [116]



(d) Climbing Morning Glory [117]



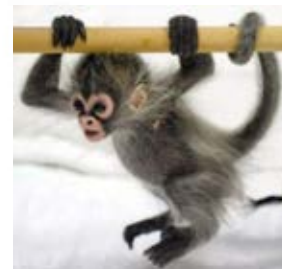
(e) Rattle Snake [118]



(f) Human Finger [119]



(g) European Eel [120]



(h) Monkey Tail [121]

Figure 2-1 Examples of Flexible Parts in Nature

Figure 2-1 (e) shows a rattle snake coiling its body. Figure 2-1 (f) shows a human figure flexion. Figure 2-1 (g) shows a European eel swimming. Figure 2-1 (h) shows a monkey wrapping its tail and fingers around a horizontal bar. In these four examples, the flexible skeleton is composed of a series of short rigid bones. Adjacent bones form a joint, and the skeleton is covered by soft muscles and skin. The joint revolution is controlled by muscle contraction. At the same time, the muscle and skin deforms along with the joint motion.

These are just a few examples, there are a lot more. Among all the examples from nature, there are two examples of special interest to us. One is the snake skeleton and the other is the octopus arm.

2.1.1 Snake Skeleton

Figure 2-2 shows the skeleton of a sliding snake and coiling snake. From an anatomical point of view, the snake body is composed of four major parts: vertebral column, muscle, skin and viscera. The muscle, skin and viscera are all soft. The snake body curve is shaped by the vertebral column. It consists of a series of similar bony vertebrae extending from the skull occipital to the tail tip. Two successive vertebrae together with the interposed intervertebral disc form a joint. These joints can rotate in both yaw direction and pitch direction. Although the rotation of each joint is small, the overall snake body deformation is large. As shown in Figure 2-1 (e) and Figure 2-2, a snake can easily bend over 360° .



Figure 2-2 Skeleton of a Rattle Snake [122, 123]: (left) Sliding; (right) Coiling

It is interesting to note that when a snake slides, its body has several apparent circular arcs. The joint rotations in each circular arc segment are nearly the same. Also, when the snake coils, its body is spiral. In Figure 2-2, the coiling shape is almost a circle. Again, the joint rotations are close to each other. In both cases,

the joint rotations change gradually. There is no abrupt change in adjacent joint rotation. This implies that the joint rotations are correlated, or there exists some constraints between the joint rotations. Where do these constraints come from? The vertebral column is composed of tens of rigid bony vertebrae. Mechanically, the joints are free of rotation. Therefore, we can infer that the constraints come from the intervertebral disc, or from the muscle and skin. They are all compliant. When deformed, they generate a restoring force like a spring. For all the intervertebral discs their dimensions are similar. The muscle and skin are continuous. Also, there is no abrupt change in cross-section dimension. Hence, the restoring forces acting on each joint are similar. It should be noted that the major function of the muscle is to actively deform the vertebral column. On the other hand, the ribs connected to the vertebrae form the body cavity which holds all the viscera. They provide a conduit, allowing food, water, etc. to pass by.

2.1.2 Octopus Arm

Octopus arms are slender and highly flexible with infinite degrees of freedom. It can bend, twist as well as elongate. Figure 2-3 shows an octopus arm and its muscle arrangement. It is seen that there are three types of muscles in an octopus arm: longitudinal muscle (*L*), transversal muscle (*T*) and external oblique muscle (*O*). There are four bundles of longitudinal muscles. They are separated by the transversal muscle and are orthogonally distributed. The oblique muscles wrap around the longitudinal muscles and transversal muscle. The shape is like a helix.

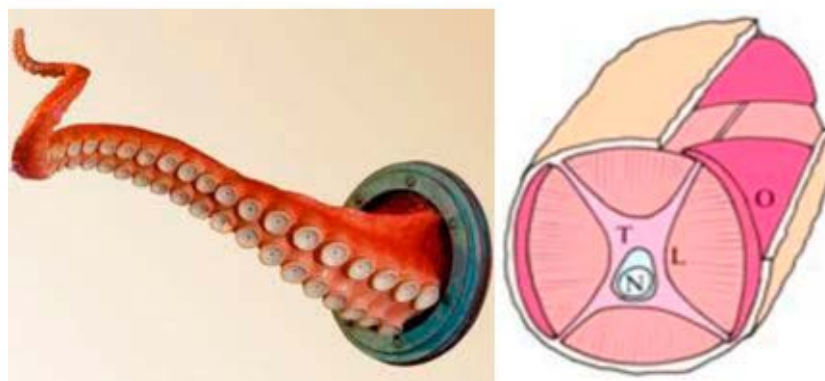


Figure 2-3 Octopus Arm and Its Muscle Arrangement[124]

From the picture, the longitudinal muscles make up the major part of the arm. They are responsible for the bending motion. As an example, when the left

muscle contracts the octopus arm bends to the left; when the right muscle contracts, the octopus arm will at first recover to the initial straight state and then bend to the right. Octopus arm elongation is achieved by transversal muscle contraction. Muscles are hydrostatic, which means that the volume does not change during the contraction. When the transversal muscle contracts, the longitudinal muscle is squeezed. As a result, the cross-sectional area of the longitudinal muscle is reduced and its length is increased. With the longitudinal muscle being elongated, the octopus arm extends as well. The oblique muscle is responsible for the twisting. As it spirals along the longitudinal muscle, when it contracts, the force exerted on the octopus arm has a tangent component and an axial component. This makes the octopus arm twist about its own axis.

2.2 Wire-Driven Mechanism Design

The snake skeleton and octopus arm muscle arrangement inspired the design of the wire-driven mechanism (WDM). The WDM has two parts: one is the flexible backbone and the other is the controlling wire pairs. The backbone structure follows the snake's skeleton, while the wire configuration is similar to an octopus arm's longitudinal muscle arrangement.

2.2.1 Flexible Backbone

As indicated previously, in nature, there are two categories of flexible structures: one is serpentine and the other is continuum. For the WDM backbone, it is the same. The backbone type is determined by the vertebra articulation. When the vertebrae are articulated by joints, the backbone is serpentine. If the vertebrae are connected by a continuous flexible beam, the backbone is continuum.

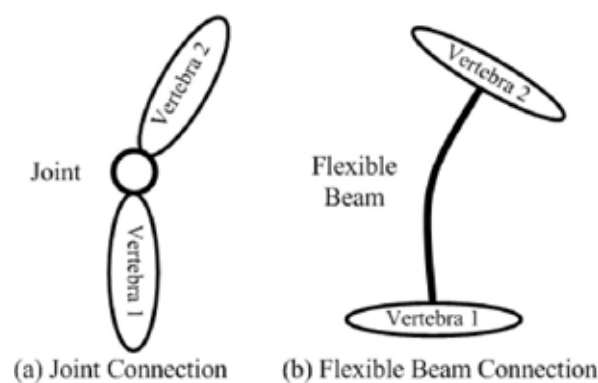


Figure 2-4 Two Types of Vertebra Articulations

Figure 2-4 illustrates the two types of vertebra articulations. Figure 2-4 (a) shows the joint connection and Figure 2-4 (b) shows that the vertebrae are connected by a continuous flexible beam. In the serpentine backbone, the vertebrae are the main body. In the continuous backbone, the vertebrae are degenerated to a spacing disc. The flexible beam is the major part of the backbone.

In nature, the motion of these flexible structures has two types. One is planar bending, such as finger flexion, and the other is spatial, such as monkey tail wrapping. This brought about the design of the planar backbone and spatial backbone. Considering the backbone structure, there are four flexible backbone categories: 1) planar serpentine backbone; 2) spatial serpentine backbone; 3) planar continuum backbone and 4) spatial continuum backbone.

1) Planar Serpentine Backbone Design

For serpentine backbone, vertebrae are critical as they are the main body and determine the joint type. For planar serpentine backbone, the vertebra motion is confined to a plane. The vertebrae are articulated by revolute joints. Figure 2-5 shows an example of planar vertebra design.

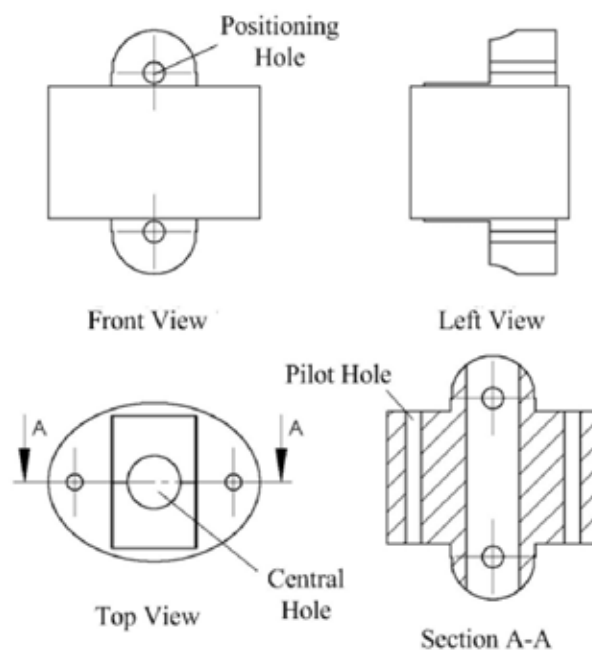


Figure 2-5 Vertebra Design - Planar Serpentine Backbone

The cross-section of the vertebra is elliptic. On the top and bottom of the vertebra, there is a semicircular stage. The positioning hole in the middle of the stage is

used to fasten the joint connection. It is also the joint rotation center. As shown in the A-A cross-section view, there are two pilot holes in the vertebra. The wires penetrate the vertebra via the pilot holes. At the middle of the vertebra, there is a central hole. All the central holes of the vertebrae form a central cavity. Just like the body cavity of a snake, it provides a passage for water, or other tools.

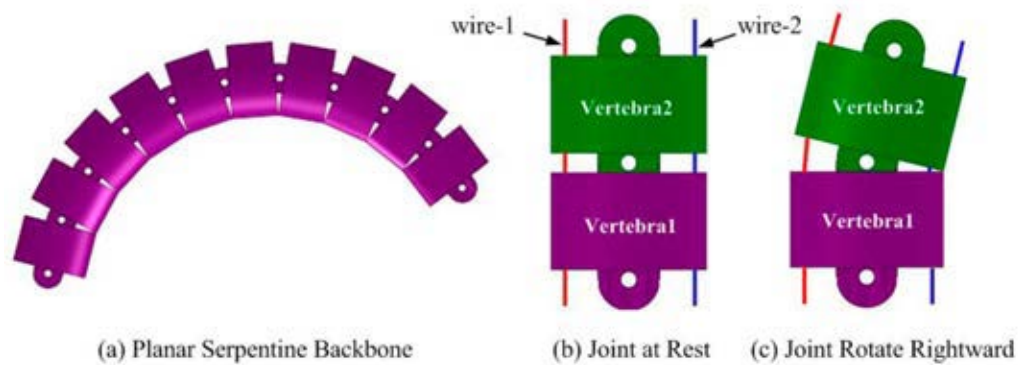


Figure 2-6 Planar Serpentine Backbone

By connecting all the joints together, the planar serpentine backbone is formed as shown in Figure 2-6 (a). Theoretically, there is no limitation on the joint number. In the figure, 10 vertebrae are shown. In the figure, the joint connection is also shown. Figure 2-6 (b) shows the joint in the rest position and Figure 2-6 (c) shows the joint rotates to the right. In this design, the vertebra is symmetrical. Hence, the left rotation and right rotation are the same. The maximum joint rotation is limited by the vertebra dimensions. To constrain the joint motion as in the snake skeleton, a compliant rubber tube is placed in the center cavity. With uniform cross section area, the tube constrains all the joint motion equally.

2) Spatial Serpentine Backbone Design

In the spatial serpentine backbone, the joints rotate in both yaw and pitch directions. A straightforward design is to use spherical joints to articulate the vertebrae. Figure 2-7 shows a design example.

The cross-section of the vertebra is a circle. A convex spherical surface is at the top of the vertebra. At the bottom, there is a concave spherical surface. The two surfaces have the same radius. Hence, two adjacent vertebrae form a spherical joint. The rolling is mechanically constrained. As a result, each joint has two DOFs, one is the yawing, and the other is the pitching. To control the two

motions, at least three wires are needed, i.e. at least three pilot holes are required. In this design, following the octopus arm longitudinal muscle arrangement, we use four wires to control the two DOFs. Each wire corresponds to a pilot hole on the vertebra. Therefore, the pilot holes are orthogonally distributed. Compared with the commonly used three-wire configuration, the four-wire configuration can decouple the two bending motions, simplifying the control. Considering that there can be more than one set of wires, the number of pilot holes is $4X$, where X is the number of wire sets. In this example, the number of pilot holes is 12. The same as the planar vertebra design, in the spatial vertebra there is a central hole, which can hold elastic rod or tube.

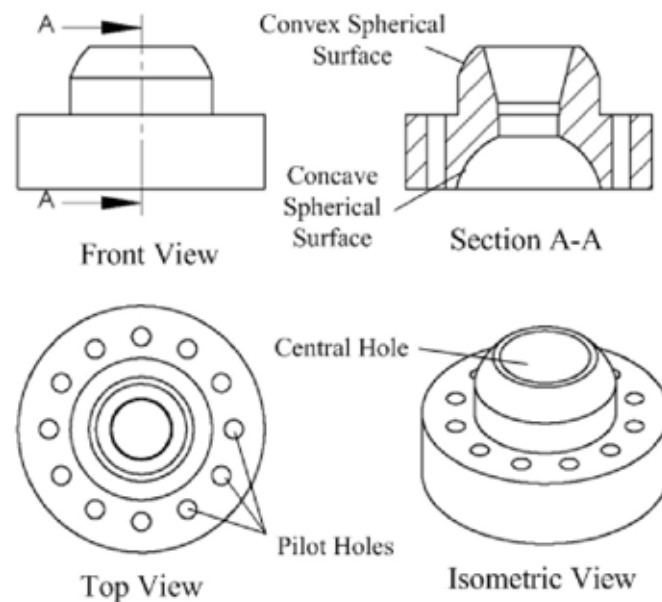


Figure 2-7 Vertebra Design - Spatial Serpentine Backbone

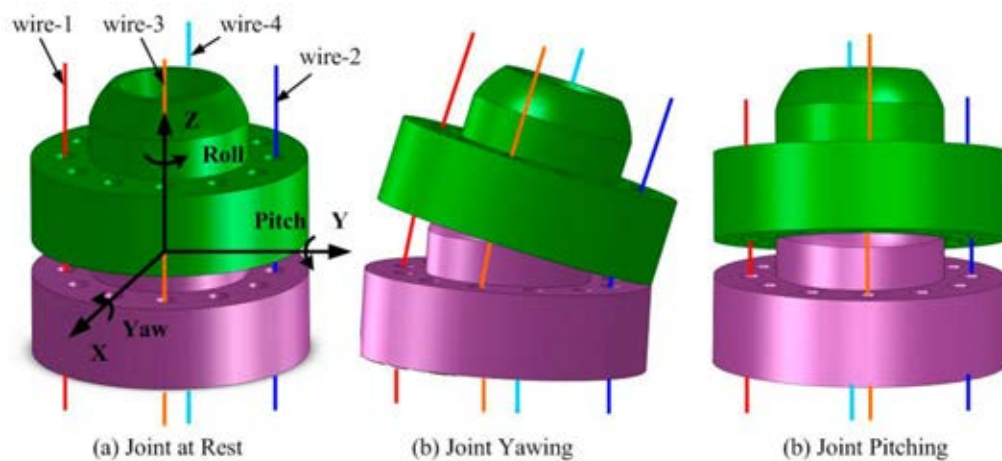


Figure 2-8 Joint Motion - Spatial Serpentine Backbone

Two successive vertebrae form a spherical joint as shown in Figure 2-8. Figure 2-8 (a) shows the joint in the rest position. Figure 2-8 (b) shows the joint yawing and Figure 2-8 (c) shows the joint pitching. The motions are illustrated in Figure 2-8 (a). As the vertebra is circularly symmetric, the yawing motion and pitching motion is basically the same. Similarly, by connecting more vertebrae together, the spatial serpentine backbone is formed. A compliant rubber tube is placed in the central cavity to constrain the joints' rotations.

3) Planar Continuum Backbone Design

Besides joint connection, another vertebra connection method is the flexible beam. For planar motion, the beam deflection is constrained in a plane. To eliminate the deflection in other directions, the second axial moment of area of the beam in the bending direction should be much smaller than the other two directions. Figure 2-9 shows an example of planar continuum backbone design using a flexible beam with rectangular cross-section. Figure 2-9 (a) shows the backbone in the rest position. Figure 2-9 (b) shows the backbone being deflected. The cross-section view of the beam is shown in Figure 2-9 (c). A number of cylindrical vertebrae are evenly distributed along the flexible beam. The same as that in the planar serpentine backbone, two pilot holes penetrate the vertebrae to guide the wires.

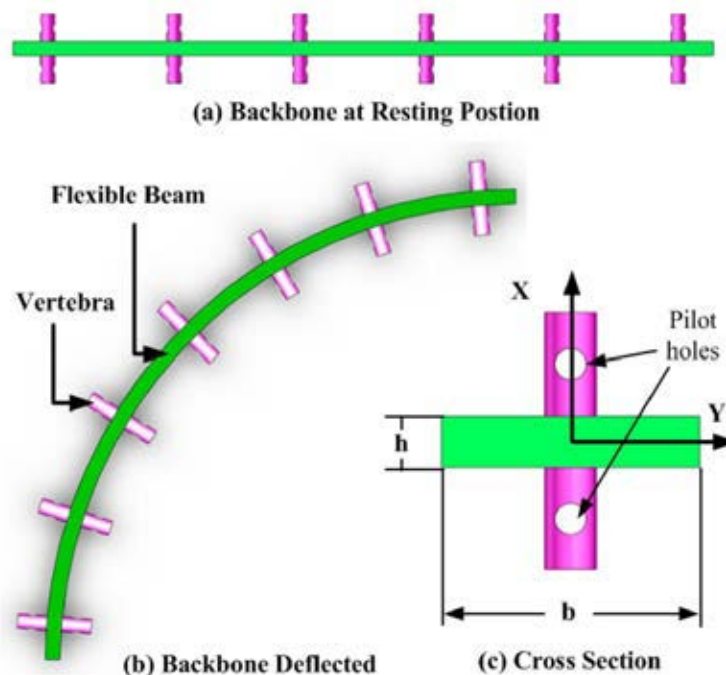


Figure 2-9 Planar Continuum Backbone

For the beam, the ratio of second axial moment of area in the X and Y directions is:

$$Ratio = \frac{I_y}{I_x} = \frac{bh^3}{12} \cdot \frac{12}{hb^3} = \left(\frac{h}{b}\right)^2 \quad (2-1)$$

The larger the ratio, the better the selection of deflection of the beam. In the given example, the width is 10 mm, and the height is 2 mm. The ratio is 25. This means the beam is 25 times easier to bend about the Y axis.

4) Spatial Continuum Backbone Design

For spatial motion, deflections in both directions are necessary. To ensure easy control of the beam deflection in arbitrary directions, the ideal cross-section of the beam is circularly symmetric, e.g. circle and ring. In this case, the second axial moment of area in all directions is the same.

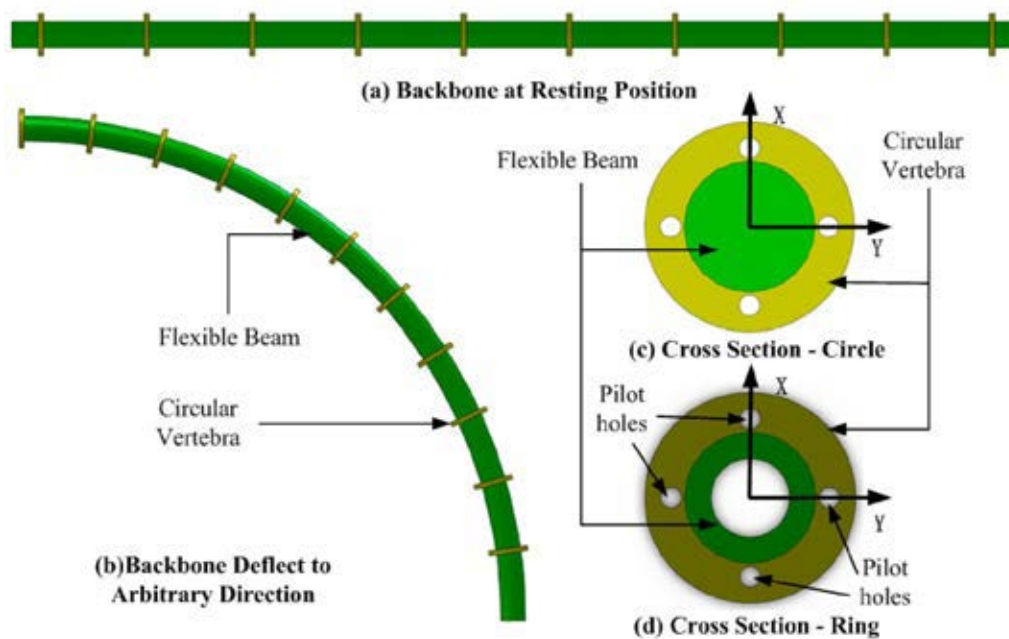


Figure 2-10 Spatial Continuum Backbone

The second axial moment of area of the beam with a circular cross-section is:

$$I_x = I_y = \frac{\pi D^4}{64} \quad (2-2)$$

The second axial moment of area of the beam with a ring cross-section is:

$$I_x = I_y = \frac{\pi(D^4 - d^4)}{64} \quad (2-3)$$

where, D is the outer diameter and d is the inner diameter of the beam.

Figure 2-10 (a) shows a spatial continuum backbone in the rest position, while Figure 2-10 (b) shows the backbone deflecting in an arbitrary direction. The cross-section of the flexible beam can be circular as shown in Figure 2-10 (c), or ring shaped as shown in Figure 2-10 (d). The circular vertebrae are evenly distributed along the beam. On each vertebra, there are four pilot holes to guide the wires. The angular spacing between each hole is 90 degrees. One can also change the number of pilot holes. However, to control the beam deflection in arbitrary directions, at least three wires are needed. It is the same as spatial serpentine backbone design.

2.2.2 Backbone Segmentation

The deflection of the backbone is controlled by the controlling wires. For the serpentine backbone, the number of DOFs is proportional to the joint number. For the continuum backbone, theoretically, there are infinite DOFs. Each pair of wires controls one DOF. To fully control all the DOFs, the number of wires needed is enormous. As discussed in the previous section, the snake skeleton is segmented. Hence, we could divide the backbone into several segments. The motion of each segment is controlled by one pair of wires. From the segmentation point of view, the WDM is categorized as single segment WDM and multi-segment WDM.

1) Single Segment WDM

In the single segment WDM, only one set of wires is used. For the planar WDM, each set contains one pair of wires; for the spatial WDM, each set contains two pairs of wires. For each wire, one end is connected to the actuator, and the other end is fastened to the end of the backbone, i.e. to the last vertebra. Each pair of wires control only one bending DOF of the backbone.

Figure 2-11 shows a single segment WDM example. In this example, the WDM has a continuum elastic beam as the backbone and one pair of wires. Along the beam are a number of evenly distributed spacing discs, which are the degenerated vertebrae. The wires are guided by the pilot holes in the spacing discs, from the backbone base to the distal end. The end of the wires is fastened to the last spacing disc. Figure 2-11 (a) shows a single segment planar WDM in the rest position. As shown in Figure 2-11 (b), the backbone is deformed into a C shape

by the wires. The distal end position and orientation is determined by the deformed backbone curve. This bending motion is fundamental in the WDM. When there are two pairs of wires orthogonally arranged, we get the single segment spatial WDM.

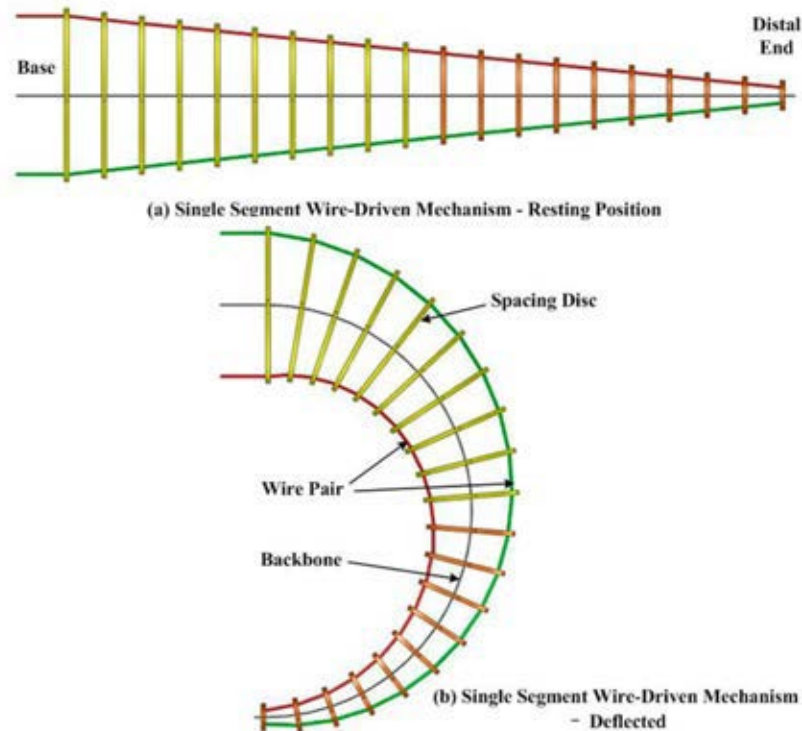


Figure 2-11 Single Segment WDM with Continuum Backbone

2) Multi-Segment WDM

In the multi-segment WDM, the backbone is segmented into several sections. Each section is a single segment WDM. The number of segments is determined by the number of wire sets. Figure 2-12 shows a two-segment planar WDM. As shown in the figure, the backbone structure is similar to that in the previous example and the number of wires is increased. There are two sets of wires. Each set contains one pair of wires. Wire pair 1's ends are connected to the last spacing disc of the first segment. Wire pair 2 passes by the first segment and their ends are fastened to the last spacing disc of the second segment. Figure 2-12 (a) shows the WDM in the rest position, which is the same as that of the single segment WDM. In Figure 2-12 (b) the backbone is deformed by the wire pairs. The bending directions of the two segments are the same. For both segments the wire at the bottom contracts. The deflected backbone curve is the same as that of the

single segment WDM. When the contracted wire in each pair is opposing, as shown in Figure 2-12 (c), the bending directions of the two segments are opposite. The deformed backbone curve is S shaped.

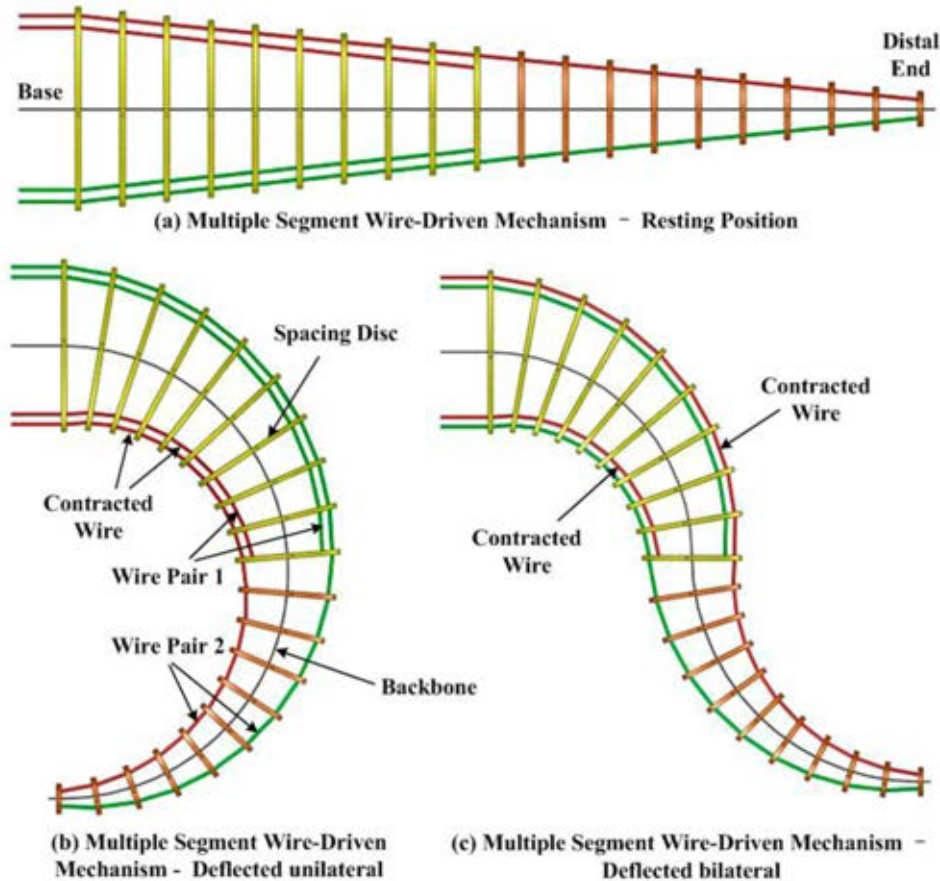


Figure 2-12 Multi-Segment WDM with Continuum Backbone

2.2.3 Wire Configuration

The backbone design is important. It determines the deflection type of the WDM. On the other hand, the wire configuration is of equal importance. The wire pairs control the deflection of the backbone. Also, the wire configuration influences the kinematics as well as the statics properties of the WDM. Details will be discussed in subsequent chapters. In the WDM, the wires are guided by the pilot holes in the vertebrae. In the previous section, one wire configuration was shown. In this section, the wire configuration is presented systematically. There are three types of wire configurations: parallel, tapered and spiral.

1) Parallel Wire Configuration

The parallel wire configuration is the most commonly used wire configuration. In this configuration, the pilot holes are of equal distance to the backbone. As a result, the wires are parallel to the backbone all the time. Also, the two wires and the backbone are coplanar, with the backbone between the two wires. In this configuration, when the backbone deforms, the extension of one wire is the same as the contraction of the other wire. The parallel wire configuration is shown in Figure 2-13. In the figure, the red curve is the contracted wire, the blue curve is the extended wire and the dark green curve is the backbone. As shown in the figure, during the deflection, the wires are always parallel to the backbone. The length changes of the two wires could be viewed as the same.

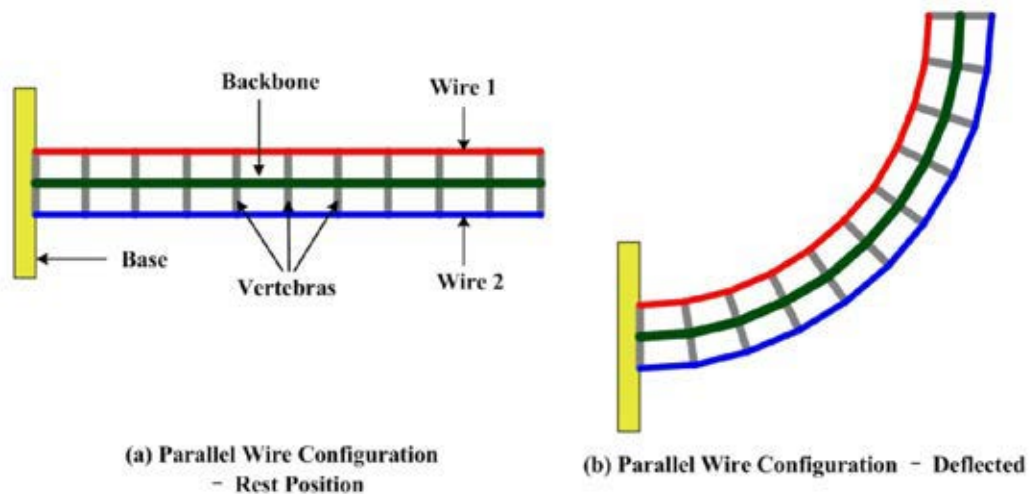


Figure 2-13 Parallel Wire Configuration

2) Tapered Wire Configuration

The tapered wire configuration is similar to the parallel wire configuration. The difference is that there is an inclined angle γ between the wires and the backbone, as shown in Figure 2-14. In the rest position, the two wires are symmetrically located at the two sides of the backbone. When the backbone is deflected, the length changes in the two wires are not the same as previously. Figure 2-14 (b) shows the deflected backbone as well as the wires. In this configuration, the wire length changes are different. The discrepancy is determined by the incline angle γ , and joint rotation angle θ . When γ and θ are small, e.g. less than 10° , the approximation error is small and length changes can be treated as the same. This is shown in Chapter 5.

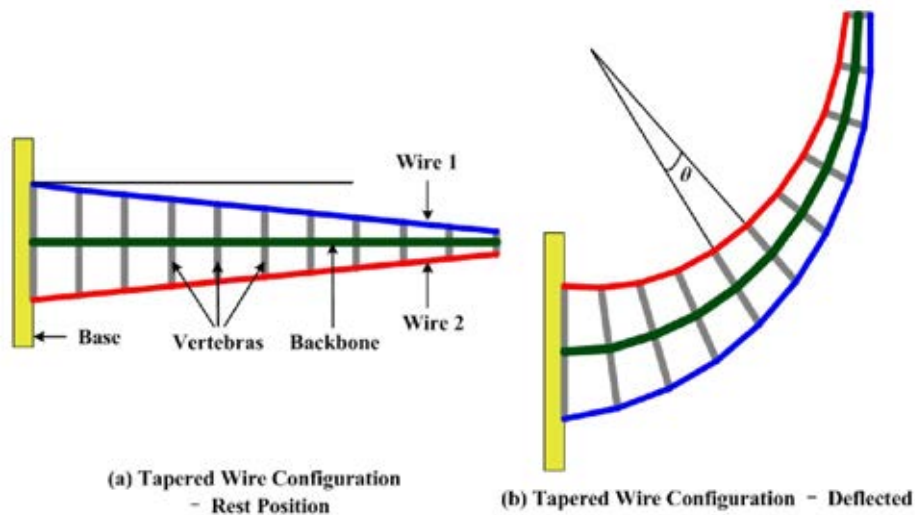


Figure 2-14 Tapered Wire Configuration

3) Spiral Wire Configuration

In the previous two configurations, the wires and the backbone were coplanar. The wires control the backbone bending in the plane without twisting. In the spiral wire configuration the wires spiral around the backbone, similar to the octopus arm oblique muscles or the double helix structure in DNA.

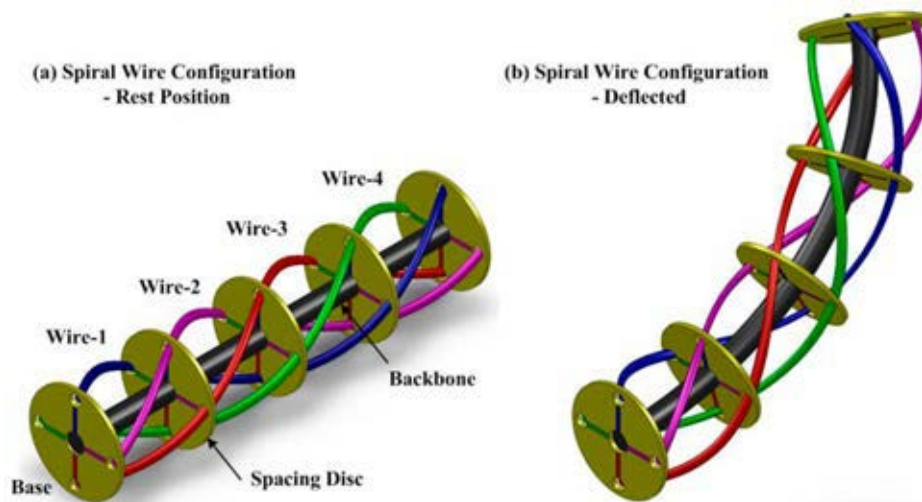


Figure 2-15 Spiral Wire Configuration

Figure 2-15 shows the spiral wire configuration. The two wires in a pair are opposed to each other. When one of the wires contracts, the force exerted on the backbone has an axial component and a tangent component. The axial force bends the backbone, while the tangent component twists the backbone. The deformed

backbone curve is complicated. Also, the length changes in the two wires are different. There is no simple analytical representation available.

2.3 Wire-Driven Mechanism Categorization

From the previous description, the WDM is composed of two parts: the flexible backbone and the wire pairs. However, there are various designs of the WDM according to the deflection type, backbone structure, backbone segmentation and wire configuration. It is beneficial if we can know the design information from the name. In this section, the WDM categorization and coding are introduced.

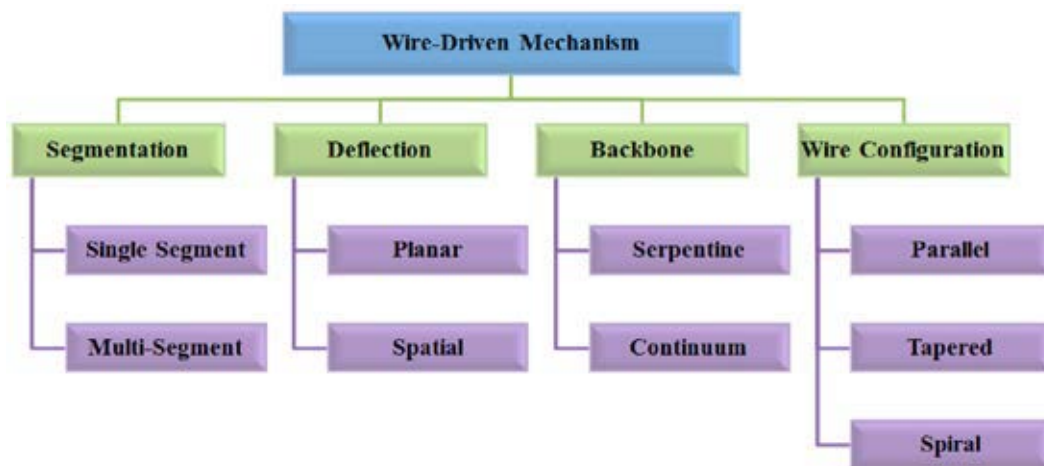


Figure 2-16 Wire-Driven Mechanism Categories

Figure 2-16 lists the categories of the WDM according to the backbone segmentation, deflection type, backbone structure and wire configuration. Based on the backbone segmentation, the WDM is categorized as single segment WDM (S) and multi-segment WDM (M). From the backbone deflection type, the WDM is categorized as planar WDM (P) and spatial WDM (S). From the backbone structure, it can be categorized as serpentine WDM (S) and continuum WDM (C). From the wire configuration with respect to the backbone, the WDM is categorized as parallel WDM (P), tapered WDM (T) and spiral WDM (S).

To simplify the naming of different WDMs, the nomenclature is introduced. Four characters are used to code the WDM as follows: “XXXX WDM”. The first character represents the backbone segmentation: there are two selections, i.e. S and M, as shown in Figure 2-16. The second character means the deflection type:

there are two selections, i.e. P and S. The third character denotes the backbone structure: there are two selections, i.e. S and C. The last character shows the wire configuration: there are three selections, i.e. P, T and S. For example: “SPSP WDM” represents “Single segment Planar Serpentine Wire-Driven Mechanism with Parallel wire configuration”; “MSCT WDM” denotes “Multi-segment Spatial Continuum Wire-Driven Mechanism with Tapered wire configuration”. In case there is no information of one category, the “X” character is used. Such as, “SPSX” represents “Single Segment Planar Serpentine Wire-Driven Mechanism”; “SXXS WDM” represents “Single Segment Serpentine Wire-Driven Mechanism”. When there are three categories’ information unknown, the WDM is named by the full information of the remaining category. Such as “SXXX WDM” is named “Single Segment Wire-Driven Mechanism”; and “XXSX WDM” is named “Serpentine Wire-Driven Mechanism” directly.

2.4 Summary

In this chapter, the biomimetic WDM is introduced. The design follows nature. It is composed of a flexible backbone and a number of wire pairs. For the serpentine WDM, the backbone is composed of several serially linked vertebrae, with adjacent vertebrae forming a joint. For the continuum WDM, the backbone is a flexible beam. The vertebrae degenerated to a spacing disc. The wires control the backbone deflection. Following the octopus arm muscle arrangement, each bending DOF is controlled by a pair of wires. WDM coding is introduced, i.e. “XXXX WDM”. The four characters represent the backbone segmentation, backbone deflection type, backbone structure and wire configuration respectively. WDM information is straightforwardly understood from the name.

Chapter 3 Kinematics and Workspace of the Wire-Driven Mechanism

In this chapter, the kinematic models of the XXSP WDM and XXCP WDM are developed. Geometry analysis is used for single segment WDM, while the extended D-H method is used for building the multi-segment WDM kinematic model. Workspace is obtained from the forward kinematic model. A novel idea of expanding the WDM workspace via obstacles or actively deploying constraints is proposed. Finally, the models are validated using a SPSP WDM manipulator and a MSSP WDM manipulator.

3.1 Kinematic Model of Single Segment WDM

In the WDM, the flexible backbone deformation is controlled by the wire pairs, and the wire lengths are controlled by the actuators, i.e., motors. Motion is transmitted from the actuators to the backbone via the wires. In general, position, orientation and velocity of the backbone distal end (or end effector) are of interest. When the backbone configuration or deformed shape is given, the position and orientation can be obtained. Velocity can also be determined from the time derivative of the distal end displacement. As a result, the kinematics can be divided into two parts [5, 125], as shown in Figure 3-1.

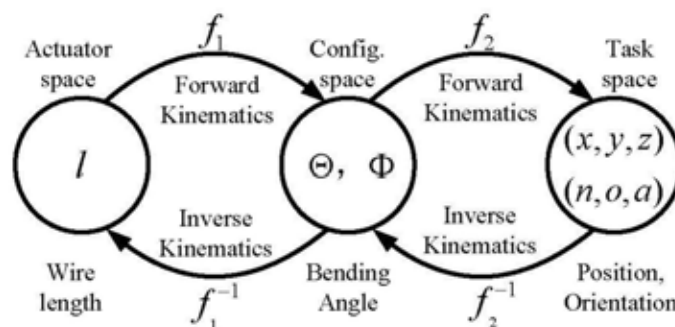


Figure 3-1 Kinematics Defined by Mappings between the Spaces

The first part is the mapping between the actuator space (i.e. wire length l) and the configuration space (i.e. WDM bending angle Θ and bending direction angle Φ). In this part the forward kinematics and inverse kinematics are defined as f_1

and f_1^{-1} respectively. Wire configuration is important in this part as it determines the joint kinematics. The second part is the mapping between the configuration space and the task space (i.e. distal end position (x, y, z) and orientation (n, o, a)). In this part, the forward kinematics and the inverse kinematics are defined as f_2 and f_2^{-1} respectively. The mapping is related to the backbone structure. In the following subsections, the kinematic model of the SXSP WDM and SXCP WDM are developed.

3.1.1 Kinematic Model of the Serpentine WDM

In this section, forward kinematics and inverse kinematics models of the SXSP WDM are presented. They are divided into two parts as shown above.

1) Mapping between the Actuator Space and the Configuration Space

In the SXSP WDM, the backbone is made up of multiple vertebrae, with two successive vertebrae forming a revolute joint. Figure 3-2 shows the joint rotation. The blocks represent the vertebrae, while the red line and blue line denote the wire pair. In the figure, D is the outer diameter of the vertebra; H is the vertebra height; d is the wire spacing distance; h_0 is the initial joint gap distance.

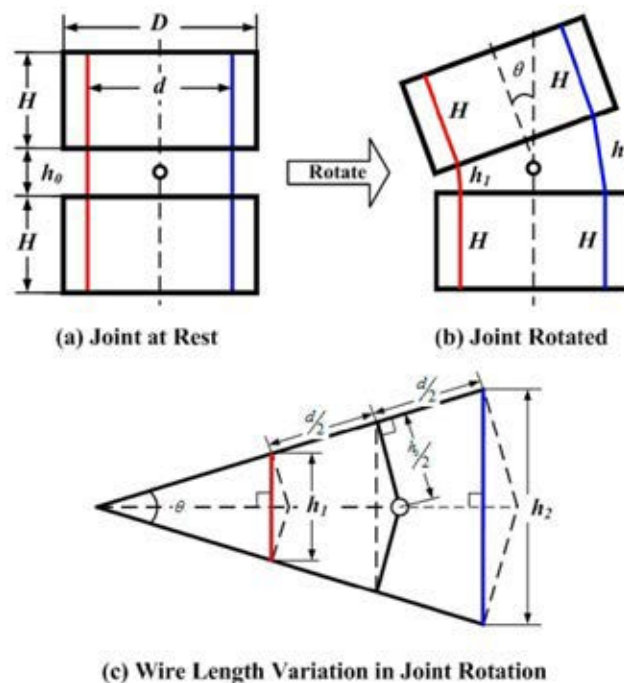


Figure 3-2 Joint of the SXSP WDM

The wire is divided into two parts: the first part is inside the vertebra. Its length is equal to the vertebra height H . It does not change with joint rotation. The second part is between the two vertebrae. Its length is related to the joint rotation angle θ . In the rest position, the length is equal to the initial joint gap distance h_0 , as shown in Figure 3-2 (a). When the joint rotates, for the two wires, their lengths are h_1 and h_2 respectively, as shown in Figure 3-2 (b).

For each vertebra, the corresponding wire length is $l_0 = H + h_0$. Assuming that there are N vertebrae, the initial wire length inside the serpentine WDM is:

$$L_{10} = L_{20} = L_0 = N \cdot (H + h_0) \quad (3-1)$$

It should be noted that, in the WDM, the first joint is formed by the base and the first vertebra. When the joint rotates θ to the left as in Figure 3-2 (b), the length of the left wire is shortened to $l_1 = H + h_1$, and the length of the right wire is increased to $l_2 = H + h_2$. The geometry relationship between the wire length and joint rotation is as shown in Figure 3-2 (c). From the figure, it is shown that after rotation, the length variations are per Equation (3-2).

$$\begin{cases} h_1 = h_0 - \Delta h_1 = h_0 - \left[d \cdot \sin\left(\frac{\theta}{2}\right) + 2h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \\ h_2 = h_0 + \Delta h_2 = h_0 + \left[d \cdot \sin\left(\frac{\theta}{2}\right) - 2h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \end{cases} \quad (3-2)$$

By summation, the total wire length after bending is shown by Equation (3-3).

$$\begin{cases} L_1 = L_0 - N \left[d \cdot \sin\left(\frac{\theta}{2}\right) + 2h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \\ L_2 = L_0 + N \left[d \cdot \sin\left(\frac{\theta}{2}\right) - 2h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \end{cases} \quad (3-3)$$

From Equation (3-3), the overall bending angle of the backbone in terms of wire length is:

$$\Theta = N \cdot \theta = 2N \cdot \arcsin\left[\frac{L_2 - L_1}{2N \cdot d}\right] \quad (3-4)$$

For computation, one can use arctan instead of arcsin. Detailed derivation of these equations is shown in appendix B.

Particularly, the maximum joint rotation angle, i.e. θ_{\max} , is constrained by the joint parameters, D and h_0 . The relationship between θ_{\max} and the vertebra parameters could be found from Figure 3-2 (c) as:

$$\theta_{\max} = 2 \arctan\left(\frac{h_0}{D}\right) \quad (3-5)$$

It is worth mentioning that, in Equation (3-2), Δh_1 and Δh_2 are the joint gap distance variations. For the serpentine WDM, the joint rotation is typically small. Hence, it is reasonable to make the following approximation: $\cos(\theta) \approx 1 - 0.5 \cdot \theta^2$ and $\sin(\theta) \approx \theta$. By approximation, the variations of joint gap distance are $\Delta h_1 \approx \Delta h_2 \approx 0.5 \cdot d \cdot \theta = \Delta h$. For example, for joint with $d=15$ mm and $h_0=2.5$ mm, when the joint rotation angle $\theta=10^\circ$, $\Delta h_1=1.317$ mm, $\Delta h_2=1.298$ mm, and the approximation is $\Delta h=1.309$ mm. The relative errors of the length approximation are 0.596% and 0.861% respectively.

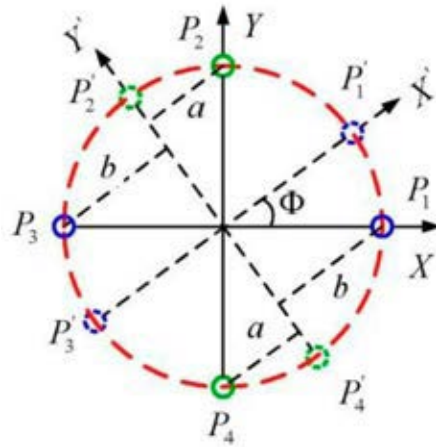


Figure 3-3 WDM Bending in Arbitrary Direction

Figure 3-3 shows the wire configuration of a serpentine WDM with arbitrary deflection direction. As shown in the figure, there are two pairs of wires as represented by P_1, P_2, P_3 and P_4 . They are evenly distributed on the vertebra. P_1 and P_3 form the horizontal pair; they control the backbone deflection about the Y axis. P_2 and P_4 make up the vertical pair; they control the backbone deflection about the X axis. As the two wire pairs are orthogonal, the two deflections are independent. Their combination allows the backbone to bend in any arbitrary

directions. For example, when the backbone bends about the Y' axis, the bending direction angle is Φ . Although, the backbone is flexed by the four wires P_1, P_2, P_3, P_4 , it is equivalent to P_1' and P_3' controlling the backbone bending while P_2' and P_4' lie in the neutral plane. The wire length change is proportional to the distance between the wire and the neutral plane. From the figure, after bending, the length change magnitude of P_1 and P_3 is the same; the length change magnitude of P_2 and P_4 is identical. The overall lengths of the four wires after bending are:

$$P_1: \quad L_1 = L_0 + 2N \left[b \cdot \sin\left(\frac{\theta}{2}\right) - h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \quad (3-6)$$

$$P_2: \quad L_2 = L_0 + 2N \left[a \cdot \sin\left(\frac{\theta}{2}\right) - h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \quad (3-7)$$

$$P_3: \quad L_3 = L_0 - 2N \left[b \cdot \sin\left(\frac{\theta}{2}\right) + h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \quad (3-8)$$

$$P_4: \quad L_4 = L_0 - 2N \left[a \cdot \sin\left(\frac{\theta}{2}\right) + h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \quad (3-9)$$

where, $a = 0.5d \cdot \sin(\Phi)$ is the distance from P_2 and P_4 to the neutral axis Y' and $b = 0.5d \cdot \cos(\Phi)$ is the distance from P_1 and P_3 to Y' .

From the wire length, the backbone bending direction angle (Φ) and the backbone bending angle (θ) can also be determined:

$$\Phi = \arctan\left(\frac{L_2 - L_4}{L_1 - L_3}\right) \quad (3-10)$$

$$\Theta = N \cdot \theta = 2N \cdot \arcsin\left[\frac{\sqrt{(L_1 - L_3)^2 + (L_2 - L_4)^2}}{2N \cdot d}\right] \quad (3-11)$$

For serpentine WDM with two pairs of wires, Equations (3-6)~ (3-9) define the inverse mapping between the actuator space and the configuration space, i.e. f_1^{-1} ; the forward mapping, i.e. f_1 , is given by Equation (3-10) and Equation (3-11). For serpentine WDM with one pair of wires, the mapping is similar. Such as keeping the horizontal wire pair, we have $L_2=L_4$, and $\Phi=0$ in the above equations.

$$\begin{cases} x = (H+h_0) \sum_{i=1}^N \sin(i \cdot \theta) \cdot \cos(\Phi) \\ y = (H+h_0) \sum_{i=1}^N \sin(i \cdot \theta) \cdot \sin(\Phi) \\ z = (H+h_0) \sum_{i=1}^N \cos(i \cdot \theta) \end{cases} \quad (3-12)$$

Note that,

$$\sum_{i=1}^N \sin(i \cdot \theta) = \frac{\sin(N\theta/2) \cdot \sin[(N+1)\theta/2]}{\sin(\theta/2)} \quad (3-13)$$

$$\sum_{i=1}^N \cos(i \cdot \theta) = \frac{\sin(N\theta/2) \cdot \cos[(N+1)\theta/2]}{\sin(\theta/2)} \quad (3-14)$$

The orientation of the distal end can also be determined from Θ and Φ . The original direction of the end effector in the inertial frame is $\hat{k} = [0 \ 0 \ 1]^T$. After the deflection, the end effector orientation is:

$$Rot_z(\Phi) \cdot Rot_y(-\Theta) \cdot \hat{k} = [\sin(\Theta)\sin(\Phi) \ \sin(\Theta)\cos(\Phi) \ \cos(\Theta)]^T \quad (3-15)$$

From the distal end position, it is not difficult to find the backbone bending angle and bending direction:

$$\Theta = N \cdot \theta = \frac{2N}{N+1} \cdot \arctan\left(\frac{\sqrt{x^2 + y^2}}{z}\right) \quad (3-16)$$

$$\Phi = \arctan\left(\frac{y}{x}\right) \quad (3-17)$$

In short, Equations (3-12) ~ (3-15) give the forward mapping between the configuration space and the task space, i.e. f_2 . Equation (3-16) and Equation (3-17) give the inverse mapping, i.e. f_2^{-1} .

3.1.2 Kinematic Model of the Continuum WDM

In the continuum WDM, the kinematics model can also be established by the mappings among the actuator space, configuration space and task space.

1) Mapping between the Actuator Space and the Configuration Space

In the continuum WDM, the wires are guided by the spacing discs. There is no apparent joint as in the serpentine case. However, in the analysis, we can treat the middle range of two adjacent discs as the joint, and analyze it similarly.

Figure 3-5 shows the joint of XPCP WDM. In the figure, D is the outer dimension of the disc, while d is the spacing distance between the two wires. In the rest position, as shown in Figure 3-5 (a), the backbone is straight and the distance between two adjacent discs is h . The thickness of the disc is small compared with h . Therefore, it is neglected in the following analysis. For the two wires, the initial lengths within the joint are the same. When one of the wires contracts (as shown by the red line) and the other extends (as shown by the blue line), the backbone deflects. The relationship between joint rotation (θ) and wire length variation (Δl_{i1} and Δl_{i2}) is shown in Figure 3-5 (b). In the analysis, the constant curvature assumption is adopted as well.

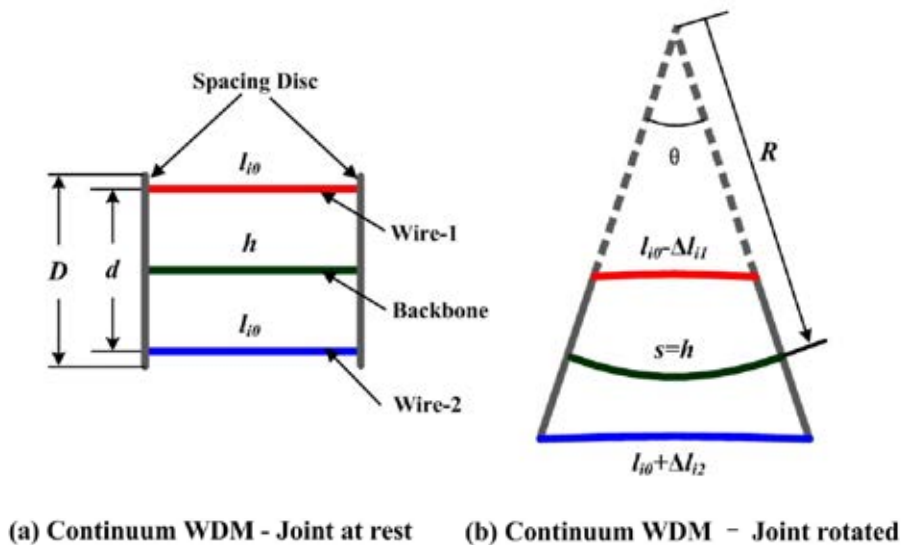


Figure 3-5 Joint of the SXCP WDM

From the figure, the wire length after backbone deflection is given by Equation (3-18). Similar to the serpentine WDM, the rotation angle of each joint in the continuum WDM is typically small. As a result, the length variations in the two wires can be treated as the same. For example, when the joint rotation $\theta=10^\circ$, $d=15$ mm and $h=5$ mm, the wire length changes are $\Delta l_{i1}=1.314$ mm and $\Delta l_{i2}=1.301$ mm. The approximation is $\Delta l_i=1.309$ mm. The relative errors of the approximation are 0.357% and 0.615% respectively.

$$\begin{cases} l_{i1} = l_{i0} - \Delta l_{i1} = l_{i0} - \left[h - 2 \left(\frac{h-d}{\theta} \right) \cdot \sin \left(\frac{\theta}{2} \right) \right] \approx l_{i0} - \frac{\theta d}{2} \\ l_{i2} = l_{i0} + \Delta l_{i2} = l_{i0} - \left[h - 2 \left(\frac{h+d}{\theta} \right) \cdot \sin \left(\frac{\theta}{2} \right) \right] \approx l_{i0} + \frac{\theta d}{2} \end{cases} \quad (3-18)$$

Theoretically, for each joint the rotation is limited by the disc dimension and spacing distance. Until the two discs collide, the joint can keep on rotating. Hence, for the joint in the continuum WDM, the maximum joint rotation is:

$$\theta_{\max} = \frac{2h}{D} \quad (3-19)$$

Assume that there are N joints in the backbone. Initially, the wire length within the WDM is:

$$L_{i0} = L_{20} = L_0 = N \cdot h \quad (3-20)$$

After deflection, the lengths of the two wires are:

$$\begin{cases} L_1 = L_0 - N \cdot \frac{\theta \cdot d}{2} \\ L_2 = L_0 + N \cdot \frac{\theta \cdot d}{2} \end{cases} \quad (3-21)$$

From Equation (3-21), the overall bending angle of the backbone in terms of wire length is:

$$\Theta = N \cdot \theta = \frac{L_2 - L_1}{d} \quad (3-22)$$

Similarly, when there are two pairs of wires orthogonally arranged as that in the serpentine WDM, the lengths of the four wires are:

$$P_1: \quad L_1 \approx L_0 + N \cdot \theta \cdot \frac{d}{2} \cdot \cos(\Phi) \quad (3-23)$$

$$P_2: \quad L_2 \approx L_0 + N \cdot \theta \cdot \frac{d}{2} \cdot \sin(\Phi) \quad (3-24)$$

$$P_3: \quad L_3 \approx L_0 - N \cdot \theta \cdot \frac{d}{2} \cdot \cos(\Phi) \quad (3-25)$$

$$P_4: \quad L_4 \approx L_0 - N \cdot \theta \cdot \frac{d}{2} \cdot \sin(\Phi) \quad (3-26)$$

From the wire length, the bending direction angle (Φ) and the bending angle (Θ) can be determined as:

$$\Phi = \arctan\left(\frac{L_2 - L_4}{L_1 - L_3}\right) \quad (3-27)$$

$$\Theta = N \cdot \theta = \frac{\sqrt{(L_1 - L_3)^2 + (L_2 - L_4)^2}}{d} \quad (3-28)$$

Equations (3-23) ~ (3-26) define the inverse mapping between the actuator space and the configuration space, i.e. f_1^{-1} , for the continuum WDM with two pairs of wires. The forward mapping, i.e. f_1 , is given by equations (3-27) and (3-28). For continuum WDM with one pair of wires, the mapping is similar. As an example, keeping the horizontal wire pair, we have $L_2=L_4$, and $\Phi=0$ in the above model.

2) Mapping between the Configuration Space and the Task Space

For the continuum backbone, the mapping between the configuration space and the task space is shown in Figure 3-6. In the figure, the coordinate frame is set at the base of the flexible backbone. Initially, the backbone is coincident with the Z axis. The cyan curve shows the deflected backbone. It is a circular arc with radius R and the center angle is Θ . It is tangential to the Z axis. The angle between the bending plane $OX'Z$ and OXZ is Φ .

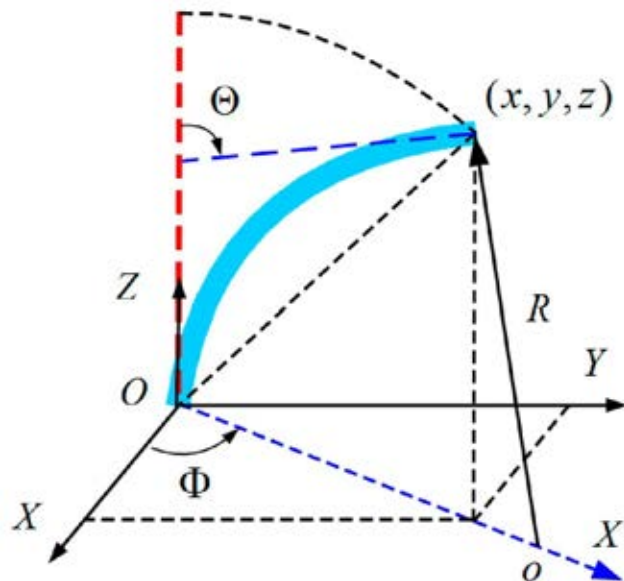


Figure 3-6 Mapping between Configuration Space and Task Space – Continuum

The distal end position can be found as:

$$\begin{cases} x = R \cdot [1 - \cos(\Theta)] \cdot \cos(\Phi) \\ y = R \cdot [1 - \cos(\Theta)] \cdot \sin(\Phi) \\ z = R \cdot \sin(\Theta) \end{cases} \quad (3-29)$$

In the equation $R=L/\Theta$ is the radius of the deflected backbone. The distal end orientation in the inertial frame is the same as that in Equation (3-15). From the distal end position, it is not difficult to find the backbone bending angle and bending direction:

$$\Theta = 2 \cdot \arccos\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \quad (3-30)$$

$$\Phi = \arctan\left(\frac{y}{x}\right) \quad (3-31)$$

In short, Equation (3-29) gives the forward mapping between the configuration space and the task space, i.e. f_2 . Equation (3-30) and Equation (3-31) give the inverse mapping, i.e. f_2^{-1} .

3.1.3 A Generalized Kinematic Model

From the previous two subsections, it is seen that the kinematic models for the serpentine WDM and continuum WDM are similar. Indeed, if the number of vertebrae is infinity, a serpentine backbone turns into a continuum backbone. In this section, a more generalized kinematic model suitable for the two types of WDM is presented.

From the previous analysis, the backbone deflection is similar for planar WDM (two wires) or spatial WDM (four wires). The only difference is the bending direction angle Φ , which can be determined from the wire lengths. It does not influence the in-plane bending. The spatial bending can be treated as: rotate the bending plane OXZ about the Z axis Φ counter clockwise; then, the backbone bends Θ in the new OXZ plane. In the following analysis, we consider the in-plane bending only.

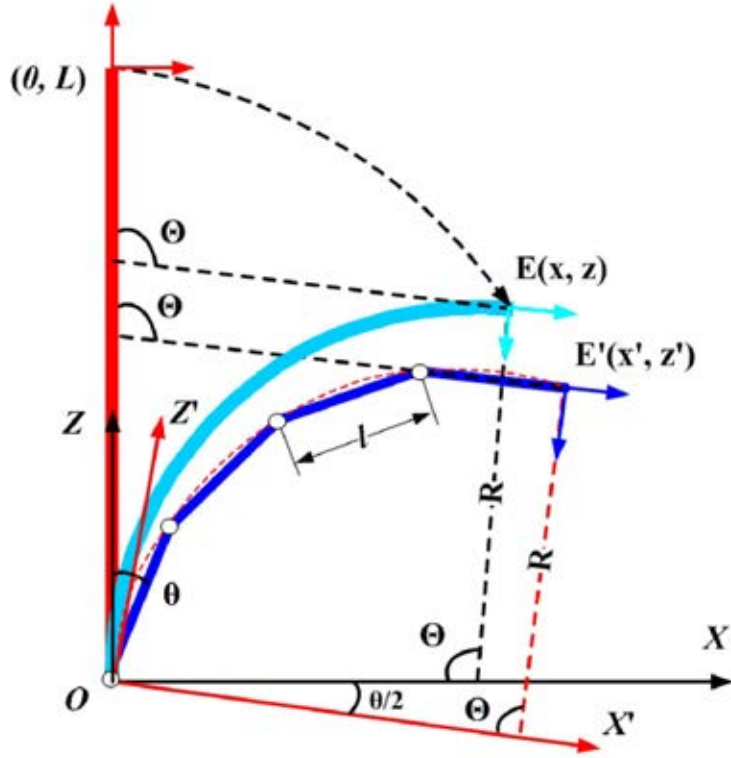


Figure 3-7 Comparison of Serpentine WDM and Continuum WDM

As shown in Figure 3-7, the bending motion is constrained in the OXZ plane. The cyan curve is the continuum backbone while the blue polyline represents the serpentine backbone. The red dashed arc is the circumcircle of the polygon. In the analysis, the constant curvature assumption is still adopted. Suppose for the two backbones, the joint number (N) and joint rotation (θ) are the same. This means that the total bending angle Θ of the two backbones are equal. The overall length of the backbone is L , and the length for each vertebra is l . From the previous section, the distal end displacement for the continuum backbone is:

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} R(1 - \cos(\Theta)) \\ R \sin(\Theta) \end{bmatrix} \quad (3-32)$$

The distal end displacement for the serpentine backbone is:

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} l \cdot \sum_{i=1}^N \sin(i \cdot \theta) \\ l \cdot \sum_{i=1}^N \cos(i \cdot \theta) \end{bmatrix} \quad (3-33)$$

From Figure 3-7 the distal end of the serpentine backbone is coincident with the endpoint of arc OE' . For the orientation, the direction of the serpentine backbone distal end is parallel to the tangent line of arc OE as shown in the figure. As a result, the position of the distal end can be represented by arc OE' , while the orientation can be represented by arc OE . In the figure arc OE is tangential with axis OZ , and arc OE' is tangential with axis OZ' . The radius of both arcs are R . The arc lengths are both L . The angle between the two axes is $\theta/2$. The distal end position and orientation of the serpentine backbone can be represented as:

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta/2) & \sin(\theta/2) \\ -\sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \begin{bmatrix} R[1 - \cos(\Theta)] \\ R \sin(\Theta) \end{bmatrix} \quad (3-34)$$

$$R = \frac{l}{2 \cdot \sin(\theta/2)} \quad (3-35)$$

$$\theta = \Theta/N \quad (3-36)$$

$$l = L/N \quad (3-37)$$

In the above equations, when $N \rightarrow \infty$ the joint angle $\theta \rightarrow 0$ and $\sin(\theta) \rightarrow 0$, $\cos(\theta) \rightarrow 1$. Hence, Equation (3-34) turns into Equation (3-32). Meanwhile, when the vertebra length $l \rightarrow 0$, by L'Hôpital's rule, the radius of the arc is $R = L/\Theta$. Since Equation (3-33) and Equation (3-34) are two representations of the same point, we can rewrite Equation (3-34) as:

$$\begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} \frac{L}{N} \cdot \frac{\sin(\Theta/2) \cdot \sin(\Theta(N+1)/2N)}{\sin(\Theta/2N)} \\ \frac{L}{N} \cdot \frac{\sin(\Theta/2) \cdot \cos(\Theta(N+1)/2N)}{\sin(\Theta/2N)} \end{bmatrix} \quad (3-38)$$

From Equation (3-38), the distal end orientation or the bending angle is solved as:

$$\Theta = \frac{2N}{N+1} \arctan\left(\frac{x}{z}\right) \quad (3-39)$$

For the single segment WDM, the end effector position and orientation are related to each other. The distal end position can be represented by the backbone parameters and orientation, i.e. the forward mapping between the configuration space and the task space f_2 . It is shown by Equation (3-34) or Equation (3-38).

When the distal end position is known, the orientation can also be determined as Equation (3-39). This is the inverse mapping between the configuration space and the task space, i.e. f_2^{-1} .

The first derivation of Equation (3-38) with respect to time gives the velocity of the WDM distal end:

$$\begin{cases} \dot{x} = \frac{L}{2N^2} \frac{N \sin(\Theta/2N) \sin[(2N+1) \cdot \Theta/2N] - \sin^2(\Theta/2)}{\sin^2(\Theta/2N)} \cdot \dot{\Theta} \\ \dot{z} = \frac{L}{2N^2} \frac{N \sin(\Theta/2N) \cos[(2N+1) \cdot \Theta/2N] - 0.5 \sin(\Theta)}{\sin^2(\Theta/2N)} \cdot \dot{\Theta} \end{cases} \quad (3-40)$$

The second derivation of Equation (3-38) with respect to time gives the acceleration of the WDM distal end. In Equation (3-41) and Equation (3-42) $\beta = \Theta / 2N$.

$$\begin{aligned} \ddot{x} = \frac{L \cdot \ddot{\Theta}}{2N^2 \sin^4(\beta)} & \left\{ 2N^2 \sin^3(\beta) \cos(\beta + \Theta) \right. \\ & + N \sin^2(\beta) (\sin(2\beta + \Theta) - \sin(\Theta)) \\ & \left. - N \sin(\beta) \sin(2\beta) \sin(\beta + \Theta) + \sin(2\beta) \sin^2(0.5\Theta) \right\} \end{aligned} \quad (3-41)$$

$$\begin{aligned} \ddot{z} = \frac{L \cdot \ddot{\Theta}}{2N^2 \sin^4(\beta)} & \left\{ -2N^2 \sin^3(\beta) \sin(\beta + \Theta) \right. \\ & + N \sin^2(\beta) (\cos(2\beta + \Theta) - \cos(\Theta)) \\ & \left. + N \sin(\beta) \sin(2\beta) \cos(\beta + \Theta) - 0.5 \sin(2\beta) \sin(\Theta) \right\} \end{aligned} \quad (3-42)$$

One important characteristic of the WDM is the leverage effect. Assume that the WDM bending velocity is $\dot{\Theta} = \pi$ rad/s, $L=150$ mm, $N=10$, and $d=10$ mm. From Equation (3-40), the distal end velocity is as shown in Figure 3-8 (a). The wire velocity can be found from Equation (3-3) or Equation (3-21). The velocity ratio with respect to the wire velocity is shown in Figure 3-8 (b). In this example, the wire velocity is constantly 0.0157 m/s. From the figure, the distal end velocity is over ten times the wire velocity. Also, the amplification ratio varies with bending angle. In the rest position, the amplification ratio is the largest. This shows the leverage effect of the WDM.

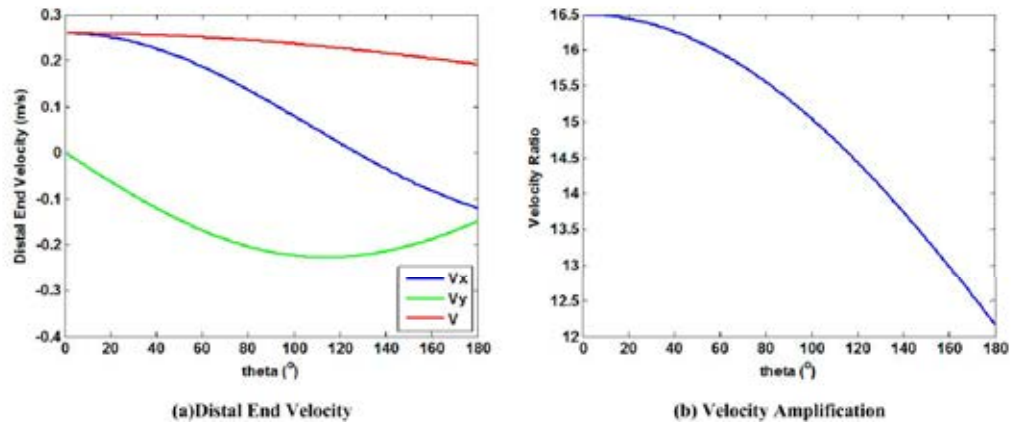


Figure 3-8 WDM Distal End Velocity and Leverage Effect

3.2 Kinematic Model of Multi-Segment WDM

Similar to traditional serial manipulators, the multi-segment WDM is composed of several serially linked single segment WDMs. The common approach used for manipulator kinematic modeling is the D-H method [126]. In this method, the end effector position and orientation is obtained by multiplying several homogeneous transformation matrixes T_i . Each transformation matrix is represented by the link length, link twist, link offset and joint angle. For the multi-segment WDM, each segment has three DOFs. Therefore, the traditional D-H method is not applicable. In this section, the kinematic model is developed using an extended D-H method.

3.2.1 Forward Kinematics

The forward kinematic model of the multi-segment WDM is developed based on the single segment WDM model. Each segment is treated as a link. However, the link here is different from the link in traditional manipulators. In the D-H method, each link has one DOF. They can either rotate about one axis or translate along the axes. For the WDM segments, they have three DOFs, i.e. two bending DOFs and a translation DOF. As a result, the traditional D-H method is not applicable for the multi-segment WDM. I.D. Walker developed a modified D-H method for multi-section continuum robots [40]. The robot has a similar structure to that of a continuum WDM. Hence, the modified D-H method can be used for continuum WDM. However, for serpentine WDMs, further modification is needed. In this section, the forward kinematics of multi-segment WDM is developed by integrating geometry analysis and the modified D-H method.

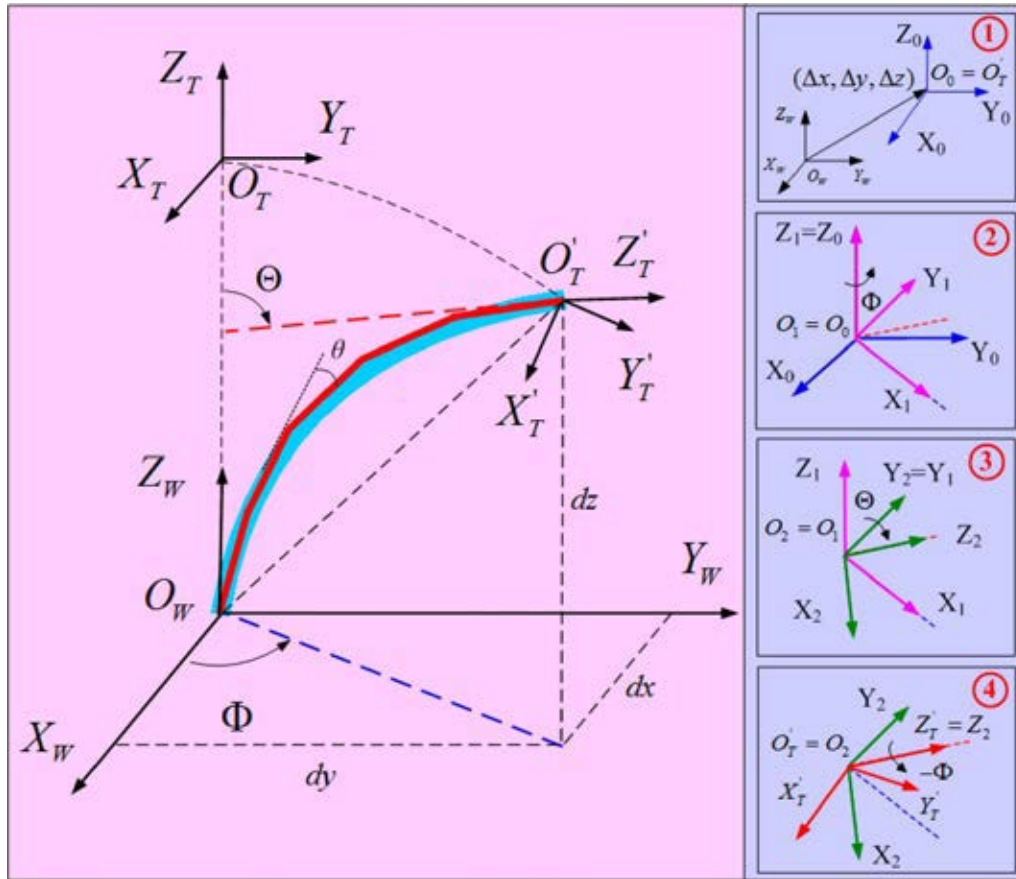


Figure 3-9 Single Segment WDM Coordinate Transform

As shown in Figure 3-9, the local coordinate frame (or inertial coordinate frame) $O_L = \{ X_L, Y_L, Z_L \}$ is set at the center of the first joint. The first joint is composed of the WDM base and the first vertebra. The WDM distal end coordinate frame $O_T = \{ X_T, Y_T, Z_T \}$, is located at the distal end of the backbone. When the WDM segment bends θ , and the bending direction angle is Φ , (the backbone bending is within the plane $O_L X'_L Z_L$), the WDM distal end moves to $O'_T = \{ X'_T, Y'_T, Z'_T \}$. As shown in Figure 3-9, the transformation between O'_T and O_L involves four steps: (1) translation of the coordinate origin from O_L to O'_T , (2) rotation of the coordinate frame about the new Z axis with angle Φ ; (3) rotation about the new Y axis with angle Θ ; and (4) rotation about the new Z axis with angle $-\Phi$. The overall transformation is described as Equation (3-43).

$${}^L T_{T'} = \text{Trans}(\Delta x, \Delta y, \Delta z) \cdot \text{Rot}(z, \Phi) \cdot \text{Rot}(y, \Theta) \cdot \text{Rot}(z, -\Phi) \quad (3-43)$$

In the equation, Δx , Δy and Δz are the displacements along X_L , Y_L , and Z_L directions. They are also the distal end coordinates in the local coordinate frame

$O_L-X_L Y_L Z_L$. As the vertebra axis is the side of a regular polygon, the distal end position can be represented by Equation (3-12) and Equation (3-29) for serpentine WDMs and continuum WDMs respectively. Or, by the generalized kinematic model for single segment WDMs, the three translations can be represented by Equation (3-44). In the equation, L is backbone length, Θ is the WDM segment bending angle, and Φ is the WDM segment bending direction as defined previously. The derivation is similar to that of appendix B.

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \frac{L}{N} \cdot \frac{\sin(\Theta/2)}{\sin(\Theta/2N)} \cdot \begin{bmatrix} \sin(\Theta(N+1)/2N)\cos(\Phi) \\ \sin(\Theta(N+1)/2N)\sin(\Phi) \\ \cos(\Theta(N+1)/2N) \end{bmatrix} \quad (3-44)$$

Therefore, for each segment, the distal end position and orientation in the local coordinate frame can be defined by the backbone structure parameters, i.e. vertebra number N and backbone length L ; and backbone deflection parameters, i.e. bending angle Θ and bending direction angle Φ .

As shown in Figure 3-10, once the transformation of each segment is determined, the forward kinematics of multi-segment WDMs can be solved using the chain rule, similar to that of the traditional D-H method. Assuming that the WDM has M segments, the distal end position and orientation in the inertial coordinate frame is:

$${}^0T_M = {}^0T_1 \cdot {}^1T_2 \cdots {}^{M-2}T_{M-1} \cdot {}^{M-1}T_M \quad (3-45)$$

where ${}^{i-1}T_i$ is the transformation between segment $i-1$ and i , as shown in Equation (3-46). In the equation, $C_\Theta = \cos(\Theta)$, $S_\Theta = \sin(\Theta)$, $C_\Phi = \cos(\Phi)$ and $S_\Phi = \sin(\Phi)$.

$${}^{i-1}T_i = \begin{bmatrix} C_\Theta C_\Phi^2 + S_\Phi^2 & (C_\Theta - 1)C_\Phi S_\Phi & S_\Theta C_\Phi & \Delta x \\ (C_\Theta - 1)C_\Phi S_\Phi & C_\Theta S_\Phi^2 + C_\Phi^2 & S_\Theta S_\Phi & \Delta y \\ -S_\Theta C_\Phi & -S_\Theta S_\Phi & C_\Theta & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-46)$$

The overall transformation matrix is a four by four matrix. The first three columns give the orientation of the distal end, while the fourth column gives the position in the inertial coordinate frame.

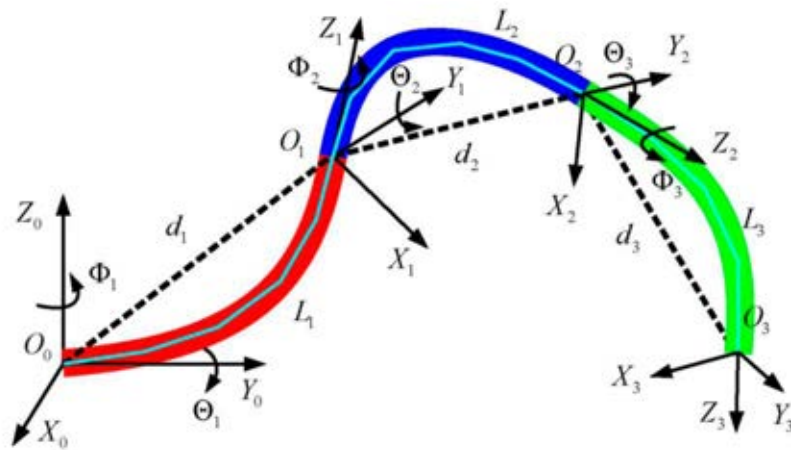
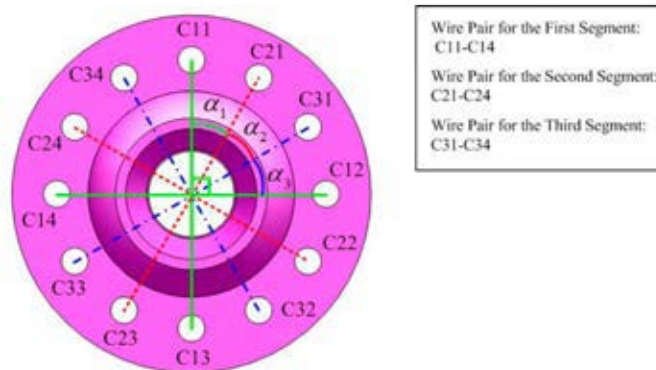


Figure 3-10 Multi-Segment WDM Coordinate Transform

Up to now, the forward mapping from the configuration space to the task space has been established. For multi-segment WDMs, the mapping between the actuation space and the configuration space is basically the same as for the single segment WDM.

For WDMs whose wire pairs of the posterior segment do not interact with the previous segments, the wire length variation for each segment is independent. They can be calculated by the method used in the single segment WDM. For WDMs whose wire pairs of the posterior segment go through the previous segments, the wire length variation has two parts. The first part is the variation within the segment. The second part is the length variation induced by the former segments. The overall length variation is the summation of the two parts. In determining the second part, the shift angle α shall be added to Φ for the latter segments. As shown in Figure 3-11, the shift angle is defined as the angular spacing among the wire pairs in different segments.



Wire Pair for the First Segment:
C11-C14
Wire Pair for the Second Segment:
C21-C24
Wire Pair for the Third Segment:
C31-C34

Figure 3-11 Wire Pair Shift Angle

3.2.2 Inverse Kinematics

The same as for a traditional robot with many DOFs, the inverse kinematics of a multi-segment WDM is complex. In most cases there is no unique solution as that can be solved for single section WDMs. I. D. Walker and his colleagues proposed a closed-form inverse solution for multi-section continuum robots [25]. In his method, the distal end position as well as the lengths of each section (i.e. d_1, d_2, d_3 in Figure 3-10) need to be known in advance. However, in general, these lengths cannot be predetermined. Since there are many ways to reach a desired position, we propose a method in which closed form inverse kinematics can be solved. It is called the uniform bending scheme.

In this scheme, the backbone radius for each segment is the same, i.e., $R_1 = \dots = R_M = R$. Also, all the bending is in the same plane, i.e., $\Phi_1 = \dots = \Phi_M = \Phi$. For the design in which each segment has the same number of vertebrae, we have $\Theta_1 = \dots = \Theta_M = \Theta$. From Equation (3-43) to Equation (3-47), the multi-segment WDM forward kinematics in this scheme becomes:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \sum_{j=1}^M \begin{bmatrix} R \cdot \sin[(j-1/2) \cdot \Theta] \cdot [1 - \cos(\Theta)] \cdot \cos(\Phi) \\ R \cdot \sin[(j-1/2) \cdot \Theta] \cdot [1 - \cos(\Theta)] \cdot \sin(\Phi) \\ R \cdot \cos[(j-1/2) \cdot \Theta] \cdot \sin(\Theta) \end{bmatrix} \quad (3-47)$$

By solving this equation, it follows that:

$$\Phi = \arctan\left(\frac{y}{x}\right) \quad (3-48)$$

$$R = \frac{x^2 + y^2 + z^2}{2\sqrt{x^2 + y^2}} \quad (3-49)$$

$$\Theta = \frac{1}{M} \arcsin\left(\frac{z}{R}\right) \quad (3-50)$$

After the configurations of each segment are determined, the wire lengths can be solved as previously. It is interesting to note that, since each segment has the same length and bending angle Θ , they can be treated as a vertebra as in the single segment WDM. Hence, the multi-segment WDM inverse kinematic problem becomes the inverse kinematic problem of single segment WDMs, which

has a unique closed-form solution. Although this simple case is one of many, it is useful in wire-driven robot manipulation.

3.3 Workspace

The WDM is a type of underactuated flexible mechanism. The workspace here denotes the positions that the distal end can reach. The workspace is derived from the forward kinematic model.

3.3.1 Workspace of Single Segment WDM

For serpentine WDMs, the joint rotation is limited by the vertebra structure. From the forward kinematic model of single segment serpentine WDMs the workspace is expressed as Equation (3-51).

$$\sqrt{x^2 + y^2 + z^2} = (H+h_0) \cdot \frac{\sin(N \cdot \theta/2)}{\sin(\theta/2)} \quad \theta \in [\theta_{\min}, \theta_{\max}] \quad (3-51)$$

For the single segment continuum WDM, the workspace is expressed as:

$$\sqrt{x^2 + y^2 + z^2} = \frac{2L}{\Theta} \cdot \sin\left(\frac{\Theta}{2N}\right) \quad \Theta \in [\Theta_{\min}, \Theta_{\max}] \quad (3-52)$$

The workspace can also be obtained from the generalized kinematic model.

$$\sqrt{x^2 + y^2 + z^2} = \frac{L}{N} \cdot \frac{\sin(\Theta/2)}{\sin(\Theta/2N)} \quad \Theta \in [\Theta_{\min}, \Theta_{\max}] \quad (3-53)$$

In the equations, L is the total length of the WDM, N is the number of joints, θ is the overall bending angle and θ is joint rotation angle. For continuum WDM, N is ∞ . The joint rotation limits are as described in section 3.1.

From the models, it is seen that the workspaces of the two types of WDM are both spheroidal surfaces. In planes parallel to the X - Y plane, the locus of the distal end is a circle, whose radius depends on the joint rotation angle and the vertebra parameters. As an example, Figure 3-12 shows the workspace of a SSSX WDM with parameters $L=150$ mm, $N=10$ and $\theta \in [-14.25^\circ, 14.25^\circ]$. The workspace of a SSCX WDM has a similar shape. Note that the workspace is circularly symmetric. It can be obtained by rotating the distal end trajectory in the X - Z plane about the Z axis 180° .

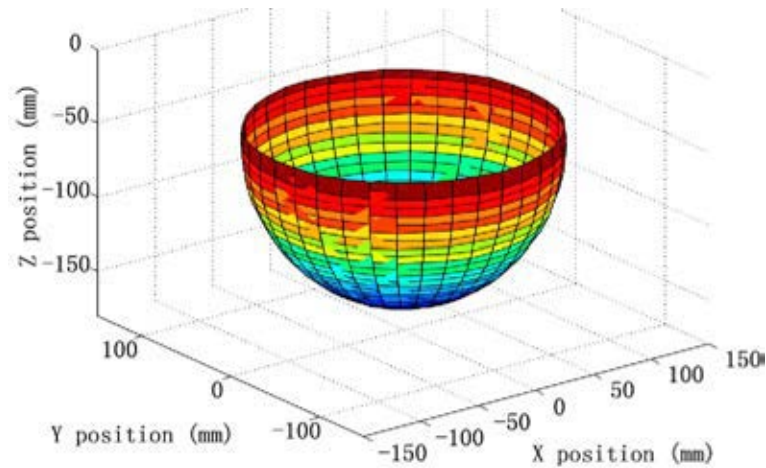


Figure 3-12 Workspace of a Single Segment Spatial WDM

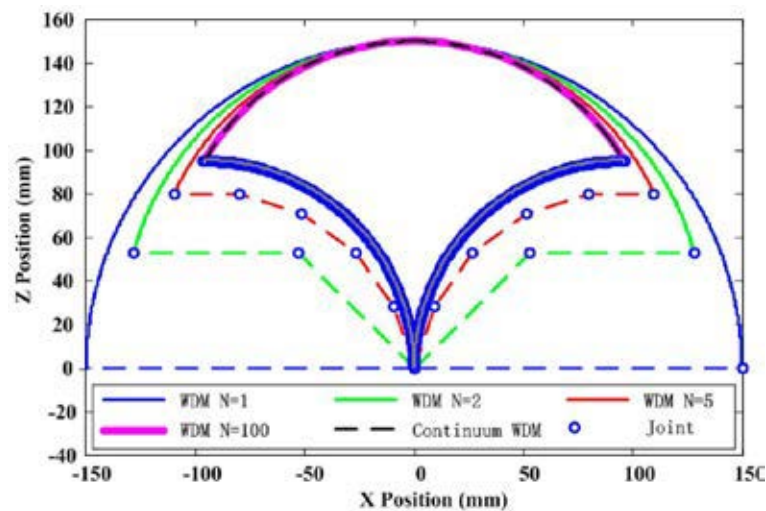


Figure 3-13 Trajectories of the WDM Distal End with Increased Joint Number

Figure 3-13 shows the trajectories of the WDMs in the X - Z plane using the generalized workspace model. The dashed lines represent the vertebra axis, and the curves show the trajectories. In the simulation, the length of the WDM is $L=150\text{ mm}$, and the rotation angle of the distal end is from -90° to 90° . From the figure, it is seen that with an increasing number of vertebrae, the trajectory of the serpentine WDM becomes closer to the trajectory of the continuum WDM. When N is 100, the two trajectories almost overlap. This also validates the generalized kinematics model.

3.3.2 Workspace of Multi-Segment WDM

From the previous section, the workspace of a single segment planar WDM is a curve, and that of a single segment spatial WDM is a surface. The workspace of a

multi-segment planar WDM is expanded to a 2D face, and that of a multi-segment spatial WDM is expanded to a 3D volume. In general, the analytical expression is very complicated and not easy to obtain. However, it can be solved numerically.

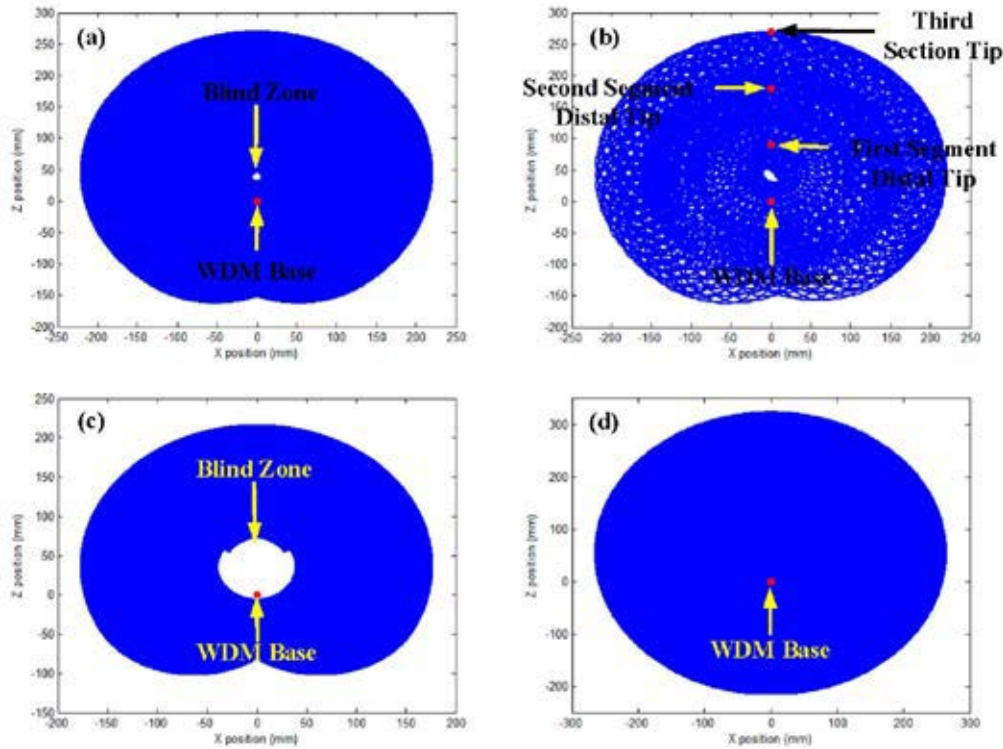


Figure 3-14 Workspace of a Three-Segment WDM in the X - Z Plane. (a) $N=10$, Simulation Interval is 2.5° ; (b) $N=10$, Simulation Interval is 10° ; (c) $N=8$, Simulation Interval is 2.5° ; (d) $N=12$, Simulation Interval is 2.5° .

Figure 3-14 (a) shows the workspace of a three-segment planar WDM. Its motion is in the X - Z plane. Each segment has 10 vertebrae and the length is 90 mm. The rotation range of each joint is $\theta \in [-14.25^\circ, 14.25^\circ]$. In the simulation, the bending interval $\Delta\theta$ is 2.5° . The blue curves indicate the reachable positions of the distal end while the red dot is the WDM base. It is seen that the workspace is symmetric about the Z axis. Figure 3-14 (b) shows the same workspace when the bending interval used in the simulation is increased to 10° . It is noted that the trajectory distribution of the WDM distal end is not uniform. The density of the curve plot indicates the number of ways of reaching the position. The denser the plot, the more ways the WDM can reach the same position, or more dexterous the

WDM is in this location. From the simulation, it is shown that the WDM is more dexterous around the second segment.

It is also noted that there is a blind zone inside the workspace, where the distal end cannot reach. The existence and size of the blind zone is determined by the backbone structure, i.e. maximum bending angle of each segment. When there are eight vertebrae in each segment and the joint rotation range remains the same, the blind zone is as shown in Figure 3-14 (c). It is larger than the previous blind zone. When the vertebra number in each section is increased to 12, the blind zone vanishes, as shown in Figure 3-14 (d). One necessary condition of illuminating the blind zone is that the WDM can reach its base, or the maximum bending angle of the whole WDM is larger than 360° .

3.4 Employing Obstacles to Expand WDM Workspace

The wire-driven mechanism is a type of underactuated flexible mechanism. The backbone deformation and the distal end motion is actively controlled by the wire pairs. However, when there is an external load acting on the backbone, its shape can also be changed, such as when there are obstacles in its way, the backbone deformation is codetermined by the wire control and obstacle location. This raises the idea of employing obstacles to expand the workspace of WDMs.

3.4.1 Constrained Kinematics Model of WDM

In the confined space, there are two types of obstacles. One is the bilateral constraint and the other is the unilateral constraint. Figure 3-15 shows the WDM bending motion in three cases: a) without constraint; b) with bilateral constraint; c) with unilateral constraint. As shown in the figure, when the WDM is under constraint, it is segmented into the anterior constrained section and the distal free section. The bilateral constraint confines the constrained section two-sided, while the unilateral constraint confines the constrained section one-sided. In this section, the kinematic models of WDMs under the two types of constraints are developed. In practice, there are several ways to find the constraint's position, such as stereo vision [76], magnetic sensor [58], etc. Therefore, in the analysis, we assume that the position of the constraint in the WDM local coordinate (x_c, z_c) is known.

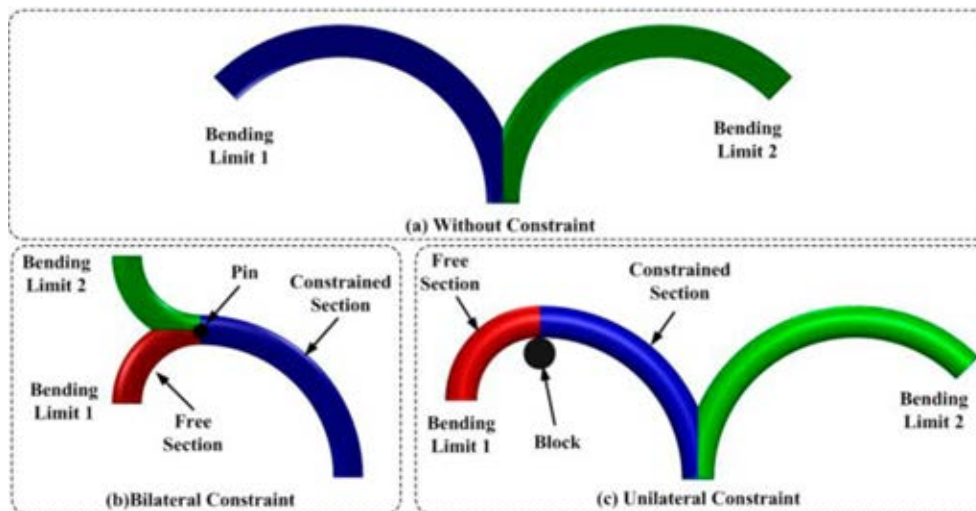


Figure 3-15 Three Types of WDM Motion: (a) without Constraint; (b) with Bilateral Constraint; (c) with Unilateral Constraint

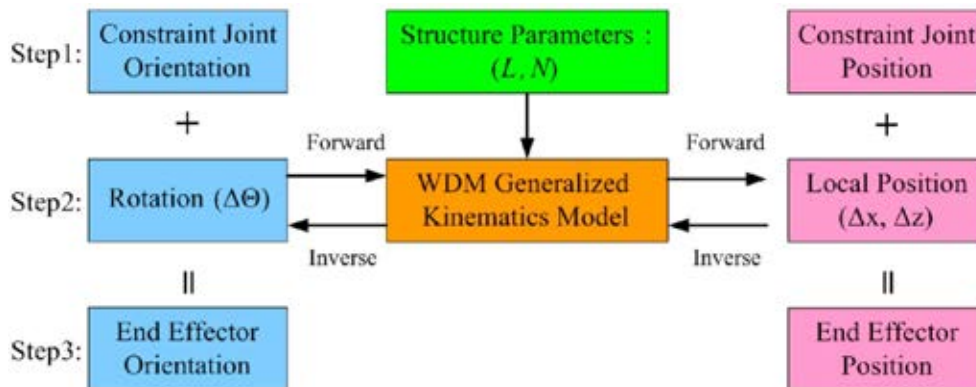


Figure 3-16 Process of Kinematics under Constraint

The kinematics under constraint can be obtained by three steps as shown in Figure 3-16. First, find the position and orientation of the constrained joint; second obtain the position and orientation of the free section with respect to the constrained joint; third, superpose the two parts to determine the final position and orientation of the distal end.

1) Constrained Forward Kinematics

In the analysis, assume that there is only one constraint in the bending plane. Figure 3-17 shows the WDM under constraint. In the figure, C is the constraint position. Assume for the constrained section the number of joints involved is N' ; the joint rotation is θ' ; the length is $L' = N' \cdot l$; the bending angle is $\Delta\Theta_{N'}$; and the location of the constraint on vertebra $P_{N'-1}P_{N'}$ is l' . For the free section, the

number of joints involved is N'' ; the joint rotation is θ'' ; the length is $L'' = L - L'$. The overall bending angle of the WDM is $\Delta\Theta$.

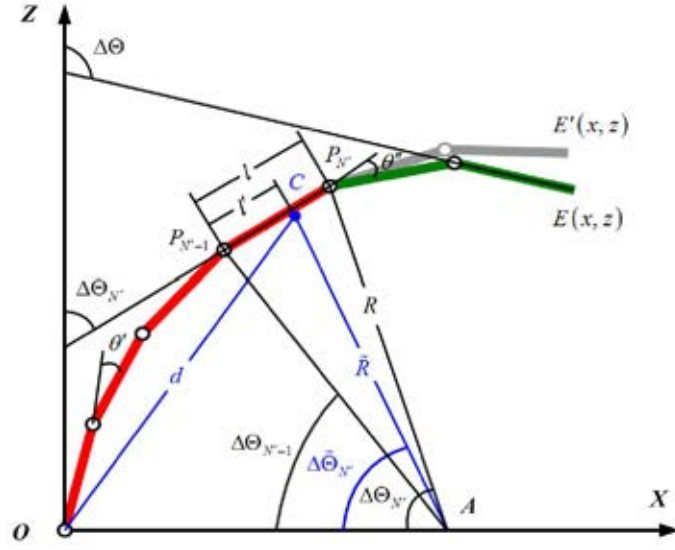


Figure 3-17 WDM with Constraint

To find the end effector position under constraint, the first step is to identify the constrained joint $P_{N'}$. Assume that the position of the constraint is (x_c, z_c) and the robot base is (x_b, z_b) . Then, the distance from the constraint to the base is:

$$d = \sqrt{(x_c - x_b)^2 + (z_c - z_b)^2} \quad (3-54)$$

The bending angle of the constraint section cannot be found directly. However, from Figure 3-17, it can be approximated as Equation (3-55). It is obvious that the approximated bending angle $\Delta\tilde{\Theta}_{N'}$ is between $\Delta\Theta_{N'}$ and $\Delta\Theta_{N'-1}$.

$$\Delta\tilde{\Theta}_{N'} = \pi - 2 \arctan\left(\frac{z_c - z_b}{x_c - x_b}\right) \quad (3-55)$$

Also, the length of the constrained section is close to the corresponding arc length of the circumcircle:

$$\tilde{L}' = \frac{d}{2 \cdot \sin(\Delta\tilde{\Theta}_{N'}/2)} \Delta\tilde{\Theta}_{N'} \quad (3-56)$$

It can be shown that \tilde{L}' is between L' and $L' + l$. As a result, the fixed joint number is the quotient of \tilde{L}' and l as shown in Equation (3-57).

$$N' = \text{floor}(\tilde{L}' / l) \quad (3-57)$$

When the constrained joint number is known, the joint rotation angle can be solved from Equation (3-58).

$$\frac{l \cdot \sin(N' \beta') \sin[(N'-1) \beta']}{x_c \cos(2N' \beta') - z_c \sin(2N' \beta')} = \sin(\beta') \quad (3-58)$$

where, $\beta' = \theta' / 2$. The rotation of the constrained joint is $\Delta\Theta_{N'} = 2N' \beta'$.

The location of the constraint on the WDM is:

$$l' = \sqrt{(z_c - z_{N'-1})^2 + (x_c - x_{N'-1})^2} \quad (3-59)$$

The constrained joint position is:

$$z_{N'} = \frac{\sin(N' \beta')}{\sin(\beta')} \cos[(N'+1) \beta'] \quad (3-60)$$

$$x_{N'} = \frac{\sin(N' \beta')}{\sin(\beta')} \sin[(N'+1) \beta'] \quad (3-61)$$

The constrained joint $P_{N'}$ is now the new base of the WDM. The free section becomes the new WDM. The forward kinematics model of the new WDM is the same as for the previous unconstrained kinematics, with the length $L'' = L - l'$ and joint number $N'' = N - N'$.

2) Constrained Inverse Kinematics

The inverse kinematics aims to answer the question of how to reach a position with given orientation. In the constraint inverse kinematics, the problem becomes how to deploy constraints/obstacles to enable the WDM to reach the position with given orientation.

Assume that the target distal end position is (x, z) and the desired rotation is $\Delta\Theta$. From Figure 3-17, it is seen that as long as the constraints lie on vertebra $P_{N'-1}P_{N'}$, the position and orientation of $P_{N'}$ is the same. As a result, in practice, we can deploy the constraints on the joint, i.e. C is coincident with $P_{N'}$. The position and orientation of $P_{N'}$ is determined by N' and θ' . Furthermore, if θ'' is known, the

end effector position and orientation can be determined. Let $\beta' = \theta' / 2$ and $\beta'' = \theta'' / 2$. From Figure 3-17, the formulas are established as Equation (3-62).

$$\left\{ \begin{array}{l} N = N' + N'' \\ \Delta\Theta = 2(N'\beta' + N''\beta'') \\ X = \frac{L \sin(N'\beta')}{N \sin(\beta')} \sin[(N'+1)\beta'] + \frac{L \sin(N''\beta'')}{N \sin(\beta'')} \sin[2N'\beta' + (N''+1)\beta''] \\ Z = \frac{L \sin(N'\beta')}{N \sin(\beta')} \cos[(N'+1)\beta'] + \frac{L \sin(N''\beta'')}{N \sin(\beta'')} \cos[2N'\beta' + (N''+1)\beta''] \end{array} \right. \quad (3-62)$$

There are four unknowns and four independent equations. Mathematically, there is a unique solution. However, due to the constraint of the WDM structure, the joint rotation is bounded. As a result, only the solution that has valid joint rotation is effective. In other words, when the end effector reaches the target position, its orientation has only $N-1$ choices. In practice, we can set the constraint joint N' from 1 to $N-1$ manually and then find all the effective orientations. Or, when orientation is more important, we can also find the $N-1$ effective positions. It is not certain that we can always meet the two requirements simultaneously. We can meet one first and make the other as close as possible.

Figure 3-18 gives an example. In the example, $N=10$, $l=15$ mm and $\Theta_{\max}=2.48$ rad. In the figures, the red dashed curve is the trajectory of the end effector without constraint. The left and right green polylines are the backbone at two bending limits. The circles denote the joints and the solid square is the constrained joint. Yellow represents the solution is effective, and cyan means in the solution the joint rotation violates the limit. The desired position and orientation are $X=66.1$ mm, $Y=120.7$ mm and $\Theta=1.32$ rad.

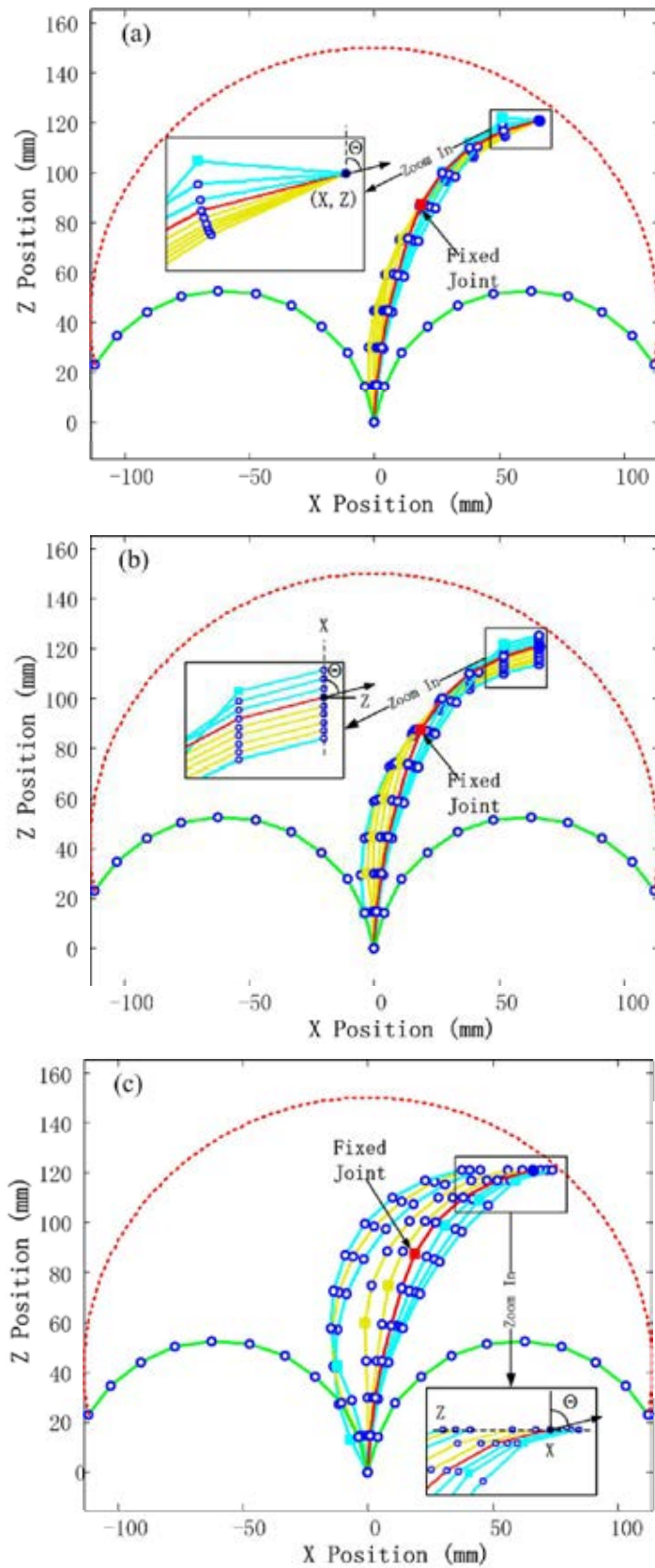


Figure 3-18 Inverse Kinematics: (a) Solution with Exact Position; (b) Solution with Exact Orientation and Exact X position; (c) Solution with Exact Orientation and Exact Z position

In Figure 3-18 (a), bilateral constraint is deployed at joints 1 to 9 successively. In all the cases, the end effector attains the desired position. The enlarged view shows the last vertebra of the WDM. Among the 9 cases, exact solution exists when the sixth joint is fixed, as shown by the red line. For the fixed joints, the rotation is 0.03 rad; the rotation of other joints is 0.12 rad. The curves in yellow represent that the solution is within the maximum joint rotation. Together with the exact solution, there are six possible orientations for the end effector to reach the target position. The curves in cyan denote that the solution violates the WDM structure constraint.

Figure 3-18 (b) shows the solutions with exact distal end orientation and X position. From the simulation results, there are five configurations (lines in yellow and red) that meet the requirement. Figure 3-18 (c) shows the solutions with exact orientation and Z position. From the simulation results, there are four configurations (lines in yellow and red) that meet the requirement. It should be noted that, for WDMs, an exact solution does not always exist. With increased number of vertebrae, it is more likely to have an exact solution. When N is infinity (i.e. continuum WDM), an exact solution is certain.

3.4.2 WDM Workspace with Constraints

The workspace of the WDM is obtained from the forward kinematics. The bilateral constraint and unilateral constraint limit the joint rotation differently. Hence, the corresponding workspaces are different.

1) Workspace with Bilateral Constraint

When the WDM is fixed in the middle, the constrained section is fixed. The free section turns into a new WDM, taking the constrained joint as the base. The distal end position and orientation can be determined as in the previous section. For the free section, the maximum joint rotation remains the same. The distal end trajectory is shown by Equation (3-63).

$$(x-x_{N'})^2+(z-z_{N'})^2=\left(\frac{L''\cdot\sin(N''\theta''/2)}{N''\cdot\sin(\theta''/2)}\right)^2 \quad \theta_{\min}\leq\theta''\leq\theta_{\max} \quad (3-63)$$

In the equation, θ_{\min} and θ_{\max} are the minimum and maximum joint rotation. The constraint position is arbitrary. Therefore, the workspace with bilateral constraint

is the collection of all the possible trajectories. The joint rotation is confined by the WDM vertebra structure. In general $\theta_{\min} = -\theta_{\max}$.

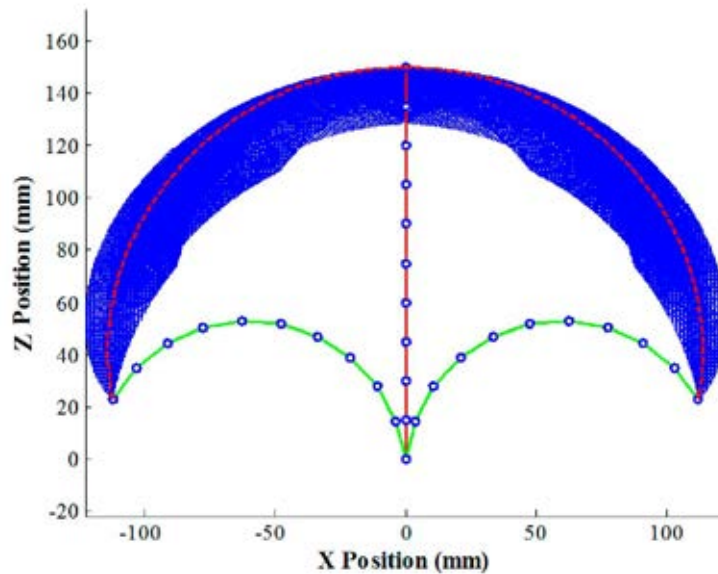


Figure 3-19 Workspace with Single Bilateral Constraint

Figure 3-19 shows the workspace of a WDM with a single bilateral constraint. The green lines show the two bending limits of the WDM. The blue circles represent the joints. The red line is the WDM in the rest position. The red dashed curve is the WDM end effector trajectory without constraint. The blue region shows the WDM workspace with single bilateral constraint. It is obvious that the workspace of the WDM is expanded a lot (blue region VS red dashed curve). It should be noted that for a specific bilateral constraint, the distal end trajectory remains a curve. The workspace is the collection of all trajectories with permissible constraint.

2) Workspace with Unilateral Constraint

When there is a unilateral constraint, such as a stone, the WDM cannot cross over the constraint but can return freely. The free section motion is partitioned into forward bending and backward bending. The forward bending is the same as that with bilateral constraint and the backward bending is the same as that without constraint. Assume that the unilateral constraint is located in the right plane, i.e. $\theta' > 0$, the end effector trajectory with unilateral constraint is represented as Equation (3-64). In the equation, θ_{\max} and θ_{\min} are the maximum and minimum

joint rotation, and θ' is determined as in the previous section. When the constraint is located in the left plane, i.e. $\theta' < 0$, the workspace has the same form with joint rotation range of: $\theta_{\min} \leq \theta'' < \theta'$ in the first part and $\theta' \leq \theta \leq \theta_{\max}$ in the second part. The workspace with unilateral constraint is also the collection of all the possible trajectories.

$$\begin{cases} (x-x_{N'})^2+(z-z_{N'})^2=\left(\frac{L''\cdot\sin(N''\theta''/2)}{N''\cdot\sin(\theta''/2)}\right)^2 & \theta' < \theta'' \leq \theta_{\max} \\ (x-x_b)^2+(z-z_b)^2=\left(\frac{L\cdot\sin(N\theta/2)}{N\cdot\sin(\theta/2)}\right)^2 & \theta_{\min} \leq \theta \leq \theta' \end{cases} \quad (3-64)$$

Figure 3-20 shows the workspace of a WDM with a single unilateral constraint. In the simulation, a single obstacle is placed on the bending plane of the WDM. The legend is the same as that in the previous figure: blue region shows the workspace with constraint; the red dashed curve is the distal end trajectory without constraint; the green lines are the two bending limits of the WDM; the blue circles represent the WDM joints; and the red line in the middle is the WDM in the rest position. In this case the workspace is also expanded a lot. It is noted that the workspace under unilateral constraint is bounded by the free path. From the representation as well as the simulation, it is known that the workspace with unilateral constraint is a subspace of the workspace with bilateral constraint.

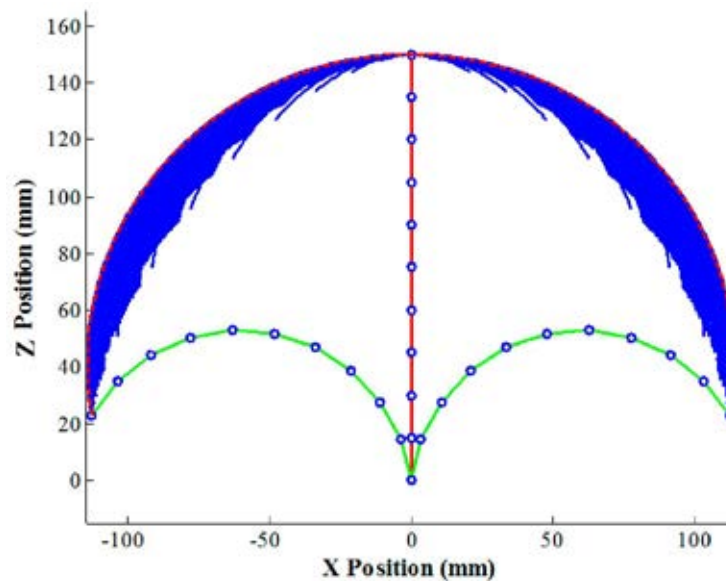


Figure 3-20 Workspace with Single Unilateral Constraint

The WDM workspace with unilateral constraint and bilateral constraint are both symmetric. When the UFM is confined by multiple constraints or hybrid constraints the forward kinematics, inverse kinematics and workspace can be solved in a similar way. In this case, the WDM is divided into multiple sections. As long as the constrained joints are found, distal end position and orientation can be solved, as well as the workspace.

3.5 Model Validation via Experiment

To validate the kinematic models, two WDM manipulators were designed and built. One was a SPSP WDM, and the other a MSSP WDM, with three segments.

3.5.1 Single Segment WDM Kinematic Model Validation

To validate the single segment WDM kinematic model, a SPSP WDM based manipulator was built as shown in Figure 3-21.

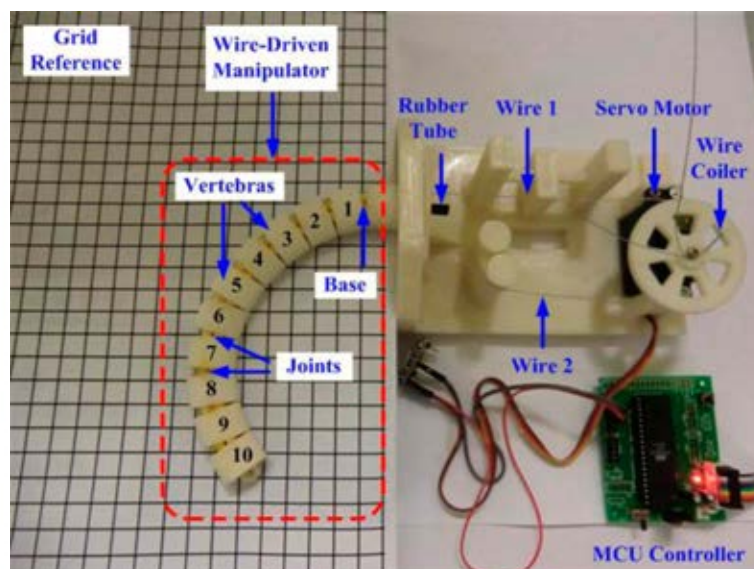


Figure 3-21 Manipulator with SPSP WDM

The vertebrae were built using Rapid Prototyping (RP). There are 10 vertebrae and 10 joints. The maximum rotation angle of the manipulator is $\theta = 142.5^\circ$ (14.25° for each joint). It is noted that the friction of the joints can be uneven, which makes the bending non-uniform. To eliminate this problem, a rubber tube is placed through the central cavity of each vertebra. It provides a restoring spring force and distributes the joints' rotations. The vertebrae together with the rubber tube serve as the backbone of the manipulator. The deformed backbone is close to

a constant curvature circular arc. The motion of the manipulator is planar, hence, only one pair of wires was used. The wires are fishing lines, and are fastened onto a wire coiler, which is driven by a servo motor. The motor is controlled by a MCU (Model: AVR ATmega 16).

The backbone is coincident with Z axis when in the rest position. During the experiment, the distal end positions are recorded at different bending angles using a grid paper. In the experiment, the distal end moved from the left extreme position to the right limit. The experiment result is summarized in Figure 3-22. In the figure, the red line is the predicted trajectory using the derived kinematic model, while the blue stars are the recorded positions. It is noted that the actual position is very close to the predicted one. As shown in Figure 3-23, the relative positioning error is generally within 2% (black dashed line). More specifically, the position error in the Z direction (red line) and error in the X direction (blue line) are less than 3% with few exceptions, such as at the two bending limits. At these positions, the absolute value is small. Although the discrepancy is small, the relative error is large. Compared to traditional rigid discrete manipulators, the positioning error is large. This may be attributed to a number of factors, such as prototyping error, modeling simplification error, non-uniform friction among the joints, etc. Nonetheless, this validates the developed kinematic model.

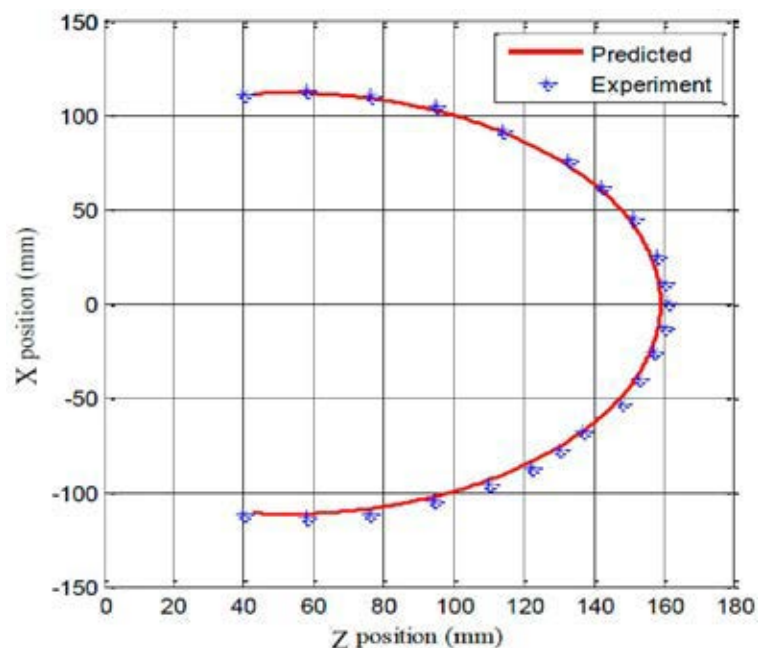


Figure 3-22 Trajectory of the WDM Manipulator End Effector

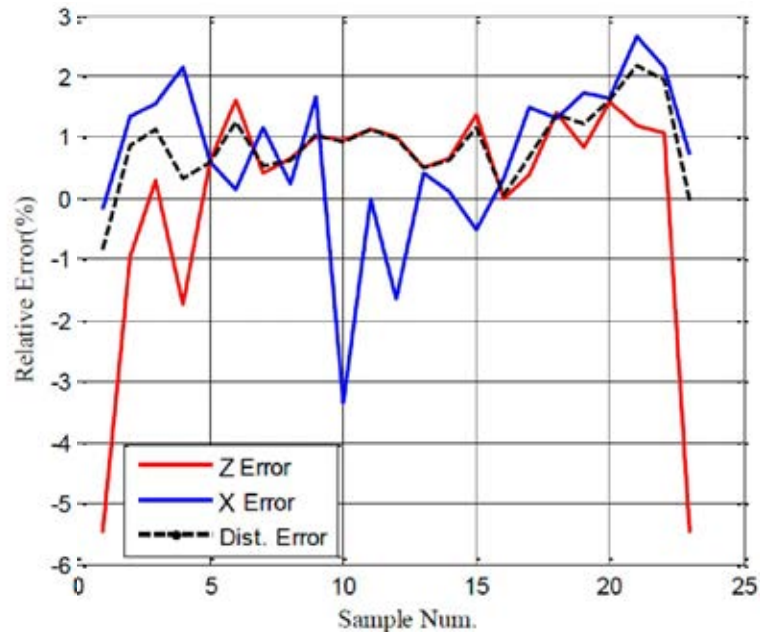


Figure 3-23 Relative Positioning Error of the Manipulator

3.5.2 Multi-Segment WDM Kinematic Model Validation

To validate the proposed multi-segment WDM kinematic model, a multi-segment manipulator was designed and built, as shown in Figure 3-24. It is based on the MSSC WDM.

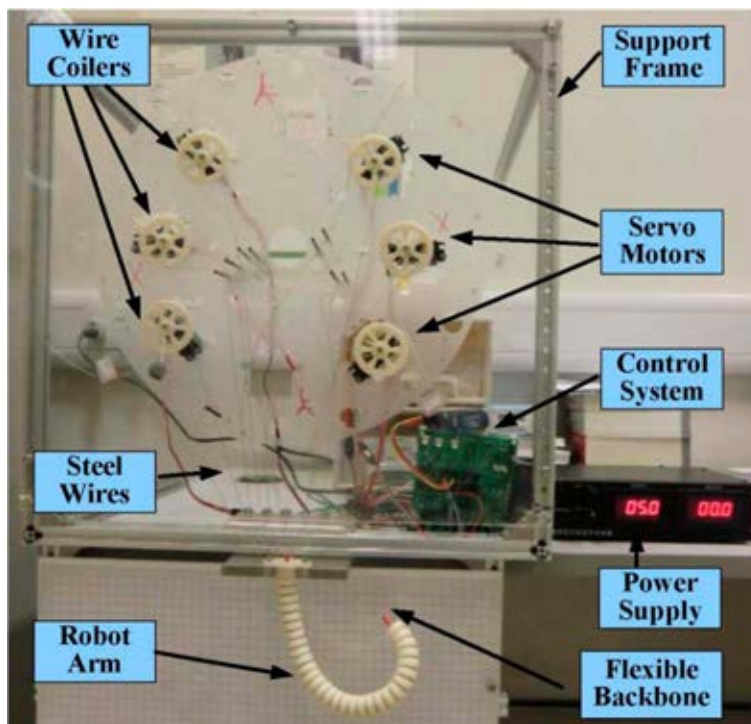


Figure 3-24 Three-Segment WDM Manipulator

It has three segments, and each segment has 10 vertebras. The vertebras are made by RP. The outer diameter of the vertebra is $D = 20$ mm, while the pilot holes are evenly distributed on a circle with a diameter of $d = 15$ mm. The height of each node $H = 6.5$ mm, and the initial gap distance $h_0 = 2.5$ mm. In this design, the maximum rotation angle for each joint is 14.25° . A 5 mm diameter rubber tube is used as the returning spring for each joint. Six groups of steel wires with 0.475 mm diameter are used to control the robot. These wires are pulled by servomotors, whose maximum torque is 13 kg·cm. The controller is developed using the commercial MCU (Model: AVR ATmega 128).

The manipulator has six controllable DOFs. Bending shapes of the manipulator can be various. Figure 3-25 (a) shows the manipulator in the rest position and Figure 3-25 (b) to Figure 3-25 (l) show various bending cases.

In particular, four cases, i.e., (b), (c), (d) and (e) are studied in detail. In these cases, three segments all bend in the X-Z plane. The bending angles for each segment are shown in Table 3-1. In case (b), only the segment near the manipulator base bends. The other two segments remain still. In case (c) only the second segment bends, and in case (d) only the third segment bends respectively. In case (e), the three segments bend together, and their bending angles are the same.

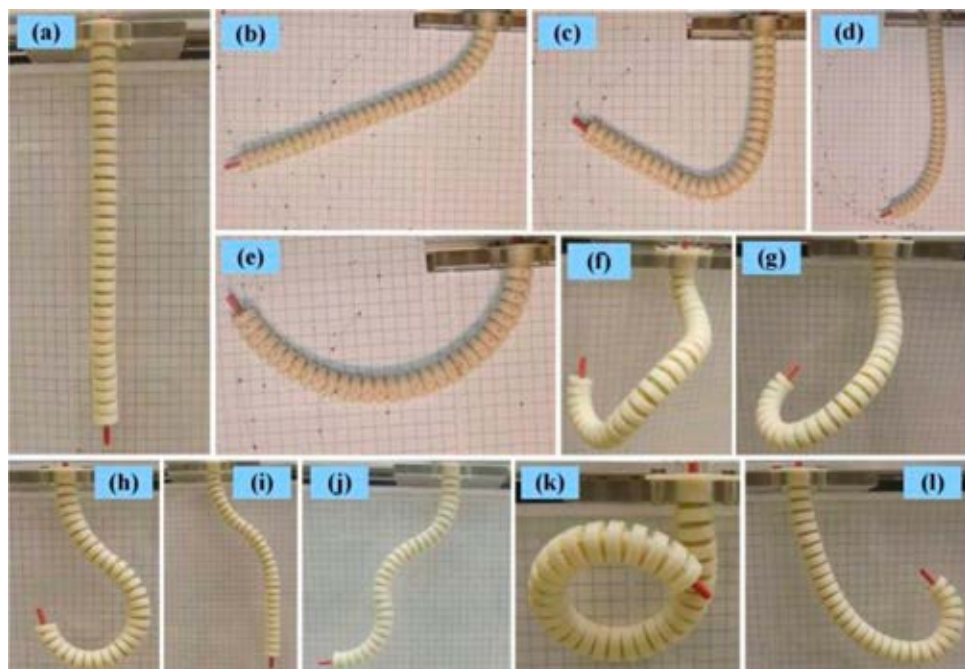


Figure 3-25 Three-Segment WDM Manipulator Bending Cases

Table 3-1 Segment Bending Angles in the Experiment

	Segment 1	Segment 2	Segment 3
Case (b)	0 ~ 142.5°	0	0
Case (c)	0	0 ~ 142.5°	0
Case (d)	0	0	0 ~ 142.5°
Case (e)	0 ~ 142.5°	0 ~ 142.5°	0 ~ 142.5°

In the test, no payload was applied to the manipulator. The power consumption of the manipulator is affected by the motion type and manipulator configuration. Generally, the power consumption of each motor is less than 1W, and the power consumption of the control system is around 0.5W. At larger bending angles, a bigger moment is needed to maintain the configuration of the manipulator. The more motors involved in the motion and the larger the bending angle, the higher the power consumption of the manipulator. In the four cases, the trajectories of the distal end were measured using grid paper, as shown in Figure 3-25, and then compared with the model predictions.

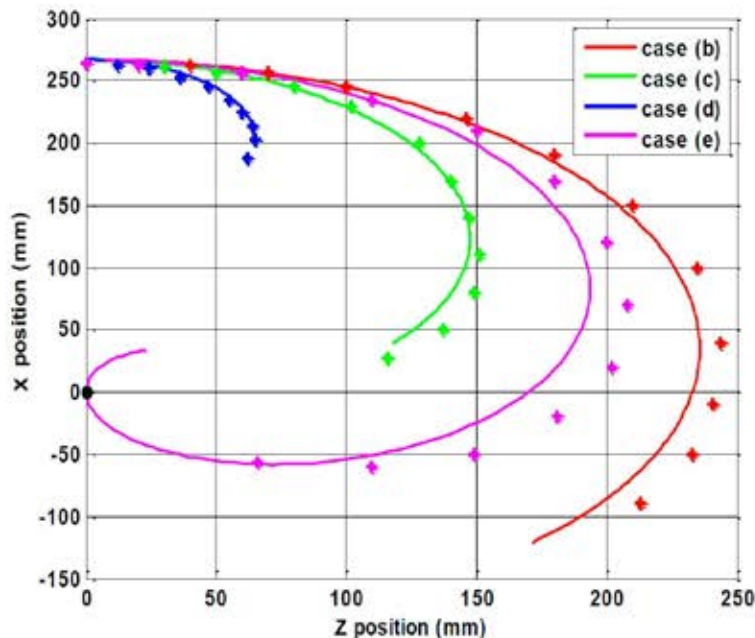


Figure 3-26 Trajectories Comparison of the Distal End

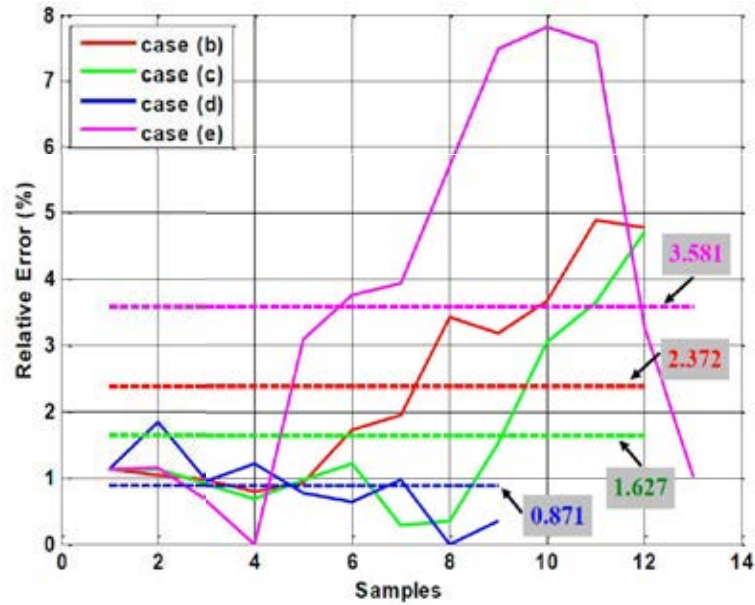


Figure 3-27 Relative Positioning Error of the Distal End

The results are shown in Figure 3-26. The curves are the predicted distal end trajectory, while the dots are the measured positions along the trajectory. Case (b) is shown in red; Case (c) is in green; Case (d) is in blue and Case (e) is in magenta. The experiment results and the model prediction match reasonably well.

Figure 3-27 shows the relative positioning error. The curves show the error in each measured position, while the dashed lines show the average error. From the figure, it is seen that the average positioning errors in the four cases are 2.372%, 1.627%, 0.871% and 3.581% respectively. These results are consistent with the results for the single segment manipulator. It should be pointed out that the positioning error is smaller than that of the continuum robot. As reported in [25], the average prediction error of OctArm V using the piecewise constant curvature model is nearly 50% of the robot length. A close examination reveals that the longer the moving part, the larger the error. The reason is: on one hand the flexibility of the robot increases with increasing moving part length; on the other hand the error near the robot base is accumulated and amplified at the distal end.

Compared with traditional rigid discrete robots, the positioning accuracy of the wire-driven robot is low. Also, it is more affected by the robot configuration and external perturbations. Meanwhile, gravity also impairs the accuracy of the piecewise constant curvature assumption. As a result, such a robot is more

suitable for inspection applications, e.g. endoscopy, and applications where gravity is trivial, e.g. space robots, underwater robots, etc.

3.5.3 Constrained Kinematic Model Validation

The single segment WDM manipulator shown in Figure 3-21 was also used to test the constrained kinematic model. Two experiments were carried out. In the first experiment, the trajectory of the end effector under bilateral constraint is recorded and compared with model predictions. In the second experiment, the end effector trajectory under unilateral constraint is compared with simulated trajectory.

1) End-effector Trajectory with Bilateral Constraint

In this experiment, the trajectory with bilateral constraint is measured and compared with model predictions. The bilateral constraint can be applied to the manipulator internally (through the tube) or externally (bilateral from outside). In the experiment, internal bilateral constraint is applied to the manipulator by inserting a steel bar into the rubber tube. External bilateral constraint is applied by pinning a needle to the joint, as shown in Figure 3-28. When the manipulator moves to a position, the joint is pinned to the bending plane by the needle.

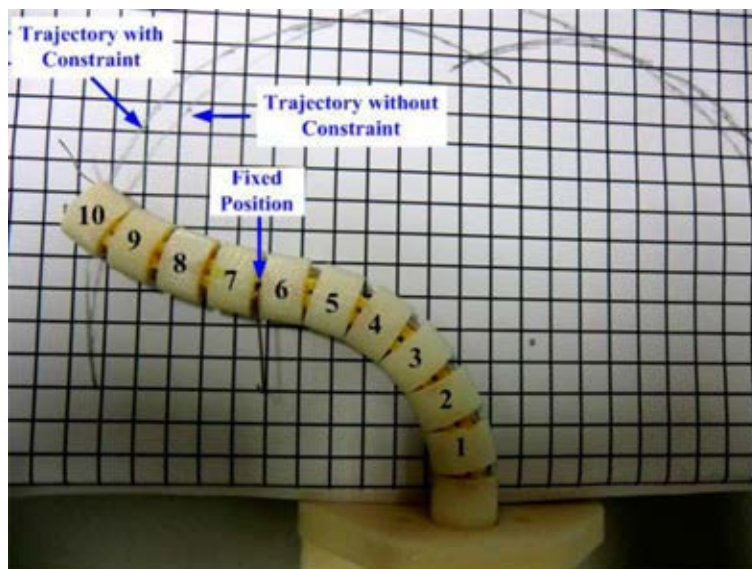


Figure 3-28 WDM Manipulator Moving with External Bilateral Constraint

Figure 3-29 shows three trajectories of the manipulator end effector with internal bilateral constraint. In the figure, curves represent the simulated trajectories; diamonds are the recorded positions on the end effector trajectories; poly lines are

the vertebrae of the manipulator at two bending limits; circles denote the joints of the backbone; the constraints are represented by squares. Three cases are shown in the same figure using different colors. Trajectory without constraint is shown in blue; trajectory with the third joint fixed is shown in green; trajectory with the fifth joint fixed is shown in red. From the results, it is seen that the proposed algorithm predicts the trajectory of the end effector well. Meanwhile, it is shown that with the constraint, the trajectory is different from the free path. It is noted that with internal bilateral constraints, the number of movable vertebrae is reduced. Hence, the trajectories are shortened. Also, the trajectories with internal bilateral constraint are bounded by the free path.

Figure 3-30 shows three cases of the manipulator end effector trajectory. In this test, the constraints are external bilateral constraints. The same as before, the curves are the simulated trajectories; the diamonds are the measurements; the lines are the manipulator vertebrae; the circles are the joints; the squares are the constraints. Manipulator trajectory without constraint is in blue; trajectory with the third joint fixed is shown in green; trajectory with the sixth joint fixed is shown in red. In the test, the WDM flexes into a configuration at the beginning. Thereafter, the joint is pinned to the work plane by the needle as shown in Figure 3-28. The trajectory of the end effector is recorded using grid paper.

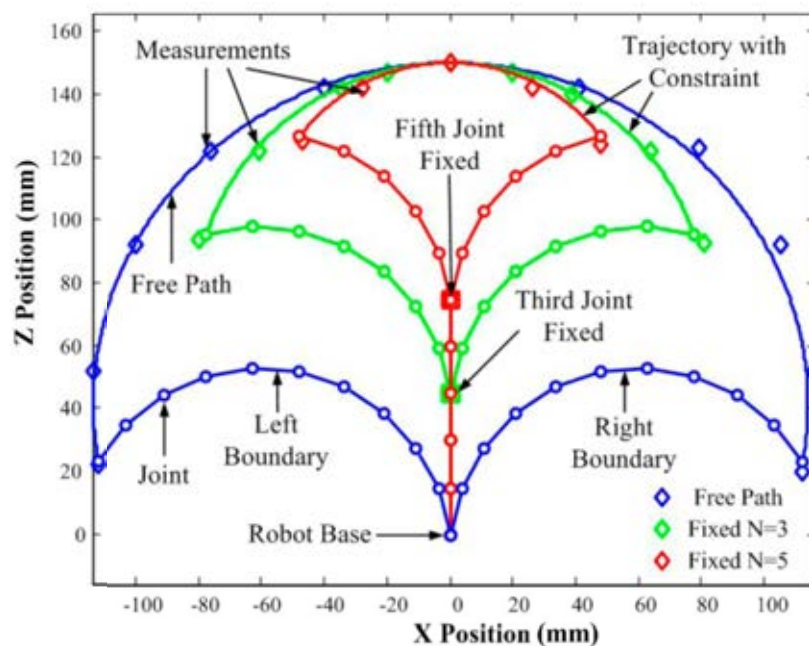


Figure 3-29 End Effector Trajectory Comparison - Internal Bilateral Constraint

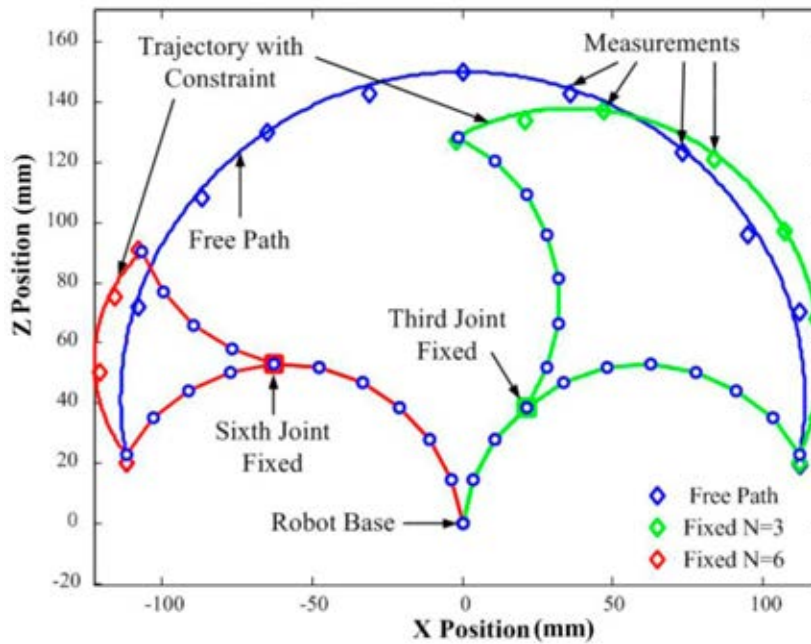


Figure 3-30 End Effector Trajectory Comparison - External Bilateral Constraint

From the results, it is seen that the proposed algorithm predicts the end effector trajectory with external bilateral constraint well. Also, the changed trajectory shows that the workspace is expanded by the constraint. However, different from that of internal bilateral constraint, the trajectory under external bilateral constraint is not bounded by the free path any more. This means that with external bilateral constraint the workspace can be expanded even more. In fact, when the external bilateral constraint is located in the WDM rest position, the end effector trajectory is the same as that under internal bilateral constraint. As a result, the workspace with internal bilateral constraint is a subspace of the workspace with external bilateral constraint.

2) End-effector Trajectory with Unilateral Constraint

In this experiment the algorithm of predicting the end effector trajectory with unilateral constraints is tested. The simulated end effector trajectory is compared with the measurements. In the simulation, multiple constraints are applied.

The results are as shown in Figure 3-31. Unilateral constraints are randomly placed on the bending plane as shown by the squares. At the beginning, the manipulator is in the rest position as shown by the magenta lines. By controlling the wire pair, the manipulator bends to both sides. When colliding with the constraint, the joints between the constraint and the manipulator base stop

rotating. The other joints rotate continuously until they reach the limit. The green poly lines show the two limits of the manipulator under unilateral constraints. As shown in the figure, in the left half plane, the first three vertebrae are confined by the unilateral constraints; in the right half plane, the first five vertebrae are confined by the unilateral constraints. The trajectory of the end effector under unilateral constraint is as shown by the red curve. The blue curve is the trajectory without constraint. In the figure, the diamonds are the recorded positions. The blue ones are for the free path and the red ones are for the trajectory with constraint. Experiment results show that the model predicts the end effector trajectory with unilateral constraint well. The new trajectory is also bounded by the free path. All the trajectories with possible block constraints make up the workspace. The workspace with unilateral constraint is also a subspace of the workspace with external bilateral constraint.

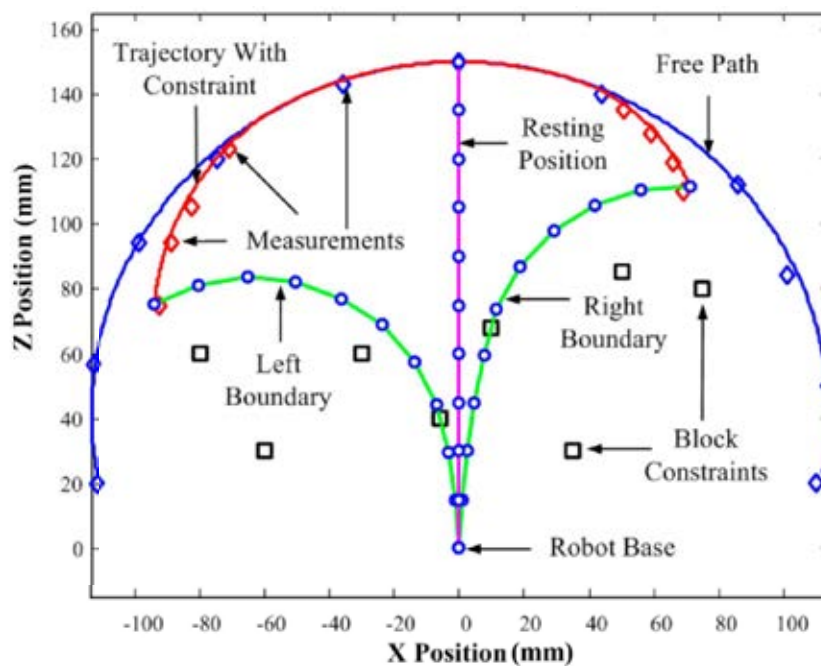


Figure 3-31 End Effector Trajectory Comparison - Unilateral Constraint

3.6 Summary

In this chapter, kinematic models of single segment serpentine WDM and single segment continuum WDM are developed from geometry analysis. The continuum WDM is a special case of serpentine WDM with an infinite number of vertebrae. As a result, a more generalized kinematic model is established. The multi-

segment WDM kinematics model is established by the extended D-H method, in which the single segment WDM is treated as a link with two rotation DOFs and one translation DOF. The workspace model of the WDM is developed from the forward kinematics model. For single segment WDM, the workspace is a spheroidal surface. For multi-segment WDM, the workspace is expanded to a 3D space. There may have a blind zone inside the workspace, depending on the WDM maximum bending angle. The WDM is highly underactuated and flexible. By this property, a novel idea of employing obstacles to expand the WDM workspace is proposed. The kinematic models are validated by two wire-driven manipulators. Results show that, the proposed models can predict the distal end trajectory well. The averaged relative error is less than 3.6%.

Chapter 4 Statics and Dynamics of the Wire-Driven Mechanism

In this chapter, the static model and dynamic model of both SPSP WDM and SPCP WDM are developed. In the static and dynamic analysis, the multi-segment WDM can be treated as a single segment WDM with different loading conditions.

4.1 Static Model of the Wire-Driven Mechanism

In this section the static models of two representative WDMs, i.e., SPSP WDM and SPCP WDM, are derived. The SPSP WDM is modeled as a multi-link structure with a torsion spring on each joint. The SPCP WDM is modeled as a Euler-Bernoulli cantilever beam, as the backbone is thin and bends only. For other WDMs, such as multi-segment WDM, spatial WDM, and WDM with tapered wire configuration, their modeling are similar. The major difference is in the boundary conditions.

4.1.1 Static Model of SPSP WDM

The SPSP WDM is a N -link manipulator with constraints on each joint. In this example, the constraint is applied via a uniform elastic tube, as shown in Figure 4-1 (a). As shown in the figure, the backbone has two parts. One is the rigid vertebrae and the other is the elastic tube. The elastic tube serves as the torsion spring on each joint. It constrains the joint rotations. The tube is uniform. As a result, the constraint on each joint is the same. On the other hand, the elastic tube deformation is also confined by the vertebrae's movement. In the rest position, the tube and the vertebra axis are colinear. When the backbone bends, the tube is coincident with the neutral axis. The tube deformation is pure bending. Its length does not change during the backbone bending, as the vertebrae are all rigid. In the analysis, the wires are not considered as they are soft and massless.

There are two categories of forces acting on the backbone. One is the external load, and the other is the controlling forces from the wires. In practice, the external load applied on the backbone can be various, such as distributed force, lumped force, and moment. From the theorem of reciprocal displacements [127],

any arbitrary external load can be viewed as a concentrated force and a pure moment at the backbone distal end. The deformations under these two loading conditions are also equivalent. Hence, in the analysis, only one external loading condition is considered, that is, a concentrated force F_{ex} , a concentrated force F_{ey} , and a pure moment M_e are applied to the distal end of the backbone. The backbone bending is actively controlled by the wires. The two wires are equally pretensioned. Therefore, in the rest position, the resultant controlling force on the backbone is an axial force without bending moment. When the two wires change lengths, the tensions in the wires are also different. Also, when external loads are applied, the tensions in the wires increase. The resultant controlling forces applied to the backbone are a concentrated force and a bending moment. Figure 4-1 (b) shows the loading condition of the SPSP WDM. In the figure, F_{ex} , F_{ey} and M_e are the external loadings, and T_1 and T_2 are the wire tensions.

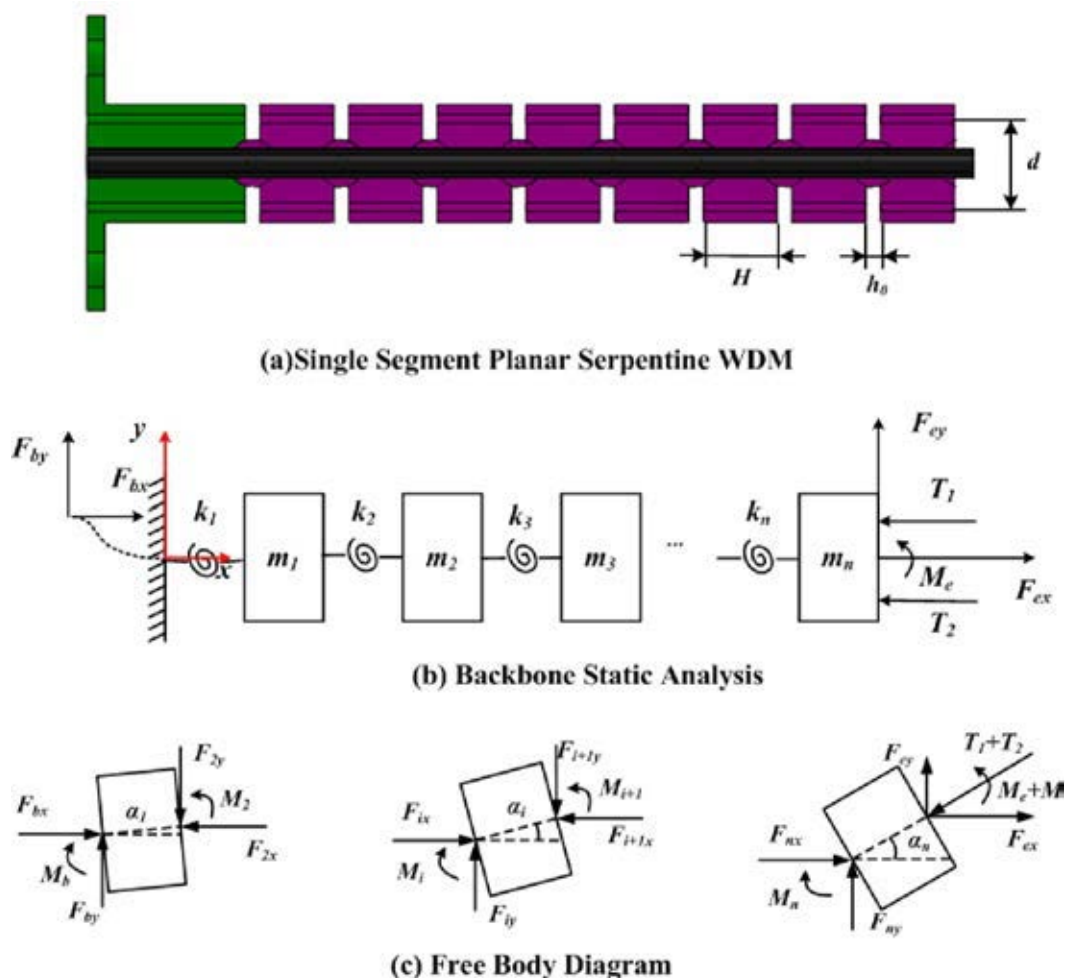


Figure 4-1 SPSP WDM Static Analysis

Figure 4-1 (c) shows the free body diagram of the vertebrae. In the figure, F_{bx} , F_{by} and M_b are the forces and moment from the WDM base; F_{ix} , F_{iy} and M_i are the forces and moments among the joints. When distal end joint rotation is α_n , or θ , the actuation moment M is:

$$M = \Delta T \cdot d \quad (4-1)$$

where, $\Delta T = T_1 - T_2$ and d is the wires' spacing distance.

Assume the Young's modulus of the elastic tube is E , and in the bending direction the second axial moment of area is I_z . For each joint, the rotation is small. Therefore, the torque applied on the joints by the tube is as per Equation (4-2). At the first joint, the torque is $M_b = EI_z \cdot \alpha_1 / h_0$.

$$M_i = EI_z \cdot (\alpha_i - \alpha_{i-1}) / h_0, \quad i=2 \text{ to } n \quad (4-2)$$

From the free body diagram, we can find the static model of the SPSP WDM as:

For the first vertebra, i.e. $i=1$:

$$\begin{cases} F_{bx} = -F_{2x} \\ F_{by} = -F_{2y} \\ M_b = M_2 + (H + h_0) [F_{2x} \sin(\alpha_1) - F_{2y} \cos(\alpha_1)] \end{cases} \quad (4-3)$$

For the middle vertebrae, where i from 2 to $n-1$:

$$\begin{cases} F_{ix} = -F_{i+1x} \\ F_{iy} = -F_{i+1y} \\ M_i = M_{i+1} + (H + h_0) [F_{i+1x} \sin(\alpha_i) - F_{i+1y} \cos(\alpha_i)] \end{cases} \quad (4-4)$$

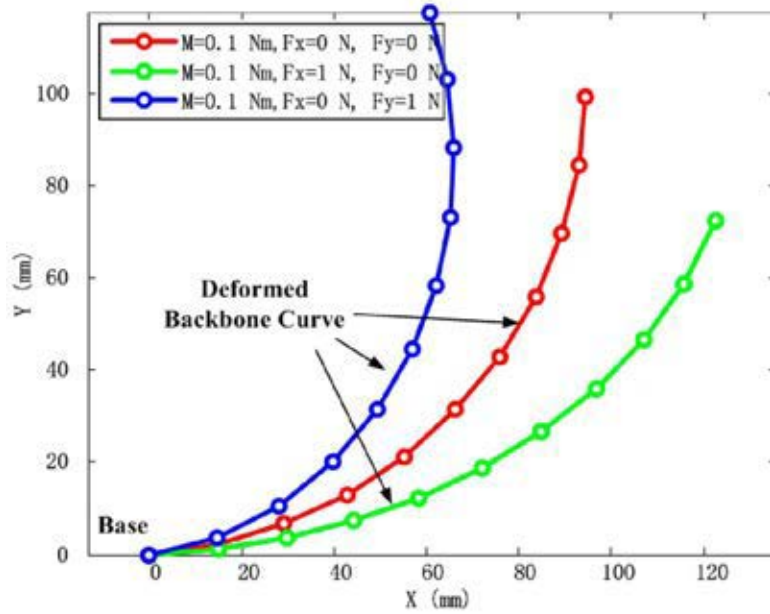
For the last vertebra, i.e. $i=n$:

$$\begin{cases} F_{nx} = (T_1 + T_2) \cos(\alpha_n) - F_{ex} \\ F_{ny} = (T_1 + T_2) \sin(\alpha_n) - F_{ey} \\ M_n = M_e + M + (H + h_0) [F_{ex} \sin(\alpha_n) - F_{ey} \cos(\alpha_n)] \end{cases} \quad (4-5)$$

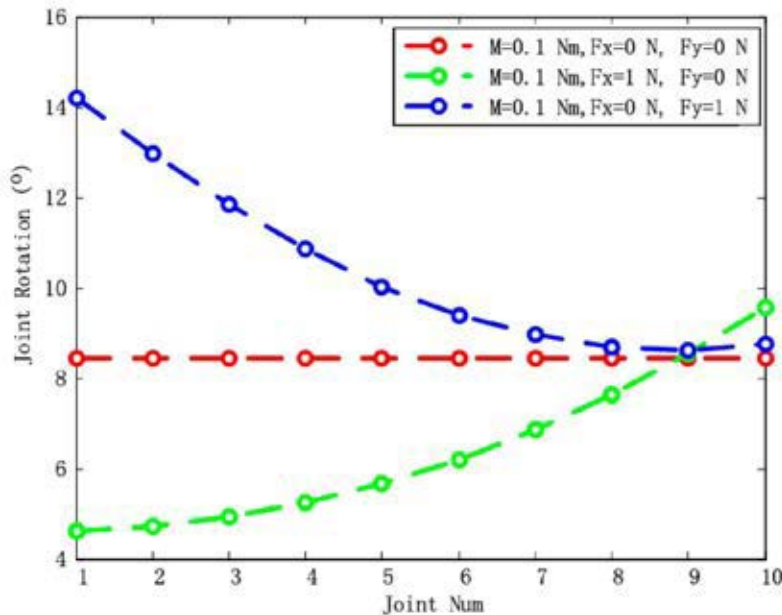
1) Deformed Backbone Curve

In the statics analysis, the general concern is the deformed shape of the backbone under given loading conditions. Assume all the forces, including the external

loads and actuation forces applied on the backbone are known. From Equation (4-2) to Equation (4-5), there are $4n$ unknowns ($M_i, F_{ix}, F_{iy}, \alpha_i$) and $4n$ independent equations. Mathematically, there is a unique solution. The deformed backbone shape is determined by the joint rotations. In the solution, the forces and moments can be viewed as intermediate variables.



(a) Deformed Backbone Curve



(b) Joint Rotation

Figure 4-2 SPSP WDM Static Analysis: (a) Deformed Backbone Curve; (b) Joint Rotations

The above shows one simulation example. In the simulation, the backbone has 10 vertebrae. For each vertebra $H=12.5$ mm and $h_0=2.5$ mm. For the elastic tube, $E=1.5$ GPa. The cross-section is ring shaped. The outer radius is $r_1=2.5$ mm, and the inner radius is $r_2=2.0$ mm. Three loading conditions are simulated: i) $M_n=0.1$ Nm, $F_{nx}=0$ N, $F_{ny}=0$ N; ii) $M_n=0.1$ Nm, $F_{nx}=1$ N, $F_{ny}=0$ N; iii) $M_n=0.1$ Nm, $F_{nx}=0$ N, $F_{ny}=1$ N. Here, M_n is the resultant bending moment; F_{nx} and F_{ny} are the resultant horizontal force and vertical force.

The results are shown in Figure 4-2. Figure 4-2 (a) shows the deformed backbone curve and Figure 4-2 (b) shows the joint rotations. From the results, when the backbone is deformed by a pure moment, as shown by the first loading condition, the rotations of each joint are the same. Under this loading condition, the constant curvature assumption is valid. However, the constant curvature assumption is invalid when there are other loadings. As shown in loading conditions ii) and iii), when $M_n=0.2$ Nm and $F_{nx}=1$ N, the joint rotations increase gradually, and when $M_n=0.2$ Nm and $F_{ny}=1$ N, the joint rotations decrease constantly.

2) Controlling Forces

In the above example, the external loadings and actuation forces are all known. In practice, one may be more concerned about how to manipulate the backbone tip into a desired position under given external loading conditions. In other words, when the tip position and external loadings are known, what are the controlling forces, and what are the joint rotations or the deformed backbone shape?

Let $P=T_1+T_2$, combined with Equation (4-1) we have:

$$\begin{cases} T_1=0.5(P+M/d) \\ T_2=0.5(P-M/d) \end{cases} \quad (4-6)$$

The backbone tip position is:

$$\begin{cases} x = \sum_{i=1}^n (H + h_0) \cdot \cos(\alpha_i) \\ y = \sum_{i=1}^n (H + h_0) \cdot \sin(\alpha_i) \end{cases} \quad (4-7)$$

From Equation (4-2) to Equation (4-5), we have:

$$\begin{cases} F_{nx} = P \cdot \cos(\alpha_n) - F_{ex} \\ F_{ny} = P \cdot \sin(\alpha_n) - F_{ey} \end{cases} \quad (4-8)$$

Let $K = EI_z/h_0$. For each vertebra, the moment equations are:

For $i=1$:

$$K \cdot \alpha_1 = K \cdot (\alpha_2 - \alpha_1) + (H + h_0) [F_{nx} \sin(\alpha_1) - F_{ny} \cos(\alpha_1)] \quad (4-9)$$

For $i=2$ to $n-1$:

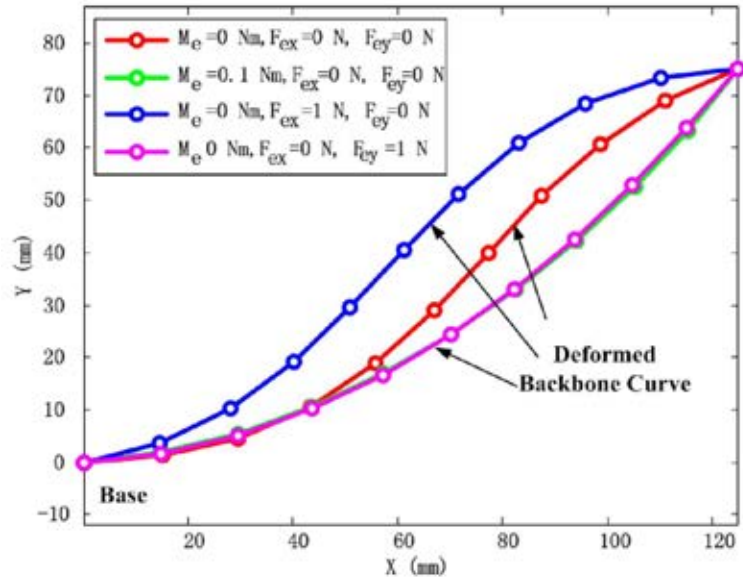
$$K \cdot (\alpha_i - \alpha_{i-1}) = K \cdot (\alpha_{i+1} - \alpha_i) + (H + h_0) [F_{nx} \sin(\alpha_i) - F_{ny} \cos(\alpha_i)] \quad (4-10)$$

For $i=n$:

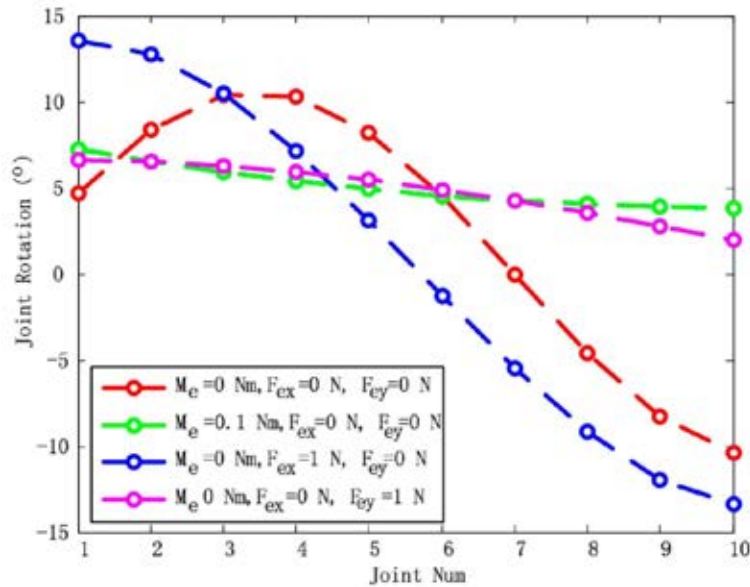
$$K(\alpha_n - \alpha_{n-1}) = M_e + M + (H + h_0) [F_{ex} \sin(\alpha_n) - F_{ey} \cos(\alpha_n)] \quad (4-11)$$

From Equation (4-7) to Equation (4-11) there are in total $n+4$ unknowns (i.e. α_i , M , P , F_{nx} , and F_{ny}) and $n+4$ independent equations. Mathematically, a unique solution exists. The wire tensions T_1 and T_2 can be solved from Equation (4-6).

An example is shown in Figure 4-3. In the simulation, the desired backbone distal end position is $x=125$ mm, and $y=75$ mm. Four external load conditions are considered: i) without external load; ii) $F_{ex}=1$ N; iii) $F_{ey}=1$ N; and iv) $M_e=0.1$ Nm. Figure 4-3 (a) shows the deformed backbone curve, and Figure 4-3 (b) shows the rotation angle of each joint. In the four loading conditions, the actuation forces are: case i) $T_1=-1.595$ N, $T_2=10.7042$ N; case ii) $T_1=-4.7125$ N, $T_2=11.0142$ N; case iii) $T_1=1.3336$ N, $T_2=0.0413$ N; and case iv) $T_1=-3.1473$ N, $T_2=2.3146$ N. In the simulation results, some of the wire tensions are negative. As we know, wires can pull and cannot push. This means wires can only provide positive tension. As a result, although a solution can be found mathematically, in reality the SPSP WDM backbone tip cannot reach arbitrary positions with external loads.



(a) Deformed Backbone Curve



(b) Joint Rotation

Figure 4-3 SPSP WDM Backbone Reaches a Desired Position: (a) Deformed Backbone Curve; (b) Joint Rotations

4.1.2 Static Model of SPCP WDM

For the SPCP WDM, the backbone is a thin beam. The deformation of the beam is pure bending. The static model of a single segment continuum WDM is established based on the nonlinear Euler-Bernoulli Beam Theory and Elastica Theory. From the theory, curvature of the deformed backbone is proportional to the bending moment. Elastica is the exact shape of the deflection curve of a flexible member [128]. From the Elastica Theory, the backbone is inextensible, or

the beam length after deflection is unchanged. In the statics analysis, the backbone is viewed as a cantilevered Euler-Bernoulli beam.

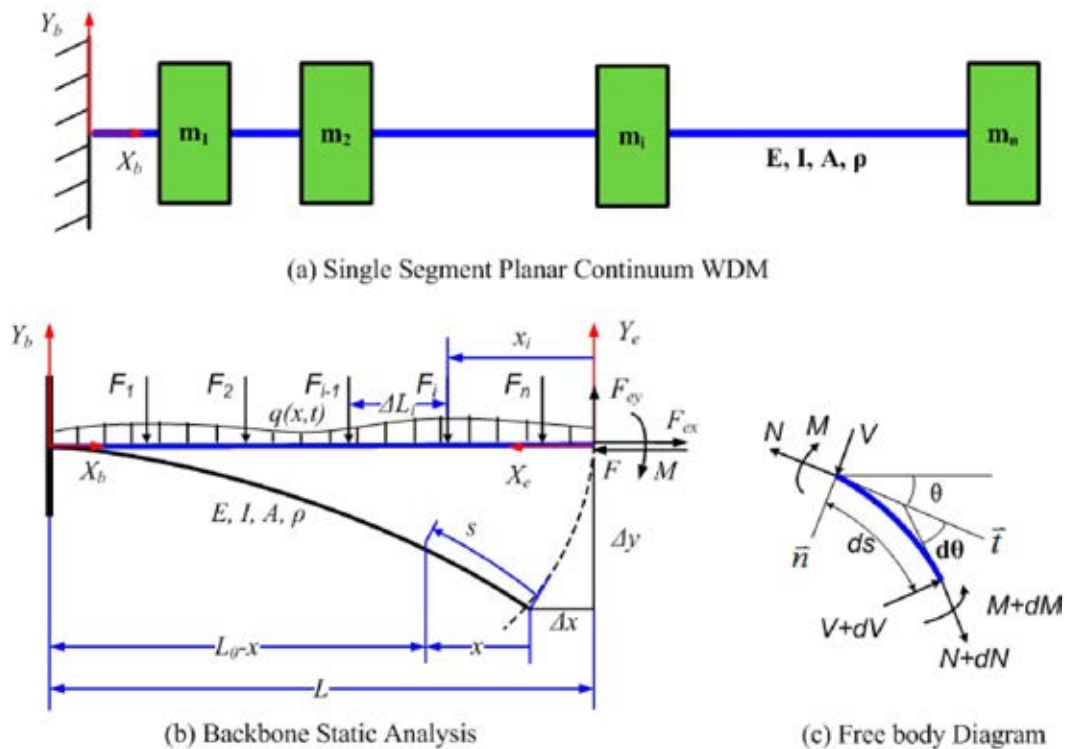


Figure 4-4 SPCP WDM Static Analysis

Figure 4-4 (a) shows the backbone subjected to general loading conditions. The structure parameters of the continuum backbone are: Young's modulus E , second axial moment of area I , cross-section area $A(x)$, length L and density ρ . The vertebrae or the spacing discs are the added mass m_i ; their distributions along the backbone are as shown in the figure. Figure 4-4 (b) shows the statics analysis of the SPCP WDM, and Figure 4-4 (c) shows the free body diagram of the backbone segment. Similarly, forces acting on the backbone are divided into external load and controlling forces. External loads include the distributed force $q(x,t)$, the gravity of the vertebrae F_i and the concentrated forces F_{ex} , F_{ey} at the distal end. Controlling forces are the concentrated force F and bending moment M at the distal tip. Two coordinate frames are set. One is at the backbone base OX_bY_b , and the other is at the distal end OX_eY_e .

The deflection of the WDM backbone is large. In the analysis, the distal end horizontal displacement Δx cannot be neglected. Assume after deflection the

distal end in the base frame is $(L-\Delta x, -\Delta y)$. In frame OX_eY_e , the distal end position is $(0, 0)$. In the following analysis, the frame OX_eY_e is used. From the Euler-Bernoulli beam theory, the deflected backbone is described as:

$$\frac{y''}{[1+(y')^2]^{3/2}} = -\frac{M(x)}{EI} = \lambda(x) \quad (4-12)$$

In the equation, the bending moment $M(x)$ is assumed to vary in an arbitrary manner. Young's modulus E and second axial moment of area I are assumed to be constant. In the equation, y' and y'' are the first and second derivative to x . To find the deformed backbone curve, we need to find $\lambda(x)$ and integrate the nonlinear differential equation.

Let $y' = p$, $y'' = p'$, then we have:

$$\frac{dp/dx}{[1+p^2]^{3/2}} = \lambda(x) \quad (4-13)$$

By multiplying both sides by dx and integrating once, we have:

$$\int \frac{dp}{[1+p^2]^{3/2}} = \int \lambda(x) dx \quad (4-14)$$

Furthermore, let $p = \tan \theta$. Then we have:

$$dp = \sec^2 \theta d\theta \quad (4-15)$$

$$\cos \theta = \frac{1}{[1+p^2]^{1/2}} \quad (4-16)$$

$$\sin \theta = \frac{p}{[1+p^2]^{1/2}} \quad (4-17)$$

By substituting Equations (4-15), (4-16) and (4-17) into Equation (4-14), we have:

$$\int \cos \theta d\theta = \int \lambda(x) dx \quad (4-18)$$

Let $\varphi(x)$ be the integration of $\lambda(x)$. By integrating Equation (4-18) once, together with Equation (4-17) we have:

$$\frac{y'}{\left[1+(y')^2\right]^{1/2}} = \varphi(x) + C \quad (4-19)$$

In the equation, C is a constant. It can be determined from the boundary condition. In the WDM, the backbone is assumed to be a cantilever beam. As a result, at the base or at $L - \Delta x$ the slope $y' = 0$, or $C = -\varphi(L - \Delta x)$.

Let $G(x) = \varphi(x) + C$, then the slope along the backbone is:

$$y'(x) = \frac{G(x)}{\sqrt{1-G(x)^2}} \quad (4-20)$$

The vertical displacement of the backbone at x is:

$$y(x) = \int_0^x \frac{G(\eta)}{\sqrt{1-G(\eta)^2}} d\eta \quad (4-21)$$

Hence, as long as $G(x)$ is known, both y' and y can be solved thereafter. In the Elastica theory, it is assumed that the beam length after deflection does not change. This yields:

$$L = \int_0^{L_0} \left[1+(y')^2\right]^{1/2} dx \quad (4-22)$$

In the equation, $L_0 = L - \Delta x$ and Δx is the backbone tip horizontal displacement. In the kinematics analysis, it is assumed the deformed backbone is a circular arc. Here, we first assume that the horizontal translation is the same as that of circular deformation as shown in Equation (4-23), where Θ is the backbone tip orientation. From the initial estimate, we can find the deformed backbone length \tilde{L} from Equation (4-22) accordingly. It is certain that error exists. However, we can adjust $\tilde{\Delta x}$ according to \tilde{L} and repeat the process again. By several iterations, Δx can be obtained with given precision.

$$\tilde{\Delta x} = L \left(1 - \frac{\sin(\Theta)}{\Theta}\right) \quad (4-23)$$

In the solution, another issue is to determine $\lambda(x)$. It is dependent on the load condition. Without losing generality, the backbone is assumed to be subject to all kinds of loads, as shown in Figure 4-4 (b). The tip position in the base frame is $(L_0, -\Delta y)$, and the orientation is Θ . The moment is:

$$M(x) = -M + (F \cdot \sin \Theta + F_{ey}) \cdot x - (F \cdot \cos \Theta - F_{ex}) \cdot y - \int_0^x f(\eta) \cdot \eta \cdot d\eta \quad (4-24)$$

In the equation, the distributed load is $f(x) = \rho A(x) + q(x, t) + F_i \cdot \delta(x - x_i)$, $\rho A(x)$ is the mass distribution along the backbone from the tip to the base, $q(x, t)$ is the distributed load, F_i is the concentrated force, x_i is the location of F_i and $\delta(x)$ is the Dirac Delta Function. As a result, we have:

$$G(x) = \int \lambda(x) dx = \frac{1}{EI} \left\{ Mx - \frac{1}{2} x^2 \left((F \sin \Theta + F_{ey}) - \left(F \cos \Theta \tan \frac{\Theta}{2} - F_{ex} \right) \right) + \int \int_0^x f(\eta) d\eta dx \right\} + C \quad (4-25)$$

By applying the boundary condition: $G(L - \Delta x) = 0$, we have:

$$C = -\varphi(L - \Delta x) = -\frac{1}{EI} \left\{ M(L - \Delta x) - \frac{1}{2} (L - \Delta x)^2 \left((F \sin \Theta + F_{ey}) - \left(F \cos \Theta \tan \frac{\Theta}{2} - F_{ex} \right) \right) + \int \int_0^{L - \Delta x} f(\eta) d\eta dx \right\} \quad (4-26)$$

The deformed backbone curve can be solved in the following steps:

- First assign an initial value to Δx . Then the length of the deflected beam is: $\tilde{L} = \int_0^{L - \Delta x} [1 - G(x)^2]^{-1/2} dx$. As the backbone is inextensible, it should be the same as the initial length, i.e. $L = \tilde{L}$. The horizontal displacement Δx can be solved by iteration with given accuracy.
- After Δx is solved, the slope along the beam $y'(x)$ can be solved.
- By integration, the deformed shape as well as the distal end position and orientation can be solved.

During the integration, Simpson's one-third rule can be used. It states that:

$$\int_a^b f(x)dx = \frac{\xi}{3}(y_0 + 4y_1 + 2y_2 + 4y_3 + \dots + 2y_{n-2} + 4y_{n-1} + y_n),$$

where a and b are the integration limits, n is an even number and $\xi = (b-a)/n$.

The following shows some simulation examples. In the simulation, the continuum backbone length is $L=0.5$ m, the Young's modulus is $E=1.5$ Gpa, the backbone thickness is 1 mm, and the width is 40 mm. The WDM is manipulated in the horizontal plane, and as a result, in the simulation, gravity is ignored.

In the first simulation, a pure moment $M=0.01$ Nm is applied at the distal end. The deformed backbone curve is predicted using the nonlinear Euler-Bernoulli model as shown by the red solid curve of Figure 4-5.

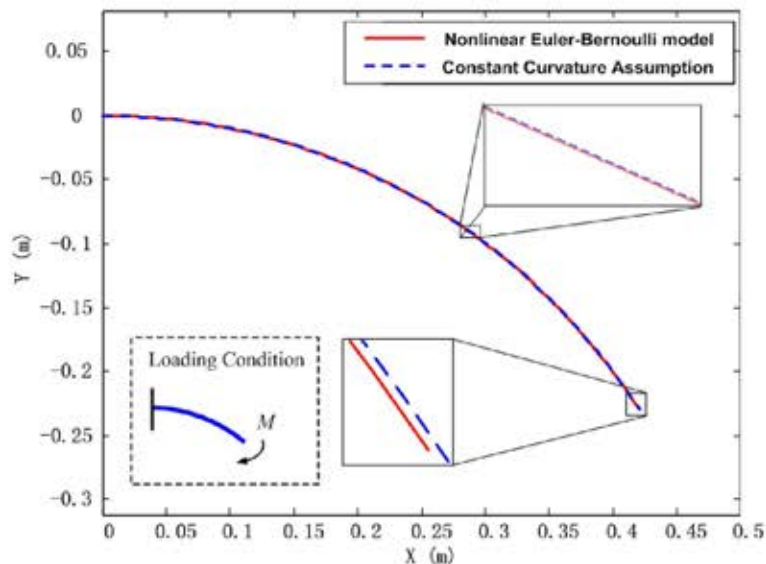


Figure 4-5 Validity of Constant Curvature Assumption at Pure Moment Loading Condition

In the figure, the red solid curve is the predicted backbone using the derived nonlinear Euler-Bernoulli model. The backbone distal end position in the base frame is $x=0.422$ m, and $y=-0.2319$ m. The backbone distal end rotation angle is $\theta=57.6^\circ$. In the constant curvature model, assume the backbone has the same bending angle. The deflected backbone is as shown by the blue dashed curve. From Figure 4-5, the two curves are very close. In the constant curvature model, the backbone tip position is $x=0.420$ m, and $y=-0.2308$ m. Compared with the nonlinear Euler-Bernoulli model, the relative errors are 0.4739% and 0.4734%,

respectively. The error is trivial and can be ignored. The result is consistent with that in the SPSP WDM analysis. It also supports the constant curvature assumption in the kinematic analysis.

In the second simulation, three loading conditions are simulated. In the first case, a pure moment $M=0.01$ Nm is applied at the distal end. In the second case, an additional force $F=0.05$ N is applied at the distal end. The direction of the force is tangent to the backbone, pointing to the base, as shown in Figure 4-6. In the third case the magnitude of the force is increased to 0.1 N. The simulation results are shown in Figure 4-6. In the first case, the backbone distal end coordinate is: $x=0.422$ m, $y=-0.2319$ m and the rotation angle is $\Theta=57.57^\circ$. In the second case, the backbone distal end coordinate is: $x=0.4736$ m, $y=-0.1340$ m and the rotation angle is $\Theta=38.34^\circ$. In the third case, the backbone distal end coordinate is: $x=0.4861$ m, $y=-0.0811$ m and the rotation angle is $\Theta=28.69^\circ$. From the results, it is clear that with additional axial force, the deformed backbone is not a circular arc any more. With the force pointing to the base, the resultant moment is reduced. As it can be seen from the figure, although the incrementals are both 0.05 N, the rate of shape changing is very different. Due to the large deformation, the system is nonlinear, which means superposition is invalid. The backbone deformation needs be solved case by case.

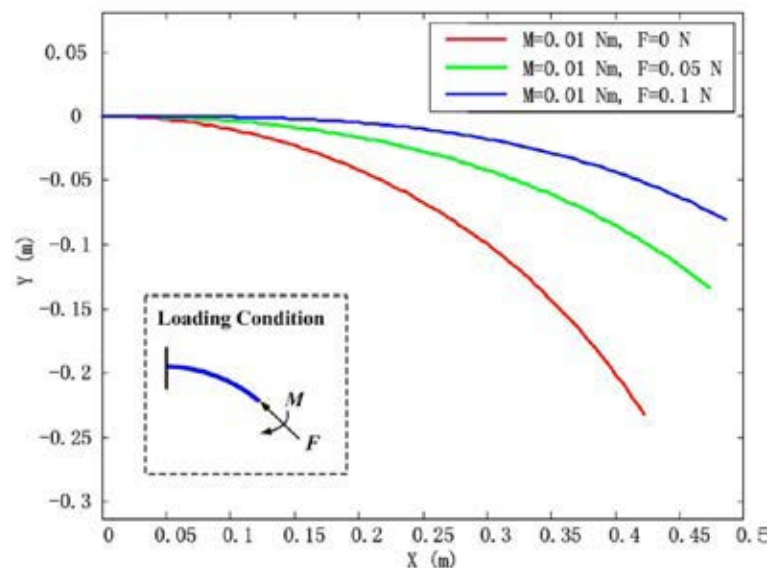


Figure 4-6 SPCP WDM Backbone Deformation under Different Loading Conditions

For more complicated loading conditions, the deformed backbone curve can also be solved using the above method. Also, for the multi-segment WDM, the static model is similar to that of the single segment WDM. The deformed curve can be viewed as a single segment WDM under different loading conditions.

4.2 Dynamic Model of the Wire-Driven Mechanism

In this section, dynamic models of the SPSP WDM and SPCP WDM are developed. The SPSP WDM dynamic model is developed using the Euler-Lagrange method. Theoretical basis of the SPCP WDM dynamic model is Hamilton's principle and Euler-Bernoulli beam theory. The dynamic models of other WDMs are similar. The main difference is the boundary condition.

4.2.1 Dynamic Model of SPSP WDM

The backbone of the SPSP WDM is modeled as an N -link structure with revolute joints and torsional springs in between, as shown in Figure 4-1 (b). The Euler-Lagrange method is used to develop the dynamic model.

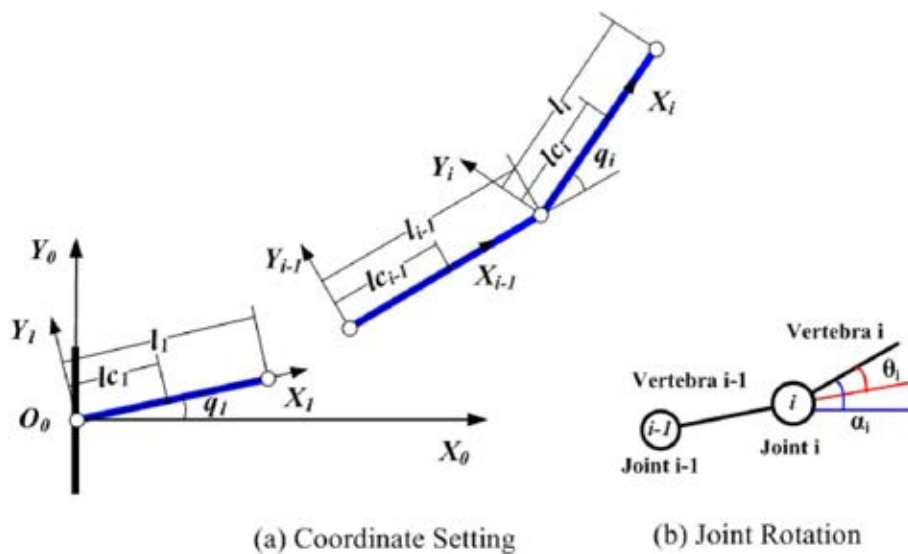


Figure 4-7 Dynamic Modeling of Serpentine WDM

Figure 4-7 (a) shows the coordinate setting. The generalized coordinate is set as the joint rotation $q_i = \theta_i$. From Figure 4-7 (a), the inertial frame $O_0 X_0 Y_0$ is set at the base. For each vertebra, a local coordinate frame $O_i X_i Y_i$ is set at the joint. The X axis is along the vertebra axis and Z axis is perpendicular to the plane of motion. For each vertebra, the length is l_i , the mass center is at l_{c_i} , the mass is m_i and the

moment of inertia is I_i . The rotation in the inertial frame is α_i , and in the local frame is θ_i , which is also chosen as the generalized coordinate.

To develop the dynamics model, the kinetic energy and potential energy of the system need to be found first. For each vertebra, the kinetic energy has two parts: one relates to the translation and the other relates to the rotation energy. As the WDM is in the horizontal plane, gravitation is ignored. The potential energy is from the elastic beam.

For each vertebra, the kinetic co-energy is:

$$K_i = \frac{1}{2} m_i v_{ci}^T v_{ci} + \frac{1}{2} \omega_i^T I_i \omega_i \quad (4-27)$$

where m_i is the total mass of the vertebra, v_{ci} is the linear velocity vector, ω_i is the angular velocity vector, and I_i is the inertia tensor expressed in the inertial frame.

We can also express the inertia tensor in the local frame as:

$$I_i = R_i \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} R_i^T \quad (4-28)$$

The rotation matrix between the local frame and inertial frame is:

$$R_i = \begin{bmatrix} c_{\alpha_i} & -s_{\alpha_i} & 0 \\ s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4-29)$$

where, $\alpha_j = \sum_{k=1}^j q_k = \sum_{k=1}^j \theta_k$, $s_{\alpha_j} = \sin(\alpha_j)$, and $c_{\alpha_j} = \cos(\alpha_j)$.

The linear velocity of the vertebra mass center is:

$$\begin{pmatrix} v_{cx_i} \\ v_{cy_i} \end{pmatrix} = J v_i \cdot \dot{q} \quad (4-30)$$

The joint angular velocity is:

$$\omega_i = J_{\omega_i} \cdot \dot{q} \quad (4-31)$$

where, J_{v_i} and J_{ω_i} are the geometric Jacobians, and $\dot{q} = [\dot{\theta}_1 \quad \dot{\theta}_2 \quad \dots \quad \dot{\theta}_n]$. For the planar serpentine WDM, their representations are:

$$J_{v_{ci}} = \begin{bmatrix} -\sum_{j=1}^{i-1} l_j s_{\alpha_j} - l_i s_{\alpha_i} & \dots & -\sum_{j=k}^{i-1} l_j s_{\alpha_j} - l_i s_{\alpha_i} & \dots & -l_i s_{\alpha_i} & 0 & \dots \\ \sum_{j=1}^{i-1} l_j c_{\alpha_j} + l_i c_{\alpha_i} & \dots & \sum_{j=k}^{i-1} l_j c_{\alpha_j} + l_i c_{\alpha_i} & \dots & l_i c_{\alpha_i} & 0 & \dots \\ 0 & \dots & 0 & \dots & 0 & 0 & \dots \end{bmatrix}_{3 \times n} \quad (4-32)$$

$$J_{\omega_i} = \begin{bmatrix} 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 \\ \underbrace{1 \quad \dots \quad 1}_i & 0 & \dots & 0 \end{bmatrix}_{3 \times n} \quad (4-33)$$

Especially, when $i=1$ and $i=n$ the Jacobian is:

$$J_{v_{c1}} = \begin{bmatrix} -l_1 s_{\alpha_1} & 0 & \dots & 0 \\ l_1 c_{\alpha_1} & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \end{bmatrix}_{3 \times n} \quad (4-34)$$

$$J_{v_{cn}} = \begin{bmatrix} -\sum_{j=1}^{n-1} l_j s_{\alpha_j} - l_n s_{\alpha_n} & \dots & -\sum_{j=k}^{i-1} l_j s_{\alpha_j} - l_i s_{\alpha_i} & \dots & -l_n s_{\alpha_n} \\ \sum_{j=1}^{n-1} l_j c_{\alpha_j} + l_n c_{\alpha_n} & \dots & \sum_{j=k}^{i-1} l_j c_{\alpha_j} + l_i c_{\alpha_i} & \dots & l_n c_{\alpha_n} \\ 0 & \dots & 0 & \dots & 0 \end{bmatrix}_{3 \times n} \quad (4-35)$$

$$J_{\omega_1} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \end{bmatrix}_{3 \times n} \quad (4-36)$$

$$J_{\omega_n} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 1 & 1 & \dots & 1 \end{bmatrix}_{3 \times n} \quad (4-37)$$

Then, the total kinetic energy of the SPSP WDM is represented as:

$$K(q, \dot{q}) = \frac{1}{2} \dot{q}^T D(q) \dot{q} \quad (4-38)$$

where $D(q)$ is the inertia matrix:

$$D(q) = \left[\sum_{i=1}^n \left\{ m_i J_{vci}(q)^T J_{vci}(q) + J_{\omega i}(q)^T R_i(q) I_i R_i(q)^T J_{\omega i}(q) \right\} \right]_{n \times n} \quad (4-39)$$

For the diagonal term of $D(q)$, we have:

$$d_{ii} = \sum_{k=1}^n m_k [J_{vck}(:,i) \cdot J_{vck}(:,i)] + \sum_{k=1}^{n-i+1} I_k \quad (4-40)$$

For $j > i$, we have:

$$d_{ij} = d_{ji} = \sum_{k=1}^n m_k [J_{vck}(:,i) \cdot J_{vck}(:,j)] + \sum_{k=1}^{n-j+1} I_k \quad (4-41)$$

The potential energy is:

$$P(q) = \frac{1}{2} \frac{EI_z}{h_0} \sum_{i=1}^n q_i^2 \quad (4-42)$$

The Lagrangian of the Serpentine WDM is:

$$L = K(q, \dot{q}) - P(q) \quad (4-43)$$

The partial derivatives of the Lagrangian with respect to q_k and \dot{q}_k are:

$$\frac{\partial L}{\partial q_k} = \frac{1}{2} \sum_{i,j} \frac{\partial d_{ij}}{\partial q_k} \dot{q}_i \dot{q}_j - \frac{\partial P}{\partial q_k} \quad (4-44)$$

$$\frac{\partial L}{\partial \dot{q}_k} = \sum_j d_{kj} \dot{q}_j \quad (4-45)$$

and

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} = \sum_j d_{kj} \ddot{q}_j + \sum_{i,j} \frac{\partial d_{kj}}{\partial q_i} \dot{q}_i \dot{q}_j \quad (4-46)$$

Referring to Figure 4-1 (b), assume the virtual displacement of each joint is δq_k .

The virtual work done by the external loading and controlling forces are:

$$\delta W = M_e \sum_{i=1}^n \delta q_i + (F_{ex} - T_1 - T_2) \delta x + F_{ex} \delta y \quad (4-47)$$

with

$$\delta x = \sum_{i=1}^n l_i \cdot \cos\left(\sum_{j=1}^i \delta q_j\right) \quad (4-48)$$

$$\delta y = \sum_{i=1}^n l_i \cdot \sin\left(\sum_{j=1}^i \delta q_j\right) \quad (4-49)$$

Hence, the generalized forces are:

$$Q_k = M_e + (F_{ex} - T_1 - T_2) \sum_{i=k}^n l_i \sin(\alpha_k) + F_{ey} \sum_{i=k}^n l_i \cos(\alpha_k) \quad k = 1, 2, \dots, n \quad (4-50)$$

Let $p_k(q) = \frac{\partial P}{\partial \dot{q}_k} = \frac{EI_z}{h_0} q_k$. Then, the Euler-Lagrange equation is:

$$\sum_{j=1}^n d_{kj}(q) \ddot{q}_j + \sum_{i=1}^n \sum_{j=1}^n c_{ijk}(q) \dot{q}_i \dot{q}_j + p_k(q) = Q_k \quad k = 1, 2, \dots, n \quad (4-51)$$

where c_{ijk} is the Christoffel symbol:

$$c_{ijk} := \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \quad (4-52)$$

4.2.2 Dynamic Model of SPCP WDM

The SPCP WDM backbone can be treated as a cantilevered Euler-Bernoulli beam as shown in Figure 4-4 (a). The dynamic model under controlling force is derived from the Euler-Bernoulli Beam theory and the extended Hamilton's principle:

$$\int_{t_1}^{t_2} (\delta K - \delta V + \delta W_{NC}) dt = 0 \quad (4-53)$$

where δK is the variation of kinetic energy, δV is the variation of potential energy, and δW_{NC} is the virtual work done by non-conservative forces.

Potential energy of the SPCP WDM is:

$$V = \frac{1}{2} \int_0^L EI \left(\frac{\partial^2 y}{\partial x^2} \right)^2 dx \quad (4-54)$$

Hence,

$$\begin{aligned}
-\int_{t_1}^{t_2} \delta V dt = & -\int_{t_1}^{t_2} EI \left(\frac{\partial^2 y}{\partial x^2} \right) \delta \left(\frac{\partial y}{\partial x} \right) \Big|_0^L dt + \int_{t_1}^{t_2} EI \left(\frac{\partial^3 y}{\partial x^3} \right) \delta y \Big|_0^L dt \\
& - \int_{t_1}^{t_2} \int_0^L EI \left(\frac{\partial^4 y}{\partial x^4} \right) \delta y dx dt
\end{aligned} \quad (4-55)$$

Compared with the backbone, the dimensions of spacing discs are generally small. Hence, in the analysis, the rotational kinetic energy of the spacing discs is neglected. As a result, kinetic energy of the SPCP WDM is:

$$K = \frac{1}{2} \int_0^L \left(\rho A + \sum_{i=1}^n m_i \delta(x - x_i) \right) \left(\frac{\partial y}{\partial t} \right)^2 dx \quad (4-56)$$

Hence,

$$\int_{t_1}^{t_2} \delta K dt = - \int_{t_1}^{t_2} \int_0^L \left(\rho A + \sum_{i=1}^n m_i \delta(x - x_i) \right) \left(\frac{\partial^2 y}{\partial t^2} \right) \delta y dx dt \quad (4-57)$$

Virtual work done by the controlling force is:

$$\delta W_{NC} = \int_0^L f(x, t) \delta y dx + \left[M \cdot \delta \left(\frac{\partial y}{\partial x} \right) - F_{\varphi} \cdot \delta y \right]_{x=L} \quad (4-58)$$

Hence:

$$\int_{t_1}^{t_2} \delta W_{NC} dt = \int_{t_1}^{t_2} \int_0^L f(x, t) \delta y dx dt + \int_{t_1}^{t_2} \left[M \cdot \delta \left(\frac{\partial y}{\partial x} \right) - F_{\varphi} \cdot \delta y \right]_{x=L} dt \quad (4-59)$$

Applying the extended Hamilton's principle, we have:

$$\begin{aligned}
\int_{t_1}^{t_2} \left\{ \int_0^L \left[- \left(\rho A + \sum_{i=1}^n m_i \delta(x - x_i) \right) \left(\frac{\partial^2 y}{\partial t^2} \right) - EI \left(\frac{\partial^4 y}{\partial x^4} \right) + f(x, t) \right] \delta y dx \right. \\
\left. - EI \left(\frac{\partial^2 y}{\partial x^2} \right) \delta \left(\frac{\partial y}{\partial x} \right) \Big|_0^L + EI \left(\frac{\partial^3 y}{\partial x^3} \right) \delta y \Big|_0^L + M \cdot \delta \left(\frac{\partial y}{\partial x} \right) \Big|_{x=L} - F_{\varphi} \delta y \Big|_{x=L} \right\} dt = 0
\end{aligned} \quad (4-60)$$

Hence, for the continuum WDM, the equation of motion is:

$$EI \left(\frac{\partial^4 y}{\partial x^4} \right) + \left(\rho A + \sum_{i=1}^n m_i \delta(x - x_i) \right) \left(\frac{\partial^2 y}{\partial t^2} \right) = f(x, t) \quad (4-61)$$

The boundary condition is:

BC I: at $x=0$

$$y(x,t) = 0 \text{ and } \frac{\partial y(x,t)}{\partial x} = 0 \quad (4-62)$$

BC II: at $x=L$

$$EI \left(\frac{\partial^3 y}{\partial x^3} \right) - F_{ey} = 0 \text{ and } EI \left(\frac{\partial^2 y}{\partial x^2} \right) \delta \left(\frac{\partial y}{\partial x} \right) - M = 0 \quad (4-63)$$

This is a partial differential equation. One can use the Galenkin's Finite Element method to discretize it to a group of ordinary differential equations, and solve for the mode frequencies and corresponding mode shapes. Or, Rayleigh's method can be used to solve for the mode frequencies.

4.3 Summary

In this chapter, the static models of SPSP WDM and SPCP WDM are studied. From the results, when the WDM is subjected to pure moment, the deformed shape is a circular arc. When there are other forces, such as gravity, the constant curvature assumption does not stand anymore. Dynamics models are also derived for the two WDMs. For SPSP WDM, the dynamics is studied using the Euler-Lagrange method, considering the backbone as an N -link structure. For SPCP WDM, the dynamics is studied with reference to Hamilton's principle and the Euler-Bernoulli beam. For other WDMs, static model and dynamic model are similar, the main difference being the boundary conditions.

Chapter 5 Application I - Wire-Driven

Robot Fish

In this chapter, based on the WDM, the designs of several underactuated wire-driven flapping propulsors are described. Four wire-driven robot fishes were built and experimented with. Compared with traditional robot fish designs, the wire-driven robot fish requires less actuators and can better resemble fish movement.

5.1 Fish Swimming Introduction

Fish are the species selected by nature to dominate water. The major reason is that fish can move effectively and efficiently in water. For a long time, fish's excellent performance has attracted researchers' attention greatly. It is known that the propulsion efficiency of fish can exceed 90% [129], the cruising speed of a sailfish can exceed 110 km/h [130], and the recorded startup acceleration of pike is higher than 249 m/s^2 [91]. Also, fish can turn sharply without decelerating. Furthermore, fish swim silently. In this section, the fish swimming categories, propulsion model, body curve model, and swimming data are reviewed.

5.1.1 Fish Swimming Categories

Fish swim in water by actively deforming their body and/or fins. Through the interaction with water fish gain thrust. On a fish's body, there are five types of fins, i.e. caudal fin, anal fin, dorsal fin, pectoral fin, and pelvic fin. Although the caudal fin is indisputably the most common fin for propulsion, there are still a lot of species which move around using other fins. Fish swimming is divided into two major categories: the body and/or caudal fin (BCF) propulsion and median and/or paired fin (MPF) propulsion, as shown in Figure 5-1 [131]. Around 85% of aquatic animals swim using BCF propulsion. BCF propulsion is further divided into several subcategories depending on the portion of the waving part. These are anguilliform, subcarangiform, carangiform, thunniform, etc. The waving part decreases from anguilliform to thunniform. In the anguilliform subcategory almost the whole body deforms into a traveling wave; in the thunniform subcategory only the part near the caudal fin oscillates. Thunniform and carangiform subcategories are adopted by most fast swimmers, such as tuna, pike,

sailfish, etc. For MPF propulsion, the waving form is even more diverse. Though, for a single fin, the waving motion can be roughly categorized as undulatory flapping and oscillatory flapping.

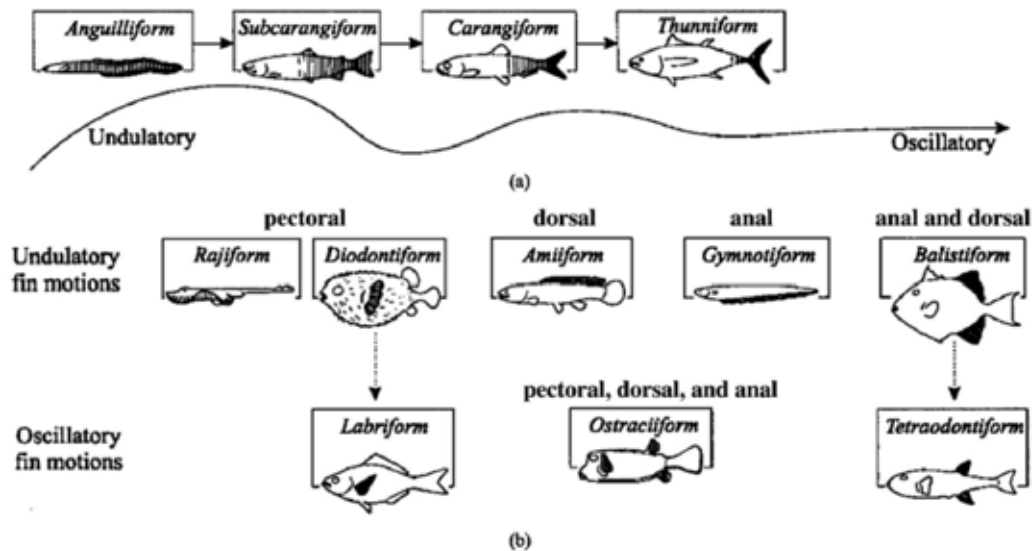


Figure 5-1 Fish Swimming Categories (a) BCF Propulsion and (b) MPF Propulsion[131]

From the above description, how fish swim can be roughly categorized as BCF and MPF. In each category, the fin waving motion is divided into oscillatory flapping and undulatory flapping. The categorization is as shown in Figure 5-2.

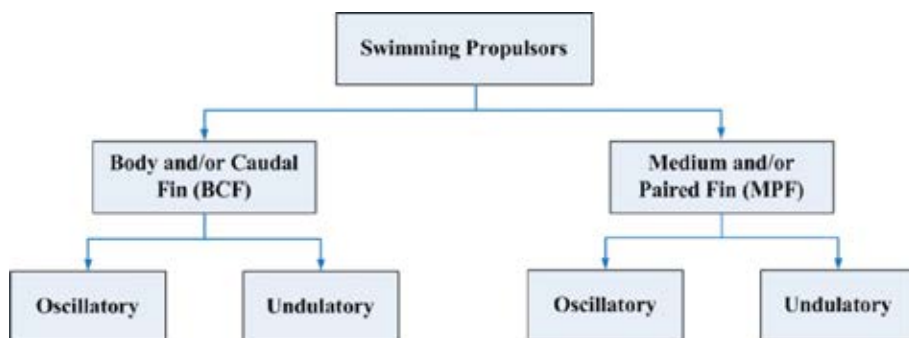


Figure 5-2 Simplified Fish Swimming Categorization

5.1.2 Body Curve Function

In the late 1960s, Sir James Lighthill proposed that the body curve of a swimming fish is a traveling wave [132]. The body curve model is composed of two parts: the first part represents the wave magnitude and the second part is the wave shape.

Among all the representations, Equation (5-1) shows a frequently used one. In the equation, c_1 and c_2 are the coefficients of the linear and quadratic term in the wave amplitude, k is the body wave number and ω is the body wave frequency [133, 134]. The body wave number k increases from oscillatory swimming to undulatory swimming.

$$y(x,t) = [c_1x + c_2x^2][\sin(kx + \omega t)] \quad (5-1)$$

Figure 5-3 shows an example of the oscillatory form swimming curve. In this example, coefficients are chosen as $c_1 = 0.1$, $c_2 = 0.2$, $k=0.5$, and $\omega = \pi$. In the figure, the fish moves in the $+X$ direction and the swimming speed is one body length (BL) per flapping cycle. Five curves are shown in the figure, representing the body curve at five instances in the flapping cycle. From the figure, it is seen that during the flapping cycle, the whole fish body is on the same side. It does not cross the centerline.

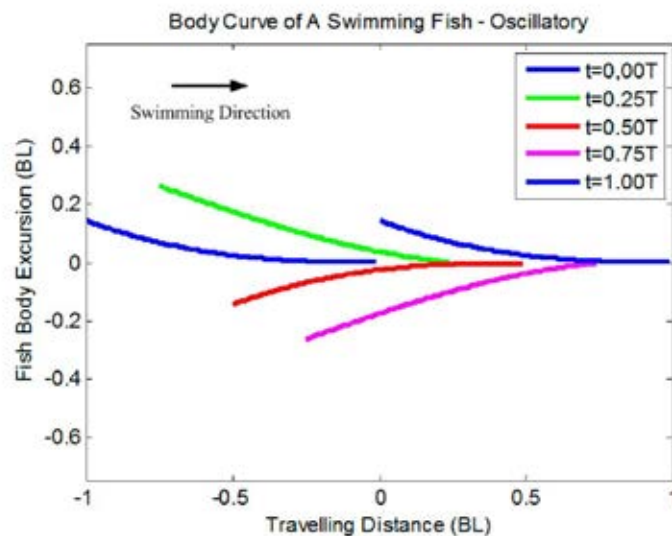


Figure 5-3 Fish Swimming Body Curve - Oscillatory

For robot fishes, the backbone curve fits the fish body curve. Among current well known robot fishes, the links in the backbone are mainly rigid. In oscillatory form swimming robot fish, the backbone typically has one or two sections. In rigid backbones, the fish body is fitted by a straight line or a polyline. In wire-driven flexible backbones, the fish body is fitted by a circular arc. Figure 5-4 shows the comparison. Figure 5-4 (a) shows a straight line and a circular arc fit the fish

body curve, respectively. From the comparison it is apparent that when the number of segments is the same, the circular arc outperforms in fitting fish body curve. Figure 5-4 (b) shows a two-segment polyline and a circular arc fit the fish body curve respectively. Compared with the single straight line version, the two-segment polyline is better. However, it is still inferior to the circular arc-shaped backbone. Hence, we can say that the wire-driven flexible backbone outperforms the rigid backbone in resembling the oscillatory form of fish swimming.

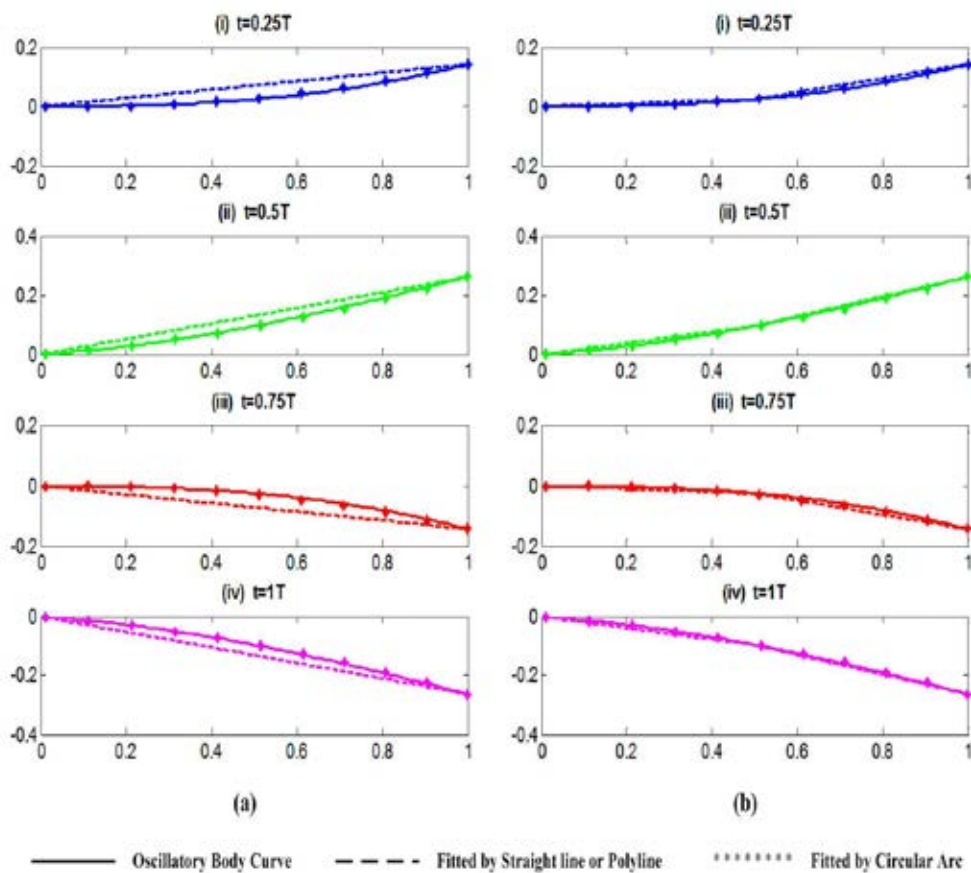


Figure 5-4 Oscillatory Body Curve Comparison: (a) One Straight Line and One Circular Arc Fitting; (b) Two Straight Lines and One Circular Arc Fitting

Figure 5-5 shows an example of fish swimming in the undulatory form. In this example, coefficients are chosen as $c_1 = 0.02$, $c_2 = 0.0835$, $k = 6.0$ and $\omega = 3.86$. These parameters are adopted by MIT's robot tuna [135]. In the figure, the fish moves in the $+X$ direction. The five curves show the fish body curve at $t=0$, $t=0.25T$, $t=0.5T$, and $t=T$, where T is the flapping period. Compared with the oscillatory form of swimming, the fish body is more like a sine wave with the

amplitude increasing from the head to the tail. During the flapping cycle, the body curve crosses the centerline in the form of undulatory flapping.

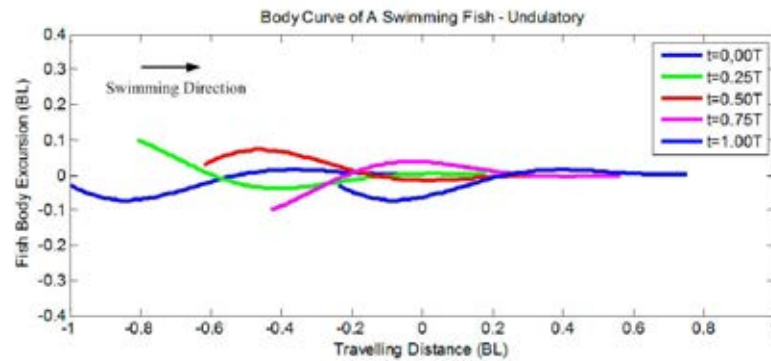


Figure 5-5 Fish Swimming Body Curve – Undulatory

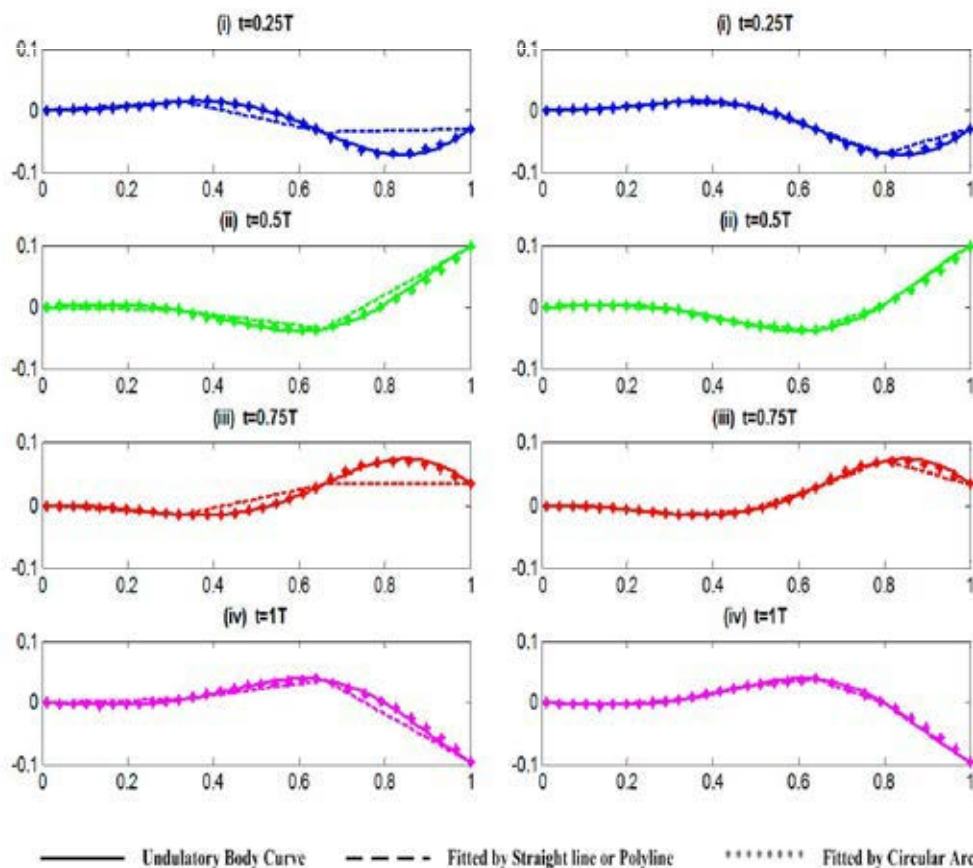


Figure 5-6 Undulatory Body Curve Comparison: (a) Three Straight Lines and Three Circular Arcs Fitting; (b) Six Straight Lines and Three Circular Arcs Fitting

To imitate the undulatory form of flapping, more sections are needed in the robot fish backbone. Typically, in rigid backbone design, three to six links are used. Figure 5-6 shows the undulatory fish body curve fitting comparison. The undulatory fish body curve is fitted by a multi-segment rigid backbone and a multi-segment wire-driven flexible backbone respectively. In the figure, the straight line represents a rigid link and the circular arc represents a wire-driven flexible section. Figure 5-6 (a) shows the body curve is fitted by three straight lines and three circular arcs; Figure 5-6 (b) shows the body curve is fitted by six straight lines and three circular arcs. From the figures, it is shown that three circular arcs can better fit the undulatory fish swimming body than three straight lines. Also, they outperform six straight lines. Therefore, a multi-segment wire-driven flexible backbone outperforms a multi-segment rigid backbone with regard to resembling the undulatory form of fish swimming.

Figure 5-7 shows the use of two circular arcs to fit the above undulatory fish swimming body curve. The fittings at four time instances are shown. From the figure, it is seen that two circular arcs can fit the undulatory fish swimming body curve well, i.e. a two-segment wire-driven flexible backbone can be used to develop the undulatory robot fish.

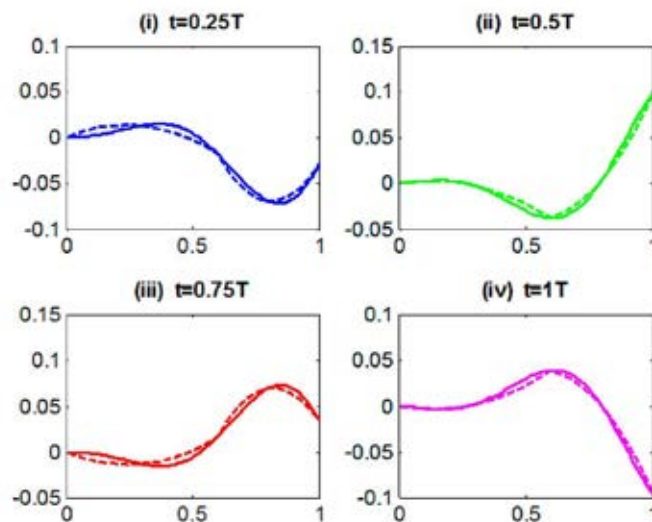


Figure 5-7 Two Circular Arcs Fit the Undulatory Fish Swimming Body Curve

5.1.3 Fish Swimming Hydrodynamics

In fish swimming the flow is unsteady and its precise modeling is complicated. In robot fish development, the three most frequently used approaches are Elongated Body Theory (EBT) [129, 132, 136, 137], 3D Waving Plate Theory (3DWPT) [138] and Computational Fluid Dynamics (CFD) [139-141]. Among all the three approaches, the EBT is simple, computationally efficient and can predict fish swimming reasonably well. As a result, in this research, we used the EBT.

EBT was first proposed by Sir James Lighthill in the late 1960s and early 1970s [136, 137]. This theory assumes that: 1) water momentum near a fish section is in a direction perpendicular to the backbone. It has a magnitude equal to the virtual mass m per unit length times the perpendicular component (w) of fish velocity in that direction, as shown in Figure 5-8; 2) Thrust can be obtained by considering the rate of momentum change within a volume enclosing the fish whose boundary at each instant includes a flat surface PI perpendicular to the caudal fin through its posterior end; 3) In the momentum balance it is necessary to take into account transfer of momentum across PI not only by convection but also by the action of the resultant $mw^2/2$ of the pressures generated by the motions within the plane PI .

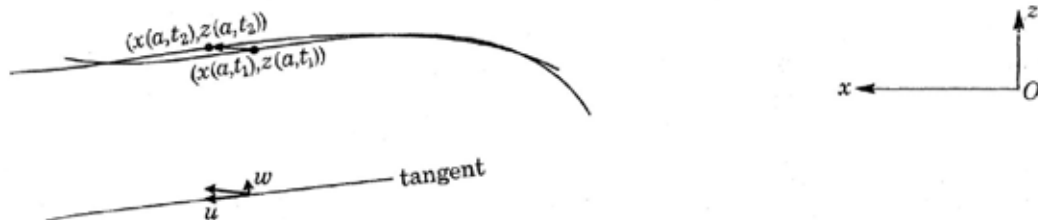


Figure 5-8 Spinal Column Configuration at Two Successive Instants [136]

In the EBT, the instantaneous lift per unit length of the fish is:

$$L(x, t) = -\rho \left(\frac{\partial}{\partial t} + U \frac{\partial}{\partial x} \right) \{ V(x, t) A(x) \} \quad (5-2)$$

Here, ρ is the density of the flow; U is the passing flow velocity or the fish forward velocity; V is the relative speed of the fish body section to the flow; $A(x)$ is the cross-sectional area of the fish body; x is the position along the fish body, from 0 to L . The virtual mass is $\rho A(x)$.

The rate of work done by the fish is:

$$\begin{aligned} W &= -\int_0^L \frac{\partial y}{\partial x} L(x,t) dx \\ &= \frac{\partial}{\partial t} \left\{ \rho \int_0^L \frac{\partial y}{\partial t} VA(x) dx - 0.5\rho \int_0^L V^2 A(x) dx \right\} + \rho U \left[\frac{\partial y}{\partial t} VA(x) \right]_0^L \end{aligned} \quad (5-3)$$

The mean over a long time of the work done by the fish is:

$$\bar{W} = \rho UA(L) \left\{ \overline{\frac{\partial y}{\partial t} \left(\frac{\partial y}{\partial t} + U \frac{\partial y}{\partial x} \right)} \right\}_{x=L} \quad (5-4)$$

Here, y is the transverse displacement of the fish body; $\partial y/\partial t$ is the traversing velocity, and $\partial y/\partial x$ is the slope of the fish body.

The energy for thrust is found by subtracting the energy wasted in generating the wake $(0.5\rho\bar{V}^2 A)U$ from the total energy. As a result, the mean thrust is:

$$\bar{T} = \left[\frac{1}{2} \rho A(L) \left(\overline{\left(\frac{\partial y(x,t)}{\partial t} \right)^2} - U^2 \overline{\left(\frac{\partial y(x,t)}{\partial x} \right)^2} \right) \right]_{x=L} \quad (5-5)$$

Drag force of a swimming fish is:

$$F_D = \frac{1}{2} C_D \rho U^2 S \quad (5-6)$$

where, C_D is the drag coefficient. For long cylinders, 0.82 can be chosen; for a cone shape, it is 0.5 [142]. When a fish cruises (swims at constant speed), the drag force and thrust are equal. Fish cruising speed is:

$$U = \left[\frac{m \cdot \overline{\left(\frac{\partial y(x,t)}{\partial t} \right)^2}}{C_D \rho S + m \cdot \overline{\left(\frac{\partial y(x,t)}{\partial x} \right)^2}} \right]_{x=L} \quad (5-7)$$

Here $m = \rho A(L)$ is the virtual mass at the fish tail tip.

The Froude efficiency of a fish is the rate of work done to achieve cruising and the total work. It is obtained by:

$$\eta = \frac{U\bar{T}}{\bar{W}/t} \times 100\% \quad (5-8)$$

An improved efficiency model for the EBT is shown in Equation (5-9). It is also noted as the improved Froude efficiency model [143].

$$\begin{cases} \eta = \frac{1}{2}(1+\beta) - \frac{1}{2}\alpha^2 \frac{\beta^2}{1+\beta} \\ \alpha = \frac{\lambda}{2\pi} \frac{h'(L)}{h(L)} = \frac{1}{2\pi} \left(\frac{\lambda}{L}\right) \left(\frac{\Delta L}{L}\right)^{-1} \left(1 - \frac{h_t}{h_m}\right) \\ \beta = U/V \end{cases} \quad (5-9)$$

In the equation, λ is wave length; h_m is amplitude at the tail end $x=L$; h_t is the amplitude at $x=L-\Delta L$.

5.1.4 Fish Swimming Data

Fish's swimming speed is related to the tail's flapping frequency. John J. Videler collected thirteen fish species's swimming speed and the corresponding flapping frequencies, flapping amplitudes, etc. in his book 'Fish Swimming' [144]. From the data, as shown in Figure 5-9, it is obvious that when the speed U is measured in body length per second, it is linearly related to the tail flapping frequency f in Hertz. The relationship is:

$$U(BL/s) = 0.71f(Hz) \quad (5-10)$$

Strouhal number describes how fast the tail is flapping relative to its forward speed or the wake behind the fish flapping tail. It is defined as:

$$St = \frac{fA}{U} \quad (5-11)$$

where A is the flapping range (the total distance between the two flapping limits), U and f are the cruising speed and flapping frequency.

The Strouhal number of the above fishes are plotted in Figure 5-10. As in the figure, most of the fishes swim with the Strouhal number between 0.2 and 0.4 as shown in the red region. Especially, for the fast swimmers, such as *Scomber scombrus* and *Oncorhynchus mykiss*, the Strouhal number is close to 0.3. It is believed by robot fish researchers that the optimal swimming efficiency achieved when the Strouhal number is close to 0.3.

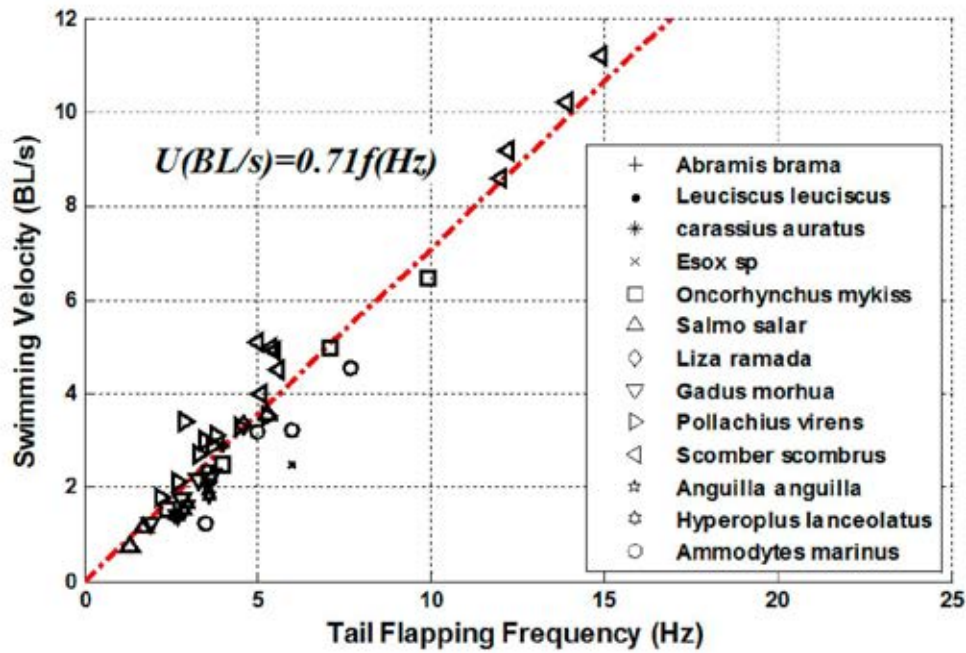


Figure 5-9 Fish Swimming Velocity Scaled to Body Length (data from [144])

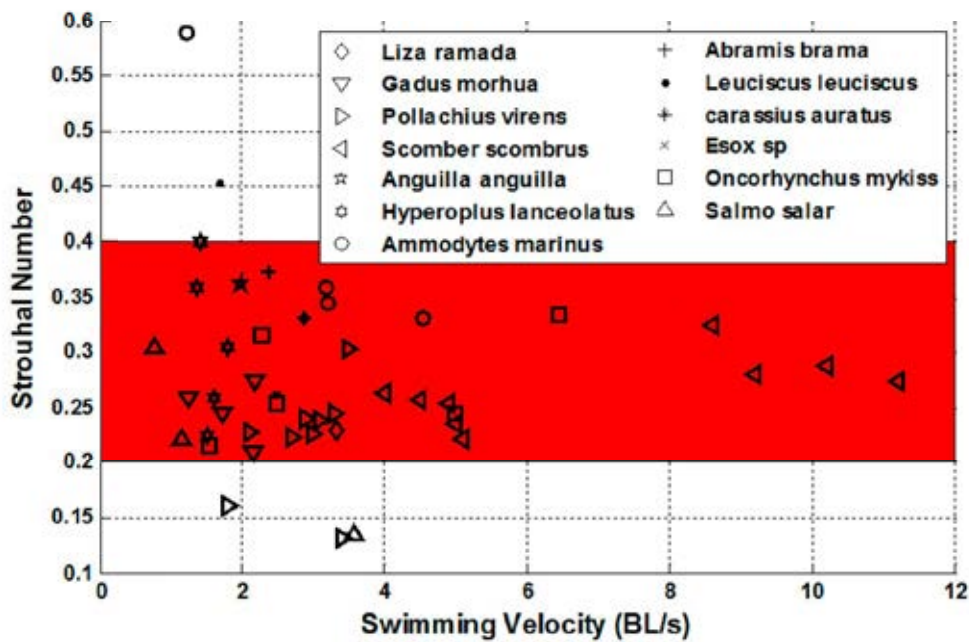


Figure 5-10 Fish Swimming Strouhal Number (data from [144])

5.2 Oscillatory Wire-Driven Robot Fish

For fish swimming in oscillatory form, the tail flaps back and forth periodically in the plane of motion. The fish body curve can be fitted by a circular arc well, as shown in the previous section. Recalling the kinematic model of the WDM, the oscillatory form of swimming can be imitated by the SPXX WDM well. This

section describes two types of oscillatory flapping robot fish which were designed and built. One is the serpentine oscillatory wire-driven robot fish and the other is the continuum oscillatory wire-driven robot fish. The robots have two parts: the fish body and wire-driven oscillatory flapping tail. The propulsion model is developed based on Lighthill's EBT. Swimming examples are given at the end.

5.2.1 Serpentine Oscillatory Wire-Driven Robot Fish Design

The serpentine oscillatory wire-driven robot fish design is divided into two parts: the serpentine oscillatory wire-driven flapping tail design and fish body design.

1) Serpentine Oscillatory Wire-Driven Flapping Tail Design

In this design, the SPSI WDM is adopted. The tail design is as shown in Figure 5-11. In the tail, there are seven serially arranged vertebrae. These vertebrae are similar. The structural parameters of all the vertebrae are shown in Table 5-1.

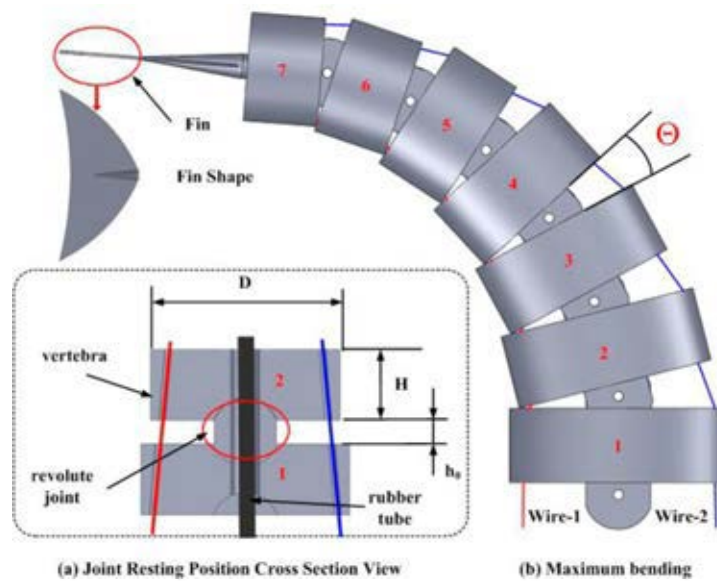


Figure 5-11 Serpentine Oscillatory Wire-Driven Flapping Tail Design

Table 5-1 Tail Vertebra Parameters

Vertebra No.	H_i (mm)	D_i (mm)	h_0 (mm)	θ_{max} (°)
1	20	44	5	13.00
2	19	40	5	14.25
3	18	36	4.5	14.25
4	17	32	4	14.25
5	16	28	3.5	14.25
6	15	24	3	14.25
7	14	20	2.5	14.25

Figure 5-11 (a) shows the cross-sectional view of the joint in the rest position. From the figure, two adjacent vertebrae form a revolute joint. A rubber tube in the central cavity also helps the joints to articulate. It deflects with the joints' rotations, and in the meantime, confines the joints' rotations. With the constraint, all the joints' rotations are the same. A pair of wires goes through all the vertebrae via the pilot holes. One end of the wire is fixed to the tail tip and the other end is connected to the actuator. The wires transmit the actuator's motion and force. They control the tail flapping. When resting, the rubber tube is straight and the wire lengths inside the tail are equal. By pulling the left wire and unwinding the right wire, as shown in Figure 5-11 (b), the tail bends to the left and vice-versa. Flapping motion is achieved by pulling and unwinding the left wire and right wire alternately. A replaceable caudal fin is connected to the tail at the distal end. The fin shape can be arbitrary.

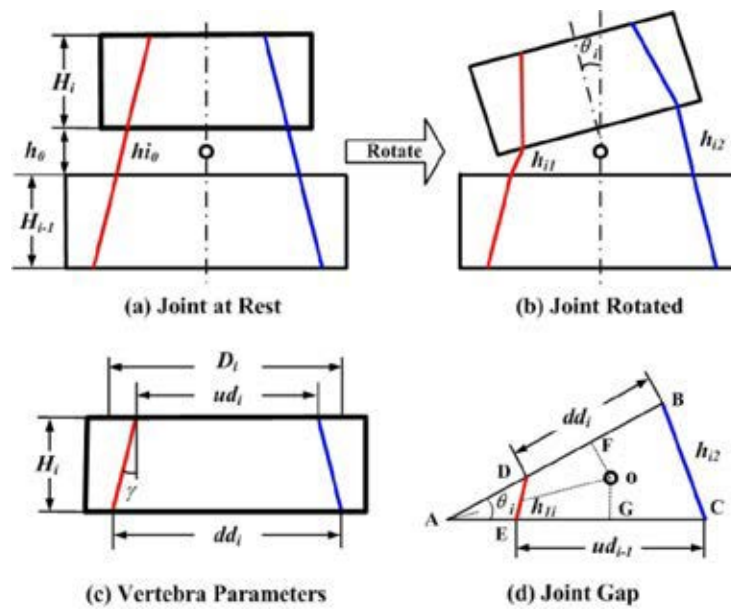


Figure 5-12 Serpentine Tail Joint Kinematics

The relationship between wire length variation and joint rotation angle is shown in Figure 5-12. By simple geometry and assuming the joint rotations are small, i.e. $\sin(\theta) \approx \theta$, the relationship is:

$$\begin{cases} h_{i1} = h_0 - \Delta h_{i1} = \sqrt{(h_0 \cdot \tan \gamma)^2 + (h_0 - \frac{dd_i \cdot \theta_i}{2})(h_0 - \frac{ud_i \cdot \theta_i}{2})} \\ h_{i2} = h_0 + \Delta h_{i2} = \sqrt{(h_0 \cdot \tan \gamma)^2 + (h_0 + \frac{dd_i \cdot \theta_i}{2})(h_0 + \frac{ud_i \cdot \theta_i}{2})} \end{cases} \quad (5-12)$$

In the equation, h_0 is the initial joint gap distance, γ is the incline angle, dd_i is the wire spacing distance at the bottom of the upper vertebra, and ud_i is the wire spacing distance on the top of the lower vertebra. When the wire inclination is small, it is reasonable to make the following approximation:

$$\Delta h_{i1} \approx -\Delta h_{i2} \approx \frac{1}{2} \cdot dd_i \cdot \theta_i \cdot \cos(\gamma) \quad (5-13)$$

In this design, the incline angle is 7.5° . Figure 5-13 shows the relative error with respect to the inclination and joint rotation of this approximation when $\gamma = 7.5^\circ$. In the simulation, the maximum joint rotation is set as 12° . From the figure, the maximum relative error is 1.66%.

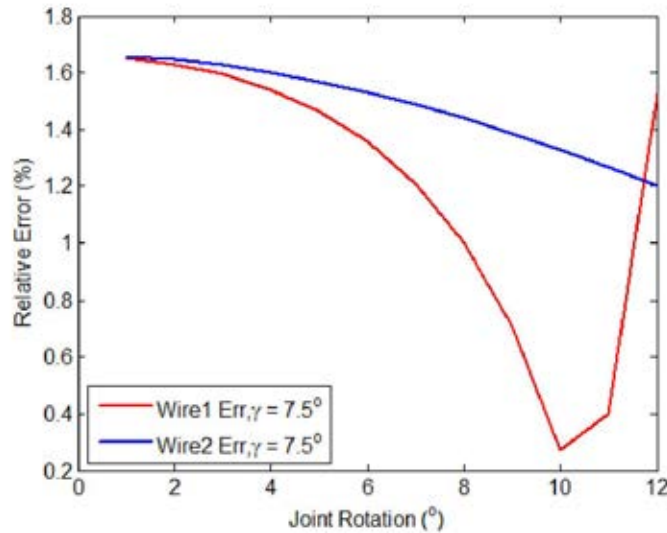


Figure 5-13 Wire Length Change Approximation Error - Serpentine

The maximum joint rotation is:

$$\theta_{i\max} = 2 \arctan \left(\frac{h_0}{D_i} \right) \quad (5-14)$$

Table 5-1 shows the maximum rotation for each joint. For the first joint, the maximum rotation angle is 13° . For other joints, the maximum rotation angle is 14.25° . In this design, the maximum tail bending angle Θ is 98.5° . When the tails bend to 98.5° , the wire length changes for the two wires are 18.37 mm and 19.06 mm, respectively. The approximated wire length change is 18.57 mm. The maximum error is less than 0.5 mm, which is small and can be ignored.

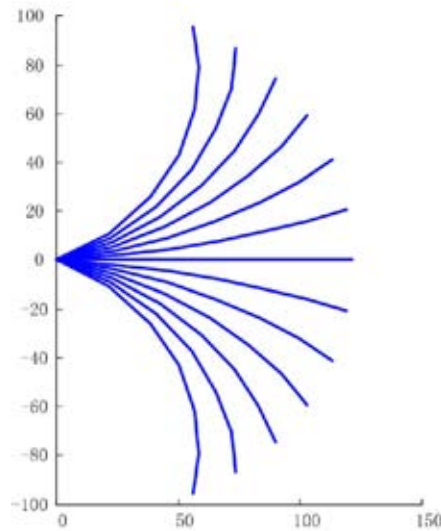


Figure 5-14 Serpentine Oscillatory Wire-Driven Tail Flapping Cycle

Figure 5-14 shows the flapping cycle of the designed serpentine oscillatory wire-driven flapping tail. The deformed shape of the tail is a seven-segment polyline. It can fit the previous oscillatory fish swimming body curve well.

2) Fish Body Design

The fish body is important as it not only houses the power supply, actuator controller, and balance weight but also serves as the tail base. Meanwhile, the body shape, especially the front shape influences the water resistance greatly.

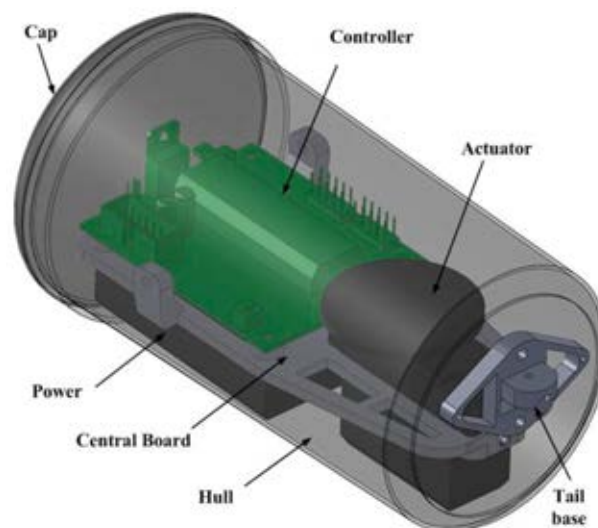


Figure 5-15 Oscillatory Wire-Driven Robot Fish Body Design

As shown in Figure 5-15, the fish body comprises the hull, central board, cap, tail base, controller, actuator, and power. The cylindroid hull is sealed by a cap. The

front of the cap is conical, which helps reduce the water resistance. The tail base is connected to the hull by two bolts. The first vertebra of the tail and the base form a revolute joint. The controller and communication unit is connected to the top of the central board by four bolts. A DC power supply is fixed at the bottom. A servo motor is used as the actuator. It is connected to the central board by four bolts. The central board is fixed to the hull by three bolts. Two pinholes are opened at the bottom of the hull. The wires can go through the pinholes and connect to the motor. To reduce water leakage, the pinhole diameter is almost the same as that of the wire.

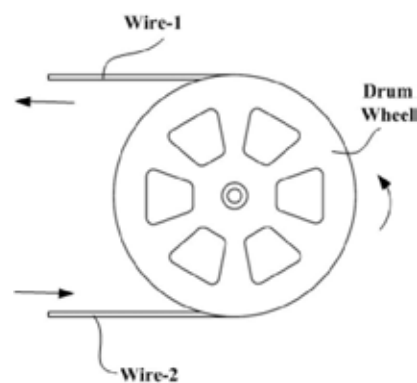


Figure 5-16 Wire Connection

From the previous analysis, the maximum absolute error of wire length change approximation is less than 0.5 mm. As a result, the wires can be connected to the actuator via a drum wheel as shown in Figure 5-16. When the wheel rotates, the wire-1 elongation and wire-2 contraction is the same.

3) Serpentine Oscillatory Wire-Driven Robot Fish

The designed robot fish is as shown in Figure 5-17. The overall length of the robot fish is 328 mm, and the tail length is 175 mm. The maximum cross-sectional area along the robot fish is 52.8 cm^2 , and the displacement of the robot fish is about 580 cm^3 .

Figure 5-18 shows the serpentine oscillatory wire-driven robot fish prototype. The tail is fabricated by Rapid Prototyping (RP). The material is ABS plastic. The maximum flapping amplitude of the tail is 98.5° as in the design. The fin is replacable, a square fin with 80 mm side length is shown in the figure. The wires are 0.475 mm diameter fishing line. The core material is steel and is coated with

resin. One end of the wire is fixed to the last vertebra and the other end is connected to the drum wheel, which rotates with the servo motor (Model: TowerPro MG995). The servo motor is controlled using a Pulse Width Modulation (PWM) signal, which is generated by the commercial Micro Controller Unit (MCU) (Model: AVR AtMega16). The commands are sent to the MCU via a Bluetooth serial com port. A plastic hull is used to house the motor, controller and power supply. The total weight of the robot fish is 484 g.

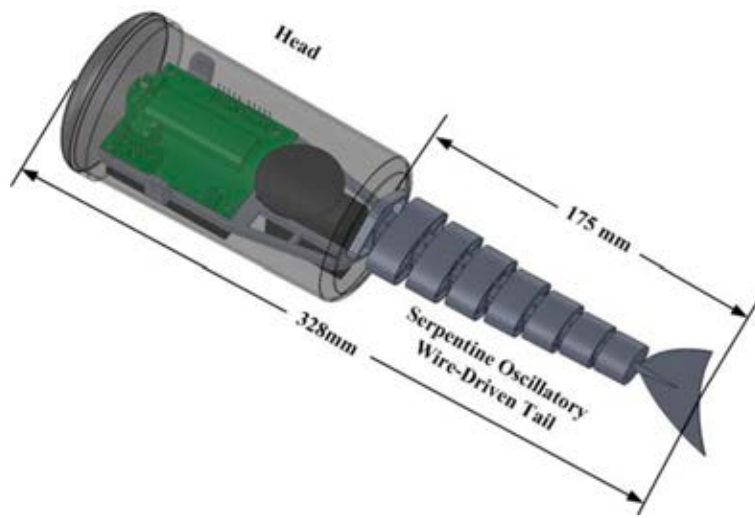


Figure 5-17 Designed Serpentine Oscillatory Robot Fish



Figure 5-18 Serpentine Oscillatory Wire-Driven Robot Fish Prototype

5.2.2 Continuum Oscillatory Wire-Driven Robot Fish Design

The continuum oscillatory wire-driven robot fish design also has two parts, i.e. the tail design and fish body design. The fish body is the same as the serpentine oscillatory wire-driven robot fish. Therefore, only the continuum oscillatory wire-driven flapping tail design is presented here.

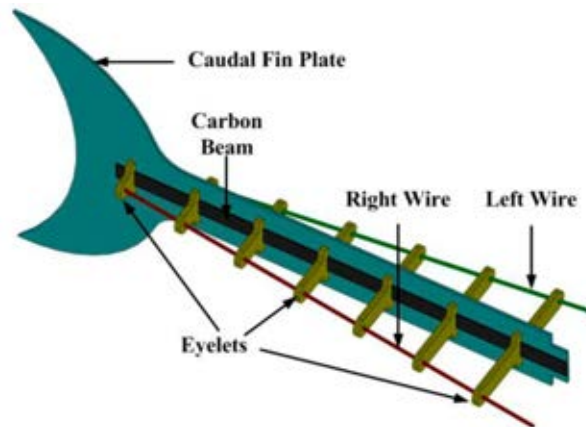


Figure 5-19 Continuum Oscillatory Wire-Driven Flapping Tail Design

The designed tail is as shown in Figure 5-19. In this design, the SPCT WDM is adopted. The backbone is composed by the caudal fin plate and carbon beam. The carbon beam is used to strengthen the backbone. Several spacing discs, (called eyelets here), are evenly distributed along the backbone. Each eyelet has a pilot hole at both ends, allowing the wires to go through. The two wires are placed on the two sides of the backbone. Similar to the previous design, a servo motor is used to reel and unwind the wires. One end of the wire is fastened to the last eyelet, and the other end is connected to the drum wheel as shown in Figure 5-16.

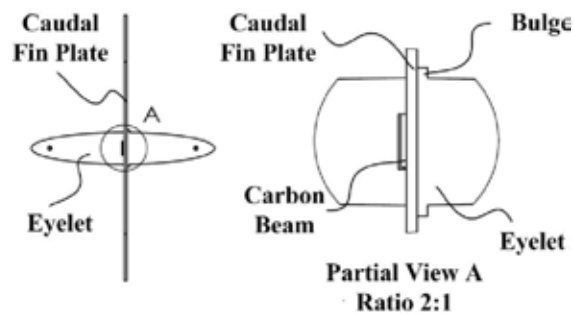


Figure 5-20 Eyelet Connection

The connection of the backbone and eyelets are as shown in Figure 5-20. From the enlarged view, the eyelet is inserted into the slot on the caudal fin plate. When the bulge on the eyelet collides with the fin plate, the slot on the eyelet is on the other side of the backbone. The carbon beam is inserted into this slot, and the three pieces are connected together like a latch. This connection is very simple but secure. When the motor rotates, the length of one wire is shortened and the length of the wire on the other side is elongated. At the same time, the backbone

length remains the same. The tension variation in the two wires produces a bending moment. The backbone is then deformed by the wires. In the design, the thickness of the backbone is much less than its width. As a result, the bending of the backbone is inclined to the thickness direction.

For the continuum WDM, the range between the eyelets is treated as a joint. In the tapered wire configuration, the joint kinematics is as shown in Figure 5-21. The wires are tensioned. As a result, in the figures they are straight lines as shown in red and blue. The backbone is flexible. Ideally, when it is loaded with a pure moment, the deformed shape is a circular arc. In the analysis, the deformed backbone is assumed to be a circular arc. It is also assumed that the spacing discs are orthogonal to the backbone as shown in Figure 5-21 (b). The relationship between the wire length change and the joint rotation is obtained by the Law of cosines:

$$\begin{cases} l_{i1} = l_{i0} - \Delta l_{i1} = \sqrt{(R - r_i)^2 + (R - r_{i-1})^2 - 2(R - r_i)(R - r_{i-1}) \cos \theta} \\ l_{i2} = l_{i0} + \Delta l_{i2} = \sqrt{(R + r_i)^2 + (R + r_{i-1})^2 - 2(R + r_i)(R + r_{i-1}) \cos \theta} \end{cases} \quad (5-15)$$

In the equation $R = h/\theta$, $r_i = d_i/2$, and $r_{i-1} = d_{i-1}/2$.

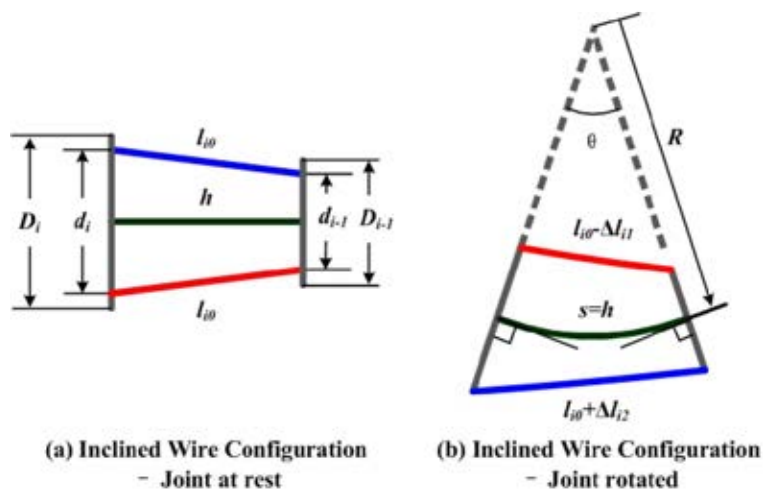


Figure 5-21 Continuum Backbone with Tapered Wire Configuration

Generally, for each joint, the rotation angle θ is small. Hence, it is reasonable to make the simplification: $\cos(\theta) \approx 1 - 0.5\theta^2$. The wire length changes are:

$$\begin{cases} \Delta l_{i1} = +l_{i0} - \sqrt{l_{i0}^2 - (r_{i-1} + r_i) \cdot h \cdot \theta + r_{i-1} \cdot r_i \cdot \theta^2} \\ \Delta l_{i2} = -l_{i0} + \sqrt{l_{i0}^2 + (r_{i-1} + r_i) \cdot h \cdot \theta + r_{i-1} \cdot r_i \cdot \theta^2} \end{cases} \quad (5-16)$$

The same as in the previous design, when the inclination is small the wire length changes can be assumed to be the same. They are approximated as:

$$\Delta l_{i1} \approx -\Delta l_{i2} \approx r_{i-1} \cdot \theta_i \cdot \cos(\gamma) \quad (5-17)$$

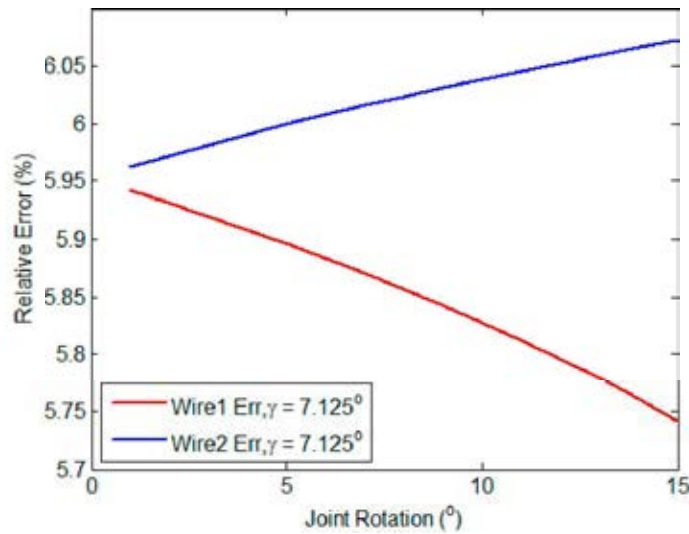


Figure 5-22 Wire Length Change Approximation Error - Continuum

In this design, the incline angle is $\gamma=7.125^\circ$. The eyelet spacing distance is $h=20$ mm, and for the first joint $d_1=39.5$ mm, $D_1=45$ mm. Figure 5-22 shows the relative wire length approximation error. The maximum joint rotation is set as 15° . From the simulation, the error is close to 6%. It is larger than that in the serpentine design. The reason is that the eyelet spacing distance is much larger than the joint initial gap distance in the serpentine joint. To reduce the error, one could use more eyelets.

Flapping motion is obtained by controlling the motor rotate to and fro. Figure 5-23 shows the flapping cycle of the continuum oscillatory wire-driven tail. It is similar to the flapping of the serpentine oscillatory wire-driven robot fish. The flapping amplitude is controlled by the rotation angle of the motor and the flapping frequency is controlled by the motor speed.

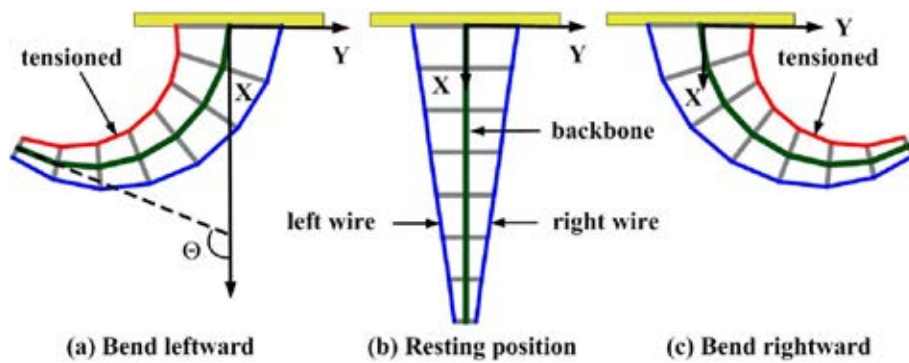


Figure 5-23 Continuum Oscillatory Wire-Driven Tail Flapping Cycle

The prototype is shown in Figure 5-24. The total length of the robot fish is 310 mm, and the tail length is 170 mm. The maximum cross-section diameter of the fish body is 80 mm. The caudal fin is made from an ABS plastic plate, and the wire-guiding eyelets are made by rapid prototyping. The cross-section diameter of the carbon beam is 0.5 mm \times 2 mm. The fish body is the same as that of the serpentine oscillatory wire-driven robot fish.

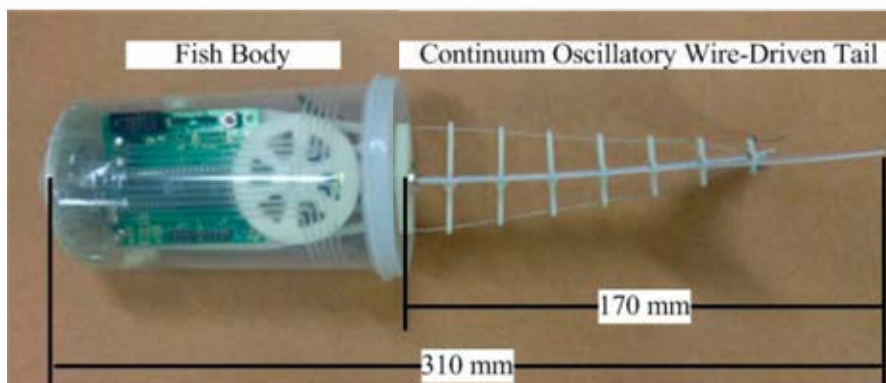


Figure 5-24 Continuum Oscillatory Wire-Driven Robot Fish Prototype

5.2.3 Oscillatory Robot Fish Propulsion Model

In this design, the oscillatory wire-driven robot fish is slender, i.e. the cross-section diameter is much less than the robot length. Also the diameter change is small. This meets the assumption in Lighthill's EBT. As a result, the oscillatory robot fish propulsion model can be established based on EBT.

From the previous introduction, it is known that in the EBT, the tail tip lateral displacement, slope, and traversing velocity determine the mean propulsion, cruising speed, and propulsion efficiency. To develop the propulsion model, one needs to obtain these items of the oscillatory flapping tail.

Table 5-2 Serpentine Oscillatory Wire-Driven Tail Motion Parameters

Lateral Displacement (y)	$y(q,t) _{q=L} = \sum_{i=1}^N l_i \cdot \sin(\sum_{j=1}^i \theta_j)$
Tail Tip Slope ($\partial y/\partial q$)	$\frac{\partial y(q,t)}{\partial q} \Big _{q=L} = \sin(\sum_{i=1}^N \theta_i)$
Lateral Velocity ($\partial y/\partial t$)	$\frac{\partial y(q,t)}{\partial t} \Big _{q=L} = -\sum_{i=1}^N l_i \cdot \left(\cos(\sum_{j=1}^i \theta_j) \cdot \sum_{j=1}^i \dot{\theta}_j \right)$

Table 5-3 Continuum Oscillatory Wire-Driven Tail Motion Parameters

Lateral Displacement (y)	$y(q,t) _{q=L} = \frac{L}{\Theta} [1 - \cos(\Theta)]$
Tail Tip Slope ($\partial y/\partial q$)	$\frac{\partial y(q,t)}{\partial q} \Big _{q=L} = \sin(\Theta)$
Lateral Velocity ($\partial y/\partial t$)	$\frac{\partial y(q,t)}{\partial t} \Big _{q=L} = \frac{L}{\Theta^2} [\Theta \cdot \sin(\Theta) + \cos(\Theta) - 1] \cdot \frac{d\Theta}{dt}$

Assume the tail overall length is L , the number of vertebrae is N , the length of each vertebra is l_i , the flapping frequency is f , and the flapping amplitude is Θ . From the previous kinematic model of the single segment serpentine WDM and single segment continuum WDM, the lateral displacement, slope, and lateral velocity at the tail tip of the serpentine oscillatory wire-driven tail are as shown in Table 5-2 and that of the continuum oscillatory wire-driven tail are shown in Table 5-3. The coordinate frame setting for the serpentine wire-driven tail is shown in Figure 5-25. For the continuum wire-driven tail, the coordinate frame setting is similar as shown in Figure 5-23. The constant curvature assumption is still used. Hence, all the joints are the same, i.e., for $j=2$ to N , θ_j are the same. This is the same for $\dot{\theta}_j$. It is noted that, as the flapping frequency is f , the average joint rotation velocity is $\bar{\dot{\theta}}_j = 4\theta_j f$. In Table 5-3, Θ is the continuum wire-driven tail distal end angle, and $d\Theta/dt$ is the distal end rotation velocity.

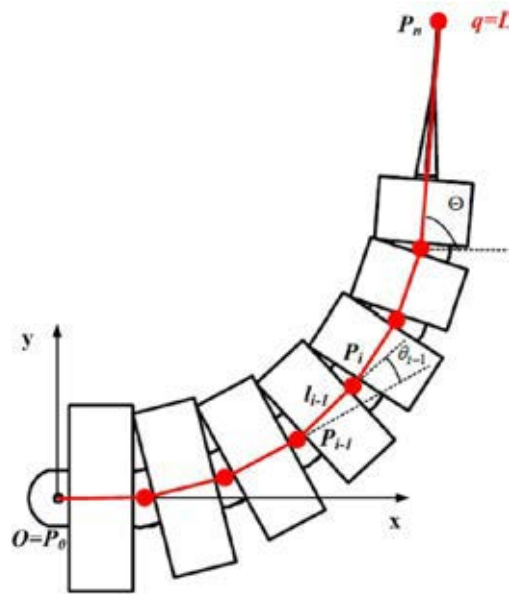


Figure 5-25 Oscillatory Flapping Tail Coordinate Frame Setting

5.2.4 Robot Fish Swimming Control

The swimming of the robot fish is controlled by the wire-driven tail. The wire-driven tail's flapping cycle is divided into four stages as shown in Figure 5-26. Stage I: Flap from the rest position to the right limit; Stage II: Flap from the right limit back to the rest position; Stage III: Flap from the rest position to the left limit; Stage IV: Flap from the left limit back to the rest position.

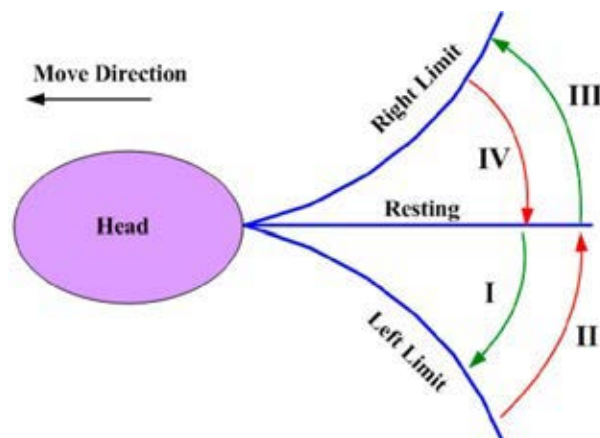


Figure 5-26 Oscillatory Flapping Cycle

The forward speed and turning performance (i.e. turning speed and turning radius) is controlled by the flapping velocity in the four stages and the amplitudes of the two limits. For the oscillatory flapping robot fish, there are three basic motion

modes, i.e. swimming forward, turning to the left and turning to the right. In the forward mode, the tail flaps symmetrically. In the left turning mode, the left amplitude is larger than the right limit, while in the right turning mode, the right amplitude is larger than the left limit. The forward speed and turning radius are both influenced by the flapping amplitude and flapping frequency.

Figure 5-27 shows the swimming control scheme. The command is sent to MCU by the operator via a remote controller. On receiving the command, the MCU generates a PWM sequence which controls the velocity and position of the servo motor. The rotation of the motor are transferred to the tail's flapping motion through the WDM and the robot fish is propelled by the thrust. The direction and magnitude of the thrust is controlled by the flapping motion. A human is in the control loop. Visual feedback is established by the operator. The operator can send different commands to control the robot fish to swim in a desired manner.

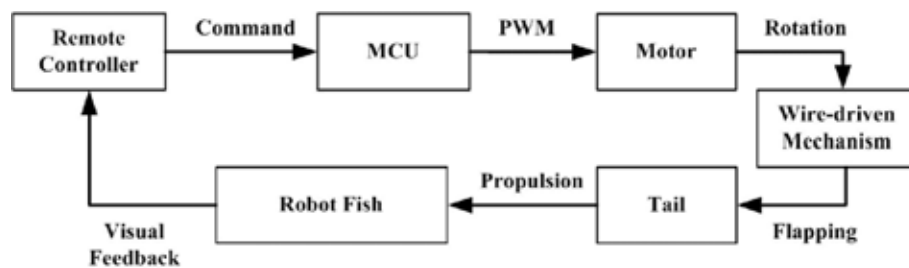


Figure 5-27 Robot Fish Swimming Control Scheme

The frequency of the PWM is 50 Hz, i.e., the period is 20 ms. The duty cycle controls the servomotor rotating position. In the robot fish development, the servomotor selected is Towerpro MG995. For this motor, the rotation range is 180°. When the PWM duty cycle is 1.5/20, or the voltage at high level continuous 1.5 ms in one period, the servomotor is in the middle position (90°). When the PWM duty cycle is 1/20, the motor shaft is at the left limit (0°); when the PWM duty cycle is 2/20, the motor shaft is at the right limit (180°). By controlling the duty cycle, the motor position is controlled. The motor speed is the maximum (0.17s for 60°) from one position to another. The speed is controlled by segmenting the rotation and setting time delays. The following table shows the control signals in the three motion modes: swimming forward, turning left and turning right. The Δt controls the amplitude of the flapping.

Table 5- 4 PWM Duty Cycle for Swimming (1/20)

Time (T)	Forward	Turn Left	Turn Right
0	1.5	1.5	1.5
0.25	$1.5-\Delta t$	$1.5-0.5\Delta t$	$1.5+0.5\Delta t$
0.5	1.5	$1.5-dt$	$1.5+dt$
0.75	$1.5+\Delta t$	$1.5-0.5\Delta t$	$1.5+0.5\Delta t$
1	1.5	1.5	1.5

5.2.5 Swimming Experiments

The experimental setup is as shown in Figure 5-28. An inflated water tank was used to test the two oscillatory wire-driven robot fishes' swimming performances. The length of the tank is 1.4 m, and the width 0.9 m. A one meter reference with 10 divisions was placed in the tank for better evaluation of the swimming performance. In the experiments, the effects of flapping amplitude and frequency on swimming velocity were tested. Also, robot fish's turning performance was tested.

1) Serpentine Oscillatory Wire-Driven Robot Fish Swimming Experiments

For the serpentine oscillatory wire-driven robot fish, three experiments were carried out: moving forward with different flapping amplitudes; moving forward with different flapping frequencies; turning performance.

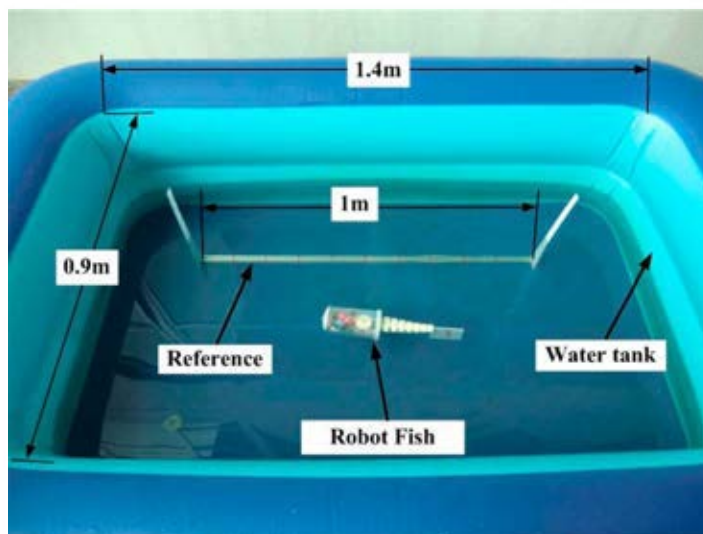


Figure 5-28 Robot Fish Swimming Experiment Setup

Experiment 1 - Forward with Different Flapping Amplitudes

The first experiment tested the effects of flapping amplitude. The experiment results are shown in Table 5-5. In the experiments, the flapping frequencies in the four stages are all 0.75 Hz. Six flapping amplitudes, i.e. 90°, 75°, 60°, 45°, 30°, and 15° were tested. The average swimming velocity (V_m) is estimated from the video frames. From the results, the cruise velocity increases with larger amplitudes. The increase rate decreases from 15° to 90°. The maximum velocity in these tests was 12.35 cm/s, which is 0.38 BL/s. The cruise velocity predicted by the propulsion model (V_p) is also shown in Table 5-5. In the model, the robot fish has a conical cap, and the drag coefficient is selected as 0.5 [142]. The wetted surface area of the robot fish is 613 cm². It is seen that the predictions are generally larger than the measurements, and the average prediction error is 14.25%.

The Froude efficiency is calculated from Equation (5-9) using V_p . It is affected little by the flapping amplitude. However, the trend is that with smaller flapping amplitude, the robot fish has a higher Froude efficiency. From the results, the maximum efficiency is 68.88%, which is achieved at 30° flapping amplitude.

Table 5-5 Influence of Flapping Amplitudes on Velocity - Serpentine

Amp (°)	Freq (Hz)	V_m (cm/s)	V_p (cm/s)	Err (%)	Efficiency (%)
90	0.75	12.35	15.28	19.18	65.68
75	0.75	11.63	13.37	13.01	66.47
60	0.75	9.09	11.19	18.77	65.79
45	0.75	8.43	8.74	3.55	68.51
30	0.75	5.71	6.01	4.99	68.88
15	0.75	2.67	3.07	13.03	68.02

Experiment 2 - Forward with Different Flapping Frequencies

In the second experiment, the influence of flapping frequency on cruising speed was tested. Table 5-6 lists the experiment conditions and results. In the experiments, the flapping amplitude was fixed as 45°. The flapping velocities in the four stages are the same. The tested frequencies are from 1.5 Hz to 0.25 Hz.

The average forward velocity increases with the frequency. In the range of 0.25 Hz to 1.25 Hz the increase rate is almost constant. From 1.25 Hz to 1.5 Hz, the increase rate is small. The maximum velocity in these cases is 13.56 cm/s, which is 0.413 BL/s. From the propulsion model, the cruise velocity should increase linearly. The same as the first experiment, the prediction is slightly larger than the measurements. The average prediction error is 16.23%. Also, the efficiency is less affected by the flapping frequency. In this experiment, the maximum efficiency was 68.51%, which is achieved at the flapping frequency $f=0.75$ Hz.

Table 5-6 Influence of Flapping Frequency on Velocity - Serpentine

Amp($^{\circ}$)	Freq (Hz)	V_m (cm/s)	V_p (cm/s)	Err (%)	Efficiency (%)
45	1.5	13.56	17.48	22.43	65.54
45	1.25	12.20	14.56	16.21	66.55
45	1	10.64	11.65	8.67	67.74
45	0.75	8.43	8.74	3.55	68.51
45	0.5	5.32	5.83	8.75	67.74
45	0.25	2.30	2.91	20.96	65.77

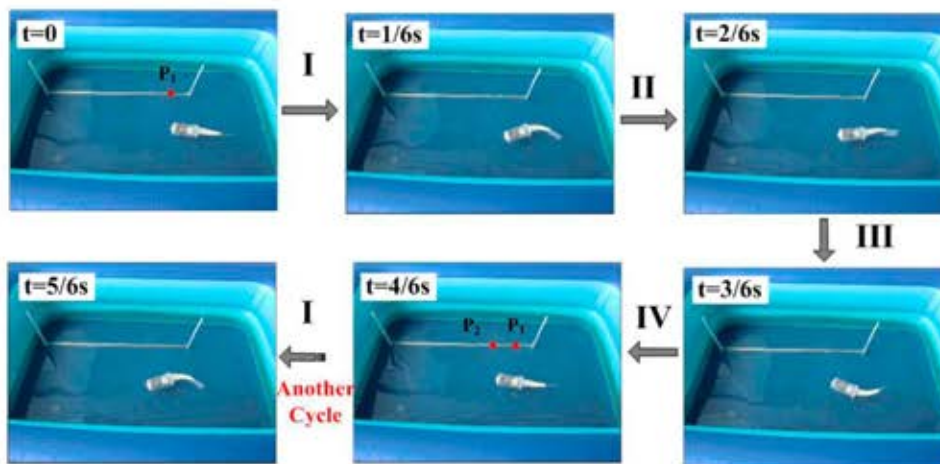


Figure 5-29 Serpentine Oscillatory Wire-Driven Robot Fish Swimming Forward

Figure 5-29 shows the robot fish swimming forward in one flapping cycle. In this example, the flapping amplitude is 45° and the flapping frequency is 0.75 Hz. At the beginning, the robot fish tail is relaxed and its head is in position P_1 . The tail flaps to the left to the amplitude at first. During this process, as shown in the figure, the head also bends to the same side. In the next stage, the tail returns to

the rest position. The head also goes back. The robot fish is as in the relaxed status. After that, the tail flaps to the right. The final step is the tail flapping back to the rest position. It is noted that the tail and head always bend to the same side. During one flapping cycle, the robot fish moves from P_1 to P_2 . The robot keeps on moving with continuous flapping cycles.

Experiment 3 - Turning

In the third experiment, turning performance of the robot fish was tested. Figure 5-30 shows one flapping cycle in the robot fish turning mode. Flapping amplitudes and flapping speeds in the four stages are different. Also, some stages may not exist. For example, when turning to the left, the tail flaps to the right first, and then returns back. The flapping cycle has only two stages. After one cycle, as shown in the figure, the robot head turns an angle α to the left. After several cycles the robot fish turns 360° and returns to the original position.

Figure 5-31 shows the trajectory of the robot fish when turning around. It is seen that the trajectory of the robot head is close to a cycle, and the radius is 8 cm (0.24 BL). In the example, the average turning speed of the robot fish is $27.5^\circ/s$.

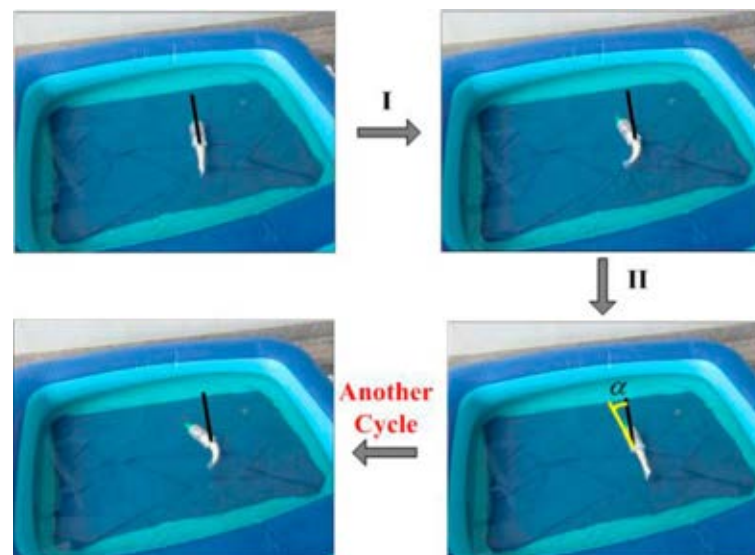


Figure 5-30 Flapping Cycle in the Turning Mode

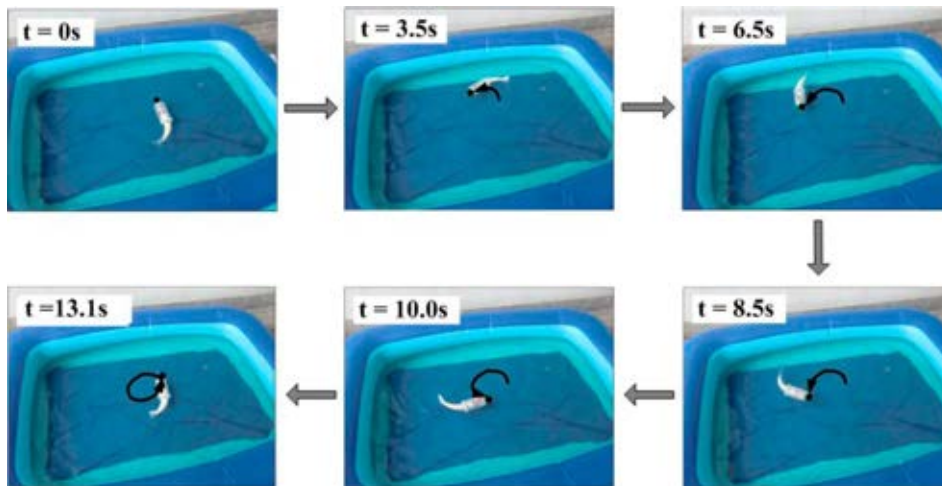


Figure 5-31 Oscillatory Flapping Wire-Driven Robot Fish Turning Trajectory

In the experiments, the serpentine oscillatory wire-driven robot fish swims forward and turns to both sides effectively. From the results, it is seen that the forward velocity is affected by both the flapping amplitude and the frequency. The predicted velocities are larger than the measurements. There could be several error sources. The first source is the modeling error. In Lighthill's model the swimmer should be slender and the diameter change along the fish body be small. For this robot fish, there is an abrupt geometry change in the connection of the head and the tail. Also, there could be discrepancy in the drag coefficient. The second source is the measuring error. The measured velocity is the average speed of the robot fish crossing the water tank, including the acceleration stage. Hence, the measured velocity is smaller than the cruising speed. The third source is the reduced flapping frequency. Due to the water resistance, the actual flapping frequency is smaller than input. As shown in [98], there could be around a 10% decrease in frequency when the tail flaps in water.

The maximum velocity recorded is 13.56 cm/s from all the experiments, which is 0.413 BL/s. The speed can be further increased. In the experiments, the flapping speeds in the four stages were the same. When the flapping speeds in stage I and III were smaller than the speeds in stage II and IV, the forward velocity of the robot fish was greater. The fin size tested was 80 mm×80 mm. A fin with other shapes such as lunate may improve the forward speed. Also, the maximum frequency tested was 1.5 Hz due to the limitation of the servo motor. A more

powerful actuator can be used to improve the robot fish's performance. In the test, the power consumption of the motor is between 0.5 watt and 1.5 watt.

2) Continuum Oscillatory Wire-Driven Robot Fish Swimming Experiments

Similar tests were carried out for the continuum oscillatory wire-driven robot fish. Figure 5-32 shows the flapping cycle in the cruising mode.

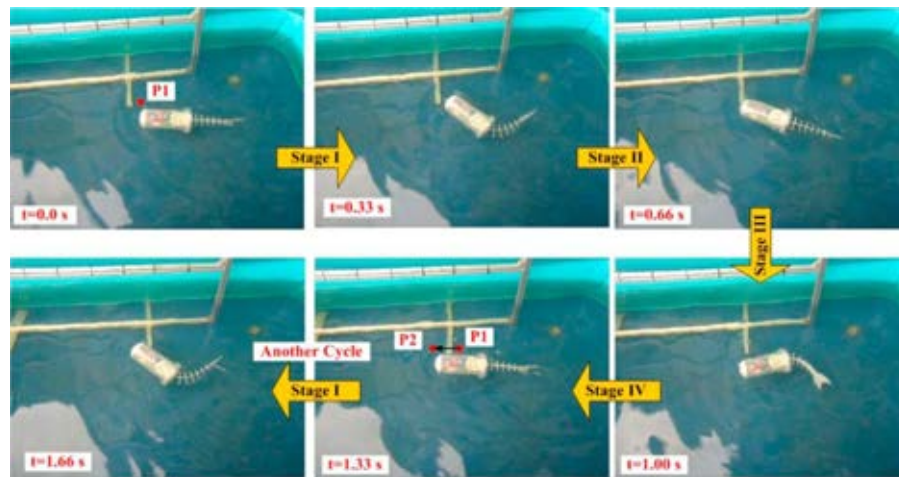


Figure 5-32 Continuum Oscillatory Wire-Driven Robot Fish Cruising Example

The Froude efficiency is calculated using the measurements. The results are summarized in Table 5-7 and Table 5-8. From the results, when the flapping amplitude increases from 30° to 90° , the cruising speed as well as the Froude efficiency increase at first and then decrease. The maximum efficiency (64.4%) is achieved at 60° , when cruising speed is 0.254 BL/s. When the amplitude is fixed at 45° and the flapping frequency increases from 0.25 Hz to 1.0 Hz, the Froude efficiency decreases from 64.2% to 58.2%.

Table 5-7 Influence of Flapping Amplitude on Velocity - Continuum

Amp($^\circ$)	Freq(Hz)	V_m (cm/s)	V_p (cm/s)	Err (%)	Efficiency (%)
30	0.75	3.30	3.87	14.73	62.0
45	0.75	4.34	5.66	23.32	60.2
60	0.75	7.89	7.3	-8.03	64.4
75	0.75	6.19	8.77	29.42	59.9
90	0.75	6.12	10.07	39.24	58.8

Table 5-8 Influence of Flapping Frequency on Velocity - Continuum

Amp(°)	Freq(Hz)	Vm (cm/s)	Vp (cm/s)	Err (%)	Efficiency (%)
45	0.25	1.98	1.89	-4.51	64.2
45	0.5	2.83	3.78	25.15	60.5
45	0.75	4.34	5.66	23.32	61.3
45	1	4.46	7.55	40.97	58.4

Figure 5-33 shows the continuum oscillatory wire-driven robot fish turning left. In the flapping cycle, the left flapping amplitude is larger than the right flapping amplitude. From the tests, the turning speed and radius is affected by the flapping amplitude more than the flapping frequency. In fact, when the flapping frequency increases the robot moves forward rapidly. This increases the turning radius as well as impairing the turning speed.

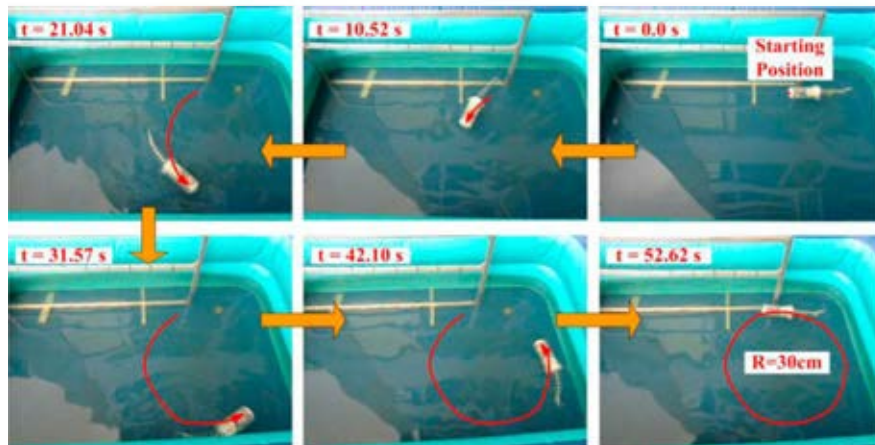


Figure 5-33 Continuum Oscillatory Wire-Driven Robot Fish Turning Example

From the results, it is shown that the continuum oscillatory wire-driven tail is also well suited to water propulsion. The backbone structure parameters, e.g. bending rigidity, are key factors in relation to propulsion performance. With high rigidity, large tension is needed for the wires. This will increase the motor power consumption. However, if the rigidity is too low, the bending shape is lost. On the other hand, the model frequency of the propulsor is low due to its low rigidity. When the actuation frequency is close to the model frequency, the deformation of the propulsor is affected greatly by the mode shape. This is different from the

serpentine oscillatory wire-driven flapping tail, whose model frequencies are usually large.

To sum up, the two types of oscillatory wire-driven robot fish both use one motor to actuate the flexible tail. The tail structure is simple and compact. The control is easy. More importantly, compared with screw propellers the efficiency is high. The tail motions resemble the oscillatory fish swimming body curve well. However, the performance of the oscillatory serpentine wire-driven robot fish is better. This is because the vertebrae are rigid, which allows large wire tension. Also, due to the small joint initial gap distance, the approximation errors of the wire length changes are smaller. Therefore, the drum wheel can control the two wires' lengths in the oscillatory flapping tail well.

5.3 Undulatory Wire-Driven Robot Fish

From the previous example, the serpentine WDM is more suitable for robot fish. In this section only the undulatory serpentine wire-driven robot fish is described.

5.3.1 Undulatory Wire-Driven Robot Fish Design

The same as the oscillatory wire-driven robot fish, the undulatory wire-driven robot fish is composed of the fish body and wire-driven propulsor, or the tail. In the tail design, the MPSP WDM is used. The undulatory flapping tail is actuated by two servo motors.

1) Serpentine Undulatory Wire-Driven Flapping Tail Design

The tail is composed of 13 vertebrae and two pairs of wires, as shown in Figure 5-34. These vertebrae are divided into two segments, as shown in the figure. The first segment has six vertebrae (numbered from 1 to 6); the second segment has seven vertebrae (numbered from 7 to 13). The profile of the tail is shaped by the fin plate. It is similar to that of a slender fish tail. The plastic fin plate also helps articulate the vertebrae and confine the joint rotations.

The wire routing is shown in Figure 5-34 (b). The two pairs of wires are coplanar. The first wire pair goes through the first five vertebrae via the pilot holes and is connected to the end of the sixth vertebra. The second wire pair passes by the first

six vertebrae via the central cavity and goes through the left seven vertebrae via the pilot holes. The end is connected to the end of the 13th vertebra.

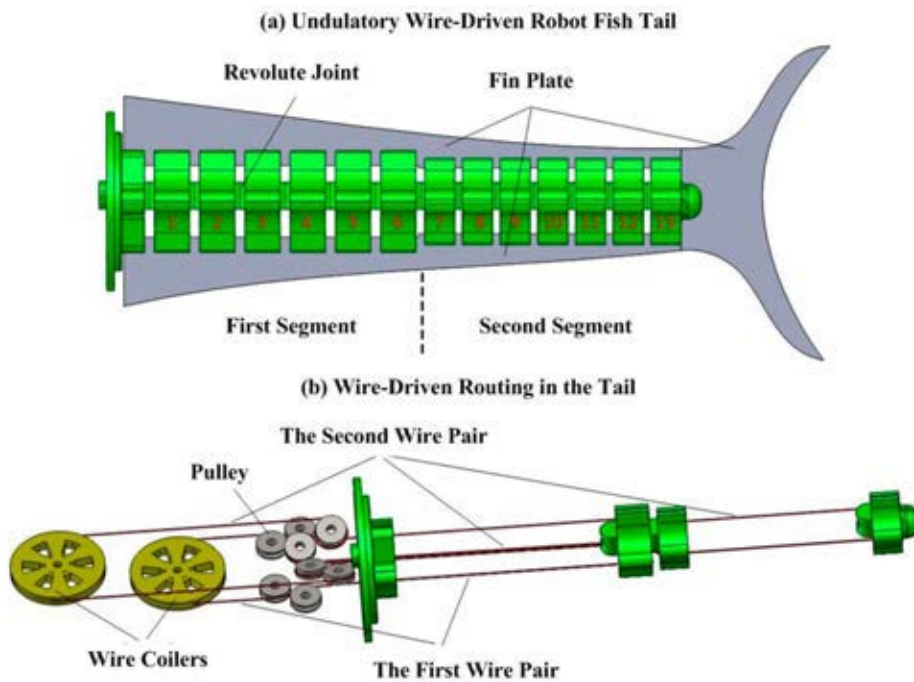


Figure 5-34 Undulatory Wire-Driven Tail Design

Detailed vertebra design and joint connection are shown in Figure 5-35. All the vertebrae in the segment are similar. The first three figures in Figure 5-35 show the sixth vertebra. For vertebrae 1-6, there are two pilot holes used to guide the wires; two cylindrical surfaces are used to form a revolute joint; one guide hole is used to pass the wires of the second group; two slots (upper slot and lower slot) connect the fin plate; and one central cavity holds the elastic tube. For vertebrae 7-13, the structure is similar. The difference is there is no guide hole. As shown in the figure, two successive vertebrae form a revolute joint. The wire routing in the sixth and seventh vertebrae is also shown in the figure.

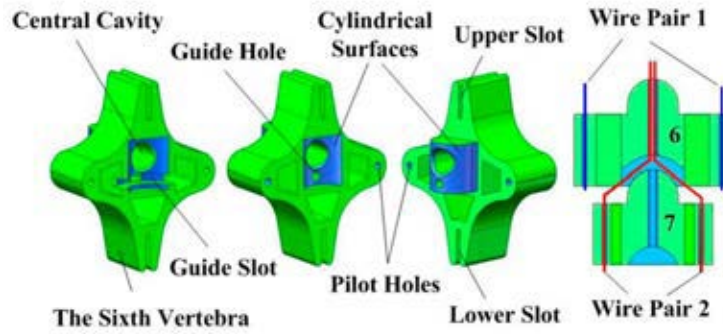


Figure 5-35 Vertebra Design and Joint Rotation

Detailed vertebra parameters are shown in Table 5-9. In the table, the meaning of H , D , d , and h_0 are the same as that in the serpentine oscillatory wire-driven robot fish. The maximum joint rotation can be found by Equation (5-14). For the first segment, the maximum rotation of each joint is 14.7° . The maximum bending angle of the first segment is 88.20° . For the second segment, the maximum joint rotation is 14.55° . The maximum bending angle of the second segment is 101.85° . The overall bending angle of the propulsor is 190.05° .

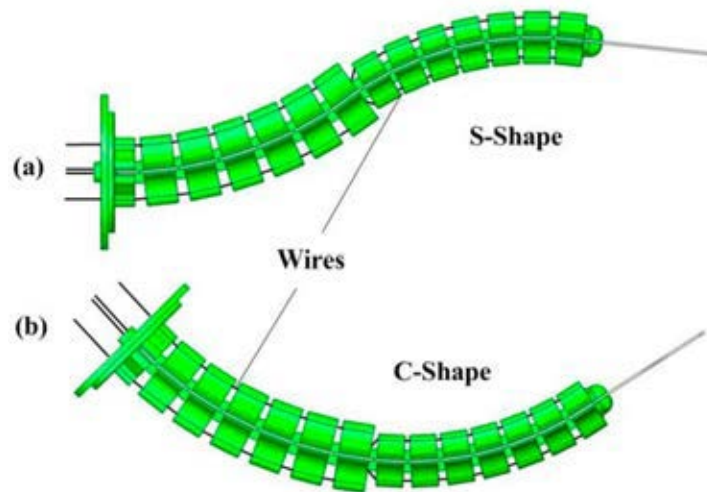


Figure 5-36 S-Shape and C-Shape Bending of the Wire-Driven Tail

Table 5-9 Vertebra Parameters

Vertebra No.	$H(\text{mm})$	$D(\text{mm})$	$d(\text{mm})$	$h_0(\text{mm})$	$\theta_{\max}(\text{^\circ})$
1~6	15	31	27	4	14.70
7~13	12	23.5	19	3	14.55

For the two segments, their motions are independently controlled. There are a couple of motion combinations. However, two motions are basic. One is the two segments bending in the same direction. The tail forms a C-Shape. The other is the two segments bending in the opposite direction. The tail looks like an S-Shape. Figure 5-36 shows the two basic motions of the tail. In swimming, if the tail deforms to the C-Shape all the time, the tail performs C-motion or the fish swims in oscillatory form. If the tail deforms into the S-Shape all the time, the tail performs S-motion or the fish swims in undulatory form. This shows that the undulatory wire-driven robot fish can swim in both oscillatory form and undulatory form. There are other motions, such as keeping the first segment relaxed and flapping the second segment only. This is called the small C-motion.

The motion of the undulatory wire-driven tail is compared with the fish undulatory swimming body curve, as shown in Figure 5-37.

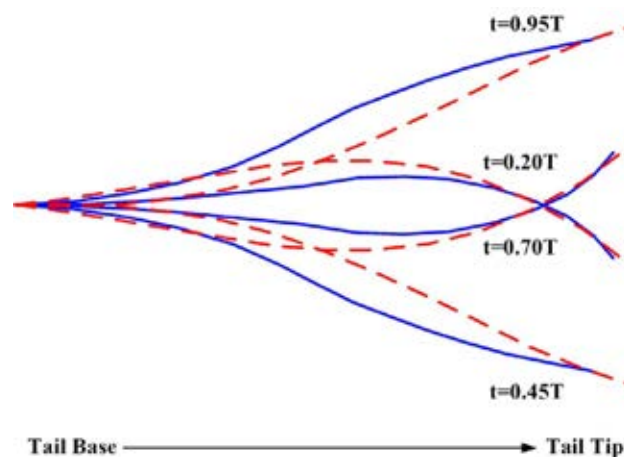


Figure 5-37 Undulatory Swimming Curve Comparison

The red dashed lines are the fish curves at four time instances in a flapping cycle according to the fish body curve model. In the model, parameters are chosen as $c_1=0.2$, $c_2=0.4175$, $k=2.4$, and $\omega=2\pi$. The blue solid lines are the undulatory wire-driven tail at the same time instances. The rotation of each angle is controlled as per Equation (5-18). The phase lag of the second segment is -0.3π . From the figure, the two set of curves match reasonably well, especially at the tail tip, which is the major factor influence fish's swimming performance according to Lighthill's EBT.

$$\theta_i = \begin{cases} 8\sin(2\pi t) = \theta_a & i = 1, 2, \dots, 6 \\ -12\sin(2\pi t - 0.3\pi) = \theta_b & i = 7, 8, \dots, 13 \end{cases} \quad (5-18)$$

2) Fish Body Design

Figure 5-38 shows the designed fish body. It comprises the airtight hull, central board, controller, motors, and DC power supply. Two servo motors are fixed to the central board. The hull is axisymmetric. The front of the hull is a paraboloid. It helps reduce water resistance. At the end of the hull, pinholes are opened to let the wires pass through. Each pair of wires is connected to a drum wheel, which rotates with the motor shaft. The propulsor is connected to the central board by screws. Waterproofing is achieved by a silicone covering. The overall length of the robot fish is 495 mm. The tail length is 280 mm.

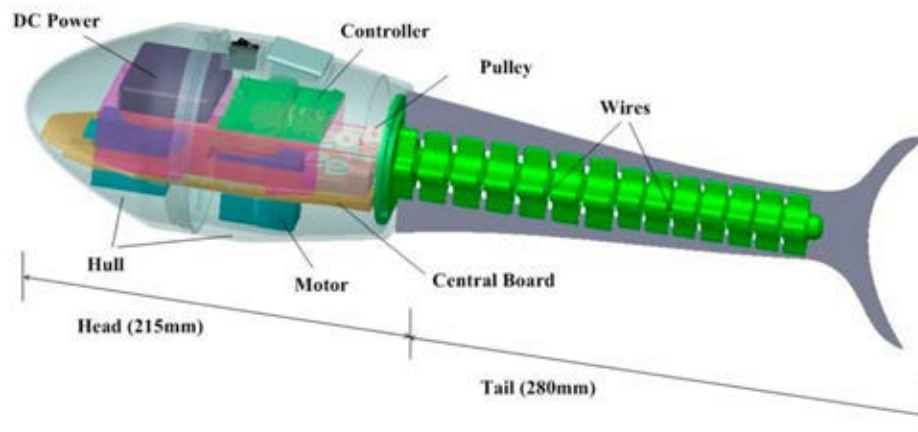


Figure 5-38 Undulatory Wire-Driven Robot Fish Body Design

3) Undulatory Wire-Driven Robot Fish Prototype

An undulatory wire-driven robot fish was built according to the design. The vertebrae were fabricated by 3D printing. The material used was ABS plastic. Steel wires with 0.475 mm diameter were used as the controlling wires. The motors, power supply control board and control scheme are the same as that used in the oscillatory wire-driven robot fish. The overall mass of the robot fish is 1256 g. Figure 5-39 shows the robot fish prototype. Figure 5-39 (a) shows the robot fish tail in the rest position; Figure 5-39 (b) shows the tail bending into a C-Shape, and Figure 5-39 (c) shows the tail bending into an S-Shape.

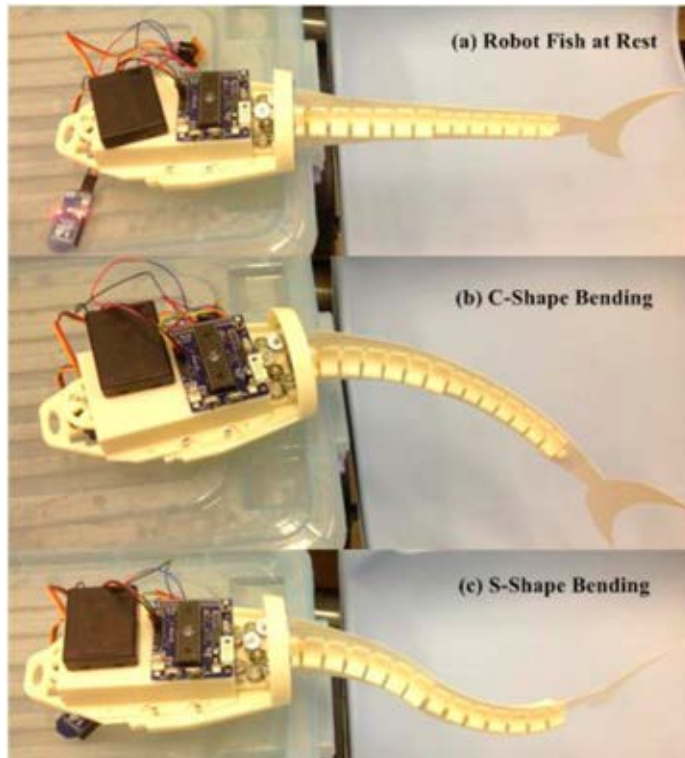


Figure 5-39 Undulatory Wire-Driven Robot Fish Prototype

5.3.2 Undulatory Wire-Driven Robot Fish Propulsion Model

The propulsion model of the undulatory wire-driven tail was developed in a similar way: by substituting the tail tip lateral displacement, lateral velocity, and tail tip slope into EBT. In the kinematic analysis, the vertebrae are treated as the line segments as shown in Figure 5-40.

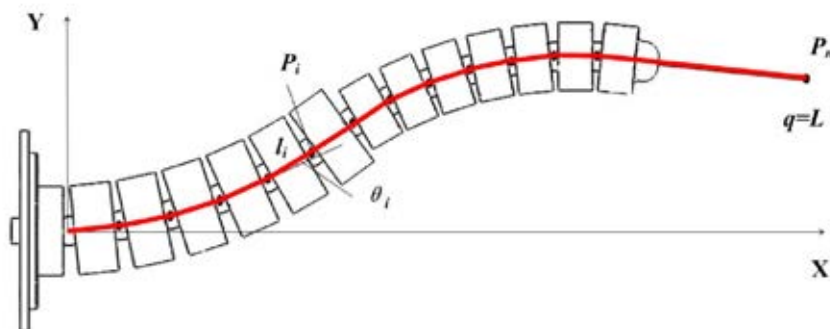


Figure 5-40 Undulatory Wire-Driven Tail Coordinate Frame Setting

In the figure, P_i is the ending point of line segment i , l_i is the length of line segment i , θ_i is the bending angle between line segment i and $i-1$, L is the total length of the propulsor, q is the generalized coordinate denoting the distance from

the tail base along the backbone curve, and i is the index of vertebrae, ranging from 1 to 13. The constant curvature assumption is still adopted here. In the wire-driven tail, the joint rotation within a segment is the same. Hence, for i from 1 to 6, θ_i are the same, as well as $\dot{\theta}_i$ and it is the same for i from 7 to 13.

The lateral displacement, slope and lateral velocity of the tail tip are as shown in Table 5-10. They can be derived from the serpentine WDM kinematic model. The propulsion model is obtained by substituting the three items into the EBT.

From the previous experiments, the fish body sways with the tail flapping. This increases the water resistance. A sway coefficient C_s is multiplied to the drag force in Equation (5-6) in calculating the cruising speed. Generally, a larger C_s is chosen for oscillatory swimming.

Table 5-10 Undulatory Wire-Driven Tail Motion Parameters

Lateral Displacement (y)	$y(q,t) _{q=L} = \sum_{i=1}^6 l_i \cdot \sin(i \cdot \theta_a) + \sum_{i=7}^{13} l_i \cdot \sin[6\theta_a + (i-6) \cdot \theta_b]$
Tail Tip Slope ($\partial y / \partial q$)	$\left. \frac{\partial y(q,t)}{\partial q} \right _{q=L} = \sin(6\theta_a + 7\theta_b)$
Lateral Velocity ($\partial y / \partial t$)	$\left. \frac{\partial y(q,t)}{\partial t} \right _{q=L} = \sum_{i=1}^6 l_i \cdot [\cos(i \cdot \theta_a) \cdot (i \cdot \dot{\theta}_a)] + \sum_{i=7}^{13} l_i \cdot (\cos(6\theta_a + (i-6) \cdot \theta_b) \cdot (6\dot{\theta}_a + (i-6) \cdot \dot{\theta}_b))$

5.3.3 Swimming Experiments

Swimming experiments were performed in an inflated water tank as shown in Figure 5-28. The robot fish swims in the water tank under the control of an operator. Figure 5-41 shows the two basic swimming motions of the robot fish: oscillatory swimming (C-Motion) and undulatory swimming (S-Motion).

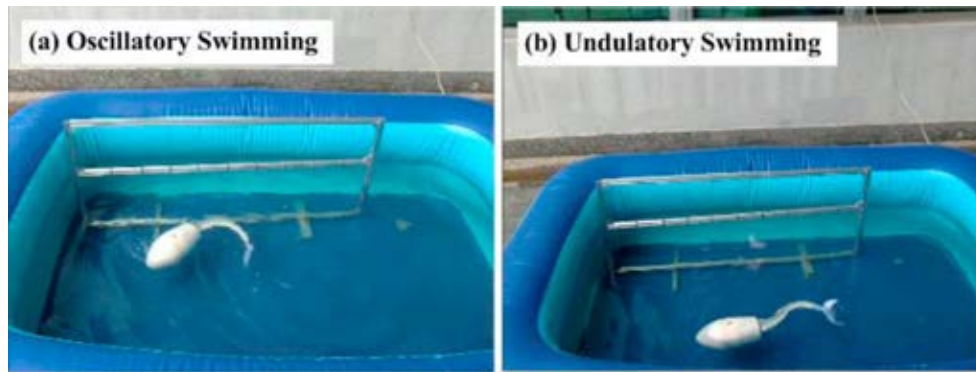


Figure 5-41 Oscillatory and Undulatory Swimming

Four experiments were performed, i.e. “Big-C-Motion” forward, “Small-C-Motion” forward, “S-Motion” forward, and turning.

Experiment 1 – “Big-C-Motion” Forward

In the first experiment the performance of “Big-C-Motion” was tested. In this swimming mode, the two segments flap synchronizely. There is no phase lag between the two segments. In the experiment, the cruising velocity of the robot fish was recorded under various bending amplitudes of the two segments and waving frequencies. The average speed is estimated by the traveling distance and time from the video frames. The traveling distance is estimated using the ruler standing in the tank, as shown in Figure 5-41. The results are shown in Table 5-11, where A_1 is the flapping amplitude of the first segment, A_2 is the flapping amplitude of the second segment, f is waving frequency, V_m is average cruising velocity, V_p is the predicted velocity. From the results, when the frequency is constant, the cruising speed increases with the increasing of flapping amplitudes. Theoratically, when the flapping amplitudes are constant, the cruising speed increases linearly with waving frequency. However, the experiment results show a different relationship. When the waving frequency f is in the range between 0 Hz and 1 Hz, the cruising speed increases with f . Beyond this range, the crusing speed does not increase anymore, it even decreases. This is due to the low frequency response of the WDM.

In this test, the maximum velocity is 300.75 mm/s, which is 0.608 BL/s (BL represents for body length). It is achieved when the bending amplitudes of the first segment is 40° and the second segment is 60° , and the waving frequency is 1 Hz. The velocities predicted by the propulsion model are generally larger than the

measured one. One reason is that, the measured velocity is the average speed, which is smaller than the actual cruising speed. Also, due to the size limitation of the pool, the robot fish didn't reach the cruising speed when it gets to the other side of the pool. When the flapping frequency is 1.5 Hz, the error is largest. One important reason is the low frequency response of the WDM, and another is the water resistance reduces the flapping frequency. In the test, the measured Froude efficiency of the robot fish swimming in oscillatory form is between 55.56% and 65.62%. This is consistent with the previous oscillatory wire-driven robot fish.

Table 5-11 Big-C-Motion Swimming Experiment Result

$A_1(^{\circ})$	$A_2(^{\circ})$	$f(\text{Hz})$	$V_m(\text{mm/s})$	$V_m(\text{BL/s})$	$V_P(\text{mm/s})$	$\eta(\%)$
20	30	1	201.73	0.408	252.50	61.99
30	45	1	260.51	0.526	325.90	62.14
40	60	1	300.75	0.608	353.60	63.04
40	60	0.5	198.87	0.402	176.80	65.62
40	60	1.5	154.00	0.311	522.30	55.56

Experiment 2 – “Small-C-Motion” Forward

In this experiment, only the second segment flaps in oscillatory form. The first segment remains still. Table 5-12 shows the results of this experiment. The relationship of cruising speed and bending angle is similar to that in the “Big-C-Motion”, as well as the speed-frequency relationship. The maximum speed in this experiment was 162.76 mm/s (0.329 BL/s), when the bending angle of the second segment was 60° and the waving frequency 1 Hz.

Table 5-12 Small-C-Motion Swimming Experiment Result

$A_1(^{\circ})$	$A_2(^{\circ})$	$f(\text{Hz})$	$V_m(\text{mm/s})$	$V_m(\text{BL/s})$	$V_P(\text{mm/s})$	$\eta(\%)$
0	30	1	91.74	0.185	134.60	61.99
0	45	1	140.85	0.285	190.10	62.75
0	60	1	162.76	0.329	233.30	62.11
0	60	0.5	143.54	0.290	116.70	68.07
0	60	1.5	136.75	0.276	346.80	57.52

The propulsion model predicts the swimming velocity reasonably well. The prediction is generally larger than the measurement. The measured Froude efficiency of the robot fish performing “Small-C-Motion” swimming is between 57.52% and 68.07%. It is a little bit higher than that of “Big-C-Motion”. Recalling the results in the previous section, it is seen that the efficiency is lower when the flapping tail is more flexible.

Experiment 3 – “S-Motion” Forward

The performance of “S-Motion” was tested similarly. Because the motion of the propulsor is the superposition of the two segments’ motions, the modes of “S-Motion” were varied according to the different phase difference between the two segments.

Table 5-13 S-Motion Swimming Experiment Result

$A_1(^{\circ})$	$A_2(^{\circ})$	$f(\text{Hz})$	$t_p(\text{T})$	$V_m(\text{mm/s})$	$V_m(\text{BL/s})$	$V_P(\text{mm/s})$	$\eta(\%)$
20	30	1	5/8	207.29	0.419	194.00	81.50
30	45	1	5/8	278.88	0.563	280.60	79.83
40	60	1	5/8	333.33	0.673	356.60	78.27
40	60	0.5	5/8	308.37	0.623	178.30	88.51
40	60	1.5	5/8	200.57	0.405	537.00	63.54
40	60	1	1/8	289.26	0.584	578.50	66.14
40	60	1	3/8	121.95	0.246	356.60	62.47
40	60	1	1/2	32.26	0.065	8.40	92.85
40	60	1	7/8	303.33	0.613	578.50	66.84

Table 5-13 shows the results from this experiment, where t_p is the number of periods that the second segment precedes the first segment. According to the experiments, when t_p is 5/8 the performance of “S-Motion” is optimal. For this reason, in the flapping amplitude and flapping frequency test, t_p is chosen as 5/8. The maximum speed in this experiment was 333.33 mm/s (0.673 BL/s), when the bending angles of the two segments were 40° and 60° respectively, the waving frequency was 1 Hz and t_p is 5/8. The velocity prediction is reasonably good. It is seen that the measured Froude efficiency is much larger than the above results.

The maximum Froude efficiency is 92.85%. The averaged efficiency from this experiment is 75.56%. From the tests, when the robot fish swims in the undulatory form, the head is more stable. This reduces the drag force a lot. Hence, it improves the fish's efficiency. This agrees well with the design idea for BUAA's SPC robot fish, i.e. "Stability first, Propulsion second, and Control third."

Experiment 4 – Turning

In the fourth experiment, the robot fish turning performance was tested. In the test, the robot fish turned using the "Big-C-Motion". Figure 5-42 shows the turning cycle of the robot to the right. The turning radius of the robot fish is around 0.7 BL, and the time used is 7 s. The turning speed is 51.4°/s.

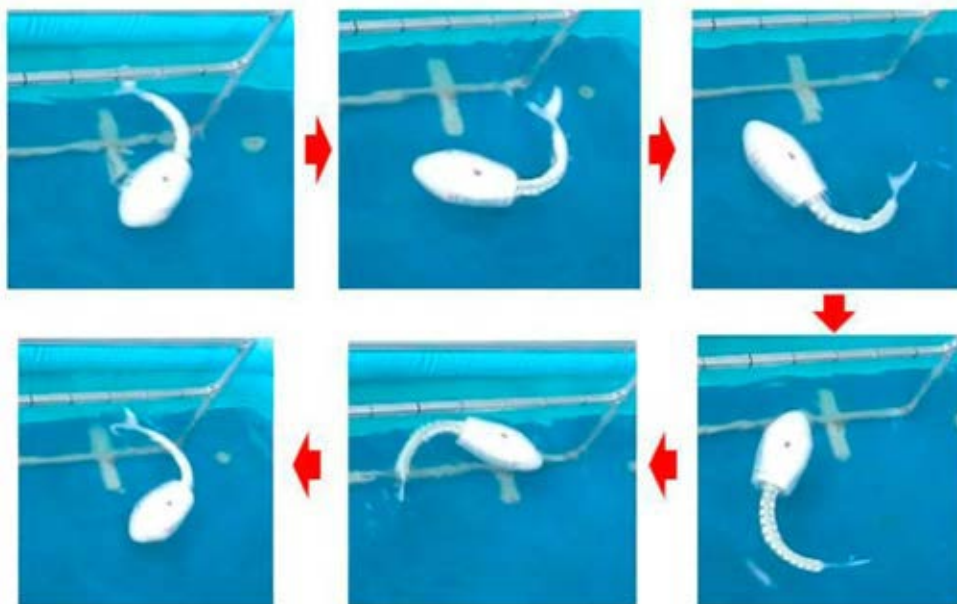


Figure 5-42 Undulatory Wire-Driven Robot Fish Turning

From the experiment results, the undulatory wire-driven robot fish swims efficiently in both oscillatory form and undulatory form. In "S-Motion" or the undulatory form of swimming, the fish body's swaying is smaller. Hence, the drag force is reduced. This helps increase swimming speed. However, compared with the "C-Motion", the "S-Motion" is inferior in relation to turning.

5.4 Vector Propelled Wire-Driven Robot Fish

The motion of oscillatory wire-driven robot fish and undulatory wire-driven robot fish are both planar. By changing the wire configuration, the WDM backbone can bend in 3D space. Taking advantage of this feature, a vector propelled wire-driven robot fish was designed and built. The robot fish comprises the vector propulsor (tail) and fish body. The uniqueness of this robot fish is that the tail can provide thrust in arbitrary directions.

5.4.1 Vector Propelled Wire-Driven Robot Fish Design

1) Vector Propulsor Design

The direction of thrust is confined to the flapping plane. To provide vector thrust, the flapping plane should be controllable. In the design, the SSSI WDM is used. The propulsor has two independent motions: one is flapping in the horizontal plane, and the other is flapping in the vertical plane.

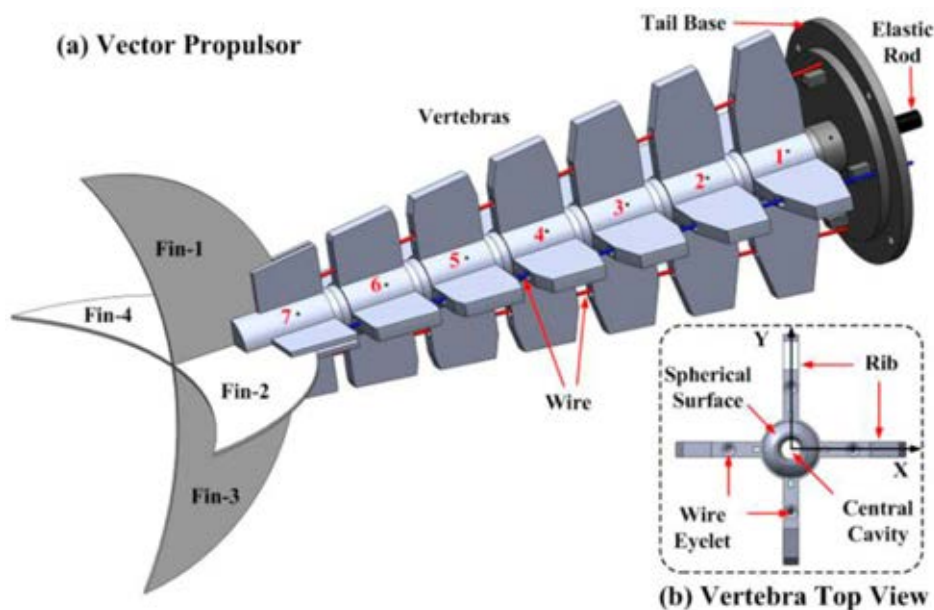


Figure 5-43 Vector Propulsor Design: (a) Vector Propulsor Isometric View; (b) Vertebra Top View

Figure 5-43 (a) shows the designed propulsor. It comprises the tail base, several vertebrae, fins, elastic rod, and the controlling wires. Four fin pieces are inserted into the last vertebra as shown in the figure. Two opposite fin pieces form a lunate caudal fin. The vertebra is shown in Figure 5-43 (b). It has four

orthogonally distributed ribs. A wire eyelet penetrates each rib. On the top of the vertebra there is a convex spherical surface, and on the bottom of the vertebra there is a concave spherical surface with the same diameter. In the middle of the vertebra is a central cavity, in which there is an elastic rod. The number of vertebrae is arbitrary. Seven vertebrae are shown in the figure. The vertebrae are articulated by a uniform elastic rod and spherical joints. The rod confines the vertebra to rotate about its own axis, and as a result the joint can only rotate about the X and Y axes, i.e. the propulsor can bend horizontally and vertically without twisting. The rotations are controlled by two pairs of wires, which are guided by the eyelets on the vertebra rib as shown in Figure 5-43 (b). As the two pairs of wires are orthogonally arranged, the horizontal rotation and vertical rotation are independent. The wires work in pairs. When one wire is pulled and the other is loosened, the propulsor will bend to the shortened wire. The load acting on the elastic rod is pure moment, and it will deform to a circular arc. Therefore, all the joint rotations are the same during propulsor flapping.

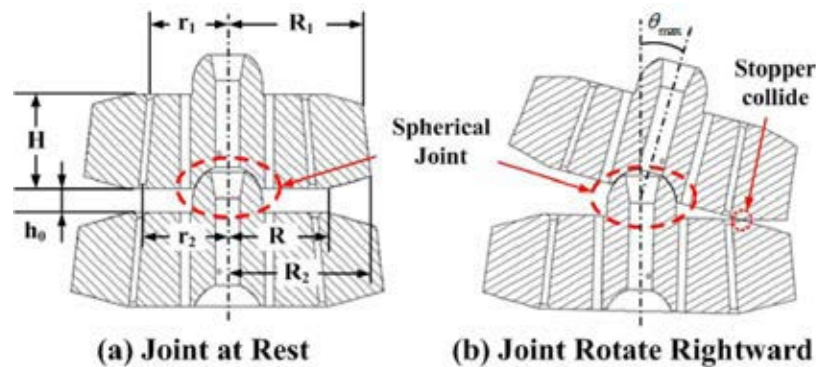


Figure 5-44 Joint Rotation: (a) Joint at Rest; (b) Joint Rotating to the Right

Table 5-14 Vertebra Dimensions

Num	$H(\text{mm})$	$h_0(\text{mm})$	$R(\text{mm})$	$R_1(\text{mm})$	$R_2(\text{mm})$	$r_1(\text{mm})$	$r_2(\text{mm})$
1	20	5	21.12	20.00	22.29	10.00	11.71
2	20	5	21.12	22.86	25.14	12.14	13.86
3	20	5	21.12	25.71	28.00	14.29	16.00
4	20	5	21.12	28.57	30.86	16.43	18.14
5	20	5	21.12	31.43	33.71	18.57	20.29
6	20	5	21.12	34.29	36.57	20.71	22.43
7	20	5	21.12	37.14	39.43	22.86	24.57

Figure 5-44 shows the cross-section view of the joint. The wire pilot hole is inclined with respect to the propulsor axis. This is helpful in reducing the wire tension [145]. In the figure, H is the rib height; h_0 is the joint gap distance; r_1 is the top wire eyelet central distance; R_1 is the vertebra top width; r_2 is the bottom eyelet central distance; R_2 is the vertebra bottom width; R is the radius of the stopper. The joint can rotate about the X and Y axes independently, and the two rotations are the same. The maximum joint rotation angle is determined by the joint gap distance h_0 and the stopper radius R as shown in Figure 5-44 (b). In the designed vector propulsor, there are seven vertebrae. The dimensions of the vertebrae are as shown in Table 5-14, and the unit is mm. Each joint can rotate up to 13.5° in both X and Y directions. The maximum bending angle is 94.5° .

2) Fish Body Design

Figure 5-45 shows the fish body design. It comprises the hull, main board, auxiliary board, servo motors, wire coilers (drum wheels), controller, battery, pulleys, etc. The hull has three pieces: hull-1, hull-2, and hull-3. Hull-1 is the base of the robot fish. The main board and tail base are fastened to Hull-1. Hull-2 is fastened to Hull-1 by four pegs. It is used to facilitate the robot fish assembly. The front of Hull-3 is a paraboloid, which is helpful in reducing water resistance. The assembly procedure is also shown in the figure: Step I, connect Hull-2 to Hull-1; step II, cover the fish body by Hull-3. The three pieces form an axisymmetric robot fish hull.

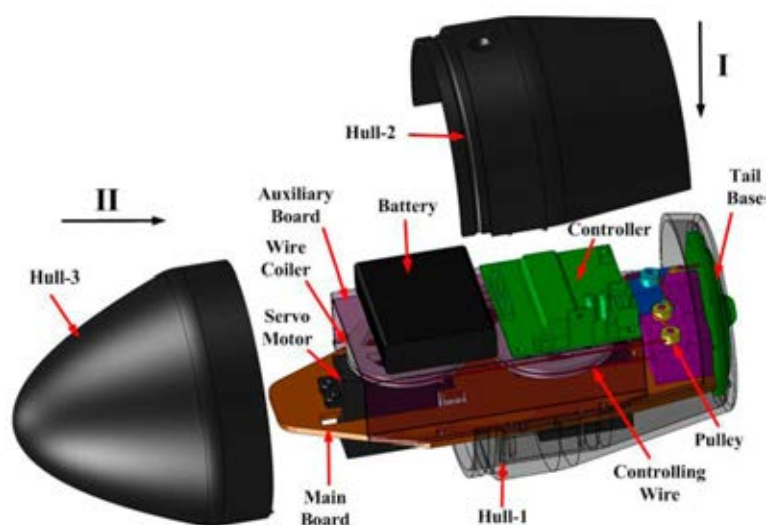


Figure 5-45 Fish Body Design

Two servo motors are used to control the wire lengths. The motor in front controls the vertical wire group, while the other one controls the horizontal wire group. The wires are guided by the pulleys. One end of the wire is fixed at the last vertebra, and the other end is connected to the wire coiler, which rotates with the servo motor. The motor motion is controlled by the MCU controller. The control scheme is similar to the oscillatory wire-driven robot fish. The command is sent out by the operator using a remote controller or using a comport via Bluetooth. On receiving the signal, the MCU generates a 50 Hz PWM sequence. The position of the servo motor is controlled by the duty cycle of the PWM, while the velocity is controlled by setting time delays between positions.

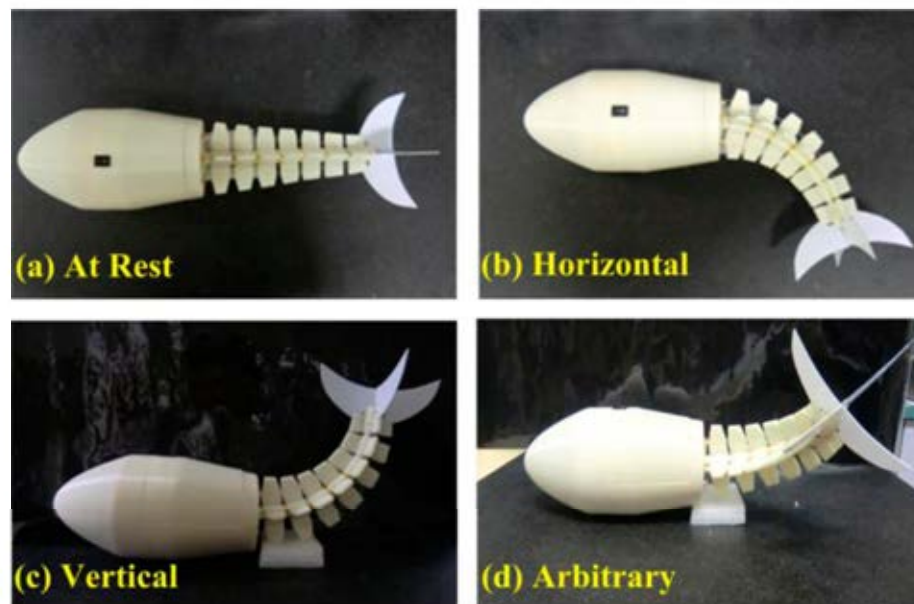


Figure 5-46 Vector Propelled Robot Fish Prototype: (a) Robot Fish in the Rest Position; (b) Propulsor Bending Horizontally; (c) Propulsor Bending Vertically; (d) Propulsor Bending in an Arbitrary Direction

The robot fish prototype is built as shown in Figure 5-46. In the prototype there are seven vertebrae, which are made by RP. The size of each vertebra is as listed in Table 5-14. The maximum rotation of each joint is 13.5° . The vertebrae are connected by a silicon rubber rod with a 5 mm diameter. To improve the rigidity, four carbon sticks of 0.5 mm diameter are connected to the tail. To reduce the friction, lubricating oil is added to all the joints. Four plastic fins are orthogonally mounted on the last vertebra. Two opposite fins make a lunate shape, which is similar to the caudal fin of a dolphin. Two servo motors are used to control the

propulsor flapping. The length of the robot fish is 425 mm. A balancing weight is used to adjust the fish's suspension in water. The overall weight of the robot fish is 1.65 kg.

5.4.2 Tail Motion Analysis

The tail flapping is illustrated in Figure 5-47 (a). The flapping amplitude Θ and flapping direction Φ are controlled by the lengths of the four wires. The flapping direction is defined as the angle between the X axis (horizontal direction) and the flapping plane. The wire configuration is as shown in Figure 5-47 (b), where P_1, P_2, P_3, P_4 denote the wire location. P_1 and P_3 are the horizontal wire group. They control the propulsor bending about the Y axis (flapping is in the horizontal plane). P_2 and P_4 are the vertical wire group. They control the propulsor bending about the X axis (flapping in in the vertical plane). When the flapping direction is Φ , it is conceived that the propulsor bends about a virtual axis Y' as shown in the figure. The distance between the wires in the vertical group and the virtual axis is a , while the distance between the wires in the horizontal group and the virtual axis is b .

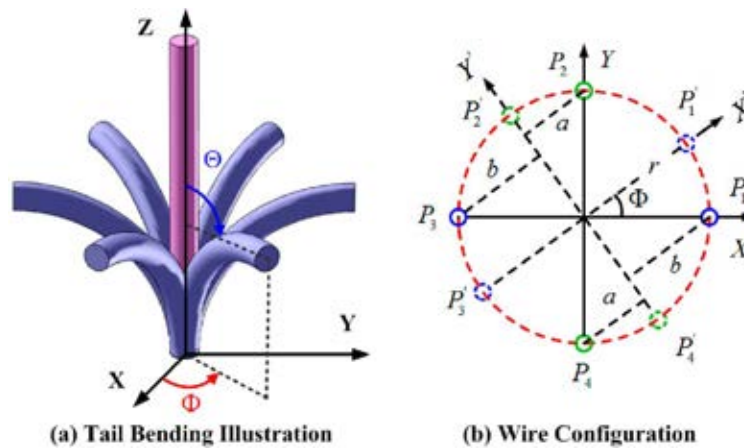


Figure 5-47 Tail Bending and Wire Configuration

The flapping plane is controlled as that in the WDM. Assuming the lengths for the four wires are $L_1, L_2, L_3,$ and L_4 , the flapping direction is as in Equation (5-19). The flapping amplitude is controlled as per Equation (5-20). Here, r is the average radius of r_1 and r_2 , as shown in Figure 5-44; N is the joint number.

$$\Phi = \arctan \left(\frac{L_2 - L_4}{L_1 - L_3} \right) \quad (5-19)$$

$$\Theta = 2N \cdot \arcsin \left[\frac{\sqrt{(L_1 - L_3)^2 + (L_2 - L_4)^2}}{4N \cdot r} \right] \quad (5-20)$$

Figure 5-48 shows some examples of the vector propelled flapping motion. Figure 5-48 (a) shows the tail flapping horizontally, or in the XOZ plane. The blue curves represent the tail location during the flapping cycle, while the red curve denotes the tail tip trajectory. Figure 5-48 (b) shows the tail flap vertically, or in the YOZ plane. Figure 5-48 (c) and (d) show the tail flap direction angle are 45° and -45° respectively. All these flapping motions are planar, i.e. during the flapping the tail is in the same plane.

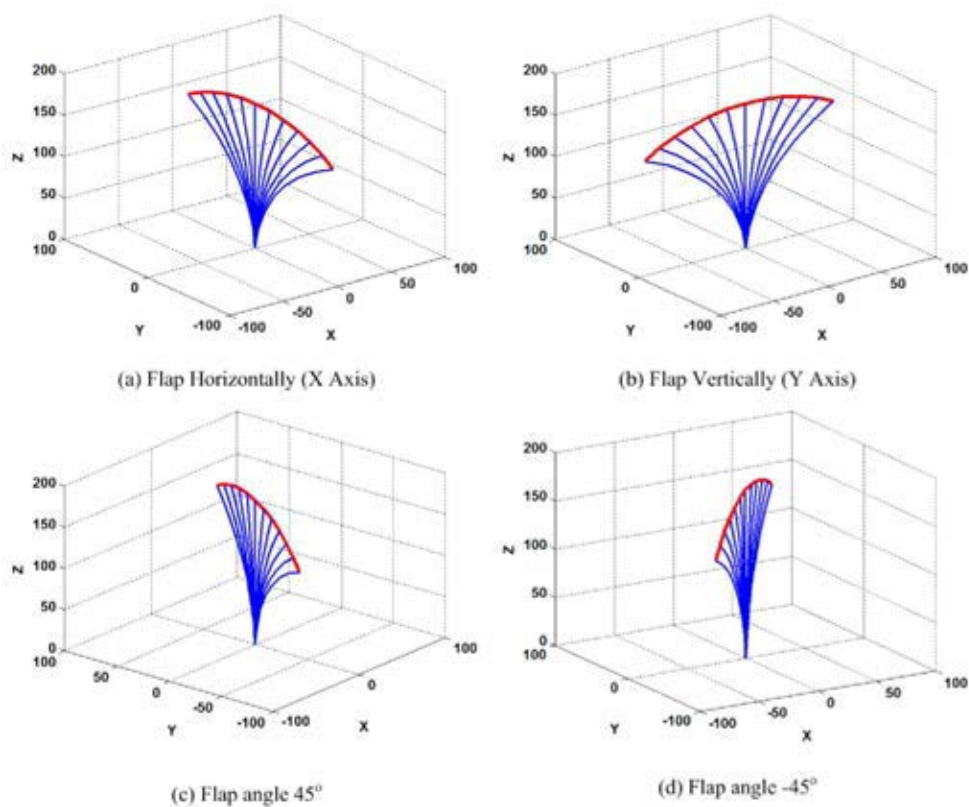


Figure 5-48 Vector Propulsor Planar Flapping

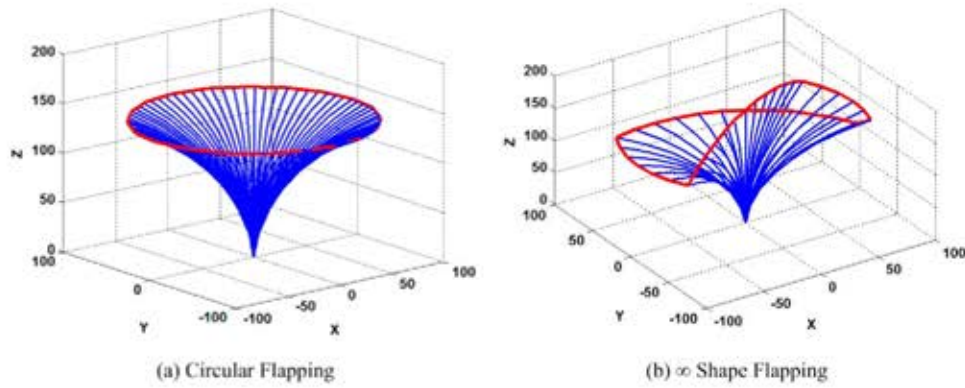


Figure 5-49 Vector Propulsor Spatial Flapping

Besides planar flapping, the vector propulsor can perform a lot more swimming motion. Such as circular flapping, ∞ shape flapping, etc. Figure 5-49 (a) shows the tail flapping motion is circular, and Figure 5-49 (b) shows the tail flapping is a ∞ shaped. Both are spatial.

5.4.3 Swimming Experiments

The robot fish was tested in an inflated swimming pool as before. The robot fish was covered with a rubber skin to waterproof it. In the experiment, swimming performance of the robot fish in still water was tested. Two sets of experiments were conducted, i.e. swimming in shark form (flap horizontally) and swimming in dolphin form (flap vertically).

1) Swimming in Shark Form

In this experiment, the vector propulsor flaps horizontally like a shark. In this mode, the back motor controls the horizontal wire group and the front motor keeps still. Fin 1 and fin 3 provide thrust, while fin 2 and fin 4 do not. The flapping frequency of the tail is $f=1\text{ Hz}$, and the flapping amplitude is 45° . At first, the robot fish is placed in the swimming pool. When the water is still, the robot fish is controlled via a Bluetooth comport flapping horizontally. One flapping cycle is shown in Figure 5-50 (a)-(e). Figure 5-50 (a) shows the robot fish in the rest position. Then it flaps to the left as shown in Figure 5-50 (b). When it reaches the left-most position, it flaps back to the rest position as shown in Figure 5-50 (c). The other half cycle follows a similar mode. The propulsor flaps to the right at first, as shown in Figure 5-50 (d), and then flaps back to the rest position. In this experiment, the left flapping amplitude and right flapping amplitude are both 45° .

Also, the flapping frequencies in the four stages are the same. It is shown that after one flapping cycle, the robot fish moves forward 148 mm, i.e. 0.35 BL. From the previous oscillatory flapping propulsion model, when the robot fish flapping frequency is 1 Hz and the amplitude is 45° , the cruising speed of the robot fish is 170.4 mm/s. The prediction error is about 13%.

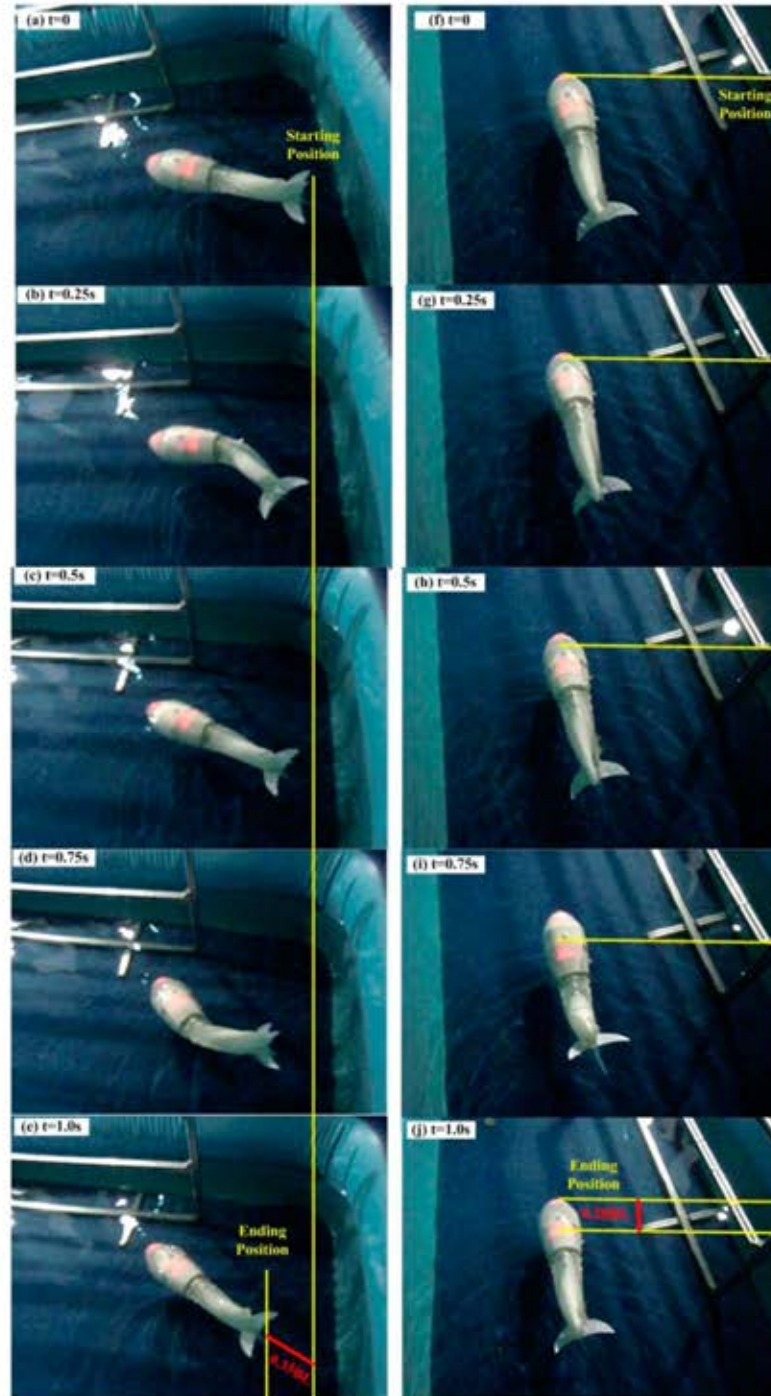


Figure 5-50 Experiment Results: (a)-(e) is the Flapping Cycle of Shark Form Swimming; (f)-(j) is the Flapping Cycle of Dolphin Form Swimming

2) Swimming in Dolphin Form

In this experiment, the vector propulsor flaps vertically like a dolphin. Fin 2 and fin 4 provide thrust in this mode, while fin 1 and fin 3 do not. The flapping frequency of the propulsor is $f = 1 \text{ Hz}$ and the flapping amplitude is 45° . The same as the former test, the robot fish is placed in still water. When the fish receives the command, it starts flapping vertically. In this mode, the front motor controls the vertical wire group and the back motor keeps still. Figure 5-50 (f)-(j) shows one flapping cycle. As shown in Figure 5-50 (f), the robot fish is in the rest position, waiting for the command. On receiving the command, the tail flaps downward first, as shown in Figure 5-50 (g). After reaching the lowest position, the tail flaps back as shown in Figure 5-50 (h). The tail does not stop in the rest position. It flaps until it reaches the highest position as shown in Figure 5-50 (i). Finally, the tail flaps back into the rest position and finishes a cycle. After a cycle, the tail does not stop. It keeps flapping and drives the robot fish forward. From the measurement, the distance the robot fish travels in one cycle is around 0.28 BL. The cruise prediction error for the dolphin form swimming is 30%.

By the experiment, it is shown that the robot fish can swim effectively in both shark form and dolphin form. This validates the vector flapping propulsor design method. In both cases, the propulsion velocities are similar. Moreover, the velocity is close to the model prediction. The error sources include prototyping error, modeling error, measuring error, etc. Although, the speed of the dolphin form is less than that of the shark form, it does not mean the shark form is superior. In fact, in the experiment, it is seen that in the dolphin form, the robot can employ gravity to glide. In these preliminary swimming tests the robot was not fully submerged in water. This limits the robot fish's 3D mobility performance.

5.5 Wire-Driven Robot Fish Performance and Discussion

5.5.1 Performance

Fish's cruising speed is related to the tail flapping frequency. John. J Videler summarized thirteen fish species' swimming speed with respect to the tail

flapping frequency [144]. From the data, a simple relationship can be found as: $U = 0.71f$, where U is in Body Length per second and f is in hertz.

As a comparison, Figure 5-51 shows the swimming speed of the four wire-driven robot fishes scaled to body length. In the figure, the red line is the speed of the real fish regressioned from Vidler's data, the squares represent the speed of the oscillatory wire-driven robot fish, the circles show the undulatory wire-driven robot fish, and the right-pointing triangle represent the speed of the vector propelled robot fish. From the figure, it is seen that the performance of the wire-driven fishes can catch up the real fish. At some frequencies, such as 0.5 Hz, the wire-driven robot fish can even surpass the real fish.

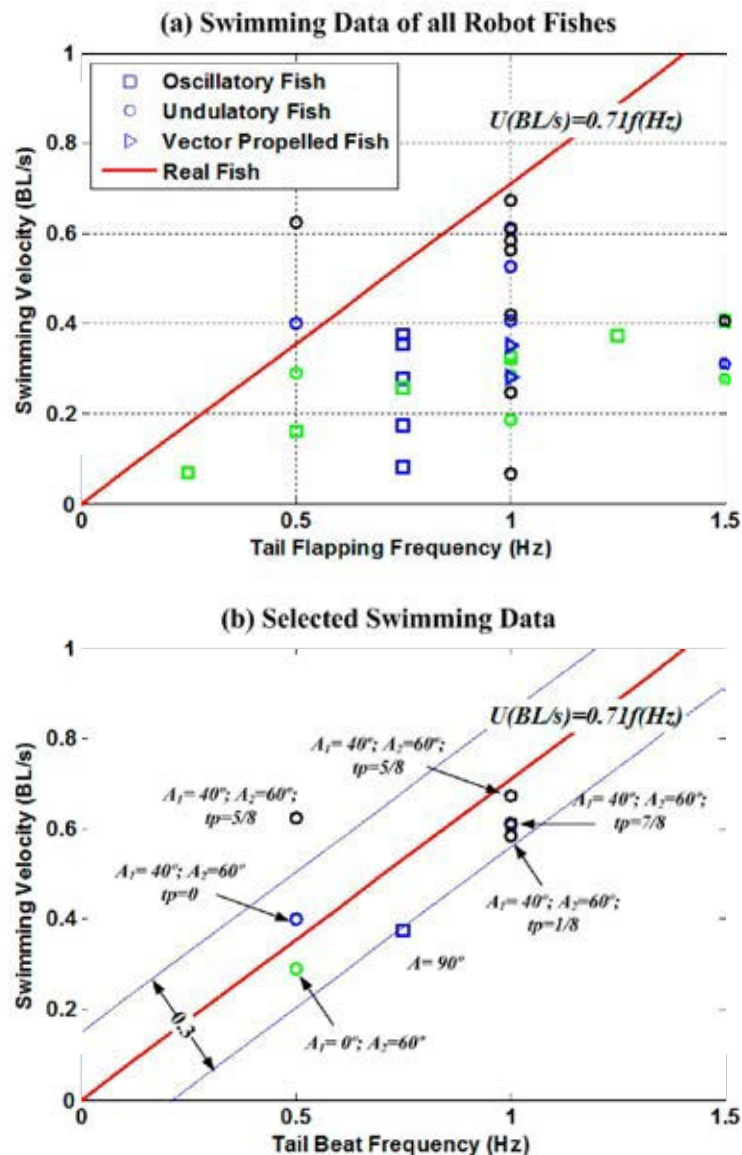


Figure 5-51 Wire-Driven Robot Fish Swimming Speed Scaled to Body Length

The Strouhal number describes how fast the tail is flapping relative to its forward speed. It is defined as $St=fA/U$, where f is the tail flapping frequency, A is the flapping amplitude and U is the cruising speed. Fishes typically flap with a Strouhal number close to 0.3, which is also shown by researchers where the optimal swimming efficiency locates [146].

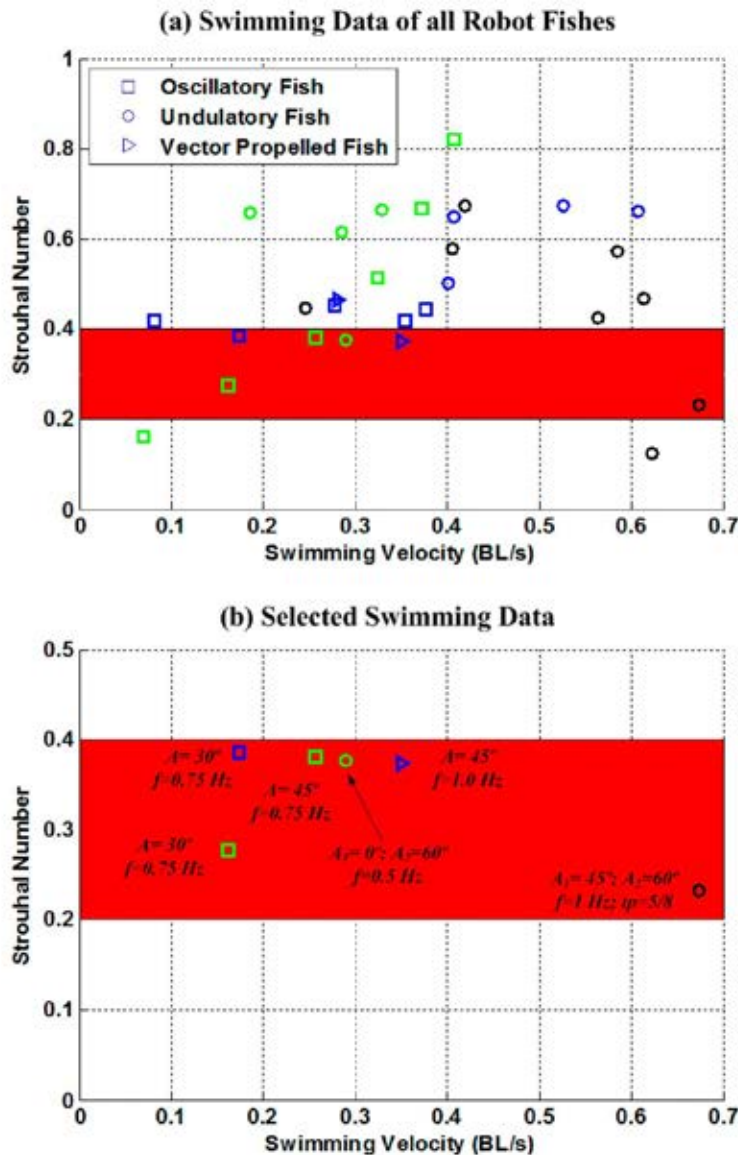


Figure 5-52 Strouhal Number of the Wire-Driven Robot Fish in Experiments

The Strouhal number of the wire-driven fishes are shown in Figure 5-52. In the figure, the blue squares show the Strouhal number of the oscillatory wire-driven robot fish in the frequency test. In the test, the tail bending amplitude of the robot fish is 45° . It is seen that, in this case the Strouhal number is around 0.4. The frequency has little effect on the Strouhal number. The green squares show the

Strouhal number of the oscillatory wire-driven robot fish in the bending amplitude test. The frequency is constantly 0.75Hz. From the figure, with greater tail bending amplitude, the larger is the Strouhal number. It is generally viewed when the Strouhal number is close to 0.3, eg. 0.2-0.4 as shown by the shadowed area, the fish swims with high efficiency. When the flapping amplitude is 30°, the Strouhal number is 0.276, which is close to 0.3. The black circles show the Strouhal number of the undulatory robot fish swimming in undulatory form. The blue circles show the Strouhal number of the Big-C motion, and the green circles show the Strouhal number of the Small-C motion. As shown in the figure, there is no significant difference between the Strouhal numbers of the undulatory form swimming and oscillatory form swimming. For undulatory form swimming, the phase difference between the two segments are important. The right-pointing triangles show the Strouhal number of the vector propelled robot fish, which is close to 0.4.

From all the data, it is seen that the all the three type of robot fishes can tune the flapping parameters and let the Strouhal number between 0.2 and 0.4. This again shows the wire-driven robot fishes can have high efficiency.

5.5.2 Discussion

Improve Speed and Efficiency

The wire-driven robot fishes show a good frequency-speed relationship. However, the maximum speed of the robot fishes is 0.333 m/s, which is too slow compared with current ships. To improve the speed, there are several ways. One is to use a more powerful actuator. The power of the servomotors used in the prototypes are around 1 W, and the speed of the motor is limited. As a result, the flapping frequency of the prototypes is limited a lot. Second, is change the transmission method. In the prototypes, the wires are connected to the motor shaft directly. The flapping is obtained by the back and forth rotation of the motor. During each cycle, there are two acceleration periods and two deceleration periods. This limits the flapping frequency as well as impairs the overall energy efficiency of the robot fish. An alternate method is to design a special gear box, which transmits the motor's unidirectional rotation to the wire's linear back and forth motion. This can improve the flapping frequency and the overall energy efficiency a lot. Third,

is to improve the fish body stability. Undulatory form is better than oscillatory form in terms of swimming speed and efficiency, the major reason is the fish swims steadily in undulatory form. To further improve the swimming stability, there are several ways. Such as, use dorsal fins, use another actuator to control the head swaying, use passive joints, etc. One simple way is to put the actuators close to the fish head. From the videos, the swaying center is the motor shaft. When the motor is put in front, the rotation center is also moved forward. As a result, the swaying portion of the head is smaller. The previous swaying generating drag force now generates propulsion. Last but not least, in the wire-driven mechanism, there are friction along the wires during the motion. As all the wires are guided by the pilot holes, the friction impairs the transmission efficiency. For wire transmission, if the friction is not considered, the efficiency is close to 95%. When the friction is considered, the transmission efficiency is lowered. In the robot fish prototype, lubricant grease is used to reduce the sideeffect of friction. The transmission efficiency is between 80% and 90%.

Elongated Body Theory Model Prediction

The elongated body theory predicts the robot fish's performance reasonably well, the prediction error is generally within 20%. However, there are also some cases with large error, such as when the undulatory wire-driven robot fish swims in oscillatory form, with $A_1=40^\circ$, $A_2=60^\circ$, $f=1.5$ Hz, the measured speed is 154 mm/s, and the prediction is 522.3 mm/s. The reason for this is manifold. At first, the elongated body theory assumes that the fish body is slim and the tail bending motion is not large. When the tail bending is very large, as in the above case, the model cannot predict the results well. Second, the flapping frequency of the tail is 1.5 Hz, which is high. For the WDM, when the elasticity coefficient of the elastic elements are small, the frequency response is slow. In the robot fish, plastic plates are used as the elastic element, which has low Young's Modulus. Hence, the elasticity coefficient is small. As a result, the propulsor cannot follow the input well. The actual bending amplitudes of the two segments are much smaller than the desired ones. This impairs the wire-driven robot fishes' performance a lot. To improve the frequency response, higher Young's Modulus material can be used. Third, the testing water pool is too small to measure the real cruising speed of the robot fish, especially when the swimming speed is fast. When the robot fish

swims across the pool, it is still accelerating. A larger swimming pool is needed to take more accurate measurement.

Wire-Driven Flapping Propulsor Scaling

In the thesis, the size of the robot fishes developed are in the range of 328 mm to 495 mm. The wire-driven mechanism is simple, and compact. The length of the backbone can range from millimeter scale to meter scale. The wire-driven flapping propulsors can scale down to millimeter scale, e.g. below 10 mm; it can also scale up to meter scale, e.g. a few meters. The factors limit scaling down include the motor size, and fabrication of the vertebrae. The major factor limits scaling up the wire-driven propulsor is the dynamic property. In the current design, kinematics model of WDM is used, statics and dynamics are not cooperated in. When the scale of the robot fish is large, the flapping forces are also large, which will affect the backbone curve a lot as indicated in the statics analysis. Hence, in scaling up the robot fish, the elastic element should be more rigid. For a large scale robot fish, the desired speed is often high, which means higher flapping frequency. The mode frequencies of the WDM backbone is low, typically a few Hertz. When the flapping frequency is large the backbone curve is not only controlled by the wires, but also the dynamic behavior of the backbone itself. In the design of high speed large scale robot fish, these factors should be taken into account. With proper adjustment, it is for sure that the WDM based robot fish can have good performance in millimeter scale and meter scale.

5.6 Summary

In this chapter, three types of robot fish, i.e. oscillatory wire-driven robot fish, undulatory wire-driven robot fish, and vector propelled wire-driven robot fish are presented. Four WDMs are used in the propulsor design, i.e. SPSI WDM, SPCT WDM, MPSP WDM and SSSI WDM. From the experiments, serpentine backbone WDM is more suitable for robot fish development. The wire-driven robot fish propulsors are compact, simple in structure, and easy to control. More importantly, they better resemble fish's swimming body curve than traditional multi-joint robot fish tails. More importantly, they have better efficiency. For the wire-driven propulsor, two segments are sufficient to imitate fish's undulatory swimming. By rearranging the wire configuration, a robot fish with vector

propulsion can be developed. It provides propulsion in arbitrary directions. Furthermore, it can perform more complicated spatial flapping motions, such as circular flapping and ∞ -shaped flapping. The vector propulsion can improve fish's maneuverability a lot.

Chapter 6 Application II - Wire-Driven LTAV – Flying Octopus

In this chapter, a Lighter-Than-Air-Vehicle (LTAV) is described. It has four independently actuated wire-driven flapping wings, which enable the LTAV to move freely in 3D space. The SPCT WDM is used in wing design.

6.1 Introduction

Before fixed-wing airplanes, Lighter-Than-Air-Vehicles (LTAVs) were the sole method of aerial transportation. Airships were the most common LTAVs. The first airship was built by a French engineer Henri Giffard in 1852 [147]. Since then, airships have been studied and built continuously. Compared with fixed wing airplanes, the LTAV has advantages in several aspects. First, it is energy efficient. LTAVs utilize buoyancy to suspend itself in the air; its payload can be very large without requiring much energy. Second, it can hover in the air and the speed can be very slow. This makes LTAVs very suitable for sightseeing, aerial photography, aerial monitoring, etc. The disadvantage is that the size of the LTAV is large, the speed is slow, and the motion is affected by wind. However, these disadvantages are not critical on some occasions, such as indoor entertainment. For indoor applications, the weight of a LTAV is typically small. As a result the size is moderate. Meanwhile, from the safety point of view, the low speed is beneficial for indoor applications.

Traditional LTAVs have streamlined bodies and are propelled by screw propellers, such as the Graf Zeppelins LZ127, LZ129 [148]. In recent decades, various shaped LTAVs have been developed. The Flying Yachts Inc. built spherical airships [149]; a British company, named Thermo Skyship built lenticular airships [150]; Advanced Technologies Group Ltd. built a double hulled airship [151]; an airship with fixed wings was also proposed [150]. For indoor applications, the LTAV shape is even more diversified, such as being animal-like, cartoon-like [152], etc. Compared with the diversity in shape, the actuation is less changed. In addition to traditional screw propellers, one important way of LTAV actuation is flapping wings. Such as the Airjelly [153]

and Air swimmers [152], they flap their wings or tails to generate propulsion. The moving direction is controlled by other systems, such as the Airjelly using a pendulum, and the air swimmers using a moving balance weight.

Flapping is widely accepted as a highly efficient way to generate thrust and propulsion. Birds and fish have adopted flapping to move around through millions years of evolution. This gives researchers inspiration. From the previous chapter, it is shown that the WDM is well suited for propulsion in water, both oscillatory form swimming and undulatory form swimming. How about in the air? This motivated the development of a LTAV with wire-driven flapping wings. It is called Flying Octopus for one reason: its outlook is similar to that of an octopus; another reason is the wire configuration in the WDM was inspired by the octopus arm muscle arrangement.

6.2 Flying Octopus Design

An octopus, from its appearance, has a round head and eight arms, as suggested by its name. The Flying Octopus also has two parts: one is the body and the other is the flapping wings. The designs are as shown below.

6.2.1 Flying Octopus Body Design

The Flying Octopus body comprises a round head and midsection. Flapping wings are connected to the midsection via the wing base adapter. The designed Flying Octopus is as shown in Figure 6-1.

For indoor entertainment, the size as well as the weight of the Flying Octopus is confined. The speed of the LTAV is not a key element. A sphere has the maximum volume to surface ratio. Hence, a spherical balloon was chosen as the head. The diameter of the balloon is 1.5m, with a volume of 1.767m^3 . When filled with helium, ideally, the buoyancy it can provide is 19.3 N, i.e., it can support a 2.02 kg mass.

The balloon is connected to the midsection on the top, while the flapping wings are connected at the bottom. The actuators, power supply, and control systems are all inside the midsection. The midsection structure should be lightweight and strong enough. For this reason a carbon plate was chosen. The design is shown in Figure 6-1 (b). The evenly distributed square slot is used to mount servo motors

and the exterior circular holes are used to hold the wing base adapter. From the figure, the structure can hold up to eight motors and has 16 ports for wings. In the figure, only four wings and motors are shown. In fact, four wings are more than enough for 3D motion. Each wing is actuated by a servo motor via the WDM. The propulsion magnitude and direction is controlled by controlling the wing flapping motion. The other holes in the carbon plate are used to fix the balloon, power supply, and control system. They also help lighten the overall weight of the Flying Octopus. The carbon plate is circumscribed by a fence-like covering, as shown in Figure 6-1 (a). It not only covers the midsection but also helps to secure the balloon. The top of the fence presses against the balloon. This increases the contact area between the balloon and the midsection. As a result, there is no relative motion between the balloon and midsection.

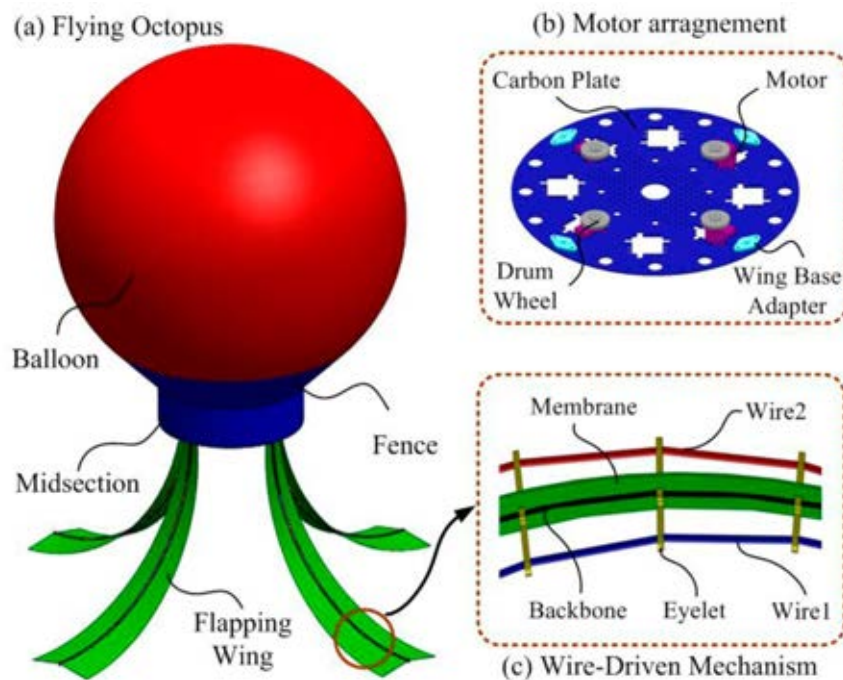


Figure 6-1 Flying Octopus Design

6.2.2 Wire-Driven Flapping Wing Design

The flapping wings are the critical part of the Flying Octopus. They not only provide thrust but also control the motion of the Flying Octopus. The design requirements include: a) large flapping motion; b) light weight; c) easy to control. To meet all these requirements, the SPCT WDM is used.

The designed flapping wing is as shown in Figure 6-2. It has four parts: membrane, backbone, eyelets, and wires. The overall length of the wing is 680 mm. The membrane defines the profile of the wing and also serves as a secondary backbone. In this design, the membrane width decreases from the wing base to the distal tip. The end of the wing is a lunate flipper. The backbone connects to the Flying Octopus body and supports the wing structure. It is slim and has a rectangular cross-section. The width is 5 mm and the thickness is 0.5 mm. The bending selectivity is 100. Hence the backbone is inclined to bend in the thickness direction.

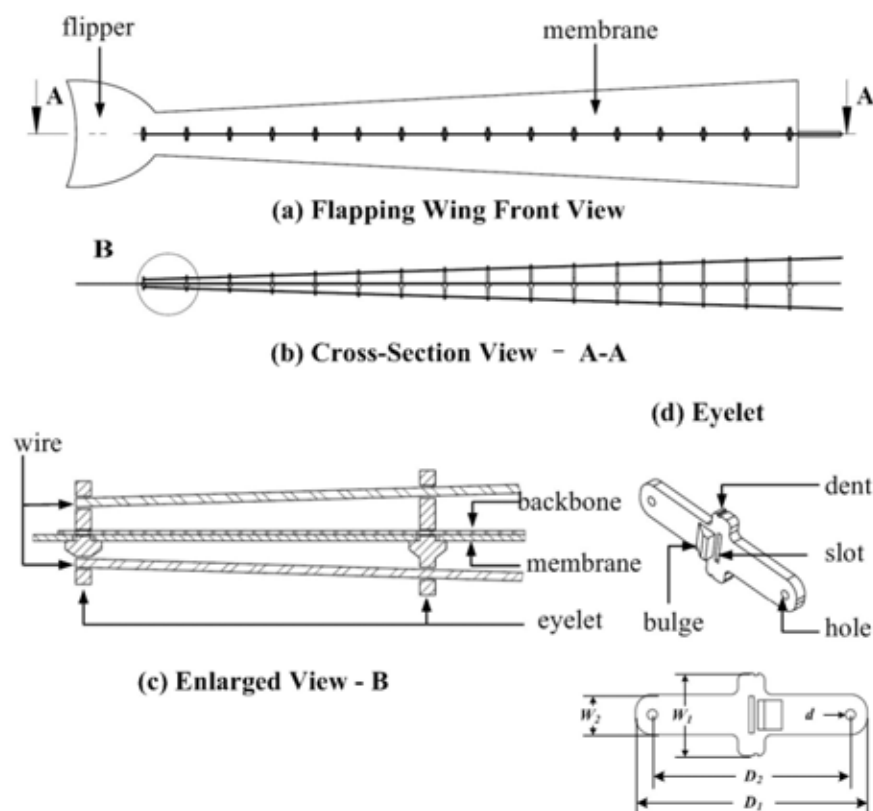


Figure 6-2 Wire-Driven Flapping Wing Design

As shown in Figure 6-2 (b), the wires are symmetrically placed on both sides of the backbone and there is an inclined angle between the backbone and the wires. The eyelets are used to guide the wires. In this design, all the eyelets in the flapping wing are similar in structure. They have the same features: bulge, slot, hole, dent, and rib, as shown in Figure 6-2 (d). From the flipper to the wing base the eyelet rib length increases gradually. The growth rate is determined by the incline angle. The key parameters of the eyelet are shown in Table 6-1. There are

16 eyelets in each wing. All the eyelets are evenly distributed along the backbone. The eyelet locations (distance from the wing base to the eyelet) are also shown in Table 6-1. As shown in Figure 6-2 (c), the connection of the four parts is similar to the method used in the continuous oscillatory wire-driven robot fish. From the enlarged cross-section view, the eyelets are placed in the rectangular slots of the membrane and are positioned by the dent on the eyelet. There is a bulge and a slot around the center of each eyelet. The bulge of the eyelet presses against the membrane, leaving the eyelet slot on the other side of the membrane. The backbone inserts into the slots of the eyelets and locks the eyelets like a latch. This makes the connection of the three parts very simple and solid. After the connection, the wires are passed through the eyelets' hole and fastened to the distal end (eyelet #16).

Table 6-1 Eyelet Parameters

Num.	D1 (mm)	D2 (mm)	W1(mm)	W2 (mm)	d (mm)	Location (mm)
1	82	77	12.5	6	1.5	10
2	78.5	73.5	12.5	6	1.5	50
3	75	70	12.5	6	1.5	90
4	71.5	66.5	12.5	6	1.5	130
5	68	63	12.5	6	1.5	170
6	64.5	59.5	12.5	6	1.5	210
7	61	56	12.5	6	1.5	250
8	57.5	52.5	12.5	6	1.5	290
9	54	49	12.5	6	1.5	330
10	50.5	45.5	12.5	6	1.5	370
11	47	42	12.5	6	1.5	410
12	43.5	38.5	12.5	6	1.5	450
13	40	35	12.5	6	1.5	490
14	36.5	31.5	12.5	6	1.5	530
15	33	28	12.5	6	1.5	570
16	29.5	24.5	12.5	6	1.5	610

6.3 Flying Octopus Motion Control

The propulsion model of the flapping wings is crucial for Flying Octopus motion control. In this section, the propulsion model is derived and 3D motion control strategy is presented.

6.3.1 Propulsion Model

As shown in Figure 6-3, the forces acted on the Flying Octopus include gravity G , drag force F_d , buoyancy F_b , thrust F_t , and lateral force F_l . By delicate adjustment, gravity and buoyancy balance each other, and the Flying Octopus can suspend itself in the air. The thrust and lateral forces are generated by the flapping wings. They are used to control the Flying Octopus' moving direction and velocity. Drag force is opposite to the moving direction.

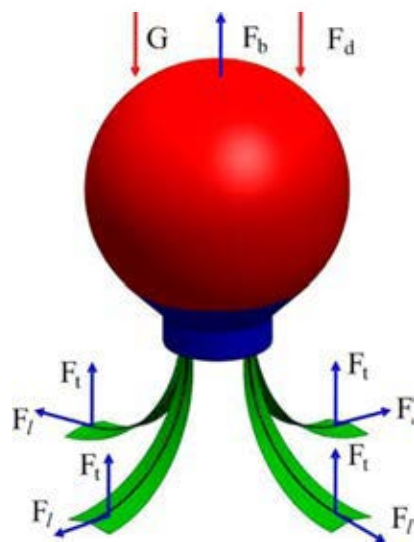


Figure 6-3 Flying Octopus Force Analysis

Table 6-2 Forces on Flying Octopus

Gravity:	$G = mg$
Drag force:	$F_d = 0.5\rho_{air}C_dAv^2$
Buoyancy:	$F_b = \rho_{air}Vg$
Thrust per wing:	$F_t = \int_0^{\ominus} [0.5\rho_{air}C_Lu^2 \sin(\alpha)] dS$
Lateral force per wing:	$F_l = \int_0^{\ominus} [0.5\rho_{air}C_Lu^2 \cos(\alpha)] dS$

The external forces acting on the Flying Octopus are as shown in Table 6-2. In the table, m is the overall mass of the Flying Octopus, g is gravity constant, ρ_{air} is the density of air, C_d is the drag coefficient, A is the projected area in the moving direction, v is the velocity of the Flying Octopus, V is the overall volume, C_L is the thrust coefficient, u is the velocity of the points on the membrane, α is the complement angle of u and v , dS is the infinitesimal area of the membrane at angle α , and Θ is the wing flapping amplitude.

As shown in Figure 6-4, the general coordinate q is set as the arc length from the wing base to the point, along the backbone. Based on the constant curvature assumption and WDM kinematics, the velocity of the point is:

$$u = \dot{\Theta} \cdot \frac{L}{\Theta^2} \sqrt{\alpha^2 - 2[\alpha \sin(\alpha) + \cos(\alpha) - 1]} \quad (6-1)$$

where $\alpha = q \cdot \Theta / L$ and L is the total length of the wing.

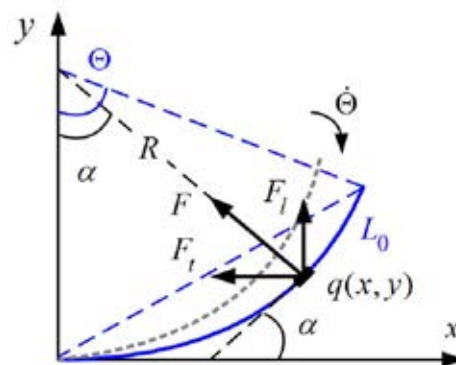


Figure 6-4 Thrust Force Illustration

6.3.2 Motion Control Strategy

The motion of the Flying Octopus is controlled by the four independently actuated flapping wings. The control scheme is similar to that used in the oscillatory wire-driven robot fish: the command is sent to the MCU by remote control. On receiving the command, the MCU generates four PWM signals, which control the positions and velocities of the four motors. The wire-driven mechanism transmits the motor's rotation to the wings' flapping. Thrust and lateral forces are generated to drive the Flying Octopus body, and control its motion.

The flapping cycle is also similar to that in the oscillatory wire-driven robot fish. A complete cycle includes four stages. Stage I: flap from rest position to the outermost position. Stage II: flap from the outermost position to the rest position. Stage III: flap from rest position to the innermost position. Stage IV: flap from the innermost position to rest position.

The Flying Octopus' motion is controlled by the four wings' flapping cycles, i.e., flapping amplitudes and the velocities in the four stages. The wings are divided into two groups: X wing group and Y wing group. There are five basic modes of motions: a) flying upward, b) flying downward, c) hovering, d) flying in the X direction and e) flying in the Y direction. The wing motions in the five motion modes are described below:

a) Flying upward: Four wings flap synchronously. Before flapping, all the wings return to the rest position. The outer bending amplitude is larger than the inner amplitude. In stages I and III, the velocity is slower than that of stages II and IV. As the four wings flap identically, the lateral forces cancel out each other. Due to the inward flapping velocity being larger than the outward flapping velocity, from the thrust representation in Table 6-2, the thrust in stages I and III is smaller than that in stages II and IV. The net thrust in one flapping cycle is upward. Hence, the Flying Octopus ascends.

b) Flying downward: The scheme is similar to flying upward. The difference is that in this mode the inward flapping velocity is smaller than outward flapping velocity. As a result, the net thrust in one flapping cycle is downward.

c) Hovering: Ideally, the Flying Octopus hovers in the air when the flapping wings keep still. However, disturbances exist. An aerometer could be used to detect the status of the Flying Octopus. Upward or downward flapping schemes are used when the flapping octopus is dropping or ascending.













d) Flying in the X direction: in this motion mode, a lateral force is needed. The four wings are grouped into X and Y pairs. For traversing in the $-X$ direction, the X wings flap toward the $-X$ direction in stages I and IV slowly and flap toward the $+X$ direction with a larger velocity in stages II and III. The generated forces are a lateral force in the $-X$ direction and an upward thrust. A downward force is generated by the other two wings. Their motions are the same as that in the

downward scheme, i.e. fast bending outward and slowly inward. In one flapping cycle the net force is in the $-X$ direction. This drives the Flying Octopus to glide in the $-X$ direction. By switching the $+X$ and $-X$ flapping velocities of the X wing pair, the Flying Octopus will move in the $+X$ direction.

e) Flying in the Y direction: The four wings are orthogonally placed on the Flying Octopus. By switching the motions of the X and Y wing pairs, the Flying Octopus will move in the Y direction.

The forces generated by the two wing groups in the five basic motions are as shown in Table 6-3. In the table, the arrow shows the force direction and the thickness of the arrow represents the magnitude of the force. With the five basic motion modes, the Flying Octopus can move in 3D space freely, since arbitrary motion can be decomposed into these basic motions.

Table 6-3 Five Basic Motions

	Forces by X wing group	Forces by Y wing group
Fly Upward		
Fly Downward		
Hovering		
Fly in the X direction	 	
Fly in the Y direction		 

6.3.3 Motion Simulation

From the propulsion analysis, it is convenient to develop a Matlab program to simulate the Flying Octopus' motion. Figure 6-5 shows the user interface of the program as well as the simulation example.

The left upper figure shows the instantaneous wing motion; the lower figure shows the simulation results. The program can simulate the backbone bending angle, bending velocity, wing distal end linear velocity, thrust force, resistance force, resultant force, Flying Octopus velocity, etc. Simulation conditions can be input from the control panel on the right-hand side. The conditions include the

Flying Octopus' structure parameters and simulation parameters. Structures are wing length, balloon radius, midsection radius, wire pair distance, Flying Octopus overall mass, air density, and drag coefficient. Simulation parameters are wing flapping velocity profile (cosine, rectangular wave, etc.), flapping amplitude, flapping frequency, and simulation periods.

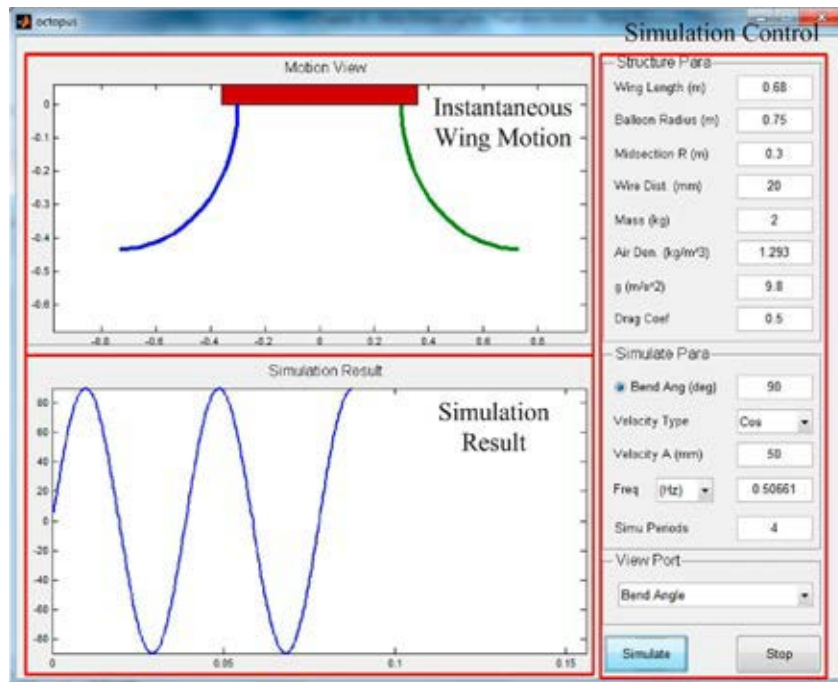


Figure 6-5 Flying Octopus Motion Simulation

Figure 6-6 shows one simulation example. In the simulation, the Flying Octopus is 1.592 kg; the wing length is 0.68 m; balloon radius is 0.75 mm; drag coefficient is 0.5. The wings' flapping amplitudes are all 60° and flapping frequency is 1 Hz. The flapping speeds in the four stages are the same.

Figure 6-6(a) shows the Flying Octopus velocity. The horizontal axis represents the time (s), and vertical axis is the velocity (m/s). From the results, in the first few flapping cycles the Flying Octopus' velocity increases with fluctuation. After 25 seconds, the speed of the Flying Octopus becomes stable. However, the velocity remains fluctuating with the wing flapping motion. The average stabilized velocity of the Flying Octopus is 0.215 m/s. Figure 6-6 (b) shows the propulsion generated by the flapping wings. It is shown that during one flapping cycle the instantaneous propulsion is between -0.12 N and 0.21 N. The average propulsion is positive, i.e. upward. Figure 6-6 (c) shows the drag force. It has a

similar trend as that of the Flying Octopus velocity, which is straightforward. Figure 6-6 (d) shows the resultant force acting on the Flying Octopus. The blue curve is the instantaneous resultant force, and the red curve is the historical average. From the result, the averaged resultant force decreases after a few seconds. This is also revealed by the velocity increase rate.

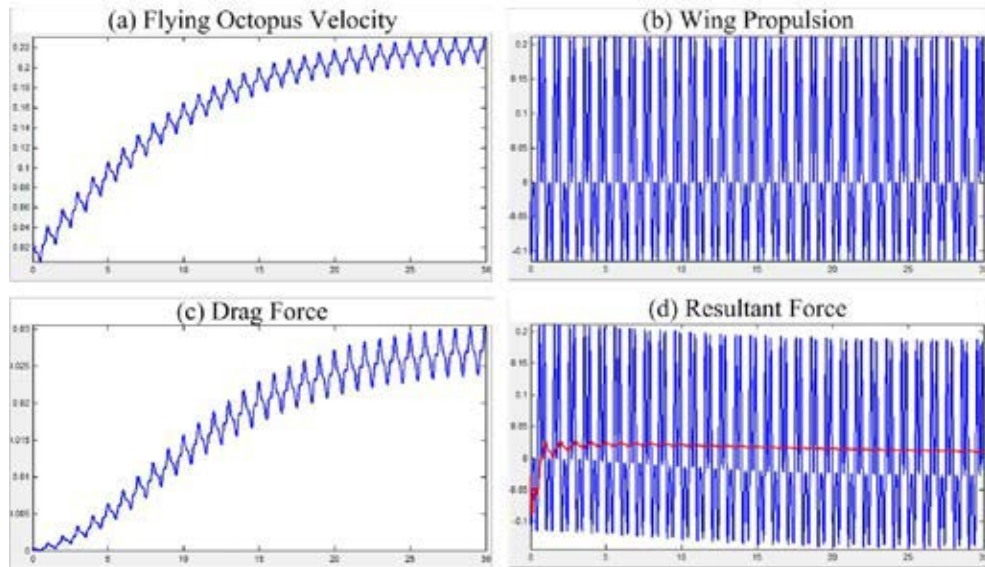


Figure 6-6 Simulation Results in 30 Flapping Cycles

6.4 Prototype and Indoor Experiments

6.4.1 Flying Octopus Prototype

To validate the design, a Flying Octopus prototype was built and tested indoor. As shown in Figure 6-7, a polyethylene (PE) balloon with 1.5 m diameter is used as the round head. The backbone in the wing is a carbon beam. Its width is 5 mm, and the thickness is 0.5 mm. The eyelets are fabricated by rapid prototyping, and the material used is ABS plastic. The membrane is made from a 0.5 mm thick ABS plate. The overall length of the flapping wing is 680 mm. The four wings are evenly distributed on the middle section. Steel wires covered with plastic sheath are used in the prototype. Four servo motors pull the wires by a rotator under the control of a commercial MCU. Each wing weighs 39 g. The total mass of the Flying Octopus is 1592 g.

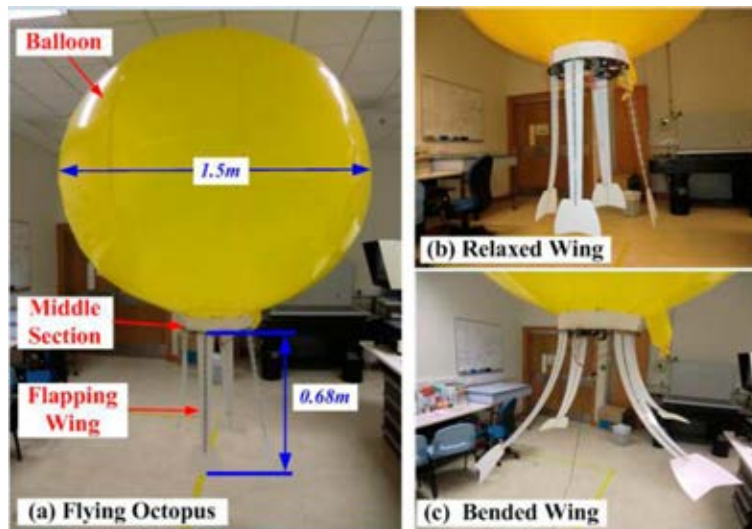


Figure 6-7 Flying Octopus Prototype

Table 6-4 Flying Octopus Bill of Materials

Component	Description	Mass (g)	Number
Balloon	1.5 m Diameter	568	1
Midsection Plate	Carbon	302	1
Motor	Towerpro MG 995	59	4
Drum Wheel	ABS Plastic	7	4
Wing Membrane	ABS Plastic	39	4
Battery Box	4 AAA battery	69	1
Control Board	MCU Atmega 128	156	1
Fence	ABS Plastic	38	1
Wire	Steel	0.25	4
Others	Bolts, etc.	38	-

The details of the Flying Octopus are shown in Table 6-4. The weight of the Flying Octopus is delicately tuned to balance the buoyancy. As a result, the Flying Octopus is able to hover in the air when all the wings are at rest. At resting, the four wings are relaxed as shown in Figure 6-7 (b). By pulling the outer wires, the wings bend outward as shown in Figure 6-7 (c). The maximum bending angle is constrained by the stiffness of the backbone. For this prototype, the maximum

bending angle is around 180° . Over bending will exert a large moment on the backbone and twist the flapping wing.

6.4.2 Indoor Experiments

Indoor flying tests were carried out. To eliminate the wind disturbance, all the air conditioners and exhaust fans were turned off. The weight of the Flying Octopus was adjusted to equal the buoyancy. From the control scheme, it is seen that in the five motion modes, the wings' motion in flying upward, flying downward and hovering are similar. Only the flapping speed in the four stages are different. Flying in the X direction and flying in the Y direction are similar. Therefore, in the experiment, two fundamental modes, i.e. flying upward and flying in the X direction are tested.

Experiment 1 – Flying upward

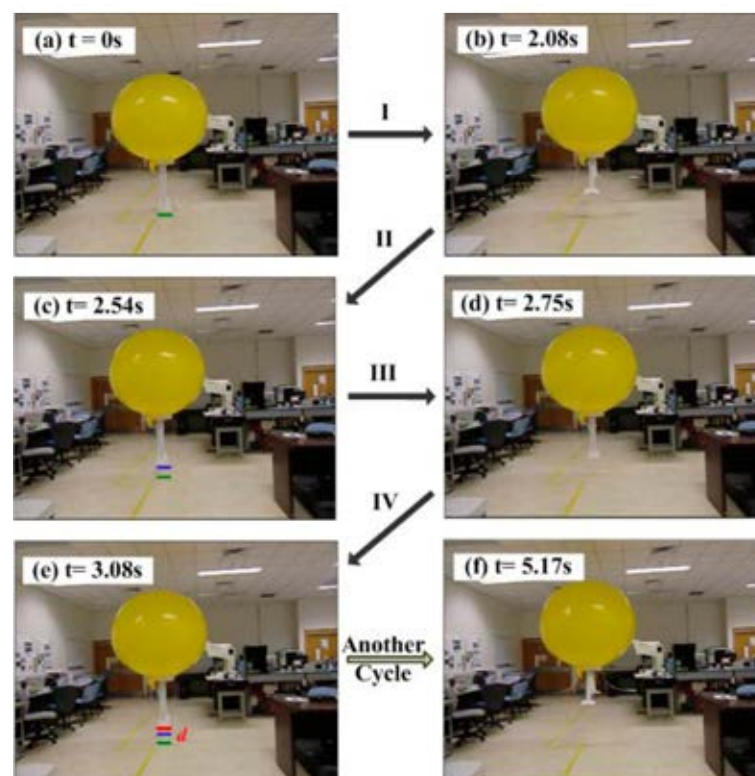


Figure 6-8 Flapping Cycle of Flying Upward

Figure 6-8 shows the four stages of the wings' flapping cycle in the flying upward mode. At the beginning, the Flying Octopus stays on the ground with all the four wings relaxed. Next, the four wings are flexed by the WDM slowly. When they have reached the outermost position, the four wings flaps back into the rest

position. From the figure, it is seen that the time for stage I takes 2.08 s, while it only takes 0.46 s for stage II. After reaching the rest position, the wings continue to bend inward. When they have reached the innermost position, the four wings flap back into the rest position and finish one flapping cycle.

The Flying Octopus flies upward 1 m with five flapping cycles. The distance it travels in one cycle is about 20.0 cm, and the average speed is about 6.5 cm/s. As the traveling distance is short, the Flying Octopus does not reach its maximum velocity. This is consistent with the simulation result as shown in Figure 6-9. From the simulation, it is shown that during the first five flapping cycles, the velocity increases cycle by cycle. The average speed is 5.3 cm/s.

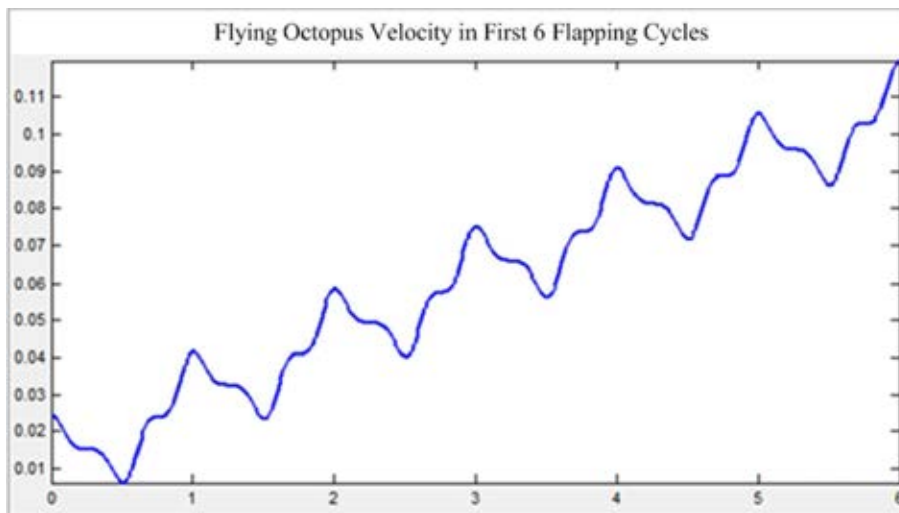


Figure 6-9 Flying Octopus Velocity in the First Six Flapping Cycles

Experiment 2 – Flying in the horizontal plane

Figure 6-10 shows the wings' flapping cycle for the mode of flying in the X direction. At the beginning, all the wings are relaxed. The wings in the X group flap in the $-X$ direction, and the wings in the Y group flap inward quickly in stage I. In stage II and stage III, the X wings flap in the $+X$ direction and the Y wings flap outward slowly. After that, the X wings flap in the $-X$ direction and the Y wings flap inward quickly in stage IV. The distance traveled in one flapping cycle is around 25 cm, and the average traversing speed is 7.4 cm/s.

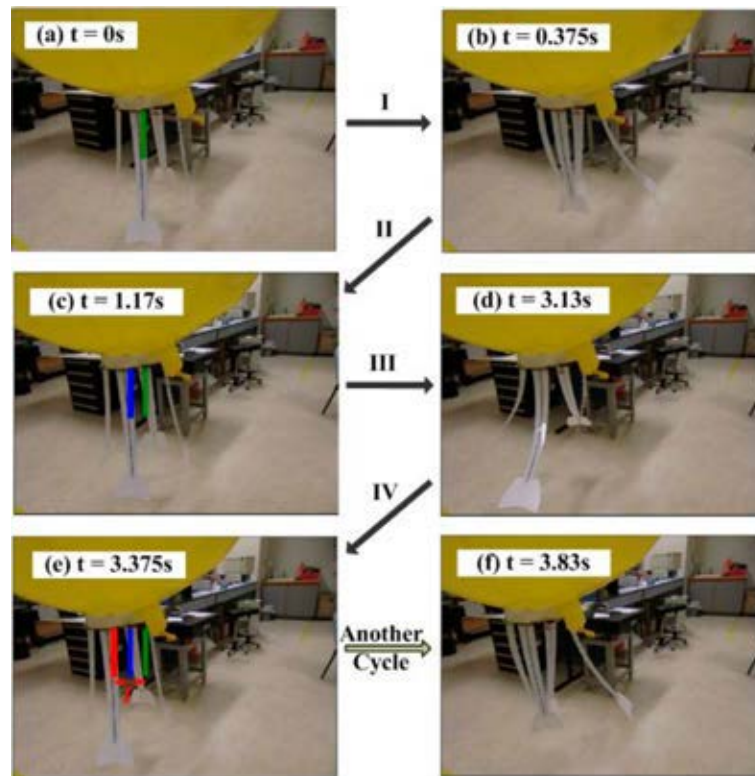


Figure 6-10 Flapping Cycle of Flying in X Direction

6.4.3 Discussion

Due to the size limitation of the room, performance of the Flying Octopus was not fully exhibited by the experiments. Such as in the ascending test, the LTAV reached the ceiling after five flapping cycles. The maximum velocity should be greater than 6.5 cm/s. It is also expected that the Flying Octopus can move faster after wing shape optimization and flapping motion parameters optimization.

The flapping wing is a new application of the WDM besides robot fish and flexible manipulator. The wire-driven flapping wing is compact, lightweight, and easy to control. Also, the flapping range is quite large. In this prototype, the maximum flapping angle is over 180° . However, it also has drawbacks. Due to the wing's low rigidity, flapping frequency is limited and the wing is inclined to twist at large bending angles. To avoid twisting, the length of the backbone and membrane should not be too large. To increase the thrust, a small web at the distal end is beneficial.

The wire tension in flapping is affected much by the mass distribution and wire configuration of the wing. The ratio between the wire tension and force acting on

the wing is equivalent to the backbone length over the first eyelet's rib length. This is usually large, e.g. in this design the ratio is 30. Hence, it is beneficial to reduce the wing's weight, especially the wing tip's weight. As a result, the tilted wire configuration is better than parallel wires. The reason is that with shorter rib length the eyelet weight is reduced while keeping the same ratio.

6.5 Summary

This chapter introduces a novel LTAV named Flying Octopus. It is actuated by four wire-driven flapping wings. The wing design follows the SPCT WDM. This made the wings compact, light-weight, and easy to control. Also, the flapping range is quite large. The maximum flapping angle of this prototype can exceed 180° . With the four independently actuated wings, the Flying Octopus can move in 3D space freely. The basic motions are flying upward, flying downward, hovering, flying in the X direction, and flying in the Y direction.

Chapter 7 Conclusions and Future Work

In this thesis, the wire-driven mechanism is studied systematically and several applications are developed, especially in relation to highly efficient propulsion in water. The contributions of this thesis are:

1) Designed the biomimetic Wire-Driven Mechanism (WDM). The WDM is inspired from snake skeleton and octopus arm muscle arrangement. It contains a flexible backbone and several pairs of controlling wires. Its features include: large flexibility, highly under-actuated, leverage effect, long range force and motion transmission. The backbone structure is simple and compact. It can bend largely in all directions under the wires' actuation. Hence, it is well suited for working in confined spaces, such as minimally invasive surgery, engineering nondestructive inspection, disaster relief, etc. Also, it is well suited to flapping propulsion, both in water and in air.

2) Developed the kinematics model for both serpentine WDM and continuum WDM. A generalized model with/without constraint was also proposed. The kinematic model was derived from geometry analysis under the constant curvature assumption. Several WDM based manipulators were built to test the kinematic model. The results show that the average prediction error is less than 3.6%. Workspaces of the single segment WDM and multi-segment WDM were both developed. With the constrained kinematic model, a novel idea of employing obstacles or actively deploying constraints to improve the workspace was proposed. The idea was also validated by a single segment WDM prototype. The results show that with the constraints, the workspace can be expanded a lot, from a curve to a large area.

3) Developed the static and dynamic models for both serpentine WDM and continuum WDM. The Newton-Euler method and Lagrange method were used to develop the static model and dynamic model of serpentine WDM respectively. The static model and dynamic model of the continuum WDM were developed based on the nonlinear Euler-Bernoulli beam theory and extended Hamilton's principle. The simulation results show that only when the WDM is subject to pure

moment is the deformed shape a circular arc. In other loading conditions, the constant curvature assumption is invalid.

4) Designed several novel fish-like flapping propulsors using the WDM. Compared with existing robot fish tails, the WDM propulsors use circular arcs to fit the fish's swimming body curve. They employ less actuators but can better resemble fish motion, such as only one motor being used in the oscillatory flapping propulsor and only two motors in the undulatory flapping propulsor. Traditionally, three to six motors are used for undulatory robotic fish tail. Besides the traditional motions, brand-new flapping motions can be generated easily, such as the wire-driven vector propulsor. It can flap in any arbitrary plane or flap spatially, such as the shark form flapping, dolphin form flapping, circling, or ∞ -shaped flapping. This enables the propulsor to provide propulsion in any arbitrary direction. Hence, improving the robot fish's maneuverability greatly.

5) Developed the propulsion model of the WDM flapping propulsors. The models were developed based on the WDM kinematic model and Lighthill's elongated body theory. The model is simple but can predict the propulsor performance well, including the cruising speed, and propulsion efficiency. For example, the average prediction error for the oscillatory serpentine robot fish is 16.73%.

6) Build several novel robot fishes using the wire-driven flapping propulsors, including a serpentine oscillatory robot fish, a continuum oscillatory robot fish, a serpentine undulatory robot fish, and a vector propelled robot fish. The performances of these robot fishes are good. For example, cruising speed of the serpentine undulatory robot fish can reach $0.67BL/s$; the maximum Froude efficiency of the serpentine undulatory robot fish is 92.85% and the average tested efficiency is 75.6%, which is far better than current screw propellers (typically below 45%); the turning radius of the serpentine oscillatory robot fish is $0.24BL$ and the turning speed is $51.4^\circ/s$; the vector propelled robot fish can mimic both shark swim and dolphin swim effectively. Factors affect the swimming performance were also studied. These include the flapping amplitude, flapping frequency, and phase lag. For example, with the increasing of flapping amplitude and frequency, the cruising speed increased at first, and after reaching an optimal value, the cruising speed decreased; when the phase lag between the two segments is $0.5T$, the cruising speed is the minimum.

7) Designed and built a novel indoor LTAV, named Flying Octopus using the WDM flapping propulsor. It suspends in the air using a helium balloon. The actuation is provided by four independently actuated continuous oscillatory flapping wings. These wings are light-weight, simple in structure and can bend over 180° . With the four wings' propulsion, the Flying Octopus can ascend, descend, transverse, and hover in the air. This shows another important application of the WDM.

The WDM and similar mechanisms are commonly used in continuum robots and other flexible robots. Besides the topics studied in this thesis, there are a lot more. Future research contains three branches.

1) Design optimization. In the current design, the joint rotations are all the same, such as for the serpentine WDM, the joint gap distances are identical, and for the continuum WDM, the eyelets' spacing is the same. Therefore the ideal deformed backbone is a circular arc. In some applications, different joint rotations may be beneficial. Also, the lengths of each segment affect the workspace. These issues generated the design's optimization.

2) Theoretical modeling and control. In the current kinematic model, the constant curvature assumption is used. As indicated by the static model, when external loading is not a pure moment, this assumption is invalid. In the future, a more accurate model integrating the static model needs to be developed. Also, in this thesis, the dynamic model was developed. However, the dynamic behavior of the WDMs was not fully studied and validated by experiment. In the future, a more detailed study is needed. The WDM is highly underactuated and the control is highly nonlinear at large bending amplitudes. A computational efficient and robust control model is critical for developing elegant applications of the WDM.

3) Application exploration. The WDM is a special mechanism with many useful features. There are a lot of applications where compliance, large deformation, leverage effect, long-range motion transmission, etc. are required. The flexible manipulator, robot fish and Flying Octopus are just one of many. More extensive applications could be identified, such as space robots, welding robots, medical robots, bio-inspired robots, etc.

Appendix A - Publication Record

Journal Publications

- [1] Baofeng Liao, **Li Zheng***, Du Ruxu, “Robot Fish with a Novel Biomimetic Wire-driven Flapping Propulsor”, *Advanced Robotics – special issue on biologically inspired robotics*, 2013 (Invited paper, Under Review)
- [2] **Li Zheng***, Du Ruxu, “Design and Analysis of a Biomimetic Wire-Driven Multi-Section Flexible Robot”, *International Journal of Advanced Robotic Systems*, pp. 209-220, Vol. 10, April, 2013
- [3] **Li Zheng***, Du Ruxu, Zhang Yi, and Li Hua, “Robot Fish with Novel Wire-Driven Continuous Flapping Propulsor”, *Applied Mechanics and Materials*, pp. 510-514, Vol. 300-301, 2013
- [4] **Li Zheng***, Du Ruxu, “Design and implementation of a biomimetic wire-driven underactuated serpentine manipulator”, *Transactions on Control and Mechanical Systems (TCMS)*, pp. 1-9, Vol. 1, No. 6, October, 2012
- [5] Du Ruxu*, **Li Zheng**, “Comparison of Manufacturing Development between China and the United States in the Last Century”, *Reformation & Strategy*, pp.161-165, Vol. 27, No.5, 2011

Conference Publications

- [1] **Li Zheng***, Zhong Yong, Du Ruxu, “Design and Implementation of a Novel Vector Propulsion Robot Fish”, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013)*, (Accepted)
- [2] Du Ruxu*, **Li Zheng**, “Under-Actuated Flexible Wire-Driven Mechanism and It's Application in Underwater Propulsion”, the 10th Cross Strait Advanced Manufacturing Technology Forum, Taiwan, Jun. 15-19, 2013.
- [3] Baofeng Liao, **Li Zheng***, Du Ruxu, “Robot Tadpole with a Novel Biomimetic Wire-driven Propulsor”, 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO 2012), Guangzhou, China, Dec. 11-14, 2012. **(Best Paper Finalists)**

[4] **Li Zheng***, Du Ruxu, Zhang Yi, and Li Hua, “Robot Fish with Novel Wire-Driven Continuous Flapping Propulsor”, The 2nd International Conference on Mechatronics and Applied Mechanics (ICMAM), Hong Kong, Dec. 6-7, 2012

[5] **Li Zheng***, Gao Wenqi, Du Ruxu, Liao Baofeng, “Design and Analysis of a Wire-Driven Robot Tadpole”, International Mechanical Engineering Congress & Exposition (IMECE 2012), Houston, Texas, USA, Nov. 9-15, 2012.

[6] **Li Zheng***, Du Ruxu, Yao Yupei, “Flying Octopus – A LTAV with Wire-Driven Flapping Wings”, International Mechanical Engineering Congress & Exposition (IMECE 2012), Houston, Texas, USA, Nov. 9-15, 2012.

[7] **Li Zheng***, Du Ruxu, “Design and Analysis of a Biomimetic Wire-Driven Flapping Propeller”, IEEE International Conference on Biomedical Robotics and Biomechatronics, (BioRob2012), Roma, Italy, June 24-27, 2012.

[8] **Li Zheng***, Du Ruxu, Lei Mancheong, Yuan Songmei, “Design and Analysis of A Biomimetic Wire-Driven Robot Arm”, International Mechanical Engineering Congress & Exposition (IMECE 2011), Denver, Colorado, USA, Nov. 11-17, 2011.

Appendix B - Derivation

B.1 Wire Length w.r.t. Joint Rotation

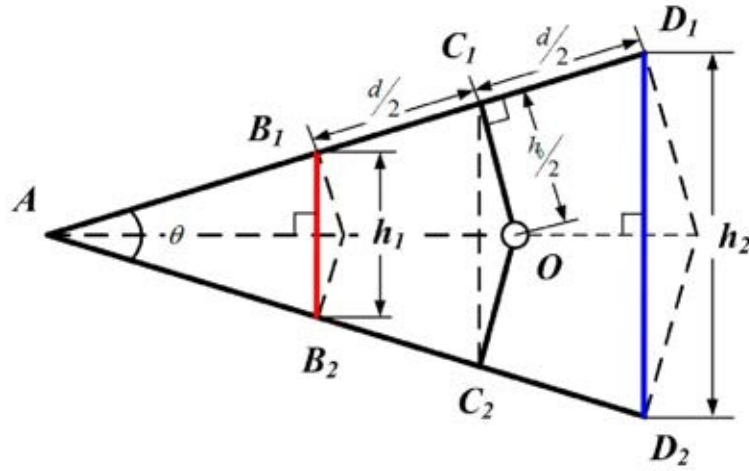


Figure B-1 Wire Length Change in Joint Rotation

Before joint rotation, the wire length B_1B_2 equals to D_1D_2 , they are both h_0 . When the joint rotates θ to left, the wire length B_1B_2 is h_1 , and the wire length D_1D_2 is h_2 .

From the geometry as shown in Figure B-1, we get:

$$|C_1C_2| = h_0 \cdot \cos\left(\frac{\theta}{2}\right) \quad (\text{B-1})$$

By using the half angle formula, we have:

$$|C_1C_2| = h_0 \cdot \left[1 - 2 \cdot \sin^2\left(\frac{\theta}{4}\right)\right] \quad (\text{B-2})$$

The wires lengths after joint rotation are:

$$\begin{cases} h_1 = |B_1B_2| = |C_1C_2| - d \cdot \sin\left(\frac{\theta}{2}\right) \\ h_2 = |D_1D_2| = |C_1C_2| + d \cdot \sin\left(\frac{\theta}{2}\right) \end{cases} \quad (\text{B-3})$$

By substituting Equation (B-2) into Equation (B-3), we have:

$$\begin{cases} h_1 = h_0 \cdot \left[1 - 2 \cdot \sin^2\left(\frac{\theta}{4}\right)\right] - d \cdot \sin\left(\frac{\theta}{2}\right) \\ h_2 = h_0 \cdot \left[1 - 2 \cdot \sin^2\left(\frac{\theta}{4}\right)\right] + d \cdot \sin\left(\frac{\theta}{2}\right) \end{cases} \quad (\text{B-4})$$

After rearrangement, we have:

$$\begin{cases} h_1 = h_0 - \Delta h_1 = h_0 - \left[d \cdot \sin\left(\frac{\theta}{2}\right) + 2h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \\ h_2 = h_0 + \Delta h_2 = h_0 + \left[d \cdot \sin\left(\frac{\theta}{2}\right) - 2h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \end{cases} \quad (\text{B-5})$$

The joint rotation can be found from Equation (B-5). After rotation, the discrepancy of the two wires' lengths is:

$$h_2 - h_1 = 2d \cdot \sin\left(\frac{\theta}{2}\right) \quad (\text{B-6})$$

Hence, the joint rotation is:

$$\theta = 2 \cdot \arcsin\left[\frac{h_2 - h_1}{2d}\right] \quad (\text{B-7})$$

In the wire-driven mechanism, there are N joints and based on the constant curvature assumption, all the joints rotate the same. Assume in the rest position, the lengths of the two wires are the same, they both equal to L_0 . By summing up all the length change in the joints, the lengths of the two wires after backbone bending are:

$$\begin{cases} L_1 = L_0 - N \cdot \Delta h_1 = L_0 - N \left[d \cdot \sin\left(\frac{\theta}{2}\right) + 2h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \\ L_2 = L_0 + N \cdot \Delta h_2 = L_0 + N \left[d \cdot \sin\left(\frac{\theta}{2}\right) - 2h_0 \cdot \sin^2\left(\frac{\theta}{4}\right) \right] \end{cases} \quad (\text{B-8})$$

The backbone bending angle is:

$$\Theta = N \cdot \theta = 2N \cdot \arcsin\left[\frac{h_2 - h_1}{2d}\right] \quad (\text{B-9})$$

Or, we can represent the backbone bending using the two wires' lengths:

$$\Theta = 2N \cdot \arcsin\left[\frac{L_2 - L_1}{2N \cdot d}\right] \quad (\text{B-10})$$

B.2 Distal End Position in Generalized Kinematics Model

In the kinematics modeling, constant curvature assumption is adopted. This means, for serpentine backbone the joints have identical rotation, and for continuum backbone, the deformed backbone curve is a circular arc. Figure B-2

shows the bending of the two types of backbone, with the same length L and bending angle Θ . Coordinate frame XOZ is located at the base or at the first joint rotation center. Initially, the backbone lies on Z axis.

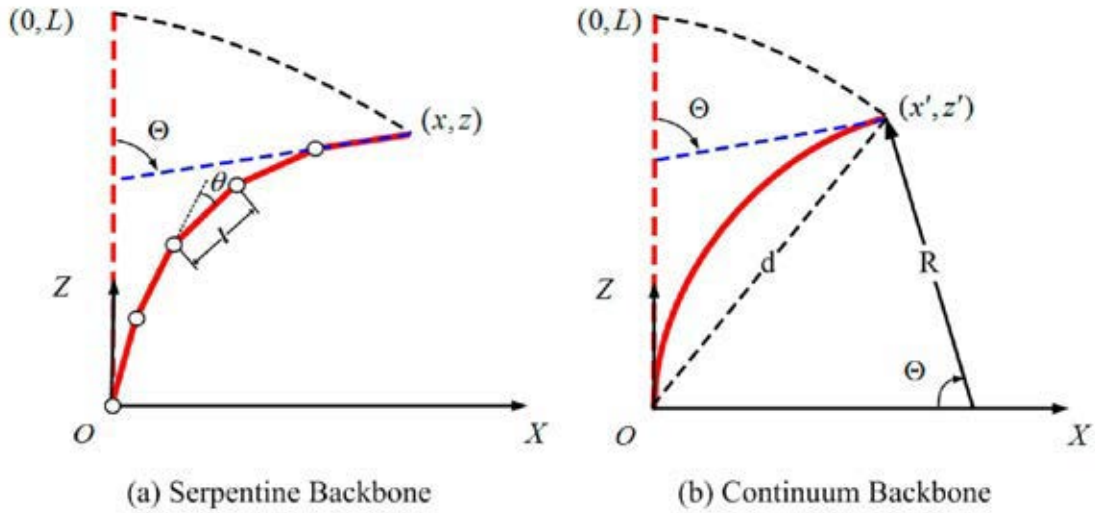


Figure B-2 Distal End Position of the Wire-Driven Mechanism

For serpentine backbone, the deformed curve is a polyline with uniform side length and interior angle. The total length of the backbone is L , and there are N vertebras with N joints as shown in the figure. The first joint is formed by the base and the first vertebra. For each vertebra, the length is $l = L/N$. The bending angle of the backbone is Θ . As all the joints rotate the same, for each joint, the rotation is $\theta = \Theta/N$. The distal end position can be found from geometry:

$$\begin{cases} x = \sum_{i=1}^N l \cdot \sin(i \cdot \theta) \\ z = \sum_{i=1}^N l \cdot \cos(i \cdot \theta) \end{cases} \quad (\text{B-11})$$

For continuum backbone, the deformed backbone is a circular arc. The arc length is L , which is the same as the backbone length. The center angle of the arc is the backbone bending angle Θ . From geometry, the distal end position is:

$$\begin{cases} x' = R[1 - \cos(\Theta)] \\ z' = R \cdot \sin(\Theta) \end{cases} \quad (\text{B-12})$$

In the equation, $R = L/\Theta$ is the arc radius.

When the backbone length L is kept constant and increases the joint number N to infinite, the two distal ends overlap. Hence, we can use Equation (B-11) to represent the continuum backbone distal end. It is noted that:

$$\begin{aligned}
\sum_{i=1}^N l \cdot \sin(i \cdot \theta) &= \frac{l}{2 \sin\left(\frac{\theta}{2}\right)} \cdot \sum_{i=1}^N 2 \sin\left(\frac{\theta}{2}\right) \cdot \sin(i \cdot \theta) \\
&= \frac{l}{2 \sin\left(\frac{\theta}{2}\right)} \cdot \sum_{i=1}^N \left[\cos\left(i \cdot \theta - \frac{\theta}{2}\right) - \cos\left(i \cdot \theta + \frac{\theta}{2}\right) \right] \\
&= \frac{l}{2 \sin\left(\frac{\theta}{2}\right)} \cdot \left[\cos\left(\frac{\theta}{2}\right) - \cos\left(N \cdot \theta + \frac{\theta}{2}\right) \right] \\
&= \frac{l}{\sin\left(\frac{\theta}{2}\right)} \cdot \sin\left(\frac{N \cdot \theta}{2}\right) \cdot \sin\left(\frac{(N+1)\theta}{2}\right)
\end{aligned} \tag{B-13}$$

Similarly, we have:

$$\sum_{i=1}^N l \cdot \cos(i \cdot \theta) = \frac{l}{\sin\left(\frac{\theta}{2}\right)} \cdot \sin\left(\frac{N \cdot \theta}{2}\right) \cdot \cos\left(\frac{(N+1)\theta}{2}\right) \tag{B-14}$$

Hence, the distal end of a wire-driven mechanism is:

$$\begin{cases} x = \frac{L \cdot \sin(\Theta/2)}{N \cdot \sin(\Theta/2N)} \cdot \sin\left(\frac{(N+1)\Theta}{2N}\right) \\ z = \frac{L \cdot \sin(\Theta/2)}{N \cdot \sin(\Theta/2N)} \cdot \cos\left(\frac{(N+1)\Theta}{2N}\right) \end{cases} \tag{B-15}$$

When $N \rightarrow \infty$, by L'Hôpital's rule we have:

$$\begin{cases} x = \lim_{N \rightarrow \infty} \left\{ \frac{L \cdot \sin(\Theta/2)}{N \cdot \sin(\Theta/2N)} \cdot \sin\left(\frac{(N+1)\Theta}{2N}\right) \right\} = \frac{L}{\Theta} [1 - \cos(\Theta)] \\ z = \lim_{N \rightarrow \infty} \left\{ \frac{L \cdot \sin(\Theta/2)}{N \cdot \sin(\Theta/2N)} \cdot \cos\left(\frac{(N+1)\Theta}{2N}\right) \right\} = \frac{L}{\Theta} \sin(\Theta) \end{cases} \tag{B-16}$$

It is the same as Equation (B-12).

Appendix C – Matlab Programs

1) Figure 3-8 WDM Distal End Velocity and Leverage Effect

```

L=150e-3; N=10;
d=10e-3; dtheta=pi;
thetamax=180*pi/180;
x=zeros(181,1); z=zeros(181,1);
vx=zeros(181,1); vz=zeros(181,1);
v=zeros(181,1); vw=zeros(181,1)
theta=zeros(181,1);
a=L/N;
b=L/2/N^2;
for i=1:181
    t(i)=(i-1)/180;
    beta(i)=(i-1)/2/N*pi/180;
    theta(i)=(i-1)*pi/180;
    sb=sin(beta(i));
    cb=cos(beta(i));
    st=sin(theta(i));
    ct=cos(theta(i));
    x(i)=a*sin(theta(i)/2)*sin(theta(i)/2+beta(i))/
    sb;
    z(i)=a*sin(theta(i)/2)*cos(theta(i)/2+beta(i))
    /sb;
    vx(i)=b*(N*sb*sin(theta(i)+beta(i))-
    sin(theta(i)/2)^2/sb^2*dtheta;
    vz(i)=b*(N*sb*cos(theta(i)+beta(i))-
    0.5*st)/sb^2*dtheta;
    v(i)=sqrt(vx(i)^2+vz(i)^2);
    vw(i)=d/2*dtheta;
end
figure % distal end velocity
plot(theta*180/pi,vx,'b','linewidth',2);
hold on;
plot(theta*180/pi,vz,'g','linewidth',2);
plot(theta*180/pi,v,'r','linewidth',2);
xlabel('theta
(^o)','fontsize',12,'fontweight','bold');
ylabel('Distal End Velocity
(m/s)','fontsize',12,'fontweight','bold');
legend('Vx','Vy','V',4);
xlim([0 180]);
set(gca,'fontsize',12);
set(gca,'fontweight','bold');
figure % velocity amplification ratio
plot(theta*180/pi,v./vw,'b','linewidth',2);
xlabel('theta
(^o)','fontsize',12,'fontweight','bold');
ylabel('Velocity
Ratio','fontsize',12,'fontweight','bold');
xlim([0 180]);
set(gca,'fontsize',12);
set(gca,'fontweight','bold');

```

2) Figure 3-12 Workspace of a Single Segment Spatial WDM

Figure 3-22 Trajectory of the WDM Manipulator End Effector

Figure 3-23 Relative Positioning Error of the Manipulator

```

% single section robot arm simulation
N=8; H=12.5 D=20;
d=15; h0=2.5;
HS=13.75;
HE=12.85+12.5;
pi=3.1415926;
A=-HE;
B=-(H+h0);
C=-HS;
% Distal end trajectory prediction
for i=1:286
    theta(i)=(-14.25+0.1*(i-1))*pi/180;
    x(i)=A*cos((N+1)*theta(i))+B*sin(N*theta(
    i)/2)/sin(theta(i)/2)*cos((N+1)*theta(i)/2)+C;
    y(i)=A*sin((N+1)*theta(i))+B*sin(N*theta(i)
    )/2)/sin(theta(i)/2)*sin((N+1)*theta(i)/2);
end
plot(-x,y,'r','linewidth',2);
xlabel('Z position
(mm)','fontsize',12,'fontweight','bold');
ylabel('X position
(mm)','fontsize',12,'fontweight','bold');
xlim([0,180]); ylim([-150,150]);
% Compare with experiment data
xexp=[40 58 76.5 95 114 132.5 142 151 158
160 161 160 157 153 148 137 130 122 110
94 76 58 40];
yexp=[111 113 110.5 105 92 76 62 45 25 11
0.1 -12.5 -26 -40 -53 -68 -78 -87 -96 -105 -
112 -114 -112];
hold on;
plot(xexp,yexp,'*b');
grid on
legend('Predicted','Experiment');
set(gca,'fontsize',12,'fontweight','bold');
% Extract from simulation
xp=[42.32 58.56 76.28 96.68 113.3 130.4
141.4 150 156.4 158.5 159.2 158.4 156.2
152.0 146.0 137 129.5 120.3 109.1 92.55
75.1 57.39 42.32];
yp=[111.2 111.5 108.8 102.8 91.47 75.89
61.28 44.89 24.59 11.38 0.1 -12.71 -25.89 -
39.95 -53.28 -67.78 -76.86 -85.88 -94.37 -
103.3 -109.1 -111.6 -111.2];
x_error=(xexp-xp)./xp;
y_error=(yexp-yp)./yp;
d_exp=sqrt(xexp.*xexp+yexp.*yexp);
d_prid=sqrt(xp.*xp+yp.*yp);

```

```

d_error=(d_exp-d_prid)/d_prid;
figure;
plot(x_error*100,'r','linewidth',2);
hold on;
plot(y_error*100,'b','linewidth',2);
plot(d_error*100,'c','linewidth',2);
xlabel('Sample
Num.','fontsize',12,'fontweight','bold');
ylabel('Relative
Error(%)','fontsize',12,'fontweight','bold');
grid on;
set(gca,'fontsize',12,'fontweight','bold');
legend('Z Error','X Error','Dist. Error',3);
% workspace
for i=1:29
    theta(i)=(-14.25+1*(i-1))*pi/180;
    temp1=A*sin((N+1)*theta(i))+B*sin(N*theta(i)/2)/sin(theta(i)/2)*sin((N+1)*theta(i)/2);
    temp2=A*cos((N+1)*theta(i))+B*sin(N*theta(i)/2)/sin(theta(i)/2)*cos((N+1)*theta(i)/2)+C;

```

```

for j=1:37
    phi(j)=(j-1)*pi/18;
    wx(i,j)=temp1*cos(phi(j));
    wy(i,j)=temp1*sin(phi(j));
    wz(i,j)=temp2;
end
end
figure;
grid on;
surf(wx,wy,wz);
xlabel('X position
(mm)','fontsize',12,'fontweight','bold');
ylabel('Y position
(mm)','fontsize',12,'fontweight','bold');
zlabel('Z position
(mm)','fontsize',12,'fontweight','bold');
xlim([-150,150]); ylim([-150,150]);
zlim([-180,0]);
set(gca,'fontsize',12,'fontweight','bold');

```

3) Figure 3-13 Trajectories of the WDM Distal End with Increased Joint Number

```

% Serpentine WDM parameter
N=[1 2 5 100];
mycolor=[0 0 1; 0 1 0; 1 0 0; 1 0 1];
mywidth=[2 2 2 4];
d=15; D=20;
h0=2.5; H=12.5;
len=H+h0;
L=150;
BTheta=pi/2;
dtheta=0.1*pi/180;
figure;
for j=1:size(N,2)
    thetamax=BTheta/N(j)
    clear theta x y;
    for i=1:ceil(2*tetamax/dtheta)+1
        theta(i) = -tetamax+(i-1)*dtheta;
        [x(i),y(i)]=UFM(L,N(j),N(j)*theta(i));
    end
    l(j)=L/N(j);
    bx(1)=0;
    by(1)=0;
    for k=1:N(j)
        bx(k+1)=bx(k)+l(j)*sin(k*tetamax);
        by(k+1)=by(k)+l(j)*cos(k*tetamax);
    end
    h(j)=plot(x,y,'color',mycolor(j,:), 'linewidth',
mywidth(j)); hold on;
    plot(bx,by, 'o','LineWidth',2,'color',mycolor(j,:),...
        'MarkerEdgeColor','b',...
        'MarkerFaceColor','w',...
        'MarkerSize',5);
    plot(-bx,by, 'o','LineWidth',2,'color',mycolor(j,:),...
        'MarkerEdgeColor','b',...
        'MarkerFaceColor','w',...

```

```

        'MarkerSize',5);
    clear bx by;
end
% Continuum WDM
clear x y i j;
for i=1:ceil(2*BTheta/dtheta)+1
    theta(i)=-BTheta+(i-1)*dtheta;
    R=L/theta(i);
    x(i)=R*(1-cos(theta(i)));
    y(i)=R*sin(theta(i));
end
for j=1:150
    R=L/BTheta;
    len(j)=j;
    alfa(j)=len(j)/L*BTheta;
    bx(j)=R*(1-cos(alfa(j)));
    by(j)=R*(sin(alfa(j)));
end
h(5)=plot(x,y,'-k','linewidth',2);
hold on;
plot(bx,by,'linewidth',2,'color',[0.5,0.5,0.5]);
plot(-bx,by,'linewidth',2,'color',[0.5,0.5,0.5]);
xlabel('X Position
(mm)','fontsize',12,'fontweight','bold');
ylabel('Z Position
(mm)','fontsize',12,'fontweight','bold');
xlim([-150 150]);
ylim([-100 160]);
axis equal;
legend(h,'WDM N=1','WDM N=2','WDM
N=5','WDM N=100','Continuum WDM',3);
set(gca,'fontsize',12,'fontweight','bold');

function [x,y]=UFM(L,N,THETA)
if N>1e6
    tempTheta=0

```



```

R=L/THETA;
else
tempTheta=THETA/2/N;
R=L/(2*N)/sin(tempTheta);
end
if abs(THETA)<1e-3
x=0;
y=L;
else

```

```

tempX=R*(1-cos(THETA));
tempY=R*sin(THETA);
x=cos(tempTheta)*tempX+sin(tempTheta)*t
empY;
y=-
sin(tempTheta)*tempX+cos(tempTheta)*tem
pY;
end

```

4) Figure 3-14 Workspace of a Three-Segment WDM in the X-Z Plane. (a) $N=10$, Simulation Interval is 2.5° ; (b) $N=10$, Simulation Interval is 10° ; (c) $N=8$, Simulation Interval is 2.5° ; (d) $N=12$, Simulation Interval is 2.5° .

```

% robot arm parameters
N=[10 10 10]; % joint number
phi = [0 0 0]*pi/180;
H = [6.5 6.5 6.5]; h0 = [2.5 2.5 2.5];
D = [20 20 20]; d = [15 15 15];
for i=1:3
thetamax(i) = 2*atan(h0(i)/D(i));
dtheta(i)=0.25*pi/180; % simulation
interval
Num(i)=floor(2*thetamax(i)/dtheta(i)+1);
end
%tip position
origin = [0 0 0 1]';
for i=1:Num(1) % section 1
theta1(i)=-thetamax(1)+(i-1)*dtheta(1);
BTheta1(i)=N(1)*theta1(i);
dist1(i)=(H(1)+h0(1))*sin(BTheta1(i))/sin(t
heta1(i));
if abs(sin(theta1(i)/2))<0.0001
dist1(i) = (H(1)+h0(1))*N(1);
end
T01= coordTrans(BTheta1(i), phi(1),
dist1(i));
for j=1:Num(2) % section 2
theta2(j)=-thetamax(2)+(j-1)*dtheta(2);
BTheta2(j)=N(2)*theta2(j);
dist2(j)=(H(2)+h0(2))*sin(BTheta2(j))/sin(t
heta2(j));
if abs(sin(theta2(j)/2))<0.0001
dist2(j) = (H(2)+h0(2))*N(2);
end
T02=T01*coordTrans(BTheta2(j), phi(2),
dist2(j));

```

```

for k=1:Num(3) % section 3
theta3(k)=-thetamax(3)+(k-
1)*dtheta(3);
BTheta3(k)=N(3)*theta3(k);
dist3(k)=(H(3)+h0(3))*sin(BTheta3(k)
)/sin(theta3(k));
if abs(sin(theta3(k)/2))<0.0001
dist3(k) = (H(3)+h0(3))*N(3);
end
T03= T02*coordTrans(BTheta3(k),
phi(3), dist3(k));
P3=T03*origin;
tipX(i,j,k)=P3(1);
tipY(i,j,k)=P3(2);
tipZ(i,j,k)=P3(3);
x(k)=P3(1);
y(k)=P3(2);
z(k)=P3(3);
end
plot(x,z,'linewidth',2);
% plot(-x,z,'linewidth',2);
hold on;
end
end
scatter(0,0,'r','filled');
xlim([-300,300]); ylim([-250,350]);
xlabel('X position
(mm)','fontSize',12,'fontWeight','b');
ylabel('Z position
(mm)','fontSize',12,'fontWeight','b');
set(gca,'fontSize',12,'fontWeight','b');

```

5) Figure 3-18 Inverse Kinematics: (a) Solution with Exact Position; (b) Solution with Exact Orientation and Exact X position; (c) Solution with Exact Orientation and Exact Z position

```

% single section robot arm parameters
N=10; d=15; D=20;
h0=2.5; H=12.5; len=H+h0;
L=N*len;
dtheta = 0.1*pi/180; % simulation interval
is 0.1 deg
thetamax=2*atan(h0/D); % maximum
bending angle for one joint
% forward kinematics

```

```

FN1=6; FN2=N-FN1;
Fbeta=0.03; Falfa=0.12;
FX1=len*(sin(FN1*Fbeta)*sin((FN1+1)*Fb
eta))/sin(Fbeta);
FY1=len*(sin(FN1*Fbeta)*cos((FN1+1)*Fb
eta))/sin(Fbeta);
FX2=len*(sin(FN2*Falfa)*sin(FN1*2*Fbet
a+(FN2+1)*Falfa))/sin(Falfa);

```

```

FY2=len*(sin(FN2*Falfa)*cos(FN1*2*Fbet
a+(FN2+1)*Falfa))/sin(Falfa);
FTheta=FN1*2*Fbeta+FN2*2*Falfa;
figure;
% target position and orientation
X=FX1+FX2; Y=FY1+FY2;
Theta=FTheta;
tmpcolor=0;
mywidth=[2 2 2];
mycolor=[0 1 1; 0.9 0.9 0; 1 0 0];
% inverse by scan (meet position
requiement first)
for i=1:N-1
    N1=i; N2=N-N1;
    eq=@(ang)
inverseFun(ang,X,Y,N1,N2,len);
[ang(i,:),fval(i,:),exitflag(i)]=fsolve(eq,[1e-6
1e-6]);
    tmpalfa(i)=ang(i,2); tmpbeta(i)=ang(i,1);
    tmptheta(i)=2*(N1*tmpbeta(i)+N2*tmpal
fa(i));
    X1(i)=len*(sin(N1*tmpbeta(i))*sin((N1+
1)*tmpbeta(i)))/sin(tmpbeta(i));
    Y1(i)=len*(sin(N1*tmpbeta(i))*cos((N1+
1)*tmpbeta(i)))/sin(tmpbeta(i));
    colorindex=1;
    if abs(tmpalfa(i))<thetamax/2 &
abs(tmpbeta(i))<thetamax/2
        beta=tmpbeta(i); alfa=tmpalfa(i);
        theta=tmptheta(i); tmpN1=N1;
        colorindex=2;
        if abs(theta-FTheta)<1e-2
            colorindex=3;
        end
    end
% plot robot arm
vx(1)=0; vy(1)=0;
for j=1:N
    if j<N1
        vx(j+1)=vx(j)+len*sin(j*2*tmpbeta(i));
        vy(j+1)=vy(j)+len*cos(j*2*tmpbeta(i));
    else
        vx(j+1)=vx(j)+len*sin(N1*2*tmpbeta(i)+(j-
N1)*2*tmpalfa(i));
        vy(j+1)=vy(j)+len*cos(N1*2*tmpbeta(i)+(j-
N1)*2*tmpalfa(i));
    end
end
plot(vx,vy,'-
o','LineWidth',mywidth(colorindex),'color',
mycolor(colorindex,:),...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);

hold on;
scatter(X1(i),Y1(i),'s','filled',...
'MarkerFaceColor',mycolor(colorindex,:));
end
vy(1)=0;
angBT=beta; angAF=alfa;

for i=1:N
    if i<tmpN1
        vx(i+1)=vx(i)+len*sin(i*2*tmpbeta(tmpN
1));
        vy(i+1)=vy(i)+len*cos(i*2*tmpbeta(tmp
N1));
    else
        vx(i+1)=vx(i)+len*sin(tmpN1*2*tmpbeta(t
mpN1)+(i-tmpN1)*2*tmpalfa(tmpN1));
        vy(i+1)=vy(i)+len*cos(tmpN1*2*tmpbeta(t
mpN1)+(i-tmpN1)*2*tmpalfa(tmpN1));
    end
end
plot(vx,vy,'-
o','LineWidth',2,'color',mycolor(3,:),...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);

hold on;
scatter(X1(tmpN1),Y1(tmpN1),'s','filled',...
'MarkerFaceColor',mycolor(3,:));
% left bending limit
leftx(1)=0; leftz(1)=0;
for i=1:N
    leftx(i+1)=leftx(i)+(H+h0)*sin(-
i*thetamax);
    leftz(i+1)=leftz(i)+(H+h0)*cos(-
i*thetamax);
end
plot(leftx,leftz,'-go','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);
% right bending limit
rightx(1)=0; rightz(1)=0;
for i=1:N
    rightx(i+1)=rightx(i)+(H+h0)*sin(i*thetama
x);
    rightz(i+1)=rightz(i)+(H+h0)*cos(i*thetama
x);
end
plot(rightx,rightz,'-go','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);
% tip trajectory without constraint
for i=1:ceil(2*thetamax/dtheta)
    theta(i) = -thetamax+(i-1)*dtheta;
    BTheta= N*theta(i);
    [tx(i),tz(i)]=UFM(N*len,N,BTheta);
end
plot(tx,tz,'--r','linewidth',2);
axis equal;
set(gca,'fontsize',12,'fontweight','bold');
xlabel('X Position
(mm)','fontsize',12,'fontweight','b');
ylabel('Z Position
(mm)','fontsize',12,'fontweight','b');
hold on;
scatter(X,Y,'filled');

```



```

% inverse by scan (meet orientation ans X
position first)
figure;
for i=1:N-1
    N1=i; N2=N-N1;
    eq=@(ang)
inverseFun2(ang,X,FTheta,N1,N2,len);

[ang(i,:),fval(i,:),exitflag(i)]=fsolve(eq,[1e-6
1e-6]);
tmpalfa(i)=ang(i,2); tmpbeta(i)=ang(i,1);
tmptheta(i)=2*(N1*tmpbeta(i)+N2*tmpal
fa(i));
X1(i)=len*(sin(N1*tmpbeta(i))*sin((N1+
1)*tmpbeta(i)))/sin(tmpbeta(i));
Y1(i)=len*(sin(N1*tmpbeta(i))*cos((N1+
1)*tmpbeta(i)))/sin(tmpbeta(i));
tmpY(i)=len*(sin(N1*tmpbeta(i))*cos((N
1+1)*tmpbeta(i)))/sin(tmpbeta(i))+...
len*(sin(N2*tmpalfa(i))*cos(2*N1*tmpbe
ta(i)+(N2+1)*tmpalfa(i)))/sin(tmpalfa(i))
colorindex=1;
if abs(tmpalfa(i))<thetamax/2 &
abs(tmpbeta(i))<thetamax/2
    beta=tmpbeta(i); alfa=tmpalfa(i);
    tmpN1=N1; colorindex=2;
    if abs(tmpY(i)-Y)<1e-2
        colorindex=3;
    end
end
end
% plot robot arm
vx(1)=0; vy(1)=0;
for j=1:N
    if j<N1
vx(j+1)=vx(j)+len*sin(j*2*tmpbeta(i));
vy(j+1)=vy(j)+len*cos(j*2*tmpbeta(i));
    else
vx(j+1)=vx(j)+len*sin(N1*2*tmpbeta(i)+(j-
N1)*2*tmpalfa(i));
vy(j+1)=vy(j)+len*cos(N1*2*tmpbeta(i)+(j-
N1)*2*tmpalfa(i));
    end
end
plot(vx,vy,'-
o','LineWidth',mywidth(colorindex),'color',
mycolor(colorindex,:),...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);

hold on;
scatter(X1(i),Y1(i),'s','filled',...
'MarkerFaceColor',mycolor(colorindex,:));
end
vy(1)=0;
angAF=alfa;
for i=1:N
    if i<tmpN1
vx(i+1)=vx(i)+len*sin(i*2*tmpbeta(tmpN1)
);

```

```

vy(i+1)=vy(i)+len*cos(i*2*tmpbeta(tmpN1)
);
    else
vx(j+1)=vx(i)+len*sin(tmpN1*2*tmpbeta(t
mpN1)+(i-tmpN1)*2*tmpalfa(tmpN1));
vy(i+1)=vy(i)+len*cos(tmpN1*2*tmpbeta(t
mpN1)+(i-tmpN1)*2*tmpalfa(tmpN1));
    end
end
plot(vx,vy,'-
o','LineWidth',2,'color',mycolor(3,:),...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);

hold on;
scatter(X1(tmpN1),Y1(tmpN1),'s','filled',...
'MarkerFaceColor',mycolor(3,:));
% left bending limit
leftx(1)=0; leftz(1)=0;
for i=1:N
    leftx(i+1)=leftx(i)+(H+h0)*sin(-
i*thetamax);
    leftz(i+1)=leftz(i)+(H+h0)*cos(-
i*thetamax);
end
plot(leftx,leftz,'-go','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);

% right bending limit
rightx(1)=0; rightz(1)=0;
for i=1:N
    rightx(i+1)=rightx(i)+(H+h0)*sin(i*thetama
x);
    rightz(i+1)=rightz(i)+(H+h0)*cos(i*thetama
x);
end
plot(rightx,rightz,'-go','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);

% tip trajectory without constraint
for i=1:ceil(2*thetamax/dtheta)
    theta(i) = -thetamax+(i-1)*dtheta;
    BTheta= N*theta(i);
    [tx(i),tz(i)]=UFM(N*len,N,BTheta);
end
plot(tx,tz,'-r','linewidth',2);
axis equal; hold on;
set(gca,'fontsize',12,'fontweight','bold');
xlabel('X Position
(mm)','fontsize',12,'fontweight','b');
ylabel('Z Position
(mm)','fontsize',12,'fontweight','b');
scatter(X,Y,'filled');
% inverse by scan (meet orientation and Y
position first)
figure
for i=1:N-1
    N1=i; N2=N-N1;

```

```

    eq=@(ang)
inverseFun3(ang,Y,FTheta,N1,N2,len);

[ang(i,:),fval(i,:),exitflag(i)]=fsolve(eq,[1e-6
1e-6]);
    tmpalfa(i)=ang(i,2);
    tmpbeta(i)=ang(i,1);
    tmptheta(i)=2*(N1*tmpbeta(i)+N2*tmpalfa(
i));
    X1(i)=len*(sin(N1*tmpbeta(i))*sin((N1+1)*
tmpbeta(i)))/sin(tmpbeta(i));
    Y1(i)=len*(sin(N1*tmpbeta(i))*cos((N1+1)
*tmpbeta(i)))/sin(tmpbeta(i));
    tmpX(i)=len*(sin(N1*tmpbeta(i))*sin((N1+
1)*tmpbeta(i)))/sin(tmpbeta(i))+...
len*(sin(N2*tmpalfa(i))*sin(2*N1*tmpbeta(
i)+(N2+1)*tmpalfa(i)))/sin(tmpalfa(i))
    colorindex=1;
    if abs(tmpalfa(i))<thetamax/2 &
abs(tmpbeta(i))<thetamax/2
        beta=tmpbeta(i);    alfa=tmpalfa(i);
        tmpN1=N1;    colorindex=2;
        if abs(tmpX(i)-X)<1e-2
            colorindex=3;
        end
    end
    % plot robot arm
    vx(1)=0;    vy(1)=0;

    for j=1:N
        if j<N1
            vx(j+1)=vx(j)+len*sin(j*2*tmpbeta(i));
            vy(j+1)=vy(j)+len*cos(j*2*tmpbeta(i));
        else
            vx(j+1)=vx(j)+len*sin(N1*2*tmpbeta(i)+(j-
N1)*2*tmpalfa(i));
            vy(j+1)=vy(j)+len*cos(N1*2*tmpbeta(i)+(j-
N1)*2*tmpalfa(i));
        end
    end
    plot(vx,vy,'-
o','LineWidth',mywidth(colorindex),'color',
mycolor(colorindex,:),...
        'MarkerEdgeColor','b',...
        'MarkerFaceColor','w',...
        'MarkerSize',5);

    hold on;
    scatter(X1(i),Y1(i),'s','filled',...
'MarkerFaceColor',mycolor(colorindex,:));
end
vy(1)=0;
angBT=beta; angAF=alfa;
for i=1:N
    if i<tmpN1
        vx(i+1)=vx(i)+len*sin(i*2*tmpbeta(tmpN1)
);
        vy(i+1)=vy(i)+len*cos(i*2*tmpbeta(tmpN1)
);
    else
        vx(i+1)=vx(i)+len*sin(tmpN1*2*tmpbeta(t
mpN1)+(i-tmpN1)*2*tmpalfa(tmpN1));
        vy(i+1)=vy(i)+len*cos(tmpN1*2*tmpbeta(t
mpN1)+(i-tmpN1)*2*tmpalfa(tmpN1));
    end
end
plot(vx,vy,'-
o','LineWidth',2,'color',mycolor(3,:),...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);

hold on;
scatter(X1(tmpN1),Y1(tmpN1),'s','filled',...
'MarkerFaceColor',mycolor(3,:));
% left bending limit
leftx(1)=0; leftz(1)=0;
for i=1:N
    leftx(i+1)=leftx(i)+(H+h0)*sin(-
i*thetamax);
    leftz(i+1)=leftz(i)+(H+h0)*cos(-
i*thetamax);
end
plot(leftx,leftz,'-go','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);

% right bending limit
rightx(1)=0; rightz(1)=0;
for i=1:N
    rightx(i+1)=rightx(i)+(H+h0)*sin(i*thetama
x);
    rightz(i+1)=rightz(i)+(H+h0)*cos(i*thetama
x);
end
plot(rightx,rightz,'-go','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);

% tip trajectory without constraint
for i=1:ceil(2*thetamax/dtheta)
    theta(i) = -thetamax+(i-1)*dtheta;
    BTheta= N*theta(i);
    [tx(i),tz(i)]=UFM(N*len,N,BTheta);
end
plot(tx,tz,'-r','linewidth',2);
axis equal; hold on;
set(gca,'fontsize',12,'fontweight','bold');
xlabel('X Position
(mm)','fontsize',12,'fontweight','b');
ylabel('Z Position
(mm)','fontsize',12,'fontweight','b');
scatter(X,Y,'filled');

```

6) Figure 3-19 Workspace with Single Bilateral Constraint


```

% single section robot arm parameters
N=10; d=15; D=20;
h0=2.5; H=6.5; len=H+h0;
dtheta = 0.1*pi/180; % simulation interval
is 0.1 deg
thetamax=2*atan(h0/D); % maximum
bending angle for one joint
figure('name','Robot Arm Motion with
External Constraint');
hold on; axis equal;
xlabel('x position
(mm)','fontsize',12,'fontweight','bold');
ylabel('z position
(mm)','fontsize',12,'fontweight','bold');
title('Workspace with Bilateral
Constraint','fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');
count=0;
for i=1:ceil(2*thetamax/dtheta) % joint
bending angle loop
    theta(i) = -thetamax+(i-1)*dtheta;
    for j=1:N % manipulator loop
        [cx,cy]=UFM(j*len,j*theta(i)); %
current fixed position
        for k=1:ceil(2*thetamax/dtheta) %
free section joint bending angle loop
            temptheta(k) = -thetamax+(k-
1)*dtheta;
            [tempcx2,tempcy2]=UFM((N-
j)*len,(N-j),(N-j)*temptheta(k)); % current
fixed position
            cx2=cos(j*theta(i))*tempcx2+sin(j*theta(i))
*tempcy2;
            cy2=-
sin(j*theta(i))*tempcx2+cos(j*theta(i))*tem
pcy2;
            tx(k)=cx+cx2; ty(k)=cy+cy2;
        end
        plot(tx,ty);
    end
end
end

```

7) Figure 3-20 Workspace with Single Unilateral Constraint

```

% single section robot arm parameters
N=10; d=15; D=20;
h0=2.5; H=12.5; len=H+h0;
L=N*len;
dtheta = 0.1*pi/180; % simulation interval
is 0.1 deg
thetamax=2*atan(h0/D); % maximum
bending angle for one joint
figure('name','Robot Arm Motion with
External Constraint');
hold on; axis equal;
xlabel('x position
(mm)','fontsize',12,'fontweight','bold');
ylabel('z position
(mm)','fontsize',12,'fontweight','bold');
title('Workspace with Unilateral
Constraint','fontsize',12,'fontweight','bold');

```

```

% in the rest position
for i=1:N
    intz(i)=h0/2+(H+h0)*(i-1);
end
intz(N+1)=(H+h0)*N;
intx=zeros(1,N+1);
plot(intx,intz,'-ro','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);
% left bending limit
leftx(1)=0; leftz(1)=0;
for i=1:N
    leftx(i+1)=leftx(i)+(H+h0)*sin(-
i*thetamax);
    leftz(i+1)=leftz(i)+(H+h0)*cos(-
i*thetamax);
end
plot(leftx,leftz,'-go','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);
% right bending limit
rightx(1)=0; rightz(1)=0;
for i=1:N
    rightx(i+1)=rightx(i)+(H+h0)*sin(i*thetama
x);
    rightz(i+1)=rightz(i)+(H+h0)*cos(i*thetama
x);
end
plot(rightx,rightz,'-go','LineWidth',2,...
'MarkerEdgeColor','b',...
'MarkerFaceColor','w',...
'MarkerSize',5);
% tip trajectory without constraint
for i=1:ceil(2*thetamax/dtheta)
    theta(i) = -thetamax+(i-1)*dtheta;
    BTheta= N*theta(i);
    [tx(i),tz(i)]=UFM(N*len,N,BTheta);
end
plot(tx,tz,'-r','linewidth',2);

```

```

set(gca,'fontsize',12,'fontweight','bold');
% workspace with unilateral constraint
dx=1; dy=1;
Nx=L/dx; Ny=L/dy;
for i=1:Nx
    xc=i*dx;
    for j=1:Ny
        yc=j*dy;
        [x y vx vy]=UFMTra(N,L,xc,yc,thetamax);
        plot(x,y,'linewidth',2);
        hold on;
        plot(-x,y,'linewidth',2);
    end
end
% robot arm at rest
for i=1:N
    intz(i)=(H+h0)*(i-1);

```

```

end
intz(N+1)=(H+h0)*N;
intx=zeros(1,N+1);
plot(intx,intz,'-ro','LineWidth',2,...
     'MarkerEdgeColor','b',...
     'MarkerFaceColor','w',...
     'MarkerSize',5);

% left bending limit
leftx(1)=0; leftz(1)=0;
for i=1:N
    leftx(i+1)=leftx(i)+(H+h0)*sin(-
i*thetamax);
    leftz(i+1)=leftz(i)+(H+h0)*cos(-
i*thetamax);
end
plot(leftx,leftz,'-go','LineWidth',2,...
     'MarkerEdgeColor','b',...
     'MarkerFaceColor','w',...
     'MarkerSize',5);

% right bending limit
rightx(1)=0; rightz(1)=0;
for i=1:N
    rightx(i+1)=rightx(i)+(H+h0)*sin(i*thetama
x);
    rightz(i+1)=rightz(i)+(H+h0)*cos(i*thetama
x);
end
plot(rightx,rightz,'-go','LineWidth',2,...
     'MarkerEdgeColor','b',...
     'MarkerFaceColor','w',...
     'MarkerSize',5);

% tip trajectory without constraint
for i=1:ceil(2*thetamax/dtheta)
    theta(i) = -thetamax+(i-1)*dtheta;
    BTheta= N*theta(i);
    [tx(i),tz(i)]=UFM(N*len,N,BTheta);
end
plot(tx,tz,'-r','linewidth',2);

```

8) Figure 3-26 Trajectories Comparison of the Distal End

Figure 3-27 Relative Positioning Error of the Distal End

```

% robot arm parameters
SectNum = 3;
N=[10 10 10];
H = [6.4 6.4 6.4];
h0 = [2.5 2.5 2.5];
D = [20 20 20];
d = [15 15 15];
dtheta = 0.1*pi/180; % simulation interval
is 0.1 deg
phi=[0 0 0];
for i=1:3
    thetamax(i) = 2*atan(h0(i)/D(i));
    BThetaMax(i) = thetamax(i)*N(i);
end
% case 1: only section 1 bending in the XZ
plane
for i=1:ceil(thetamax(1)/dtheta)
    theta(i) = i*dtheta;
    BTheta(1)= N(1)*theta(i);
    dist(1) = (H(1)+h0(1))*sin(BTheta(1)/2)/sin(theta(i)/2);
    BTheta(2)=0;
    BTheta(3)=0;
    dist(2)=(H(2)+h0(2))*N(2);
    dist(3)=(H(3)+h0(3))*N(3);
    T01 = coordTrans(BTheta(1), phi(1),
dist(1));
    T12 = coordTrans(BTheta(2), phi(2),
dist(2));
    T23 = coordTrans(BTheta(3), phi(3),
dist(3));
    T02=T01*T12; T03=T02*T23;
    c1x(i)=T03(1,4); c1y(i)=T03(2,4);
    c1z(i)=T03(3,4);
    tip1x(i)=T01(1,4); tip1y(i)=T01(2,4);
    tip1z(i)=T01(3,4);
    tip2x(i)=T02(1,4); tip2y(i)=T02(2,4);
    tip2z(i)=T02(3,4);
    dist12(i)=sqrt((tip2x(i)-
tip1x(i))^2+(tip2y(i)-tip1y(i))^2+(tip2z(i)-
tip1z(i))^2);
    dist23(i)=sqrt((tip2x(i)-
c2x(i))^2+(tip2y(i)-c2y(i))^2+(tip2z(i)-
c2z(i))^2);
end
% case 2: only section 2 bending in the XZ
plane
for i=1:ceil(thetamax(2)/dtheta)
    theta(i) = i*dtheta;
    BTheta(1)= 0;
    dist(1) = (H(1)+h0(1))*N(1);
    BTheta(2)= N(2)*theta(i);
    dist(2)=(H(2)+h0(2))*sin(BTheta(2)/2)/si
n(theta(i)/2);
    BTheta(3)=0;
    dist(3)=(H(3)+h0(3))*N(3);
    T01=coordTrans(BTheta(1), phi(1), dist(1));
    T12=coordTrans(BTheta(2), phi(2), dist(2));
    T23=coordTrans(BTheta(3), phi(3), dist(3));
    T02=T01*T12; T03=T02*T23;
    c2x(i)=T03(1,4); c2y(i)=T03(2,4);
    c2z(i)=T03(3,4);
    tip1x(i)=T01(1,4); tip1y(i)=T01(2,4);
    tip1z(i)=T01(3,4);
    tip2x(i)=T02(1,4); tip2y(i)=T02(2,4);
    tip2z(i)=T02(3,4);
    dist12(i)=sqrt((tip2x(i)-
tip1x(i))^2+(tip2y(i)-tip1y(i))^2+(tip2z(i)-
tip1z(i))^2);
    dist23(i)=sqrt((tip2x(i)-
c2x(i))^2+(tip2y(i)-c2y(i))^2+(tip2z(i)-
c2z(i))^2);
end
% case 3: only section 3 bending in the XZ
plane

```



```

for i=1:ceil(thetamax(3)/dtheta)
    theta(i) = i*dtheta;
    BTheta(1)= 0;
    dist(1) = (H(1)+h0(1))*N(1);
    BTheta(2)= 0;
    dist(2)=(H(2)+h0(2))*N(2);
    BTheta(3)=N(3)*theta(i);
    dist(3)=(H(3)+h0(3))*sin(BTheta(3)/2)/sin(theta(i)/2);
    T01=coordTrans(BTheta(1), phi(1), dist(1));
    T12=coordTrans(BTheta(2), phi(2), dist(2));
    T23=coordTrans(BTheta(3), phi(3), dist(3));
    T02=T01*T12;   T03=T02*T23;
    c3x(i)=T03(1,4);   c3y(i)=T03(2,4);
    c3z(i)=T03(3,4);
    tip1x(i)=T01(1,4);   tip1y(i)=T01(2,4);
    tip1z(i)=T01(3,4);
    tip2x(i)=T02(1,4);   tip2y(i)=T02(2,4);
    tip2z(i)=T02(3,4);
    dist12(i)=sqrt((tip2x(i)-
    tip1x(i))^2+(tip2y(i)-tip1y(i))^2+(tip2z(i)-
    tip1z(i))^2);
    dist23(i)=sqrt((tip2x(i)-
    c3x(i))^2+(tip2y(i)-c3y(i))^2+(tip2z(i)-
    c3z(i))^2);
end
% case 4: three sections bend together
for i=1:ceil(thetamax(3)/dtheta)
    theta(i) = i*dtheta;
    BTheta(1)= N(1)*theta(i);
    dist(1) = (H(1)+h0(1))*sin(BTheta(1)/2)/sin(theta(i)/2);
    BTheta(2)= N(2)*theta(i);
    dist(2)=(H(2)+h0(2))*sin(BTheta(2)/2)/sin(theta(i)/2);
    BTheta(3)=N(3)*theta(i);
    dist(3)=(H(3)+h0(3))*sin(BTheta(3)/2)/sin(theta(i)/2);
    T01=coordTrans(BTheta(1), phi(1), dist(1));
    T12=coordTrans(BTheta(2), phi(2), dist(2));
    T23=coordTrans(BTheta(3), phi(3), dist(3));
    T02=T01*T12;   T03=T02*T23;
    c4x(i)=T03(1,4);   c4y(i)=T03(2,4);
    c4z(i)=T03(3,4);
    tip1x(i)=T01(1,4);   tip1y(i)=T01(2,4);
    tip1z(i)=T01(3,4);
    tip2x(i)=T02(1,4);   tip2y(i)=T02(2,4);
    tip2z(i)=T02(3,4);
    dist12(i)=sqrt((tip2x(i)-
    tip1x(i))^2+(tip2y(i)-tip1y(i))^2+(tip2z(i)-
    tip1z(i))^2);
    dist23(i)=sqrt((tip2x(i)-
    c4x(i))^2+(tip2y(i)-c4y(i))^2+(tip2z(i)-
    c4z(i))^2);
end
figure; hold on
% final experiment
case3_x=[264 262 260 253 245 235 225 214
202 188];
case2_x=[0 30 50 80 102 128 140 147 151
149 137 116];
case2_z=[264 261 256 246 230 200 170 140
110 80 50 27];
case4_x=[0 20 60 110 150 180 200 208 202
181 149 110 66];
case4_z=[264 263 257 234 210 170 120 70
20 -20 -50 -60 -56];
case1_x=[0 40 70 100 146 180 210 235 244
241 233 213];
case1_z=[264 262 256 246 220 190 150 100
40 -10 -50 -89];
% trajectory comparison
plot(c1x,c1z,'r','linewidth',2);
plot(c2x,c2z,'g','linewidth',2);
plot(c3x,c3z,'b','linewidth',2);
plot(c4x,c4z,'m','linewidth',2);
legend('case 1','case 2','case 3','case 4');
plot(case1_x,case1_z,'*r','linewidth',2);
plot(case2_x,case2_z,'*g','linewidth',2);
plot(case3_x,case3_z,'*b','linewidth',2);
plot(case4_x,case4_z,'*m','linewidth',2);
scatter(0,0,'k','filled');
xlabel('X position (mm)','fontsize',12,'fontweight','bold');
ylabel('Z position (mm)','fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');
% relative error
clear; clc;
simu1_x=[0 42.48 72.57 97.88 144.2 175.4
204.8 225.9 235.7 231.9 220.8 202.9];
simu1_z=[267 263.2 255.7 246 218.3 189.7
149.7 101 41.76 -6.401 -45.34 -81.23];
dist_simu1=sqrt(simu1_x.*simu1_x+simu1_z.*simu1_z);
case1_x=[0 40 70 100 146 180 210 235 244
241 233 213];
case1_z=[264 262 256 246 220 190 150 100
40 -10 -50 -89];
dist_case1=sqrt(case1_x.*case1_x+case1_z.*case1_z);
simu2_x=[0 30.07 50.22 79.37 99.6 124.8
139.6 146.6 147 141.6 129.7 117.9];
simu2_z=[267 264 258.4 244.3 229.1 199.7
169.4 139.2 110.8 83.4 56.41 39.44];
dist_simu2=sqrt(simu2_x.*simu2_x+simu2_z.*simu2_z);
case2_x=[0 30 50 80 102 128 140 147 151
149 137 116];
case2_z=[264 261 256 246 230 200 170 140
110 80 50 27];
dist_case2=sqrt(case2_x.*case2_x+case2_z.*case2_z);
simu3_x=[0 13.97 24.22 36.44 47.45 55.6
60.2 64 64.57];
simu3_z=[267 266.5 262.5 256.2 247 236.6
227.6 214 202.8];

```



```

dist_simu3=sqrt(simu3_x.*simu3_x+simu3_z.*simu3_z);
case3_x=[0 12 24 36 47 55 60 64 65];
case3_z=[264 262 260 253 245 235 225 214 202];
dist_case3=sqrt(case3_x.*case3_x+case3_z.*case3_z);
simu4_x=[0 20.93 61.76 110 142.3 170 190.1 193.3 184.3 167.2 143.6 108 67.85];
simu4_z=[267 265.9 257.2 234 207 170 116.5 74.2 29.19 -4.377 -30.52 -51.51 -57.95];
dist_simu4=sqrt(simu4_x.*simu4_x+simu4_z.*simu4_z);
case4_x=[0 20 60 110 150 180 200 208 202 181 149 110 66];
case4_z=[264 263 257 234 210 170 120 70 20 -20 -50 -60 -56];
dist_case4=sqrt(case4_x.*case4_x+case4_z.*case4_z);
det1=sqrt((case1_x-simu1_x).*(case1_x-simu1_x)+(case1_z-simu1_z).*(case1_z-simu1_z));
det2=sqrt((case2_x-simu2_x).*(case2_x-simu2_x)+(case2_z-simu2_z).*(case2_z-simu2_z));
det3=sqrt((case3_x-simu3_x).*(case3_x-simu3_x)+(case3_z-simu3_z).*(case3_z-simu3_z));
det4=sqrt((case4_x-simu4_x).*(case4_x-simu4_x)+(case4_z-simu4_z).*(case4_z-simu4_z));
H=6.4; h0=2.5;
L=(H+h0)*30;
error1 = det1/L*100;
error2 = det2/L*100;
error3 = det3/L*100;
error4 = det4/L*100;

figure; hold on;
plot(error1,'r','linewidth',2);
plot(error2,'g','linewidth',2);
plot(error3,'b','linewidth',2);
plot(error4,'m','linewidth',2);
xlabel('Samples','fontsize',12,'fontweight','bold');
ylabel('Relative Error (%)','fontsize',12,'fontweight','bold');
legend('case 1','case 2','case 3','case 4',2);
plot(mean(error1)*ones(size(dist_simu1)),'--r','linewidth',2);
plot(mean(error2)*ones(size(dist_simu2)),'--g','linewidth',2);
plot(mean(error3)*ones(size(dist_simu3)),'--b','linewidth',2);
plot(mean(error4)*ones(size(dist_simu4)),'--m','linewidth',2);
grid on
set(gca,'fontsize',12,'fontweight','bold');

```

9) Figure 3-29 End Effector Trajectory Comparison - Internal Bilateral Constraint

```

% experiment data
% trajectory without constraint
tx=[-112 -114 -100 -76 -40 0 41 79 105 116 112];
ty=[22 52 92 122 142 150 142 123 92 52 20];
% trajectory with the third joint fixed
cx1=[-80 -61 -35 -20 0 20 39 64 81];
cy1=[94 122 142 147 150 147 140 122 93];
% trajectory with the fifth joint fixed
cx2=[-47 -28 0 26 48];
cy2=[125 142 150 142 124];
figure
plot(tx,ty,'d','linewidth',2);
hold on; axis equal;
plot(cx1,cy1,'dg','linewidth',2);
plot(cx2,cy2,'dr','linewidth',2);
% simulation
N=10; d=15; D=20;
h0=2.5; len=15;
dtheta = 0.1*pi/180; % simulation interval is 0.1 deg
thetamax=2*atan(h0/D); % maximum bending angle for one joint
for i=1:ceil(2*thetamax/dtheta)
    theta(i) = -thetamax+(i-1)*dtheta;
    [sx(i),sy(i)]=UFM(N*len,N,N*theta(i)); % end effector position
    [tmpsx3(i),tmpsy3(i)]=UFM((N-3)*len,(N-3),(N-3)*theta(i)); % end effector position
    [tmpsx5(i),tmpsy5(i)]=UFM((N-5)*len,(N-5),(N-5)*theta(i)); % end effector position
end
plot(sx,sy,'b','linewidth',2); % free path
plot(tmpsx3,tmpsy3+3*len,'g','linewidth',2);
plot(tmpsx5,tmpsy5+5*len,'r','linewidth',2);
xlabel('X Position','fontsize',12,'fontweight','b');
ylabel('Y Position','fontsize',12,'fontweight','b');
set(gca,'fontsize',12,'fontweight','b');
scatter(0,3*len,'sg','linewidth',3);
scatter(0,5*len,'sr','linewidth',3);
% backbone
vx(1)=0; vy(1)=0;
for i=1:N
    vx(i+1)=vx(i); vy(i+1)=vy(i)+len;
end
plot(vx,vy,'-o','LineWidth',2,'color',[1 0 1],...
'MarkerEdgeColor',[1 0 1],...
'MarkerFaceColor','w',...

```

```

        'MarkerSize',5);
% backbone
vx(1)=0; vy(1)=0; Num=3;
for i=1:N
    if i<Num
        vx(i+1)=vx(i)+0;    vy(i+1)=vy(i)+len;
    else
        vx(i+1)=vx(i)+len*sin(Num*0+(i-
Num)*thetamax);
        vy(i+1)=vy(i)+len*cos(Num*0+(i-
Num)*thetamax);
    end
end
plot(vx,vy,'-o','LineWidth',2,'color',[0 1 0],...
     'MarkerEdgeColor',[0 1 0],...
     'MarkerFaceColor','w',...
     'MarkerSize',5);
plot(-vx,vy,'-o','LineWidth',2,'color',[0 1 0],...
     'MarkerEdgeColor',[0 1 0],...
     'MarkerFaceColor','w',...
     'MarkerSize',5);
% backbone
vx(1)=0; vy(1)=0; Num=5;
for i=1:N
    if i<Num
        vx(i+1)=vx(i)+0;    vy(i+1)=vy(i)+len;
    else
        vx(i+1)=vx(i)+len*sin(Num*0+(i-
Num)*thetamax);
        vy(i+1)=vy(i)+len*cos(Num*0+(i-
Num)*thetamax);
    end
end

```

```

end
plot(vx,vy,'-o','LineWidth',2,'color',[1 0 0],...
     'MarkerEdgeColor',[1 0 0],...
     'MarkerFaceColor','w',...
     'MarkerSize',5);
plot(-vx,vy,'-o','LineWidth',2,'color',[1 0 0],...
     'MarkerEdgeColor',[1 0 0],...
     'MarkerFaceColor','w',...
     'MarkerSize',5);
% backbone
vx(1)=0; vy(1)=0; Num=0;
for i=1:N
    if i<Num
        vx(i+1)=vx(i)+0;    vy(i+1)=vy(i)+len;
    else
        vx(i+1)=vx(i)+len*sin(Num*0+(i-
Num)*thetamax);
        vy(i+1)=vy(i)+len*cos(Num*0+(i-
Num)*thetamax);
    end
end
plot(vx,vy,'-o','LineWidth',2,'color',[0 0 1],...
     'MarkerEdgeColor',[0 0 1],...
     'MarkerFaceColor','w',...
     'MarkerSize',5);
plot(-vx,vy,'-o','LineWidth',2,'color',[0 0 1],...
     'MarkerEdgeColor',[0 0 1],...
     'MarkerFaceColor','w',...
     'MarkerSize',5);
legend('Free Path','Fixed N=3','Fixed
N=5',3);

```

10) Figure 3-30 End Effector Trajectory Comparison - External Bilateral Constraint

```

% experiment data
% trajectory without constraint
tx=[-112 -108 -87 -65 -31 0 36 73 95 112
112];
ty=[20 72 108 130 143 150 143 123 96 70
16];
% trajectory with the sixth joint fixed at left
limit
cx1=[-112 -121 -116 -108];
cy1=[20 50 75 91];
% trajectory with the sixth joint fixed at
right limit
cx2=[112 119 123 109];
cy2=[16 32 63 86];
% trajectory with the third joint fixed at left
limit
cx3=[-112 -116 -101 -73 -53 -28 -4 2];
cy3=[20 63 101 128 137 140 135 127];
% trajectory with the third joint fixed at
right limit
cx4=[-2 21 47 84 107 118 118 112];
cy4=[127 134 137 121 97 68 35 16];

figure
plot(tx,ty,'d','linewidth',2);

```

```

hold on; axis equal;
plot(cx1,cy1,'Dr','linewidth',2);
plot(cx4,cy4,'Dg','linewidth',2);

% simulation
N=10; d=15; D=20;
h0=2.5; len=15;
dtheta = 0.1*pi/180; % simulation interval
is 0.1 deg
thetamax=2*atan(h0/D); % maximum
bending angle for one joint
for i=1:ceil(2*thetamax/dtheta)
    theta(i) = -thetamax+(i-1)*dtheta;
    [sx(i),sy(i)]=UFM(N*len,N,N*theta(i)); %
end effector position

[tmplx1(i),tmpsy1(i)]=UFM((N-3)*len,(N-
3),(N-3)*theta(i)); % end effector position
csx1(i)=tmplx1(i)*cos(-
3*thetamax)+tmpsy1(i)*sin(-3*thetamax);
csy1(i)=-tmplx1(i)*sin(-
3*thetamax)+tmpsy1(i)*cos(-3*thetamax);
[tmplx2(i),tmpsy2(i)]=UFM((N-3)*len,(N-
3),(N-3)*theta(i)); % end effector position

```



```

csx2(i)=tmpsx2(i)*cos(3*thetamax)+tmpsy2
(i)*sin(3*thetamax);
csy2(i)=-tmpsx2(i)*sin(3*thetamax)+
tmpsy2(i)*cos(3*thetamax);
[tmpsx3(i),tmpsy3(i)]=UFM((N-6)*len,(N-
6),(N-6)*theta(i)); % end effector position
csx3(i)=tmpsx3(i)*cos(-
6*thetamax)+tmpsy3(i)*sin(-6*thetamax);
csy3(i)=-tmpsx3(i)*sin(-
6*thetamax)+tmpsy3(i)*cos(-6*thetamax);
[tmpsx4(i),tmpsy4(i)]=UFM((N-6)*len,(N-
6),(N-6)*theta(i)); % end effector position
csx4(i)=tmpsx4(i)*cos(6*thetamax)+tmpsy4
(i)*sin(6*thetamax);
csy4(i)=-tmpsx4(i)*sin(6*thetamax)+
tmpsy4(i)*cos(6*thetamax);
end
[bx1,by1]=UFM(3*len,3,-3*thetamax); %
end effector position
[bx2,by2]=UFM(3*len,3,3*thetamax); %
end effector position
[bx3,by3]=UFM(6*len,6,-6*thetamax); %
end effector position
[bx4,by4]=UFM(6*len,6,6*thetamax); %
end effector position
scatter(bx2,by2,'sg','linewidth',3);
scatter(bx3,by3,'sr','linewidth',3);
% figure
plot(sx,sy,'b','linewidth',2); % free path
plot(csx2+bx2,csy2+by2,'g','linewidth',2);
plot(csx3+bx3,csy3+by3,'r','linewidth',2);
xlabel('X Position (mm)');
'fontsize',12,'fontweight','b');
ylabel('Z Position (mm)');
'fontsize',12,'fontweight','b');
set(gca,'fontsize',12,'fontweight','b');
% backbone 2
vx(1)=0; vy(1)=0; vx1(1)=0; vy1(1)=0;
Num=6;
for i=1:N
    if i<Num
        vx(i+1)=vx(i)+len*sin(-i*thetamax);
        vy(i+1)=vy(i)+len*cos(-i*thetamax);
        vx1(i+1)=vx1(i)+len*sin(-i*thetamax);
        vy1(i+1)=vy1(i)+len*cos(-i*thetamax);
    else
        vx(i+1)=vx(i)+len*sin(-
Num*thetamax+(i-Num)*thetamax);
        vy(i+1)=vy(i)+len*cos(-
Num*thetamax+(i-Num)*thetamax);
        vx1(i+1)=vx1(i)+len*sin(-
Num*thetamax-(i-Num)*thetamax);
        vy1(i+1)=vy1(i)+len*cos(-
Num*thetamax-(i-Num)*thetamax);
    end
end
plot(vx,vy,'-o','LineWidth',2,'color',[1 0 0],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',5);
plot(vx1,vy1,'-o','LineWidth',2,'color',[1 0
0],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',5);
% backbone 3
vx(1)=0; vy(1)=0; vx1(1)=0; vy1(1)=0;
Num=3;
for i=1:N
    if i<Num
        vx(i+1)=vx(i)+len*sin(i*thetamax);
        vy(i+1)=vy(i)+len*cos(i*thetamax);
        vx1(i+1)=vx1(i)+len*sin(i*thetamax);
        vy1(i+1)=vy1(i)+len*cos(i*thetamax);
    else
        vx(i+1)=vx(i)+len*sin(Num*thetamax+(i-
Num)*thetamax);
        vy(i+1)=vy(i)+len*cos(Num*thetamax+(i-
Num)*thetamax);
        vx1(i+1)=vx1(i)+len*sin(Num*thetamax-(i-
Num)*thetamax);
        vy1(i+1)=vy1(i)+len*cos(Num*thetamax-(i-
Num)*thetamax);
    end
end
plot(vx,vy,'-o','LineWidth',2,'color',[0 1 0],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',5);
plot(vx1,vy1,'-o','LineWidth',2,'color',[0 1
0],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',5);

```

11) Figure 3-31 End Effector Trajectory Comparison - Unilateral Constraint

```

% experiment data
% trajectory with constraint
tx=[-112 -113 -99 -75 -36 0 44 86 101 114
110];
ty=[20 57 94 120 143 150 140 112 84 50
20];
% left trajectory with block
cx1=[-93 -89 -83 -71];
cy1=[75 94 105 123];
% right trajectory with block
cx2=[69 66 59 51];
cy2=[110 119 128 135];
figure
plot(tx,ty,'d','linewidth',2);
hold on; axis equal;
plot(cx1,cy1,'dr','linewidth',2);
plot(cx2,cy2,'dr','linewidth',2);
% simulation
N=10; d=15; D=20; h0=2.5; len=15;

```

```

dtheta = 0.1*pi/180; % simulation interval
is 0.1 deg
thetamax=2*atan(h0/D); % maximum
bending angle for one joint
for i=1:ceil(2*thetamax/dtheta)
    theta(i) = -thetamax+(i-1)*dtheta;
    [sx(i),sy(i)]=UFM(N*len,N,N*theta(i)); %
end effector position
end
% figure
plot(sx,sy,'b','linewidth',2); % free path
xlabel('X Position
(mm)','fontsize',12,'fontweight','b');
ylabel('Z Position
(mm)','fontsize',12,'fontweight','b');
set(gca,'fontsize',12,'fontweight','b');
% simulation
% left part
clear sx sy;
lcx=-6; lcy=40;
[Num,Ltheta,ln,xn,yn]=solveConstraint(N,N
*len,-lcx,lcy,thetamax)
Ltheta=-Ltheta;
xn=-xn;
for i=1:ceil((thetamax+Ltheta)/dtheta)+1
    theta(i) = -thetamax+(i-1)*dtheta;
    [tmpsx(i),tmpsy(i)]=UFM((N-
Num)*len,(N-Num),(N-Num)*theta(i)); %
end effector position
sx(i)=cos(Num*Ltheta)*tmpsx(i)+sin(Num*
Ltheta)*tmpsy(i);
sy(i)=-
sin(Num*Ltheta)*tmpsx(i)+cos(Num*Lthet
a)*tmpsy(i);
end
plot(xn+sx,yn+sy,'r','linewidth',2);
hold on;
scatter(lcx,lcy,'sk','linewidth',2);
% backbone
vx(1)=0; vy(1)=0;
for i=1:N
    if i<Num
        vx(i+1)=vx(i)+len*sin(i*Ltheta);
        vy(i+1)=vy(i)+len*cos(i*Ltheta);
    else
        vx(i+1)=vx(i)+len*sin(Num*Ltheta-(i-
Num)*thetamax);
        vy(i+1)=vy(i)+len*cos(Num*Ltheta-(i-
Num)*thetamax);
    end
end
plot(vx,vy,'-o','LineWidth',2,'color',[0 1 0],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',5);
% right part

```

12) Figure 4-2 SPSP WDM Static Analysis: (a) Deformed Backbone Curve; (b) Joint Rotations

% serpentine WDM statics

```

clear sx sy xn yn tmpsx tmpsy theta;
rcx=10; rcy=68;
[Num,Rtheta,ln,xn,yn]=solveConstraint(N,N
*len,rcx,rcy,thetamax);
for i=1:ceil((thetamax-Rtheta)/dtheta)+1
    theta(i) = Rtheta+(i-1)*dtheta;
    [tmpsx(i),tmpsy(i)]=UFM((N-
Num)*len,(N-Num),(N-Num)*theta(i)); %
end effector position
sx(i)=cos(Num*Rtheta)*tmpsx(i)+sin(Num*
Rtheta)*tmpsy(i);
sy(i)=-
sin(Num*Rtheta)*tmpsx(i)+cos(Num*Rthet
a)*tmpsy(i);
end
plot(xn+sx,yn+sy,'r','linewidth',2);
hold on;
scatter(rcx,rcy,'sk','linewidth',2);
% backbone
vx(1)=0; vy(1)=0;
for i=1:N
    if i<Num
        vx(i+1)=vx(i)+len*sin(i*Rtheta);
        vy(i+1)=vy(i)+len*cos(i*Rtheta);
    else
        vx(i+1)=vx(i)+len*sin(Num*Rtheta+(i-
Num)*thetamax);
        vy(i+1)=vy(i)+len*cos(Num*Rtheta+(i-
Num)*thetamax);
    end
end
plot(vx,vy,'-o','LineWidth',2,'color',[0 1 0],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',5);
% middle part
clear theta sx sy;
for i=1:ceil((Rtheta-Ltheta)/dtheta)
    theta(i) = Ltheta+(i-1)*dtheta;
    [sx(i),sy(i)]=UFM(N*len,N,N*theta(i)); %
end effector position
end
plot(sx,sy,'r','linewidth',2);
% backbone
vx(1)=0; vy(1)=0;
for i=1:N
    vx(i+1)=vx(i);    vy(i+1)=vy(i)+len;
end
plot(vx,vy,'-o','LineWidth',2,'color',[1 0 1],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',5);
% other constraints
block_x=[-30 35 -60 50 75 -80];
block_y=[60 30 30 85 80 60];
scatter(block_x,block_y,'sk','linewidth',2);

```

% parameters:


```

% external loads
Fex=0.1; Fey=0.2; Me=0.015;
% control force
T1=0; T2=0; T=T1+T2;
% vertebra parameters
H=12.5e-3; h0=2.5e-3;
len=H+h0; d=10e-3; N=10;
% elastic tube parameters
E=1.5e9; pi=3.1415926;
r1=2.5e-3; r2=2.0e-3;
I=pi*(r1^4-r2^4)/64;
K=E*I/h0;
% intended angle;
BTheta = 45*pi/180;
alfa = BTheta/N;
% from kinematics;
L0=N*(H+h0)
dL1=N*(d*sin(alfa/2)+2*h0*sin(alfa/4)^2);
dL2=N*(d*sin(alfa/2)-2*h0*sin(alfa/4)^2);
xc=zeros(N+1,1);
yc=zeros(N+1,1);
% constant curve assumption
for i=1:N
    xc(i+1)=xc(i)+len*cos(i*alfa);
    yc(i+1)=yc(i)+len*sin(i*alfa);
end
% initialization
theta=zeros(N,1); Mo = zeros(N,1);
Fx=zeros(N,1); Fy=zeros(N,1);

syms alfa1 alfa2 alfa3 alfa4 alfa5 alfa6 alfa7
alfa8 alfa9 alfa10;
syms F1x F2x F3x F4x F5x F6x F7x F8x
F9x F10x;
syms F1y F2y F3y F4y F5y F6y F7y F8y
F9y F10y;
syms theta1 theta2 theta3 theta4 theta5
theta6 theta7 theta8 theta9 theta10;
global num
num=0;
result1=fsolve('deformedshape1',zeros(10,1)
);
result2=fsolve('deformedshape2',zeros(10,1)
);
result3=fsolve('deformedshape3',zeros(10,1)
);
x1=zeros(N+1,1); y1=zeros(N+1,1);
x2=zeros(N+1,1); y2=zeros(N+1,1);
x3=zeros(N+1,1); y3=zeros(N+1,1);
% constant curve assumption
for i=1:N
    x1(i+1)=x1(i)+len*cos(result1(i));
    y1(i+1)=y1(i)+len*sin(result1(i));
    x2(i+1)=x2(i)+len*cos(result2(i));
    y2(i+1)=y2(i)+len*sin(result2(i));
    x3(i+1)=x3(i)+len*cos(result3(i));
    y3(i+1)=y3(i)+len*sin(result3(i));
end
figure;

plot(x1*1000,y1*1000,'-
o','LineWidth',2.5,'color',[1 0 0],...
'MarkerEdgeColor',[1 0 0],...
'MarkerFaceColor','w',...
'MarkerSize',6);
hold on;
plot(x2*1000,y2*1000,'-
o','LineWidth',2.5,'color',[0 1 0],...
'MarkerEdgeColor',[0 1 0],...
'MarkerFaceColor','w',...
'MarkerSize',6);
plot(x3*1000,y3*1000,'-
o','LineWidth',2.5,'color',[0 0 1],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',6);
axis equal;
xlabel('X
(mm)','FontSize',12,'fontweight','bold');
ylabel('Y
(mm)','FontSize',12,'fontweight','bold');
legend('M=0.1 Nm,Fx=0 N, Fy=0 N',...
'M=0.1 Nm,Fx=1 N, Fy=0 N',...
'M=0.1 Nm,Fx=0 N, Fy=1 N',2);
set(gca,'FontSize',12,'fontweight','bold');
figure;
ang1=result1; ang2=result2; ang3=result3;
for i=2:10
    ang1(i)=result1(i)-result1(i-1);
    ang2(i)=result2(i)-result2(i-1);
    ang3(i)=result3(i)-result3(i-1);
end
plot(ang1*180/pi,'--
o','LineWidth',2.5,'color',[1 0 0],...
'MarkerEdgeColor',[1 0 0],...
'MarkerFaceColor','w',...
'MarkerSize',6);
hold on;
plot(ang2*180/pi,'--
o','LineWidth',2.5,'color',[0 1 0],...
'MarkerEdgeColor',[0 1 0],...
'MarkerFaceColor','w',...
'MarkerSize',6);
plot(ang3*180/pi,'--
o','LineWidth',2.5,'color',[0 0 1],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',6);
xlabel('Joint
Num','FontSize',12,'fontweight','bold');
ylabel('Joint
Rotation
(^o)','FontSize',12,'fontweight','bold');
legend('M=0.1 Nm,Fx=0 N, Fy=0 N',...
'M=0.1 Nm,Fx=1 N, Fy=0 N',...
'M=0.1 Nm,Fx=0 N, Fy=1 N',1);
set(gca,'fontSize',12,'fontweight','bold');

function eq=deformedshape1(Theta)
global num
num=num+1;

```



```

H=12.5e-3; h0=2.5e-3;
len=H+h0; d=10e-3;
E=1.5e9;
pi=3.1415926;
r1=2.5e-3; r2=2.0e-3;
I=pi*(r1^4-r2^4)/64;
K=E*I/h0;
Fex=0; Fey=0.0; Me=0.1;
T1=0; T2=0;
T=T1+T2;
M=(T1-T2)*d;
Fx=-Fex+(T1+T2)*cos(Theta(10));
Fy=-Fey+(T1+T2)*sin(Theta(10));

```

```

eq(1)=K*Theta(1)-K*(Theta(2)-Theta(1))-
len*(Fx*sin(Theta(1))-Fy*cos(Theta(1)));
eq(2)=K*(Theta(2)-Theta(1))-K*(Theta(3)-
Theta(2))-len*(Fx*sin(Theta(2))-
Fy*cos(Theta(2)));
eq(3)=K*(Theta(3)-Theta(2))-K*(Theta(4)-
Theta(3))-len*(Fx*sin(Theta(3))-
Fy*cos(Theta(3)));
eq(4)=K*(Theta(4)-Theta(3))-K*(Theta(5)-
Theta(4))-len*(Fx*sin(Theta(4))-
Fy*cos(Theta(4)));
eq(5)=K*(Theta(5)-Theta(4))-K*(Theta(6)-
Theta(5))-len*(Fx*sin(Theta(5))-
Fy*cos(Theta(5)));
eq(6)=K*(Theta(6)-Theta(5))-K*(Theta(7)-
Theta(6))-len*(Fx*sin(Theta(6))-
Fy*cos(Theta(6)));
eq(7)=K*(Theta(7)-Theta(6))-K*(Theta(8)-
Theta(7))-len*(Fx*sin(Theta(7))-
Fy*cos(Theta(7)));
eq(8)=K*(Theta(8)-Theta(7))-K*(Theta(9)-
Theta(8))-len*(Fx*sin(Theta(8))-
Fy*cos(Theta(8)));
eq(9)=K*(Theta(9)-Theta(8))-K*(Theta(10)-
Theta(9))-len*(Fx*sin(Theta(9))-
Fy*cos(Theta(9)));
eq(10)=K*(Theta(10)-Theta(9))-Me-
len*(Fex*sin(Theta(10))-
Fey*cos(Theta(10)));

```

```

function eq=deformedshape2(Theta)
global num
num=num+1;
H=12.5e-3; h0=2.5e-3;
len=H+h0;
d=10e-3;
E=1.5e9;
pi=3.1415926;
r1=2.5e-3; r2=2.0e-3;
I=pi*(r1^4-r2^4)/64;
K=E*I/h0;
Fex=1; Fey=0.0; Me=0.1;
T1=0; T2=0;
T=T1+T2;
M=(T1-T2)*d;
Fx=-Fex+(T1+T2)*cos(Theta(10));

```

```

Fy=-Fey+(T1+T2)*sin(Theta(10));
eq(1)=K*Theta(1)-K*(Theta(2)-Theta(1))-
len*(Fx*sin(Theta(1))-Fy*cos(Theta(1)));
eq(2)=K*(Theta(2)-Theta(1))-K*(Theta(3)-
Theta(2))-len*(Fx*sin(Theta(2))-
Fy*cos(Theta(2)));
eq(3)=K*(Theta(3)-Theta(2))-K*(Theta(4)-
Theta(3))-len*(Fx*sin(Theta(3))-
Fy*cos(Theta(3)));
eq(4)=K*(Theta(4)-Theta(3))-K*(Theta(5)-
Theta(4))-len*(Fx*sin(Theta(4))-
Fy*cos(Theta(4)));
eq(5)=K*(Theta(5)-Theta(4))-K*(Theta(6)-
Theta(5))-len*(Fx*sin(Theta(5))-
Fy*cos(Theta(5)));
eq(6)=K*(Theta(6)-Theta(5))-K*(Theta(7)-
Theta(6))-len*(Fx*sin(Theta(6))-
Fy*cos(Theta(6)));
eq(7)=K*(Theta(7)-Theta(6))-K*(Theta(8)-
Theta(7))-len*(Fx*sin(Theta(7))-
Fy*cos(Theta(7)));
eq(8)=K*(Theta(8)-Theta(7))-K*(Theta(9)-
Theta(8))-len*(Fx*sin(Theta(8))-
Fy*cos(Theta(8)));
eq(9)=K*(Theta(9)-Theta(8))-K*(Theta(10)-
Theta(9))-len*(Fx*sin(Theta(9))-
Fy*cos(Theta(9)));
eq(10)=K*(Theta(10)-Theta(9))-Me-
len*(Fex*sin(Theta(10))-
Fey*cos(Theta(10)));

```

```

function eq=deformedshape3(Theta)
global num
num=num+1;
H=12.5e-3; h0=2.5e-3;
len=H+h0;
d=10e-3;
E=1.5e9;
pi=3.1415926;
r1=2.5e-3; r2=2.0e-3;
I=pi*(r1^4-r2^4)/64;
K=E*I/h0;
Fex=0; Fey=1; Me=0.1;
T1=0; T2=0;
T=T1+T2;
M=(T1-T2)*d;
Fx=-Fex+(T1+T2)*cos(Theta(10));
Fy=-Fey+(T1+T2)*sin(Theta(10));
eq(1)=K*Theta(1)-K*(Theta(2)-Theta(1))-
len*(Fx*sin(Theta(1))-Fy*cos(Theta(1)));
eq(2)=K*(Theta(2)-Theta(1))-K*(Theta(3)-
Theta(2))-len*(Fx*sin(Theta(2))-
Fy*cos(Theta(2)));
eq(3)=K*(Theta(3)-Theta(2))-K*(Theta(4)-
Theta(3))-len*(Fx*sin(Theta(3))-
Fy*cos(Theta(3)));
eq(4)=K*(Theta(4)-Theta(3))-K*(Theta(5)-
Theta(4))-len*(Fx*sin(Theta(4))-
Fy*cos(Theta(4)));

```

```

eq(5)=K*(Theta(5)-Theta(4))-K*(Theta(6)-
Theta(5))-len*(Fx*sin(Theta(5))-
Fy*cos(Theta(5)));
eq(6)=K*(Theta(6)-Theta(5))-K*(Theta(7)-
Theta(6))-len*(Fx*sin(Theta(6))-
Fy*cos(Theta(6)));
eq(7)=K*(Theta(7)-Theta(6))-K*(Theta(8)-
Theta(7))-len*(Fx*sin(Theta(7))-
Fy*cos(Theta(7)));

```

```

eq(8)=K*(Theta(8)-Theta(7))-K*(Theta(9)-
Theta(8))-len*(Fx*sin(Theta(8))-
Fy*cos(Theta(8)));
eq(9)=K*(Theta(9)-Theta(8))-K*(Theta(10)-
Theta(9))-len*(Fx*sin(Theta(9))-
Fy*cos(Theta(9)));
eq(10)=K*(Theta(10)-Theta(9))-Me-
len*(Fex*sin(Theta(10))-
Fey*cos(Theta(10)));

```

13) Figure 4-3 SPSP WDM Backbone Reaches a Desired Position: (a) Deformed Backbone Curve; (b) Joint Rotations

```

%% Controlling forces
pi=3.1415926;
N=10; len=15e-3; d=10e-3;
global num
num=0;
result1=fsolve('actuationload1',zeros(14,1));
result2=fsolve('actuationload2',zeros(14,1));
result3=fsolve('actuationload3',zeros(14,1));
result4=fsolve('actuationload4',zeros(14,1));

T11=0.5*(result1(11)+result1(14)/d);
T12=0.5*(result1(11)-result1(14)/d);
T21=0.5*(result2(11)+result2(14)/d);
T22=0.5*(result2(11)-result2(14)/d);
T31=0.5*(result3(11)+result3(14)/d);
T32=0.5*(result3(11)-result3(14)/d);
T41=0.5*(result4(11)+result4(14)/d);
T42=0.5*(result4(11)-result4(14)/d);

x1=zeros(N+1,1); y1=zeros(N+1,1);
x2=zeros(N+1,1); y2=zeros(N+1,1);
x3=zeros(N+1,1); y3=zeros(N+1,1);
x4=zeros(N+1,1); y4=zeros(N+1,1);
% constant curve assumption
for i=1:N
    x1(i+1)=x1(i)+len*cos(result1(i));
    y1(i+1)=y1(i)+len*sin(result1(i));
    x2(i+1)=x2(i)+len*cos(result2(i));
    y2(i+1)=y2(i)+len*sin(result2(i));
    x3(i+1)=x3(i)+len*cos(result3(i));
    y3(i+1)=y3(i)+len*sin(result3(i));
    x4(i+1)=x4(i)+len*cos(result4(i));
    y4(i+1)=y4(i)+len*sin(result4(i));
end
figure;
plot(x1*1000,y1*1000,'-
o','LineWidth',2.5,'color',[1 0 0],...
'MarkerEdgeColor',[1 0 0],...
'MarkerFaceColor','w',...
'MarkerSize',6);
hold on;
plot(x2*1000,y2*1000,'-
o','LineWidth',2.5,'color',[0 1 0],...
'MarkerEdgeColor',[0 1 0],...
'MarkerFaceColor','w',...
'MarkerSize',6);
plot(x3*1000,y3*1000,'-
o','LineWidth',2.5,'color',[0 0 1],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',6);
plot(x4*1000,y4*1000,'-
o','LineWidth',2.5,'color',[1 0 1],...
'MarkerEdgeColor',[1 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',6);

```

```

'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',6);
plot(x4*1000,y4*1000,'-
o','LineWidth',2.5,'color',[1 0 1],...
'MarkerEdgeColor',[1 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',6);
axis equal;
xlabel('X
(mm)','fontsize',12,'fontweight','bold');
ylabel('Y
(mm)','fontsize',12,'fontweight','bold');
legend('M_e=0 Nm,F_e_x=0 N, F_e_y=0 N',...
'M_e=0.1 Nm,F_e_x=0 N, F_e_y=0 N',...
'M_e=0 Nm,F_e_x=1 N, F_e_y=0 N',...
'M_e=0 Nm,F_e_x=0 N, F_e_y=1 N',2);
set(gca,'fontsize',12,'fontweight','bold');
figure;
ang1=result1;
ang2=result2; ang3=result3; ang4=result4;
for i=2:10
    ang1(i)=result1(i)-result1(i-1);
    ang2(i)=result2(i)-result2(i-1);
    ang3(i)=result3(i)-result3(i-1);
    ang4(i)=result4(i)-result4(i-1);
end
plot(ang1(1:10)*180/pi,'-
o','LineWidth',2.5,'color',[1 0 0],...
'MarkerEdgeColor',[1 0 0],...
'MarkerFaceColor','w',...
'MarkerSize',6);
hold on;
plot(ang2(1:10)*180/pi,'-
o','LineWidth',2.5,'color',[0 1 0],...
'MarkerEdgeColor',[0 1 0],...
'MarkerFaceColor','w',...
'MarkerSize',6);
plot(ang3(1:10)*180/pi,'-
o','LineWidth',2.5,'color',[0 0 1],...
'MarkerEdgeColor',[0 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',6);
plot(ang4(1:10)*180/pi,'-
o','LineWidth',2.5,'color',[1 0 1],...
'MarkerEdgeColor',[1 0 1],...
'MarkerFaceColor','w',...
'MarkerSize',6);

```



```

        'MarkerSize',6);
xlabel('Joint
Num','fontsize',12,'fontweight','bold');
ylabel('Joint
Rotation
(^o)','fontsize',12,'fontweight','bold');
legend('M_e=0 Nm,F_e_x=0 N, F_e_y=0
N',...
'M_e=0.1 Nm,F_e_x=0 N, F_e_y=0 N',...
'M_e=0 Nm,F_e_x=1 N, F_e_y=0 N',...
'M_e=0 Nm,F_e_x=0 N, F_e_y=1 N',3);
set(gca,'fontsize',12,'fontweight','bold');

```

```

function eq=actuationload1(Theta)
global num
num=num+1;
% desired position and orientation
x=125e-3; y=75e-3; ang=1.1161;
% external load
Fex=0; Fey=0; Me=0.0;
H=12.5e-3; h0=2.5e-3; len=H+h0;
d=10e-3; E=1.5e9; pi=3.1415926;
r1=2.5e-3; r2=2.0e-3;
I=pi*(r1^4-r2^4)/64;
K=E*I/h0;
% vertebra (1-9)
eq(1)=K*Theta(1)-K*(Theta(2)-Theta(1))-
len*(Theta(12)*sin(Theta(1))-
Theta(13)*cos(Theta(1)));
eq(2)=K*(Theta(2)-Theta(1))-K*(Theta(3)-
Theta(2))-len*(Theta(12)*sin(Theta(2))-
Theta(13)*cos(Theta(2)));
eq(3)=K*(Theta(3)-Theta(2))-K*(Theta(4)-
Theta(3))-len*(Theta(12)*sin(Theta(3))-
Theta(13)*cos(Theta(3)));
eq(4)=K*(Theta(4)-Theta(3))-K*(Theta(5)-
Theta(4))-len*(Theta(12)*sin(Theta(4))-
Theta(13)*cos(Theta(4)));
eq(5)=K*(Theta(5)-Theta(4))-K*(Theta(6)-
Theta(5))-len*(Theta(12)*sin(Theta(5))-
Theta(13)*cos(Theta(5)));
eq(6)=K*(Theta(6)-Theta(5))-K*(Theta(7)-
Theta(6))-len*(Theta(12)*sin(Theta(6))-
Theta(13)*cos(Theta(6)));
eq(7)=K*(Theta(7)-Theta(6))-K*(Theta(8)-
Theta(7))-len*(Theta(12)*sin(Theta(7))-
Theta(13)*cos(Theta(7)));
eq(8)=K*(Theta(8)-Theta(7))-K*(Theta(9)-
Theta(8))-len*(Theta(12)*sin(Theta(8))-
Theta(13)*cos(Theta(8)));
eq(9)=K*(Theta(9)-Theta(8))-K*(Theta(10)-
Theta(9))-len*(Theta(12)*sin(Theta(9))-
Theta(13)*cos(Theta(9)));
% last vertebra
eq(10)=K*(Theta(10)-Theta(9))-Me-
Theta(14)+len*(Fex*sin(Theta(10))-
Fey*cos(Theta(10))); % M
eq(11)=-Theta(12)-
Fex+Theta(11)*cos(Theta(10)); % Fx
eq(12)=-Theta(13)-
Fey+Theta(11)*sin(Theta(10)); % Fy

```

```

% boundary condition
eq(13)=x-len*sum(cos(Theta(1:10)));
eq(14)=y-len*sum(sin(Theta(1:10)));

```

```

function eq=actuationload2(Theta)
global num
num=num+1;
% desired position and orientation
x=125e-3; y=75e-3; ang=1.1161;
% external load
Fex=0; Fey=0; Me=0.1;
H=12.5e-3; h0=2.5e-3;
len=H+h0; d=10e-3;
E=1.5e9;
pi=3.1415926;
r1=2.5e-3; r2=2.0e-3;
I=pi*(r1^4-r2^4)/64;
K=E*I/h0;
% vertebra (1-9)
eq(1)=K*Theta(1)-K*(Theta(2)-Theta(1))-
len*(Theta(12)*sin(Theta(1))-
Theta(13)*cos(Theta(1)));
eq(2)=K*(Theta(2)-Theta(1))-K*(Theta(3)-
Theta(2))-len*(Theta(12)*sin(Theta(2))-
Theta(13)*cos(Theta(2)));
eq(3)=K*(Theta(3)-Theta(2))-K*(Theta(4)-
Theta(3))-len*(Theta(12)*sin(Theta(3))-
Theta(13)*cos(Theta(3)));
eq(4)=K*(Theta(4)-Theta(3))-K*(Theta(5)-
Theta(4))-len*(Theta(12)*sin(Theta(4))-
Theta(13)*cos(Theta(4)));
eq(5)=K*(Theta(5)-Theta(4))-K*(Theta(6)-
Theta(5))-len*(Theta(12)*sin(Theta(5))-
Theta(13)*cos(Theta(5)));
eq(6)=K*(Theta(6)-Theta(5))-K*(Theta(7)-
Theta(6))-len*(Theta(12)*sin(Theta(6))-
Theta(13)*cos(Theta(6)));
eq(7)=K*(Theta(7)-Theta(6))-K*(Theta(8)-
Theta(7))-len*(Theta(12)*sin(Theta(7))-
Theta(13)*cos(Theta(7)));
eq(8)=K*(Theta(8)-Theta(7))-K*(Theta(9)-
Theta(8))-len*(Theta(12)*sin(Theta(8))-
Theta(13)*cos(Theta(8)));
eq(9)=K*(Theta(9)-Theta(8))-K*(Theta(10)-
Theta(9))-len*(Theta(12)*sin(Theta(9))-
Theta(13)*cos(Theta(9)));
% last vertebra
eq(10)=K*(Theta(10)-Theta(9))-Me-
Theta(14)+len*(Fex*sin(Theta(10))-
Fey*cos(Theta(10))); % M
eq(11)=-Theta(12)-
Fex+Theta(11)*cos(Theta(10)); % Fx
eq(12)=-Theta(13)-
Fey+Theta(11)*sin(Theta(10)); % Fy
% boundary condition
eq(13)=x-len*sum(cos(Theta(1:10)));
eq(14)=y-len*sum(sin(Theta(1:10)));

```

```

function eq=actuationload3(Theta)
global num

```

```

num=num+1;
% desired position and orientation
x=125e-3; y=75e-3; ang=1.1161;
% external load
Fex=1; Fey=0; Me=0.0;
H=12.5e-3; h0=2.5e-3;
len=H+h0; d=10e-3;
E=1.5e9; pi=3.1415926;
r1=2.5e-3; r2=2.0e-3;
I=pi*(r1^4-r2^4)/64;
K=E*I/h0;
% vertebrae (1-9)
eq(1)=K*Theta(1)-K*(Theta(2)-Theta(1))-
len*(Theta(12)*sin(Theta(1))-
Theta(13)*cos(Theta(1)));
eq(2)=K*(Theta(2)-Theta(1))-K*(Theta(3)-
Theta(2))-len*(Theta(12)*sin(Theta(2))-
Theta(13)*cos(Theta(2)));
eq(3)=K*(Theta(3)-Theta(2))-K*(Theta(4)-
Theta(3))-len*(Theta(12)*sin(Theta(3))-
Theta(13)*cos(Theta(3)));
eq(4)=K*(Theta(4)-Theta(3))-K*(Theta(5)-
Theta(4))-len*(Theta(12)*sin(Theta(4))-
Theta(13)*cos(Theta(4)));
eq(5)=K*(Theta(5)-Theta(4))-K*(Theta(6)-
Theta(5))-len*(Theta(12)*sin(Theta(5))-
Theta(13)*cos(Theta(5)));
eq(6)=K*(Theta(6)-Theta(5))-K*(Theta(7)-
Theta(6))-len*(Theta(12)*sin(Theta(6))-
Theta(13)*cos(Theta(6)));
eq(7)=K*(Theta(7)-Theta(6))-K*(Theta(8)-
Theta(7))-len*(Theta(12)*sin(Theta(7))-
Theta(13)*cos(Theta(7)));
eq(8)=K*(Theta(8)-Theta(7))-K*(Theta(9)-
Theta(8))-len*(Theta(12)*sin(Theta(8))-
Theta(13)*cos(Theta(8)));
eq(9)=K*(Theta(9)-Theta(8))-K*(Theta(10)-
Theta(9))-len*(Theta(12)*sin(Theta(9))-
Theta(13)*cos(Theta(9)));
% last vertebra
eq(10)=K*(Theta(10)-Theta(9))-Me-
Theta(14)+len*(Fex*sin(Theta(10))-
Fey*cos(Theta(10))); % M
eq(11)=-Theta(12)-
Fex+Theta(11)*cos(Theta(10)); % Fx
eq(12)=-Theta(13)-
Fey+Theta(11)*sin(Theta(10)); % Fy
% boundary condition
eq(13)=x-len*sum(cos(Theta(1:10)));
eq(14)=y-len*sum(sin(Theta(1:10)));

function eq=actuationload4(Theta)

```

14) Figure 4-5

Figure 4-6 SPCP WDM Backbone Deformation under Different Loading Conditions

```

% continuum statics
% system parameters
L=500e-3; E=1.5e9;
h=1e-3; w=40e-3;

```

```

global num
num=num+1;
% desired position and orientation
x=125e-3; y=75e-3; ang=1.1161;
% external load
Fex=0; Fey=1; Me=0.0;
H=12.5e-3; h0=2.5e-3;
len=H+h0; d=10e-3;
E=1.5e9; pi=3.1415926;
r1=2.5e-3; r2=2.0e-3;
I=pi*(r1^4-r2^4)/64;
K=E*I/h0;
% vertebrae (1-9)
eq(1)=K*Theta(1)-K*(Theta(2)-Theta(1))-
len*(Theta(12)*sin(Theta(1))-
Theta(13)*cos(Theta(1)));
eq(2)=K*(Theta(2)-Theta(1))-K*(Theta(3)-
Theta(2))-len*(Theta(12)*sin(Theta(2))-
Theta(13)*cos(Theta(2)));
eq(3)=K*(Theta(3)-Theta(2))-K*(Theta(4)-
Theta(3))-len*(Theta(12)*sin(Theta(3))-
Theta(13)*cos(Theta(3)));
eq(4)=K*(Theta(4)-Theta(3))-K*(Theta(5)-
Theta(4))-len*(Theta(12)*sin(Theta(4))-
Theta(13)*cos(Theta(4)));
eq(5)=K*(Theta(5)-Theta(4))-K*(Theta(6)-
Theta(5))-len*(Theta(12)*sin(Theta(5))-
Theta(13)*cos(Theta(5)));
eq(6)=K*(Theta(6)-Theta(5))-K*(Theta(7)-
Theta(6))-len*(Theta(12)*sin(Theta(6))-
Theta(13)*cos(Theta(6)));
eq(7)=K*(Theta(7)-Theta(6))-K*(Theta(8)-
Theta(7))-len*(Theta(12)*sin(Theta(7))-
Theta(13)*cos(Theta(7)));
eq(8)=K*(Theta(8)-Theta(7))-K*(Theta(9)-
Theta(8))-len*(Theta(12)*sin(Theta(8))-
Theta(13)*cos(Theta(8)));
eq(9)=K*(Theta(9)-Theta(8))-K*(Theta(10)-
Theta(9))-len*(Theta(12)*sin(Theta(9))-
Theta(13)*cos(Theta(9)));
% last vertebra
eq(10)=K*(Theta(10)-Theta(9))-Mc-
Theta(14)+len*(Fex*sin(Theta(10))-
Fey*cos(Theta(10))); % M
eq(11)=-Theta(12)-
Fex+Theta(11)*cos(Theta(10)); % Fx
eq(12)=-Theta(13)-
Fey+Theta(11)*sin(Theta(10)); % Fy
% boundary condition
eq(13)=x-len*sum(cos(Theta(1:10)));
eq(14)=y-len*sum(sin(Theta(1:10)));

```

```

I=w*h^3/12;
% Pure Moment at the end
M=10E-3;
% Calculate Horizontal displacement

```



```

Gyita=inline('M*(L-deta).*(yita-
1)/E/I','M','L','deta','E','I','yita');
Gx=inline('M.*(x-
L+deta)/E/I','M','L','deta','E','I','x');
dydx=inline('G./sqrt(1-G.^2)','G');
Fyita=inline('1./sqrt(1-Gyita.^2)','Gyita');
Fx=inline('sqrt(1+Gx.^2)','Gx');
Fg=inline('G./sqrt(1-G.^2)','G');
% compute deta
tmpdeta=0.1*L; % assumption
snum=10; % simpson's one third rule
tmpL=0;
% compute for maximum M
count=0;
while(abs(L-tmpL)>0.001*L)
    lamda=(L-tmpdeta)/snum;
    x=linspace(0,L-tmpdeta,snum+1);
    G=Gyita(M,L,tmpdeta,E,I,x/(L-tmpdeta));
% dy=dydx(G);
    F=Fyita(G);
    tmpL=(L-
tmpdeta)*Simpson10(F,lamda/(L-tmpdeta));
    deta=tmpdeta;
    if L<tmpL
        tmpdeta=tmpdeta+0.382*tmpdeta;
    else
        tmpdeta=tmpdeta-0.382*tmpdeta;
    end
    disp(['L=' num2str(tmpL)]);
    disp(['deta=' num2str(deta)]);
    disp(['percent(%)='
num2str(deta/L*100)]);
    count=count+1
    if count>1000
        break;
    end
end
% tip orientation
G0=Gx(M,L,deta,E,I,0);
dydx0=dydx(G0);
theta0=atan(dydx0)*180/pi;
% plot deformed curve - integration
xL=0; yL=0;
xp=linspace(0,L-deta,101);
for i=1:101 % calculate vertical
displacement and orientation
    lamda=xp(i)/snum;
    tmpx=linspace(0,xp(i),11);
    G=Gx(M,L,deta,E,I,tmpx);
    F=Fg(G);
    yp(i)=simpson10(F,lamda);
end
figure
x1=flipr(xp); y1=min(yp)-yp;
plot(x1,y1,'color','r','linewidth',2);
% deformed curve - constant curvature
model
hold on;
THETA=-theta0*pi/180;
theta=linspace(0,THETA,101);

```

```

R=abs(L/THETA);
for i=1:101
    x2(i)=R*sin(theta(i));
    y2(i)=-R*(1-cos(theta(i)));
end
plot(x2,y2,'b--','linewidth',2);
legend('Nonlinear Euler-Bernoulli
model','Constant Curvature Model');
xlabel('X
(m)','fontsize',12,'fontweight','bold');
ylabel('Y
(m)','fontsize',12,'fontweight','bold');
title('model compare');
xlim([0 L]); axis equal;
set(gca,'fontsize',12,'fontweight','bold');
% force and moment (M=0.01, F=0)
clear;
% system parameters
L=500e-3; E=1.5e9;
h=1e-3; w=40e-3;
I=w*h^3/12;
M=10E-3; F=0;
% Calculate Horizontal displacement
deta=0.1*L;
% anonymous function
solve_theta=@(theta0)theta0*(L-deta)-
L*sin(theta0);
theta0=fsolve(solve_theta,0.5);
C=-(M*(L-deta)-0.5*F*(L-
deta)^2*(sin(theta0)-
cos(theta0)*tan(theta0/2)))/E/I;
Gx=@(x,theta0,C) (M.*x-
0.5*F*x.^2*(sin(theta0)-
cos(theta0)*tan(theta0/2)))/E/I+C;
dydx=inline('G./sqrt(1-G.^2)','G');
Fx=inline('sqrt(1+Gx.^2)','Gx');
% compute deta
snum=10; % simpson method number
tmpL=0; count=0;
tmpdeta=deta;
while(abs(L-tmpL)>0.005*L)
    lamda=(L-tmpdeta)/snum;
    x=linspace(0,L-tmpdeta,snum+1);
    solve_theta=@(theta0)theta0*(L-
tmpdeta)-L*sin(theta0);
    theta0=fsolve(solve_theta,0.55);
    C=-(M*(L-tmpdeta)-0.5*F*(L-
tmpdeta)^2*(sin(theta0)-
cos(theta0)*tan(theta0/2)))/E/I;
    G=Gx(x,theta0,C);
    dy=dydx(G);
    FX=Fx(dy);
    tmpL=(L-
tmpdeta)*Simpson10(FX,lamda/(L-
tmpdeta));
    deta=tmpdeta;
    if L<tmpL
        tmpdeta=tmpdeta+0.382*tmpdeta;
    else
        tmpdeta=tmpdeta-0.382*tmpdeta;
    end
end

```



```

end
disp(['L=' num2str(tmpL)]);
disp(['deta=' num2str(deta)]);
disp(['percent(%)='
num2str(deta/L*100)]);
count=count+1
if count>500
break;
end
end
% plot deformed curve - integration
xL=0; yL=0;
xp=linspace(0,L-deta,101);
for i=1:101 % calculate vertical
displacement and orientation
lamda=xp(i)/snum;
tmpx=linspace(0,xp(i),11);
G=Gx(tmpx,theta0,C);
dy=dydx(G);
yp(i)=simpson10(dy,lamda);
end
figure
x1=fliplr(xp); y1=min(yp)-yp;
plot(x1,y1,'r','linewidth',2);
hold on;
% force and moment (M=0.01, F=0.05)
clear;
% system parameters
L=500e-3; E=1.5e9;
h=1e-3; w=40e-3;
I=w*h^3/12;
M=10E-3; F=0.05;
% Calculate Horizontal displacement
deta=0.1*L;
% anonymous function
solve_theta=@(theta0)theta0*(L-deta)-
L*sin(theta0);
theta0=fsolve(solve_theta,0.5);
C=-(M*(L-deta)-0.5*F*(L-
deta)^2*(sin(theta0)-
cos(theta0)*tan(theta0/2)))/E/I;
Gx=@(x,theta0,C) (M.*x-
0.5*F*x.^2*(sin(theta0)-
cos(theta0)*tan(theta0/2)))/E/I+C;
dydx=inline('G./sqrt(1-G.^2)','G');
Fx=inline('sqrt(1+Gx.^2)','Gx');
% compute deta
snum=10; % simpson method number
tmpL=0; count=0;
tmpdeta=deta;
while (abs(L-tmpL)>0.005*L)
lamda=(L-tmpdeta)/snum;
x=linspace(0,L-tmpdeta,snum+1);
solve_theta=@(theta0)theta0*(L-
tmpdeta)-L*sin(theta0);
theta0=fsolve(solve_theta,0.55);
C=-(M*(L-tmpdeta)-0.5*F*(L-
tmpdeta)^2*(sin(theta0)-
cos(theta0)*tan(theta0/2)))/E/I;
G=Gx(x,theta0,C);
dy=dydx(G);
FX=Fx(dy);
tmpL=(L-
tmpdeta)*Simpson10(FX,lamda/(L-
tmpdeta));
deta=tmpdeta;
if L<tmpL
tmpdeta=tmpdeta+0.382*tmpdeta;
else
tmpdeta=tmpdeta-0.382*tmpdeta;
end
disp(['L=' num2str(tmpL)]);
disp(['deta=' num2str(deta)]);
disp(['percent(%)='
num2str(deta/L*100)]);
count=count+1
if count>500
break;
end
end
% plot deformed curve - integration
xL=0; yL=0;
xp=linspace(0,L-deta,101);
for i=1:101 % calculate vertical
displacement and orientation
lamda=xp(i)/snum;
tmpx=linspace(0,xp(i),11);
G=Gx(tmpx,theta0,C);
dy=dydx(G);
yp(i)=simpson10(dy,lamda);
end
x1=fliplr(xp); y1=min(yp)-yp;
plot(x1,y1,'g','linewidth',2);
% force and moment (M=0.01, F=0.1)
clear;
% system parameters
L=500e-3; E=1.5e9;
h=1e-3; w=40e-3;
I=w*h^3/12;
M=10E-3; F=0.1;
% Calculate Horizontal displacement
deta=0.1*L;
% anonymous function
solve_theta=@(theta0)theta0*(L-deta)-
L*sin(theta0);
theta0=fsolve(solve_theta,0.5);
C=-(M*(L-deta)-0.5*F*(L-
deta)^2*(sin(theta0)-
cos(theta0)*tan(theta0/2)))/E/I;
Gx=@(x,theta0,C) (M.*x-
0.5*F*x.^2*(sin(theta0)-
cos(theta0)*tan(theta0/2)))/E/I+C;
dydx=inline('G./sqrt(1-G.^2)','G');
Fx=inline('sqrt(1+Gx.^2)','Gx');
% compute deta
snum=10; % simpson method number
tmpL=0; count=0;
tmpdeta=deta;
while (abs(L-tmpL)>0.005*L)
lamda=(L-tmpdeta)/snum;

```

```

x=linspace(0,L-tmpdeta,snum+1);
solve_theta=@(theta0)theta0*(L-
tmpdeta)-L*sin(theta0);
theta0=fsolve(solve_theta,0.55);
C=-M*(L-tmpdeta)-0.5*F*(L-
tmpdeta)^2*(sin(theta0)-
cos(theta0)*tan(theta0/2))/E/I;
G=Gx(x,theta0,C);
dy=dydx(G);
FX=Fx(dy);
tmpL=(L-
tmpdeta)*Simpson10(FX,lamda/(L-
tmpdeta));
deta=tmpdeta;
if L<tmpL
    tmpdeta=tmpdeta+0.382*tmpdeta;
else
    tmpdeta=tmpdeta-0.382*tmpdeta;
end
disp(['L=' num2str(tmpL)]);
disp(['deta=' num2str(deta)]);
disp(['percent(%)='
num2str(deta/L*100)]);
count=count+1
if count>500
    break;
end
end
end

```

```

% plot deformed curve - integration
xL=0; yL=0;
xp=linspace(0,L-deta,101);
for i=1:101 % calculate vertical
displacement and orientation
    lamda=xp(i)/snum;
    tmpx=linspace(0,xp(i),11);
    G=Gx(tmpx,theta0,C);
    dy=dydx(G);
    yp(i)=simpson10(dy,lamda);
end
x1=fliplr(xp); y1=min(yp)-yp;
plot(x1,y1,'b','linewidth',2);
legend('M=0.01Nm','M=0.01Nm,
F=0.05N','M=0.01Nm, F=0.1N');
title('model compare');
xlim([0 L]); axis equal;
xlabel('X(m)','fontsize',12,'fontweight',
'bold');
ylabel('Y(m)','fontsize',12,'fontweight',
'bold');
set(gca,'fontsize',12,'fontweight','bold');

function S=Simpson10(Y,lamda)
S=lamda/3*(Y(1)+4*Y(2)+2*Y(3)+...
4*Y(4)+2*Y(5)+4*Y(6)+2*Y(7)+...
4*Y(8)+2*Y(9)+4*Y(10)+Y(11));

```

15) Figure 5-3 Fish Swimming Body Curve - Oscillatory

```

% Oscillatory Swimming
% Oscillatory Body Curve Function
L=1; dL=0.01;
T=2; dt=T/4;
lamda=1.048;
c1=0.1; c2=0.2; k=0.5;
omega=pi; U=0.5;
linecolor(1,:) = [0 0 1];
linecolor(2,:) = [0 1 0];
linecolor(3,:) = [1 0 0];
linecolor(4,:) = [1 0 1];
linecolor(5,:) = [0 0 1];
for i=1:1*T/dt+1
    t(i)=(i-1)*dt;
    for j=1:L/dL
        x(j)=j*dL-U*L*t(i); % traveling wave
        xloc(j)=j*dL; % local coordinate
        y(i,j)=(c1*xloc(j)+c2*xloc(j)^2)*sin(k*xloc
(j)+omega*t(i));
    end
end

```

```

plot(-
x,y(i,:), 'linewidth',3,'color',linecolor(i,:));
hold on
end
axis equal;
alfa=15/180*pi; phi=45/180*pi;
d=0.0267*L;
H=c1*(L)+c2*(L)^2
TIP=2*(H-d*sin((atan((H)*omega/U)-
alfa)*sin(pi/2+phi)))
title('Body Curve of A Swimming Fish -
Oscillatory','FontSize',12);
xlabel('Travelling Distance
(BL)','FontSize',12,'fontweight','bold');
ylabel('Fish Body Excursion
(BL)','FontSize',12,'fontweight','bold')
legend('t=0.00T','t=0.25T','t=0.50T','t=0.75T',
't=1.00T');
set(gca,'fontsize',12,'fontweight','bold');
xlim([-1,1]); ylim([-0.75,0.75]);

```

16) Figure 5-4 Oscillatory Body Curve Comparison: (a) One Straight Line and One Circular Arc Fitting; (b) Two Straight Lines and One Circular Arc Fitting

```

% Oscillatory Body Curve Comparison
% Oscillatory Body Curve Function
L=1; dL=0.01; T=2;
dt=T/4; c1=0.1; c2=0.2;
kk=0.5; omega=pi; U=0.5;

```

```

linecolor(1,:) = [0 0 1];
linecolor(2,:) = [0 1 0];
linecolor(3,:) = [1 0 0];
linecolor(4,:) = [1 0 1];
linecolor(5,:) = [0 0 1];

```



```

str={'i','ii','iii','iv'};
for i=1:1*T/dt
    t(i)=(i-1)*dt;
    for j=1:L/dL
        x(j)=j*dL-U*L*t(i); % traveling wave
        xloc(j)=j*dL; % local coordinate
        y(i,j)=(c1*xloc(j)+c2*xloc(j)^2)*sin(kk*xloc
        (j)+omega*t(i));
    end
    x_bc=xloc;
    y_bc(i,:)=y(i,:);
    subplot(4,2,2*i-1);
    plot(x_bc,y_bc(i,:), 'linewidth',2,'color',linec
    olor(i,:));
    text=strcat(' ',str(i), 'y', 't=', num2str(i/4), 'T');
    title(text, 'fontsize',12, 'fontweight', 'bold');
    set(gca, 'fontsize',12, 'fontweight', 'bold');
    xlim([0,1]); hold on;
    % 1 rigid link
    for k=1:L/dL
        x_rl(k)=k*dL; % local coordinate
        if k<L/dL+1
            y_rl(i,k)=y_bc(i,L/dL)*k/(L/dL);
        end
    end
    plot(x_rl,y_rl(i,:), 'linewidth',2,'color',linec
    olor(i,:), 'linestyle','-');
    % 1 WD
    Jnum=floor(L/dL/3);
    [x_wd(i,:),y_wd(i,:)] = ArcPPP(x_bc(1),y_bc
    (i,1),x_bc(Jnum),y_bc(i,Jnum),x_bc(L/dL),y
    _bc(i,L/dL));
    plot(x_wd(i,:),y_wd(i,:), 'linewidth',2,'color',l
    inecolor(i,:), 'linestyle','*');
    % 2 rigid links
    for k=1:L/dL
        x_rl(k)=k*dL; % local coordinate
        if k<L/dL/2+1
            y_rl(i,k)=y_bc(i,L/dL/2)*k/(L/dL/2);
        else
            y_rl(i,k)=y_rl(i,L/dL/2)-...
            (y_rl(i,L/dL/2)-
            y_bc(i,L/dL))*k/(L/dL/2)-1);
        end
    end
    subplot(4,2,2*i);
    plot(x_rl,y_rl(i,:), 'linewidth',2,'color',linecol
    or(i,:), 'linestyle','-');
    text=strcat(' ',str(i), 'y', 't=', num2str(i/4), 'T');
    title(text, 'fontsize',12, 'fontweight', 'bold');
    set(gca, 'fontsize',12, 'fontweight', 'bold');
    xlim([0,1]);
    hold on;
    plot(x_wd(i,:),y_wd(i,:), 'linewidth',2,'colo
    r',linecolor(i,:), 'linestyle','*');
end
xlim([0,1]);

```

17) Figure 5-5 Fish Swimming Body Curve – Undulatory

```

% Robot Tuna Body Curve Function
L=1; dL=0.01; T=1.626;
dt=T/4; lamda=1.048;
c1=0.020; c2=0.0835;
K=2*pi/lamda;
omega=2*pi/T;
U=0.4687;
% travelling wave
linecolor(1,:) = [0 0 1];
linecolor(2,:) = [0 1 0];
linecolor(3,:) = [1 0 0];
linecolor(4,:) = [1 0 1];
linecolor(5,:) = [0 0 1];
for i=1:1*T/dt+1
    t(i)=(i-1)*dt;
    for j=1:L/dL
        x(j)=j*dL-U*L*t(i); % travelling wave
        xloc(j)=j*dL; % local coordinate
        y(i,j)=(c1*xloc(j)+c2*xloc(j)^2)*sin(K*xloc
        (j)+omega*t(i));
    end
    plot(-
    x,y(i,:), 'linewidth',3,'color',linecolor(i,:));
    hold on
end
axis equal;
alfa=15/180*pi; phi=45/180*pi;
d=0.0267*L;
H=c1*(L)+c2*(L)^2
TIP=2*(H-d*sin((atan((H)*omega/U)-
alfa)*sin(pi/2+phi)))
title('Body Curve of A Swimming Fish -
Undulatory', 'fontsize',12);
xlabel('Travelling Distance
(BL)', 'fontsize',12, 'fontweight', 'bold');
ylabel('Fish Body Excursion
(BL)', 'fontsize',12, 'fontweight', 'bold')
legend('t=0.00T', 't=0.25T', 't=0.50T', 't=0.75T
', 't=1.00T', 1);
xlim([-1,1]); ylim([-0.4,0.4]);
set(gca, 'fontsize',12, 'fontweight', 'bold');

```

18) Figure 5-6 Undulatory Body Curve Comparison: (a) Three Straight Lines and Three Circular Arcs Fitting; (b) Six Straight Lines and Three Circular Arcs Fitting

Figure 5-7 Two Circular Arcs Fit the Undulatory Fish Swimming Body Curve

```

h1=figure('name','undulatory swimming
body curve comparison');
h2=figure('name','Fit by two-segment
WDM');
str={'i','ii','iii','iv'};
linecolor(1,:) = [0 0 1];
linecolor(2,:) = [0 1 0];
linecolor(3,:) = [1 0 0];
linecolor(4,:) = [1 0 1];
linecolor(5,:) = [0 0 1];

L=1; dL=0.01; dt=0.4;
T=1.626; lamda=1.048;
c1=0.020; c2=0.0835;
K=2*pi/lamda;
omega=2*pi/T;
U=0.4687;
for i=1:1*T/dt
    t(i)=(i-1)*dt;
    for j=1:L/dL
        x_bc(j)=j*dL; % local coordinate
        y_bc(i,j)=(c1*x_bc(j)+c2*x_bc(j)^2)*sin(K
*x_bc(j)+omega*t(i));
    end
    figure(1); subplot(4,2,2*i-1);
    plot(x_bc,y_bc(i,:), 'linewidth',2,'color',lin
ecolor(i,:));
    hold on;
    xlim([0,1]);
    text=strcat(' ',str(i),' ','
t=',num2str(i/4),'T');
    title(text,'fontsize',12,'fontweight','bold');
    set(gca,'fontsize',12,'fontweight','bold');
    % three links
    for k=1:L/dL
        x_rl3(k)=k*dL; % local coordinate
        Jnum=floor(L/dL/3);
        if k<Jnum+1
            y_rl3(i,k)=y_bc(i,Jnum)*k/Jnum;
        else if k<2*Jnum+1
            y_rl3(i,k)=y_rl3(i,Jnum)-...
            (y_rl3(i,Jnum)-
y_bc(i,2*Jnum))*(k/Jnum-1);
        else
            y_rl3(i,k)=y_rl3(i,2*Jnum)-...
            (y_rl3(i,2*Jnum)-
y_bc(i,L/dL))*((k-2*Jnum)/(L/dL-2*Jnum));
        end
    end
end

plot(x_rl3,y_rl3(i,:), 'linewidth',2,'color',linec
olor(i,:), 'linestyle','-');
% 3 WD
Jnum=floor(L/dL/6);
% segment 1
[x_wd31,y_wd31]=ArcPPP(x_bc(1),y_bc
(i,1),x_bc(Jnum),y_bc(i,Jnum),x_bc(2*Jnum
),y_bc(i,2*Jnum));
% segment 2
[x_wd32,y_wd32]=ArcPPP(x_bc(2*Jnum),y
_bc(i,2*Jnum),x_bc(3*Jnum),y_bc(i,3*Jnu
m),x_bc(4*Jnum),y_bc(i,4*Jnum));
% segment 3
[x_wd33,y_wd33]=ArcPPP(x_bc(4*Jnum),y
_bc(i,4*Jnum),x_bc(5*Jnum),y_bc(i,5*Jnu
m),x_bc(L/dL),y_bc(i,L/dL));

x_wd3(i,:)=cat(2,x_wd31,x_wd32,x_wd33);
y_wd3(i,:)=cat(2,y_wd31,y_wd32,y_wd33);

plot(x_wd3(i,:),y_wd3(i,:), 'linewidth',2,'colo
r',linecolor(i,:), 'linestyle','*');

% six links
for k=1:L/dL
    x_rl(k)=k*dL; % local coordinate
    Jnum=floor(L/dL/6);
    if k<Jnum+1
        y_rl(i,k)=y_bc(i,Jnum)*k/Jnum;
    else if k<2*Jnum+1
        y_rl(i,k)=y_rl(i,Jnum)-...
        (y_rl(i,Jnum)-
y_bc(i,2*Jnum))*(k/Jnum-1);
    else if k<3*Jnum+1
        y_rl(i,k)=y_rl(i,2*Jnum)-...
        (y_rl(i,2*Jnum)-
y_bc(i,3*Jnum))*(k/Jnum-2);
    else if k<4*Jnum+1
        y_rl(i,k)=y_rl(i,3*Jnum)-...
        (y_rl(i,3*Jnum)-
y_bc(i,4*Jnum))*(k/Jnum-3);
    else if k<5*Jnum+1
        y_rl(i,k)=y_rl(i,4*Jnum)-...
        (y_rl(i,4*Jnum)-
y_bc(i,5*Jnum))*(k/Jnum-4);
    else
        y_rl(i,k)=y_rl(i,5*Jnum)-...
        (y_rl(i,5*Jnum)-
y_bc(i,L/dL))*((k-5*Jnum)/(L/dL-5*Jnum));
    end
end
end
end
end
end
subplot(4,2,2*i);
plot(x_rl,y_rl(i,:), 'linewidth',2,'color',linecol
or(i,:), 'linestyle','-');
hold on;
xlim([0,1]);
text=strcat(' ',str(i),' ','
t=',num2str(i/4),'T');
title(text,'fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');
xlim([0,1]);

plot(x_wd3(i,:),y_wd3(i,:), 'linewidth',2,'colo
r',linecolor(i,:), 'linestyle','*');

```



```
plot(x_bc,y_bc(i,:), 'linewidth',2,'color',linecolor(i,:));
```

```

% 2 WD
Jnum=floor(L/dL/10);
% segment 1
[x_wd21,y_wd21]=ArcPPP(x_bc(1),y_bc(i,1),x_bc(3*Jnum),y_bc(i,3*Jnum),x_bc(6*Jnum),y_bc(i,6*Jnum));
% segment 2
[x_wd22,y_wd22]=ArcPPP(x_bc(6*Jnum),y_bc(i,6*Jnum),x_bc(8*Jnum),y_bc(i,8*Jnum),x_bc(L/dL),y_bc(i,L/dL));
x_wd2(i,:)=cat(2,x_wd21,x_wd22);
y_wd2(i,:)=cat(2,y_wd21,y_wd22);

```

```
figure(2);
h2=subplot(2,2,i);
```

```

plot(x_wd2(i,:),y_wd2(i,:), 'linewidth',2,'color',linecolor(i,:), 'linestyle','-');
hold on;
set(gca,'fontsize',12,'fontweight','bold');
xlim([0,1]);
text=strcat(' ',str(i),' ');
t=num2str(i/4,'T');
title(text,'fontsize',12,'fontweight','bold');
plot(x_bc,y_bc(i,:), 'linewidth',2,'color',linecolor(i,:));
end

```

19) Figure 5-13 Wire Length Change Approximation Error - Serpentine

```

% error plot - chapter 2 serpentine tapered
h0=5; ddi=39; DD=44;
thetamax=2*atan(h0/DD)*180/pi;
figure;
mycolor1 = [1 0 0; 1 0 1];
mycolor2 = [0 0 1; 0 1 0];
for i=5:5 % gama
    gama(i)=7.5*pi/180;
    udi1(i)=ddi+2*h0*tan(gama(i));
    hi0(i)=sqrt(h0^2+(h0*tan(gama(i)))^2);
    for j=1:floor(thetamax) % theta
        theta(i,j)=j*pi/180;
        a=(h0/2/tan(theta(i,j)/2)-ddi/2);
        b=(h0/2/tan(theta(i,j)/2)-udi1(i)/2);
        A=(h0/2/tan(theta(i,j)/2)+ddi/2);
        B=(h0/2/tan(theta(i,j)/2)+udi1(i)/2);
        dh1(i,j)=hi0(i)-sqrt(a^2+b^2-2*a*b*cos(theta(i,j)));
        dh2(i,j)=-hi0(i)+sqrt(A^2+B^2-2*A*B*cos(theta(i,j)));
        dh(i,j)=0.5*(ddi)*theta(i,j)*cos(gama(i));
    end
end

```

```

err1(i,j)=abs(dh1(i,j)-dh(i,j))/dh1(i,j)*100;
err2(i,j)=abs(dh2(i,j)-dh(i,j))/dh2(i,j)*100;
end
plot(theta(i,:)*180/pi,err1(i,:), 'color',mycolor1(i/5,:), 'linewidth',2);
hold on;
plot(theta(i,:)*180/pi,err2(i,:), 'color',mycolor2(i/5,:), 'linewidth',2);
end
legend('Wire1 Err,\gamma = 7.5^o','Wire2 Err,\gamma = 7.5^o',3);
ylabel('Relative Error (%)','fontsize',12,'fontweight','bold');
xlabel('Joint Rotation (^o)','fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');
xlim([0 12]);

```

20) Figure 5-14 Serpentine Oscillatory Wire-Driven Tail Flapping Cycle

```

H = [20 19 18 17 16 15 14];
h = [5 5 4.5 4 3.5 3 2.5];
D = [44 40 36 32 28 24 20];
for j=1:7
    thetaMax(j)=2*atan(h(j)/D(j));
    len(j)=H(j)+h(j);
end
num=6;
dtheta=thetaMax/num;
for i=1:2*num+1
    theta(i,:)=(-num+i-1)*dtheta;
    x(i,1:7)=0; y(i,1:7)=0;
end

```

```

for j=1:7
    alfa(j) = sum(theta(i,1:j));
    x(i,j+1)=x(i,j)+len(j)*cos(alfa(j));
    y(i,j+1)=y(i,j)+len(j)*sin(alfa(j));
end
plot(x(i,:),y(i,:), 'linewidth',2);
hold on;
end
axis equal
set(gca,'fontsize',12,'fontweight','bold');
xlim([0 150]);
ylim([-100 100]);

```

21) Figure 5-22 Wire Length Change Approximation Error - Continuum

```

%% error plot - continuum tapered
h=20; ri=39.5/2; DDi1=45;
thetamax=2*h/DDi1*180/pi;
figure;
mycolor1 = [1 0 0; 1 0 1];

```

```

mycolor2 = [0 0 1; 0 1 0];
for i=5:5 % gama
    gama(i)=7.125*pi/180;
    ri(i)=ri1+h*tan(gama(i));
    li0(i)=sqrt(h^2+(h*tan(gama(i)))^2);
end

```



```

for j=1:15 % theta
    theta(i,j)=j*pi/180;
    dh1(i,j)=li0(i)-sqrt(li0(i)^2-
    (ri1+ri(i))*h*theta(i,j)+ri1*ri(i)*theta(i,j)^2);
    dh2(i,j)=-
    li0(i)+sqrt(li0(i)^2+(ri1+ri(i))*h*theta(i,j)+ri
    1*ri(i)*theta(i,j)^2);
    dh(i,j)=ri1*theta(i,j)*cos(gama(i));
    err1(i,j)=abs(dh1(i,j)-
    dh(i,j))/dh1(i,j)*100;
    err2(i,j)=abs(dh2(i,j)-
    dh(i,j))/dh2(i,j)*100;
end
plot(theta(i,:)*180/pi,err1(i,:), 'color',mycolor
1(i/5,:), 'linewidth',2);

```

```

hold on;
plot(theta(i,:)*180/pi,err2(i,:), 'color',myco
lor2(i/5,:), 'linewidth',2);
end
legend('Wire1 Err,\gamma =
7.125^o','Wire2 Err,\gamma = 7.125^o',3);
ylabel('Relative Error
(%)', 'fontsize',12, 'fontweight', 'bold');
xlabel('Joint Rotation
(^o)', 'fontsize',12, 'fontweight', 'bold');
set(gca, 'fontsize',12, 'fontweight', 'bold')
xlim([0 15]); ylim([5.7 6.1]);

```

22) Figure 5-37 Undulatory Swimming Curve Comparison

```

% undulatory wire-driven tail body curve
H1 = 15; h1 = 4; H2 = 12;
h2 = 3; D1 = 31; D2=23.5;
thetaMax1=2*atan(h1/D1);
thetaMax2=2*atan(h2/D2);
len1=H1+h1;
len2=H2+h2;
N1=6; N2=7;
T=1; dT=1/4;
pi=3.1415926;
for i=1:T/dT
    t(i)=(i)*dT+0.2*T;
    theta1(i)=8*sin(2*pi*t(i))*pi/180;
    theta2(i)=-12*sin(2*pi*t(i)-
    0.3*pi)*pi/180;
    x(i,1:N1+N2)=0;
    y(i,1:N1+N2)=0;
    for j=1:N1
        alfa(j)=j*theta1(i);
        x(i,j+1)=x(i,j)+len1*cos(alfa(j));
        y(i,j+1)=y(i,j)+len1*sin(alfa(j));
    end
    for k=N1+1:N1+N2
        alfa(k)=alfa(N1)+(k-N1)*theta2(i);
        x(i,k+1)=x(i,k)+len2*cos(alfa(k));
        y(i,k+1)=y(i,k)+len2*sin(alfa(k));
    end
    plot(x(i,:),y(i,:), 'linewidth',2);

```

```

hold on;
end
% Body Curve from the swimming model
L=N1*len1+N2*len2;
dL=L/13; dt=0.25;
T=1; c1=0.2/L;
c2=0.4175/L/L;
K=2.4/L;
omega=2*pi/T;
for i=1:L/dt
    t(i)=(i-1)*dt+0.20*T;
    for j=1:L/dL+1
        x_bc(j)=(j-1)*dL; % local
        coordinate
        y_bc(i,j)=(c1*x_bc(j)+c2*x_bc(j)^2)*sin(K
        *x_bc(j)+omega*t(i));
    end
    plot(x_bc,y_bc(i,:)*L, '-r', 'linewidth',2);
    str=num2str((i-1)*dt+0.20);
    gtext(strcat('t=',str, 'T'), 'fontsize',12, 'fontweig
    ht', 'bold');
end
set(gca, 'fontsize',12, 'fontweight', 'bold');
xlabel('X(mm)', 'fontsize',12, 'fontweight', 'bol
d');
ylabel('Y(mm)', 'fontsize',12, 'fontweight', 'bol
d');
xlim([0 200]);

```

23) Figure 5-48 Vector Propulsor Planar Flapping

```

% flap in horizontal plane
l=25;
figure;
phi = 0; Amp = 60; Num=12;
det = 2*Amp/Num;
for i=1:Num % cycle
    theta(i)=(-Amp+(i-1)*det)*pi/180;
    dtheta= theta(i)/7;
    x(i,1)=0; y(i,1)=0; z(i,1)=0;
    for j=1:7 % link
        q(i,j)=l;
        xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi);
        yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi);

```

```

        zloc(i,j)=q(i,j)*cos(j*dtheta);
        x(i,j+1)=xloc(i,j)+x(i,j);
        y(i,j+1)=yloc(i,j)+y(i,j);
        z(i,j+1)=zloc(i,j)+z(i,j);
    end
    plot3(x(i,:),y(i,:),z(i,:), 'b', 'linewidth',2);
    hold on; grid on;
end
plot3(x(:,8),y(:,8),z(:,8), 'r', 'linewidth',3);
xlim([-100,100]); ylim([-100,100]);
zlim([0,200]);
xlabel('X', 'fontsize',12, 'fontweight', 'bold');
ylabel('Y', 'fontsize',12, 'fontweight', 'bold');

```

```
zlabel('Z','fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');
```

```
% flap in vertical plane
```

```
l=25;
figure;
phi = 90*pi/180; Amp = 60*pi/180;
Num=12; det = 2*Amp/Num;
for i=1:Num % cycle
    theta(i)=(-Amp+(i-1)*det);
    dtheta= theta(i)/7;
    x(i,1)=0; y(i,1)=0; z(i,1)=0;
    for j=1:7 % link
        q(i,j)=1;
        xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi);
        yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi);
        zloc(i,j)=q(i,j)*cos(j*dtheta);
        x(i,j+1)=xloc(i,j)+x(i,j);
        y(i,j+1)=yloc(i,j)+y(i,j);
        z(i,j+1)=zloc(i,j)+z(i,j);
    end
    plot3(x(i,:),y(i,:),z(i:,:),'b','linewidth',2);
    hold on; grid on;
    xlim([-100,100]); ylim([-100,100]);
    zlim([0,200]);
```

```
end
plot3(x(:,8),y(:,8),z(:,8),'r','linewidth',3);
xlabel('X','fontsize',12,'fontweight','bold');
ylabel('Y','fontsize',12,'fontweight','bold');
zlabel('Z','fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');
```

```
% flap in 45 deg
```

```
l=25;
figure;
phi = 45*pi/180; Amp = 60*pi/180;
Num=12; det = 2*Amp/Num;
for i=1:Num % cycle
    theta(i)=(-Amp+(i-1)*det);
    dtheta= theta(i)/7;
    x(i,1)=0; y(i,1)=0; z(i,1)=0;
    for j=1:7 % link
        q(i,j)=1;
        xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi);
        yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi);
```

```
zloc(i,j)=q(i,j)*cos(j*dtheta);
x(i,j+1)=xloc(i,j)+x(i,j);
y(i,j+1)=yloc(i,j)+y(i,j);
z(i,j+1)=zloc(i,j)+z(i,j);
end
plot3(x(i,:),y(i,:),z(i:,:),'b','linewidth',2);
hold on; grid on;
xlim([-100,100]); ylim([-100,100]);
zlim([0,200]);
```

```
end
plot3(x(:,8),y(:,8),z(:,8),'r','linewidth',3);
xlabel('X','fontsize',12,'fontweight','bold');
ylabel('Y','fontsize',12,'fontweight','bold');
zlabel('Z','fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');
```

```
% flap in -45 deg
```

```
l=25;
figure;
phi = 60*pi/180; Amp = 60*pi/180;
Num=12; det = 2*Amp/Num;
for i=1:Num % cycle
    theta(i)=(-Amp+(i-1)*det);
    dtheta= theta(i)/7;
    x(i,1)=0; y(i,1)=0; z(i,1)=0;
    for j=1:7 % link
        q(i,j)=1;
        xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi);
        yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi);
        zloc(i,j)=q(i,j)*cos(j*dtheta);
        x(i,j+1)=xloc(i,j)+x(i,j);
        y(i,j+1)=yloc(i,j)+y(i,j);
        z(i,j+1)=zloc(i,j)+z(i,j);
    end
    plot3(x(i,:),y(i,:),z(i:,:),'b','linewidth',2);
    hold on; grid on;
    xlim([-100,100]); ylim([-100,100]);
    zlim([0,200]);
```

```
end
plot3(x(:,8),y(:,8),z(:,8),'r','linewidth',3);
xlabel('X','fontsize',12,'fontweight','bold');
ylabel('Y','fontsize',12,'fontweight','bold');
zlabel('Z','fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');
```

24) Figure 5-49 Vector Propulsor Spatial Flapping

```
% circling
```

```
l=25;
figure;
phi = 90*pi/180; Amp = 60*pi/180;
Num=72; det = 2*Amp/Num;
for i=1:Num % cycle
    theta(i)=(-Amp+(i-1)*det);
    phi(i)=(i-1)*2*pi/Num;
    dtheta= Amp/7;
    x(i,1)=0; y(i,1)=0; z(i,1)=0;
    for j=1:7 % link
        q(i,j)=1;
        xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi(i));
        yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi(i));
```

```
zloc(i,j)=q(i,j)*cos(j*dtheta);
x(i,j+1)=xloc(i,j)+x(i,j);
y(i,j+1)=yloc(i,j)+y(i,j);
z(i,j+1)=zloc(i,j)+z(i,j);
end
plot3(x(i,:),y(i,:),z(i:,:),'b','linewidth',2);
hold on; grid on;
xlim([-100,100]); ylim([-100,100]);
zlim([0,200]);
```

```
end
phi2 = 0:pi/10:2*pi;
xt=sqrt(x(1,8)^2+y(1,8)^2)*cos(phi2);
yt=sqrt(x(1,8)^2+y(1,8)^2)*sin(phi2);
zt=z(1,8)*ones(1,21);
```



```

plot3(xt,yt,zt,'r','linewidth',3);
xlabel('X','fontsize',12,'fontweight','bold');
ylabel('Y','fontsize',12,'fontweight','bold');
zlabel('Z','fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');

```

```
% 8 shape flapping
```

```

l=25;
figure;
phi = 90*pi/180; Amp = 60*pi/180;
Num=36; det = 2*Amp/Num;
alfa = 30*pi/180;
Num2=alfa/2*pi*Num;
Num3=(Num-4*Num2)/2;
for i=1:Num % cycle
    x(i,1)=0; y(i,1)=0; z(i,1)=0;
    if i<2*Num2+1 % first arc
        phi(i)=alfa-(i-1)*2*pi/Num;
        dtheta= Amp/7;
        for j=1:7 % link
            q(i,j)=l;

xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi(i));

yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi(i));
zloc(i,j)=q(i,j)*cos(j*dtheta);
x(i,j+1)=xloc(i,j)+x(i,j);
y(i,j+1)=yloc(i,j)+y(i,j);
z(i,j+1)=zloc(i,j)+z(i,j);
        end
    else if i<2*Num2+Num3/2+1 % first line
        det=2*Amp/Num3;
        theta(i)=(Amp-(i-2*Num2)*det);
        dtheta= theta(i)/7;
        phi(i)=-alfa;
        for j=1:7 % link
            q(i,j)=l;

xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi(i));

yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi(i));
zloc(i,j)=q(i,j)*cos(j*dtheta);
x(i,j+1)=xloc(i,j)+x(i,j);
y(i,j+1)=yloc(i,j)+y(i,j);
z(i,j+1)=zloc(i,j)+z(i,j);
        end
    else if i<Num/2+1 % second line
        det=2*Amp/Num3;
        theta(i)=(Amp-(i-2*Num2)*det);
        dtheta= theta(i)/7;
        phi(i)=-alfa;
        for j=1:7 % link
            q(i,j)=l;

xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi(i));

yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi(i));
zloc(i,j)=q(i,j)*cos(j*dtheta);
x(i,j+1)=xloc(i,j)+x(i,j);
y(i,j+1)=yloc(i,j)+y(i,j);
        end
    end
end

```

```

z(i,j+1)=zloc(i,j)+z(i,j);
        end
    else if i<Num/2+2*Num2+1 %
second arc
        dtheta = Amp/7;
        phi(i)= pi-alfa+(i-
Num/2)*2*pi/Num;
        for j=1:7 % link
            q(i,j)=l;

xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi(i));

yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi(i));
zloc(i,j)=q(i,j)*cos(j*dtheta);
x(i,j+1)=xloc(i,j)+x(i,j);
y(i,j+1)=yloc(i,j)+y(i,j);
z(i,j+1)=zloc(i,j)+z(i,j);
        end
    else if i<Num-Num3/2+1% third line
        det=2*Amp/Num3;
        theta(i)=(Amp-(i-
(4*Num2+Num3))*det);
        dtheta = theta(i)/7;
        phi(i)=pi+alfa;
        for j=1:7 % link
            q(i,j)=l;

xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi(i));

yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi(i));
zloc(i,j)=q(i,j)*cos(j*dtheta);
x(i,j+1)=xloc(i,j)+x(i,j);
y(i,j+1)=yloc(i,j)+y(i,j);
z(i,j+1)=zloc(i,j)+z(i,j);
        end
    else % fourth line
        det=2*Amp/Num3;
        theta(i)=(-Amp+(i-
(4*Num2+Num3))*det);
        dtheta = theta(i)/7;
        phi(i)=alfa;
        for j=1:7 % link
            q(i,j)=l;

xloc(i,j)=q(i,j)*sin(j*dtheta)*cos(phi(i));

yloc(i,j)=q(i,j)*sin(j*dtheta)*sin(phi(i));
zloc(i,j)=q(i,j)*cos(j*dtheta);
x(i,j+1)=xloc(i,j)+x(i,j);
y(i,j+1)=yloc(i,j)+y(i,j);
z(i,j+1)=zloc(i,j)+z(i,j);
        end
    end
end
end
end
end
plot3(x(i,:),y(i,:),z(i:,:), 'b','linewidth',2);
hold on; grid on;
xlim([-100,100]); ylim([-100,100]);

```

```

    zlim([0,200]);
end
plot3(x(:,8),y(:,8),z(:,8),'r','linewidth',3);
xlabel('X','fontsize',12,'fontweight','bold');

```

```

ylabel('Y','fontsize',12,'fontweight','bold');
zlabel('Z','fontsize',12,'fontweight','bold');
set(gca,'fontsize',12,'fontweight','bold');

```

25) Undulatory Propulsion model and swimming experiments

```

% robot fish Froude efficiency
% tail parameters
N1=6; N2=7; L1=19e-3; L2=15e-3;
L=280e-3;
Lend=L-N1*L1-N2*L2;
%% simulation input - exp1
A1=[20 30 40 40 40]*pi/180;
A2=[30 45 60 60 60]*pi/180;
f=[1 1 1 0.5 1.5];
UM=[0.408 0.526 0.608 0.402 0.311]*0.495;
lag=[5/8 5/8 5/8 5/8 5/8]*0;
lamda=4*L*ones(size(A1,2),1);
U=zeros(size(A1,2),1);
E=zeros(size(A1,2),1);
for i=1:size(A1,2)
    result=EBT(A1(i),
    A2(i),f(i),lamda(i),lag(i),UM(i),10);
    U(i)=result(1);
    E(i)=result(2);
end
%% simulation input - exp2
A1=[00 00 00 00 00]*pi/180;
A2=[30 45 60 60 60]*pi/180;
f=[1 1 1 0.5 1.5];
UM=[0.185 0.285 0.329 0.290 0.276]*0.495;
lamda=4*(L-L1*N1)*ones(size(A1,2),1);
lag=[5/8 5/8 5/8 5/8 5/8]*0;
U=zeros(size(A1,2),1);
E=zeros(size(A1,2),1);
for i=1:size(A1,2)
    result=EBT(A1(i),
    A2(i),f(i),lamda(i),lag(i),UM(i),10);
    U(i)=result(1); E(i)=result(2);
end
%% simulation input - exp3
A1=[20 30 40 40 40 40 40 40]*pi/180;
A2=[30 45 60 60 60 60 60 60]*pi/180;
f=[1 1 1 0.5 1.5 1 1 1 1];
UM=[0.419 0.563 0.673 0.623 0.405 0.584
0.246 0.065 0.613]*0.495;
lag=[5/8 5/8 5/8 5/8 5/8 1/8 3/8 4/8 7/8];
lamda=L*ones(size(A1,2),1);
U=zeros(size(A1,2),1);
E=zeros(size(A1,2),1);
for i=1:size(A1,2)
    result=EBT(A1(i),
    A2(i),f(i),lamda(i),lag(i),UM(i),3);
    U(i)=result(1); E(i)=result(2);
end
function result =
EBT(A1,A2,f,lamda,lag,UM,sway)
% parameter input
amp1 = A1; amp2 = A2;
freq=f; lamda=lamda;

```

```

pi = 3.1416;
phaselag=lag*2*pi;
% tail parameters
N1=6; N2=7; L1=19e-3; L2=15e-3;
L=280e-3;
Lend=L-N1*L1-N2*L2;
den = 1000; coef_d = 0.5;
sc = 110e-3; m = 0.25*pi*den*sc^2;
front_area=0.25*pi*sc^2;
front_area=sway*front_area;
drag=0.5*den*coef_d*front_area;
damp = 5*pi/180;
omega1=2*pi*freq; omega2=2*pi*freq;
T=1/freq; dt=1/100/freq;
% lighthill model
for i=1:T/dt
    % first segment
    t(i)=i*dt;
    theta1(i)=amp1/N1*sin(omega1*t(i));

theta2(i)=amp2/N2*sin(omega2*t(i)+phaselag);
% tip
Ttheta(i)=N1*theta1(i)+N2*theta2(i);
y1(i)=L1*sum_sin(N1,theta1(i),0);

y2(i)=L2*sum_sin(N2,theta2(i),N1*theta1(i));
y(i)=y1(i)+y2(i)+Lend*sin(Ttheta(i));
dtheta1(i)=omega1*amp1/N1*cos(omega1*t(i));
dtheta2(i)=omega2*amp2/N2*cos(omega2*t(i)+phaselag);
dyl1(i)=0;
for j=1:N1
dyl1(i)=dyl1(i)+L1*cos(j*theta1(i))*j*dtheta1(i);
end
dyl2(i)=dyl1(i);
for k=1:N2
dyl2(i)=dyl2(i)+L2*cos(N1*theta1(i)+j*theta2(i))*(N1*dtheta1(i)+j*dtheta2(i));
end
dyl(i)=dyl2(i)+Lend*cos(Ttheta(i))*(N1*dtheta1(i)+N2*dtheta2(i));
dyx(i)=sin(Ttheta(i));
ht(i)=y(i);
hm(i)=y1(i)+y2(i);
end
ave_yx=mean(abs(dyx));
ave_yt=mean(abs(dyt));
ave_yt2=ave_yt^2;
ave_yx2=ave_yx^2;

```

```

U=sqrt(m*ave_yt2/(coef_d*den*front_area
+m*ave_yx2))
V=sqrt(UM^2+ave_yt2);
detL=Lend;
Ht=mean(abs(ht)); Hm=mean(abs(hm));
beta=UM/V;
alfa=lamda/2/pi*(mean(abs(dyx))/Ht);
effi=froude(alfa, beta)*100

```

```

% result output
result(1)=U; result(2)=effi;

function ssin=sum_sin(N,theta,lag)
ssin=0;
for i=1:N
    ssin=ssin+sin(i*theta+lag);
end

```


References

- [1] *US strategy.* Available: http://www.foreignpolicy.com/articles/2012/07/17/the_future_of_manufacturing_is_in_america_not_china
- [2] *Foxcon Plan.* Available: <http://www.ftchinese.com/story/001039907>
- [3] *NASA.* Available: http://www.nasa.gov/mission_pages/msl/index.html
- [4] *Jiaolong.* Available: <http://www.xinhuanet.com/jiadian/zt/index22.htm>
- [5] R. J. Webster and B. A. Jones, "Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review," *International Journal of Robotics Research*, vol. 29, pp. 1661-1683, Nov 2010.
- [6] *SHOAL.* Available: <http://www.roboshoal.com/>
- [7] *Robot Tadpole.* Available: <http://news.nationalgeographic.com/news/2010/10/101026-bp-gulf-oil-spill-robots-science-nsf/>
- [8] V. C. Anderson and R. C. Horn, "Tensor arm manipulator design," *ASME Transactions*, 1967.
- [9] V. C. Anderson and R. C. Horn, "Tensor arm manipulator," United States Patent, 1970.
- [10] H. Asada and K. Youcef-Toumi, "Analysis and design of semi-direct-drive robot arms," in *American Control Conference, 1983*, 1983, pp. 757-766.
- [11] K. Salisbury, W. Townsend, B. Ebrman, and D. DiPietro, "Preliminary design of a whole-arm manipulation system (WAMS)," in *Robotics and Automation, 1988. Proceedings., 1988 IEEE International Conference on*, 1988, pp. 254-260.
- [12] H. Asada and J. Granito, "Kinematic and static characterization of wrist joints and their optimal design," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, 1985, pp. 244-250.
- [13] G. Robinson, J. Davies, and J. Jones, "Development of the Amadeus dextrous robot end-effectors," in *OCEANS'98 Conference Proceedings*, 1998, pp. 703-707.
- [14] J. Davies, D. Lane, G. Robinson, D. O'Brien, M. Pickett, M. Sfakiotakis, *et al.*, "Subsea applications of continuum robots," in *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on*, 1998, pp. 363-369.
- [15] I. D. Walker and M. W. Hannan, "A novelelephant's trunk'robot," in *Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on*, 1999, pp. 410-415.
- [16] I. A. Gravagne and I. D. Walker, "On the kinematics of remotely-actuated continuum robots," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000, pp. 2544-2550.
- [17] B. A. Jones and I. D. Walker, "Three-dimensional modeling and display of continuum robots," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 5872-5877.
- [18] W. McMahan, V. Chitrakaran, M. Csencsits, D. Dawson, I. Walker, B. Jones, *et al.*, "Field trials and testing of the OctArm continuum manipulator," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, 2006, pp. 2336-2341.

- [19] D. Braganza, D. M. Dawson, I. D. Walker, and N. Nath, "A neural network controller for continuum robots," *Ieee Transactions on Robotics*, vol. 23, pp. 1270-1277, Dec 2007.
- [20] B. A. Jones and I. D. Walker, "Limiting-case analysis of continuum trunk kinematics," in *Robotics and Automation, 2007 IEEE International Conference on*, 2007, pp. 1363-1368.
- [21] S. Neppalli, B. Jones, W. McMahan, V. Chitrakaran, I. Walker, M. Pritts, *et al.*, "OctArm-A soft robotic manipulator," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, 2007, pp. 2569-2569.
- [22] E. Tatlicioglu, I. D. Walker, and D. M. Dawson, "New dynamic models for planar extensible continuum robot manipulators," *2007 Ieee/Rsj International Conference on Intelligent Robots and Systems, Vols 1-9*, pp. 1491-1496, 2007.
- [23] S. Neppalli, M. A. Csencsits, B. A. Jones, and I. Walker, "A Geometrical Approach to Inverse Kinematics for Continuum Manipulators," *2008 Ieee/Rsj International Conference on Robots and Intelligent Systems, Vols 1-3, Conference Proceedings*, pp. 3565-3570, 2008.
- [24] W. McMahan and I. D. Walker, "Octopus-inspired grasp-synergies for continuum manipulators," in *Robotics and Biomimetics, 2008. ROBIO 2008. IEEE International Conference on*, 2009, pp. 945-950.
- [25] S. Neppalli, M. A. Csencsits, B. A. Jones, and I. D. Walker, "Closed-Form Inverse Kinematics for Continuum Manipulators," *Advanced Robotics*, vol. 23, pp. 2077-2091, 2009.
- [26] A. D. Kapadia, I. D. Walker, D. M. Dawson, and E. Tatlicioglu, "A Model-Based Sliding Mode Controller for Extensible Continuum Robots," *Ispra '09: Proceedings of the 9th Wseas International Conference on Signal Processing, Robotics and Automation*, pp. 113-120, 2010.
- [27] A. Kapadia and I. D. Walker, "Task-Space Control of Extensible Continuum Manipulators," *2011 Ieee/Rsj International Conference on Intelligent Robots and Systems*, pp. 1087-1092, 2011.
- [28] I. D. Walker, "Continuum Robot Appendages for Traversal of Uneven Terrain In In Situ Exploration," *2011 Ieee Aerospace Conference*, 2011.
- [29] A. D. Kapadia, I. Walker, and E. Tatlicioglu, "Teleoperation control of a redundant continuum manipulator using a non-redundant rigid-link master," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3105-3110.
- [30] K. C. Walker and D. W. L. Wang, "Analytical Modelling of Deformable Objects for Haptics Virtual Environments," *International Journal of Robotics & Automation*, vol. 27, pp. 92-100, 2012.
- [31] I. A. Gravagne and I. D. Walker, "Kinematic transformations for remotely-actuated planar continuum robots," in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, 2000, pp. 19-26.
- [32] I. A. Gravagne, C. D. Rahn, and I. D. Walker, "Good vibrations: a vibration damping setpoint controller for continuum robots," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2001, pp. 3877-3884.
- [33] I. A. Gravagne and I. D. Walker, "Manipulability and force ellipsoids for continuum robot manipulators," *Iros 2001: Proceedings of the 2001*

- Ieee/Rjs International Conference on Intelligent Robots and Systems, Vols 1-4*, pp. 304-311, 2001.
- [34] I. A. Gravagne and I. D. Walker, "Uniform regulation of a multi-section continuum manipulator," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, 2002, pp. 1519-1524.
- [35] I. Gravagne and I. D. Walker, "Kinematics for constrained continuum robots using wavelet decomposition," *Robotics 2000, Proceedings*, pp. 292-298, 2000.
- [36] M. Csencsits, B. A. Jones, W. McMahan, V. Iyengar, and I. D. Walker, "User interfaces for continuum robot arms," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 3123-3130.
- [37] M. D. Grissom, V. Chitrakaran, D. Dienno, M. Csencsits, M. Pritts, B. Jones, *et al.*, "Design and experimental testing of the OctArm soft robot manipulator," in *Defense and Security Symposium*, 2006, pp. 62301F-62301F-10.
- [38] W. McMahan, B. A. Jones, and I. D. Walker, "Design and implementation of a multi-section continuum robot: Air-Octor," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 2578-2585.
- [39] B. A. Jones and I. D. Walker, "A new approach to Jacobian formulation for a class of multi-section continuum robots," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005, pp. 3268-3273.
- [40] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *Ieee Transactions on Robotics*, vol. 22, pp. 43-55, Feb 2006.
- [41] B. A. Jones and I. D. Walker, "Practical kinematics for real-time implementation of continuum robots," *Ieee Transactions on Robotics*, vol. 22, pp. 1087-1099, Dec 2006.
- [42] M. D. Grissom, V. Chitrakaran, D. Dienno, M. Csencsits, M. Pritts, B. Jones, *et al.*, "Design and experimental testing of the OctArm soft robot manipulator," *Unmanned Systems Technology Viii, Pts 1 and 2*, vol. 6230, 2006.
- [43] B. A. Jones, W. McMahan, and I. D. Walker, "Practical kinematics for real-time implementation of continuum robots," *2006 Ieee International Conference on Robotics and Automation (Icra), Vols 1-10*, pp. 1840-1847, 2006.
- [44] I. D. Walker, D. M. Dawson, T. Flash, F. W. Grasso, R. T. Hanlon, B. Hochner, *et al.*, "Continuum robot arms inspired by cephalopods," *Unmanned Ground Vehicle Technology VII*, vol. 5804, pp. 303-314, 2005.
- [45] M. W. Hannan and I. D. Walker, "Real-time shape estimation for continuum robots using vision," *Robotica*, vol. 23, pp. 645-651, Sep-Oct 2005.
- [46] V. K. Chitrakaran, A. Behal, D. M. Dawson, and I. D. Walker, "Setpoint regulation of continuum robots using a fixed camera," *Proceedings of the 2004 American Control Conference, Vols 1-6*, pp. 1504-1509, 2004.
- [47] C. Laschi, B. Mazzolai, V. Mattoli, M. Cianchetti, and P. Dario, "Design of a biomimetic robotic octopus arm," *Bioinspiration & Biomimetics*, vol. 4, p. 015006, 2009.

- [48] E. Guglielmino, N. Tsagarakis, and D. G. Caldwell, "An octopus anatomy-inspired robotic arm," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 3091-3096.
- [49] E. Guglielmino, L. Zullo, M. Cianchetti, M. Follador, D. Branson, and D. G. Caldwell, "The application of embodiment theory to the design and control of an octopus-like robotic arm," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, 2012, pp. 5277-5282.
- [50] *Snake Arm*. Available: <http://www.ocrobotics.com/>
- [51] T. Mahl, A. Hildebrandt, and O. Sawodny, "Forward kinematics of a compliant pneumatically actuated redundant manipulator," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, 2012, pp. 1267-1273.
- [52] J. A. Ding, K. Xu, R. Goldman, P. Allen, D. Fowler, and N. Simaan, "Design, Simulation and Evaluation of Kinematic Alternatives for Insertable Robotic Effectors Platforms in Single Port Access Surgery," *2010 Ieee International Conference on Robotics and Automation (Icra)*, pp. 1053-1058, 2010.
- [53] K. Xu and N. Simaan, "Analytic Formulation for Kinematics, Statics, and Shape Restoration of Multibackbone Continuum Robots Via Elliptic Integrals," *Journal of Mechanisms and Robotics-Transactions of the Asme*, vol. 2, Feb 2010.
- [54] K. Xu and N. Simaan, "An investigation of the intrinsic force sensing capabilities of continuum robots," *Ieee Transactions on Robotics*, vol. 24, pp. 576-587, Jun 2008.
- [55] K. Xu, R. E. Goldman, J. N. Ding, P. K. Allen, D. L. Fowler, and N. Simaan, "System Design of an Insertable Robotic Effector Platform for Single Port Access (SPA) Surgery," *2009 Ieee-Rsj International Conference on Intelligent Robots and Systems*, pp. 5546-5552, 2009.
- [56] H. T. Sen, N. Deshmukh, R. Goldman, P. Kazanzides, R. H. Taylor, E. Boctor, *et al.*, "Enabling Technologies for Natural Orifice Transluminal Endoscopic Surgery (NOTES) using Robotically Guided Elasticity Imaging," *Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling*, vol. 8316, 2012.
- [57] A. Bajo and N. Simaan, "Kinematics-Based Detection and Localization of Contacts Along Multisegment Continuum Robots," *Robotics, IEEE Transactions on*, vol. 28, pp. 291-302, 2012.
- [58] A. Bajo and N. Simaan, "Finding Lost Wrenches: Using Continuum Robots for Contact Detection and Estimation of Contact Location," *2010 Ieee International Conference on Robotics and Automation (Icra)*, pp. 3666-3673, 2010.
- [59] A. Bajo and N. Simaan, "Kinematics-Based Detection and Localization of Contacts Along Multisegment Continuum Robots," *Ieee Transactions on Robotics*, vol. 28, pp. 291-302, Apr 2012.
- [60] A. Reiter, A. Bajo, K. Iliopoulos, N. Simaan, and P. K. Allen, "Learning-based configuration estimation of a multi-segment continuum robot," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, 2012, pp. 829-834.
- [61] A. Reiter, R. E. Goldman, A. Bajo, K. Iliopoulos, N. Simaan, and P. K. Allen, "A Learning Algorithm for Visual Pose Estimation of Continuum

- Robots," *2011 Ieee/Rsj International Conference on Intelligent Robots and Systems*, 2011.
- [62] A. Bajo and N. Simaan, "Finding lost wrenches: Using continuum robots for contact detection and estimation of contact location," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 3666-3673.
- [63] S. Tully, A. Bajo, G. Kantor, H. Choset, and N. Simaan, "Constrained Filtering with Contact Detection Data for the Localization and Registration of Continuum Robots in Flexible Environments," *2012 Ieee International Conference on Robotics and Automation (Icra)*, pp. 3388-3394, 2012.
- [64] A. Bajo, R. E. Goldman, and N. Simaan, "Configuration and joint feedback for enhanced performance of multi-segment continuum robots," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 2905-2912.
- [65] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and Control of Concentric-Tube Robots," *Ieee Transactions on Robotics*, vol. 26, pp. 209-225, Apr 2010.
- [66] T. Anor, J. R. Madsen, and P. Dupont, "Algorithms for Design of Continuum Robots Using the Concentric Tubes Approach: A Neurosurgical Example," *2011 Ieee International Conference on Robotics and Automation (Icra)*, 2011.
- [67] C. Bedell, J. Lock, A. Gosline, and P. E. Dupont, "Design Optimization of Concentric Tube Robots Based on Task and Anatomical Constraints," *2011 Ieee International Conference on Robotics and Automation (Icra)*, 2011.
- [68] H. L. Ren, N. V. Vasilyev, and P. E. Dupont, "Detection of Curved Robots using 3D Ultrasound," *2011 Ieee/Rsj International Conference on Intelligent Robots and Systems*, 2011.
- [69] A. H. Gosline, N. V. Vasilyev, E. J. Butler, C. Folk, A. Cohen, R. Chen, *et al.*, "Percutaneous intracardiac beating-heart surgery using metal MEMS tissue approximation tools," *International Journal of Robotics Research*, vol. 31, pp. 1081-1093, Aug 2012.
- [70] J. Lock, G. Laing, M. Mahvash, and P. E. Dupont, "Quasistatic Modeling of Concentric Tube Robots with External Loads," *Ieee/Rsj 2010 International Conference on Intelligent Robots and Systems (Iros 2010)*, pp. 2325-2332, 2010.
- [71] E. J. Butler, R. Hammond-Oakley, S. Chawarski, A. H. Gosline, P. Codd, T. Anor, *et al.*, "Robotic neuro-endoscope with concentric tube augmentation," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 2941-2946.
- [72] M. Mahvash and P. E. Dupont, "Stiffness control of a continuum manipulator in contact with a soft environment," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 2010, pp. 863-870.
- [73] M. Mahvash and P. E. Dupont, "Stiffness control of surgical continuum manipulators," *Robotics, IEEE Transactions on*, vol. 27, pp. 334-345, 2011.
- [74] H. L. Ren and P. E. Dupont, "Tubular Enhanced Geodesic Active Contours for Continuum Robot Detection using 3D Ultrasound," *2012*

- Ieee International Conference on Robotics and Automation (Icra)*, pp. 2907-2912, 2012.
- [75] R. J. Webster, J. M. Romano, and N. J. Cowan, "Mechanics of precurved-tube continuum robots," *Robotics, IEEE Transactions on*, vol. 25, pp. 67-78, 2009.
- [76] J. M. Croom, D. C. Rucker, J. M. Romano, and R. J. Webster, "Visual Sensing of Continuum Robot Shape Using Self-Organizing Maps," *2010 Ieee International Conference on Robotics and Automation (Icra)*, pp. 4591-4596, 2010.
- [77] D. C. Rucker and R. J. Webster, "Statics and Dynamics of Continuum Robots With General Tendon Routing and External Loading," *Ieee Transactions on Robotics*, vol. 27, pp. 1033-1044, Dec 2011.
- [78] D. C. Rucker and R. J. Webster, "Parsimonious Evaluation of Concentric-Tube Continuum Robot Equilibrium Conformation," *Ieee Transactions on Biomedical Engineering*, vol. 56, pp. 2308-2311, Sep 2009.
- [79] H. Su, D. C. Cardona, W. J. Shang, A. Camilo, G. A. Cole, D. C. Rucker, *et al.*, "A MRI-Guided Concentric Tube Continuum Robot with Piezoelectric Actuation: A Feasibility Study," *2012 Ieee International Conference on Robotics and Automation (Icra)*, pp. 1939-1945, 2012.
- [80] D. C. Rucker and R. Webster, "Mechanics-based modeling of bending and torsion in active cannulas," in *Biomedical Robotics and Biomechanics, 2008. BioRob 2008. 2nd IEEE RAS & EMBS International Conference on*, 2008, pp. 704-709.
- [81] D. C. Rucker and R. J. Webster, "Mechanics of bending, torsion, and variable precurvature in multi-tube active cannulas," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 2009, pp. 2533-2537.
- [82] R. Webster, J. M. Romano, and N. J. Cowan, "Kinematics and calibration of active cannulas," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 2008, pp. 3888-3895.
- [83] D. C. Rucker, B. A. Jones, and R. J. Webster, "A Geometrically Exact Model for Externally Loaded Concentric-Tube Continuum Robots," *Ieee Transactions on Robotics*, vol. 26, pp. 769-780, Oct 2010.
- [84] D. C. Rucker, R. J. Webster, G. S. Chirikjian, and N. J. Cowan, "Equilibrium Conformations of Concentric-tube Continuum Robots," *International Journal of Robotics Research*, vol. 29, pp. 1263-1280, Sep 2010.
- [85] D. C. Rucker and R. J. Webster, "Deflection-Based Force Sensing for Continuum Robots: A Probabilistic Approach," *2011 Ieee/Rsj International Conference on Intelligent Robots and Systems*, pp. 3764-3769, 2011.
- [86] D. Rucker and R. Webster, "Computing Jacobians and compliance matrices for externally loaded continuum robots," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 2011, pp. 945-950.
- [87] J. Burgner, P. J. Swaney, D. C. Rucker, H. B. Gilbert, S. T. Nill, P. T. Russell, *et al.*, "A Bimanual Teleoperated System for Endonasal Skull Base Surgery," *2011 Ieee/Rsj International Conference on Intelligent Robots and Systems*, 2011.

- [88] R. Webster, A. M. Okamura, and N. J. Cowan, "Toward active cannulas: Miniature snake-like surgical robots," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 2857-2863.
- [89] M. S. Triantafyllou and G. S. Triantafyllou, "An efficient swimming machine," *Scientific american*, vol. 272, pp. 64-71, 1995.
- [90] *Nature Culture Discover*. Available: <http://australianmuseum.net.au/Sailfish-Istiophorus-platypterus>
- [91] P. Domenici and R. Blake, "The kinematics and performance of fish fast-start swimming," *Journal of Experimental Biology*, vol. 200, pp. 1165-1178, 1997.
- [92] J. Z. Yu, M. Tan, S. Wang, and E. Chen, "Development of a biomimetic robotic fish and its control algorithm," *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 34, pp. 1798-1810, Aug 2004.
- [93] H. Hu, "Biologically Inspired Design of Autonomous Robotic Fish at Essex," *Proceedings of the IEEE SMC UK-RI Chapter Conference 2006 on Advances in Cybernetic Systems*, pp. 1-8, September 2006.
- [94] Y. R. Cai, S. S. Bi, and L. C. Zheng, "Design Optimization of a Bionic Fish with Multi-Joint Fin Rays," *Advanced Robotics*, vol. 26, pp. 177-196, 2012.
- [95] M. Ajallooeian, M. N. Ahmadabadi, B. N. Araabi, and H. Moradi, "Design, implementation and analysis of an alternation-based Central Pattern Generator for multidimensional trajectory generation," *Robotics and Autonomous Systems*, vol. 60, pp. 182-198, Feb 2012.
- [96] J. Shao, L. Wang, and J. Yu, "Development of an artificial fish-like robot and its application in cooperative transportation," *Control Engineering Practice*, vol. 16, pp. 569-584, May 2008.
- [97] D. T. Roper, S. Sharma, R. Sutton, and P. Culverhouse, "A review of developments towards biologically inspired propulsion systems for autonomous underwater vehicles," *Proceedings of the Institution of Mechanical Engineers Part M-Journal of Engineering for the Maritime Environment*, vol. 225, pp. 77-96, May 2011.
- [98] 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, p. 1579, 2005.
- [99] E. Mbemmo, Z. Chen, S. Shatara, and X. B. Tan, "Modeling of Biomimetic Robotic Fish Propelled by an Ionic Polymer-Metal Composite Actuator," in *IEEE International Conference on Robotics and Automation*, Pasadena, CA, USA, 2008
- [100] Z. Chen, S. Shatara, and X. B. 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, Jun 2010.
- [101] 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, 2008.
- [102] Z. L. Wang, G. R. Hang, Y. W. Wang, J. Li, and W. Du, "Embedded SMA wire actuated biomimetic fin: a module for biomimetic underwater propulsion," *Smart Materials & Structures*, vol. 17, Apr 2008.
- [103] C. Rossi, J. Colorado, W. Coral, and A. Barrientos, "Bending continuous structures with SMAs: a novel robotic fish design," *Bioinspiration & Biomimetics*, vol. 6, Dec 2011.

- [104] W. S. Chu, K. T. Lee, S. H. Song, M. W. Han, J. Y. Lee, H. S. Kim, *et al.*, "Review of biomimetic underwater robots using smart actuators," *International Journal of Precision Engineering and Manufacturing*, vol. 13, pp. 1281-1292, Jul 2012.
- [105] C. H. Le, Q. S. Nguyen, and H. C. Park, "A SMA-based actuation system for a fish robot," *Smart Structures and Systems*, vol. 10, pp. 501-515, Dec 2012.
- [106] S. Heo, T. Wiguna, H. C. Park, and N. S. Goo, "Effect of an Artificial Caudal Fin on the Performance of a Biomimetic Fish Robot Propelled by Piezoelectric Actuators," *Journal of Bionic Engineering*, vol. 4, pp. 151-158, Sep 2007.
- [107] Q. S. Nguyen, S. Heo, H. C. Park, N. S. Goo, T. Kang, K. J. Voon, *et al.*, "A Fish Robot Driven by Piezoceramic Actuators and a Miniaturized Power Supply," *International Journal of Control Automation and Systems*, vol. 7, pp. 267-272, Apr 2009.
- [108] T. Wiguna, S. Heo, H. C. Park, and N. S. Goo, "Design and Experimental Parametric Study of a Fish Robot Actuated by Piezoelectric Actuators," *Journal of Intelligent Material Systems and Structures*, vol. 20, pp. 751-758, Apr 2009.
- [109] Q. S. Nguyen, S. Heo, H. C. Park, and D. Byun, "Performance evaluation of an improved fish robot actuated by piezoceramic actuators," *Smart Materials & Structures*, vol. 19, Mar 2010.
- [110] D. Barrett, M. Triantafyllou, D. Yue, M. Grosenbaugh, and M. Wolfgang, "Drag reduction in fish-like locomotion," *Journal of Fluid Mechanics*, vol. 392, pp. 183-212, 1999.
- [111] *Essex Robot Fish*. Available: <http://dces.essex.ac.uk/staff/hhu/jliua/index.htm>
- [112] *BUAA Robot Fish*. Available: http://news.buaa.edu.cn/dispnews.php?type=5&nid=1920&table=news_txt
- [113] P. V. y. Alvarado, "Design of Biomimetic Compliant Devices for Locomotion in Liquid Environments," Doctor of Philosophy, Mechanical Engineering, Massachusetts Institute of Technology, 2007.
- [114] *White lined sphinx moth*. Available: <http://www.richard-seaman.com/Wallpaper/Nature/Caterpillars/WhiteLinedSphinxMothCaterpillar.jpg>
- [115] *Octopus*. Available: <http://www.worldbook.com/world-book-explains/item/1331-how-smart-are-octopi>
- [116] *Giraffe tongue*. Available: <http://thehusbandblog.wordpress.com/2008/12/03/fact-and-opinion-6/>
- [117] *Morning Glory*. Available: <http://nipthebud.com/article.asp?articleid=24718>
- [118] *Rattle snake*. Available: <http://kmagnes.blogspot.hk/2011/06/rattlesnake.html>
- [119] *Human finger*. Available: <http://jsj-holds.blogspot.hk/2007/10/hold-index-to-harmonise-fear.html>
- [120] *European Eel*. Available: <http://www.farnhamanglingociety.com/species/eel.php>

- [121] *Monkey*. Available: <http://www.telegraph.co.uk/news/picturegalleries/picturesoftheday/8497449/Pictures-of-the-day-6-May-2011.html?image=5>
- [122] *Rattle snake skeleton - sliding*. Available: <http://www.calpoison.com/public/snakebite.html>
- [123] *Rattle snake skeleton - coiling*. Available: http://scream.wikia.com/wiki/File:Nature_-_rattlesnake_skeleton.jpg
- [124] C. Laschi, B. Mazzolai, V. Mattoli, M. Cianchetti, and P. Dario, "Design of a biomimetic robotic octopus arm," *Bioinspiration & Biomimetics*, vol. 4, Mar 1 2009.
- [125] Z. Li, R. Du, M. C. Lei, and S. M. Yuan, "Design and Analysis of a Biomimetic Wire-Driven Robot Arm," in *Proceedings of the ASME 2011 International Mechanical Engineering Congress & Exposition*, 2011, pp. 11-17.
- [126] R. Hartenberg and J. Denavit. (1964). *Kinematic Synthesis of Linkages*.
- [127] I. A. Karnovsky and O. I. Lebed, *Formulas Structural Dynamics*. New York: McGraw Hill, 2000.
- [128] D. G. Fertis, *Nonlinear Mechanics*. United States: CRC Press, 1998.
- [129] Lighthill.Mj, "Aquatic Animal Propulsion Of High Hydromechanical Efficiency," *Journal Of Fluid Mechanics*, vol. 44, pp. 265-&, 1970.
- [130] *Nature Culture Discover: Istiophorus-platypterus*. Available: <http://australianmuseum.net.au/Sailfish->
- [131] J. E. Colgate and K. M. Lynch, "Mechanics and control of swimming: A review," *Ieee Journal of Oceanic Engineering*, vol. 29, pp. 660-673, Jul 2004.
- [132] Lighthill.Mj, "Hydromechanics Of Aquatic Animal Propulsion," *Annual Review Of Fluid Mechanics*, vol. 1, pp. 413-&, 1969.
- [133] J. Yu, S. Wang, and M. Tan, "A simplified propulsive model of biomimetic robot fish and its realization," *Robotica*, vol. 23, pp. 101-107, 2005.
- [134] Z. Li and R. Du, "Design and analysis of a biomimetic wire-driven flapping propeller," in *Biomedical Robotics and Biomechanics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, 2012, pp. 276-281.
- [135] D. S. Barrett, "The design of a flexible hull undersea vehicle propelled by oscillatoy foil," MPhil, MIT, 1994.
- [136] Lighthill.Mj, "Large-Amplitude Elongated-Body Theory Of Fish Locomotion," *Proceedings Of the Royal Society B-Biological Sciences*, vol. 179, pp. 125-+, 1971.
- [137] M. J. Lighthill, "note on the swimming of slender fish," *Journal of Fluid Mechanics*, vol. 9, pp. 305-317, 1960.
- [138] J. Y. Cheng, L. X. Zhuang, and B. G. Tong, "Analysis Of Swimming 3-Dimensional Waving Plates," *Journal Of Fluid Mechanics*, vol. 232, pp. 341-355, Nov 1991.
- [139] L. A. Yang and Y. M. Su, "CFD Simulation of Flow Features and Vorticity Structures in Tuna-Like Swimming," *China Ocean Engineering*, vol. 25, pp. 73-82, 2011.

- [140] H. Liu, R. J. Wassersug, and K. Kawachi, "A computational fluid dynamics study of tadpole swimming," *Journal Of Experimental Biology*, vol. 199, pp. 1245-1260, Jun 1996.
- [141] H. Suzuki and N. Kato, "A numerical study on unsteady flow around a mechanical pectoral fin," *International Journal Of Offshore And Polar Engineering*, vol. 15, pp. 161-167, Sep 2005.
- [142] B. MC CORMICK, "Aerodynamics, aeronautics and flight mechanics, ISE," 1995.
- [143] J. Y. Cheng and R. Blickhan, "Note on the Calculation Of Propeller Efficiency Using Elongated Body Theory," *Journal Of Experimental Biology*, vol. 192, pp. 169-177, Jul 1994.
- [144] J. J. Videler, *Fish Swimming*. London: Chapman & Hall, 1993.
- [145] Z. Li and R. Du, "Design and implementation of a biomimetic wire-driven underactuated serpentine manipulator," *Transaction on Control and Mechanical Systems*, vol. 1, 2012.
- [146] G. V. Lauder and E. D. Tytell, "Hydrodynamics of undulatory propulsion," *Fish physiology*, vol. 23, pp. 425-468, 2005.
- [147] L. Liao and I. Pasternak, "A review of airship structural research and development," *Progress In Aerospace Sciences*, vol. 45, pp. 83-96, May-Jul 2009.
- [148] "Airships: The Hindenburg and other Zeppelins," ed, 2009.
- [149] *Flying Yachts*. Available: <http://www.theflyingyacht.com/index.php/about/>
- [150] G. A. Houry and J. D. Gillett, "Airship Technology," ed. Cambridge, UK: Cambridge University press, 1999.
- [151] B. M and e. Lando M, "Peculiar performance of a new lighter-than-air platform for monitoring," in *AIAA 4th Aviation Technology, Integration and Operations Forum*, Chicago, IL, USA, Sept. 2004, pp. 1-14.
- [152] *Air Swimmer*. Available: <http://airswimmers.com/>
- [153] *Air Jelly*. Available: http://www.festo.com/cms/en_corp/9771.htm