

CRANFIELD UNIVERSITY

SAMUEL PARFITT

SMART SNAKE RECONNAISSANCE SYSTEM [SASAR]

SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING
Aerospace Engineering

Doctor of Philosophy
Academic Year: 2021 - 2022

Supervisor: Dr. H. Jia
Associate Supervisor: Dr. C. Lawson
January 2022

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND
MANUFACTURING
Aerospace Engineering

Doctorate of Philosophy

Academic Year 2021 - 2022

SAMUEL PARFITT

Smart Snake Reconnaissance System

Supervisor: Dr. H. Jia
Associate Supervisor: Dr. C. Lawson
January 2022

© Cranfield University 2022. All rights reserved. No part of this
publication may be reproduced without the written permission of the
copyright owner.

ABSTRACT

The motivation for this thesis is to develop a new, innovative snake-like robot platform: the Smart Snake Reconnaissance System (SASAR). The snake is chosen as the basis of our design due to their natural adaptability and the versatility of environments they are able to inhabit and traverse through. We therefore design a robot capable of a wide range of applications, from maintenance, to exploration, to photography, in both defence and commercial scenarios. This PhD research has especially focused on three core design pillars: Modularity, Semi-Autonomy, and Novelty.

Four prototype joints to actuate the new robot were designed, following an investigation which examined and assessed existing snake-like robots, and other innovative robotic mechanisms. We analysed their physical characteristics and investigated how the number of degrees of freedom within the mechanism affect their performance. Additionally, research and development were conducted into different methods of control for a snake-like robot. This led to the creation of both 2D and 3D simulations to analyse the motion of the new robot design.

Finally, we present our new snake-like robot, consisting of four independent, low cost, 3D printed modules. Our design implements modularity through the use of a common electrical and mechanical connector, plus we utilise a CAN bus to communicate between the modules of the robot. Semi-autonomy has been implemented by developing a kinematic model for the control of the robot, which takes two simple inputs and calculates the motion required for each link in the chain. The novelty mainly reflects in the design and manufacture of the unique “Quaternion” style Joint module, which uses a complex three-bar parallel linkage to actuate the spine of the robot. The performance of this Joint Module is then analysed, compared to previous snake-robot designs. We therefore manage to achieve our initial goal of creating a novel, modular, snake-like robot.

KEYWORDS

Snake-like Robots, Robotics, Maintenance, Inspection, Hyper-Redundant Manipulators.

ACKNOWLEDGEMENTS

This research is culmination of four years of hard work in both academia and industry, and it could not have been achieved without the help of several amazing people from both Cranfield University and BAE Systems Applied Intelligence Laboratories.

Firstly, from Cranfield, I would like to thank Dr Huamin Jia for his continued support and advice throughout my time in Cranfield. I would additionally like to thank James Whidborne for his initial help with simulation and modelling, and in particular Ben Hopper and Andrea Giampiccolo for their assistance and guidance with the physical testing. Lastly, I would like to thank Shakeb Deane and the rest of the Cranfield University Football team, for giving me an outlet at the end of every working day.

From BAE Systems, I would like to express thanks to Julian Gronland and Richard Harper for their support and supervision over not just the last four years, but our previous time together in Great Baddow. I would like to make special mention for Dan Crane, who designed and painstakingly hand-assembled all the PCBs for our final prototype.

Most of all am grateful for the support of my family, and my partner. Sorry for spending so much time at my desk! I could not have done this without your continued support.

TABLE OF CONTENTS

ABSTRACT	i
KEYWORDS	i
ACKNOWLEDGEMENTS.....	iii
LIST OF FIGURES.....	viii
LIST OF TABLES	xii
LIST OF EQUATIONS.....	xiii
LIST OF ABBREVIATIONS	15
1 INTRODUCTION.....	16
1.1 Snake-Like Robots.....	16
1.1.1 Context.....	16
1.1.2 Snake-Like Robot Designs.....	17
1.1.3 Snake Robot Motion.....	20
1.2 Potential Benefits of Snake-Like Robots.....	22
1.2.1 Advantages of Snake-like Robots	22
1.2.2 Applications.....	23
1.2.3 Disadvantages of Snake-like Robots.....	25
1.3 Research Aims and Objectives	25
1.3.1 Modularity.....	26
1.3.2 Semi-Autonomy.....	26
1.3.3 Novelty	26
1.4 Thesis Structure.....	27
2 EXISTING SNAKE-LIKE ROBOT DESIGNS.....	29
2.1 Introduction	29
2.2 Conventional Snake-like Robot Mechanisms.....	30
2.2.1 Snake-Like Robot Coordinate System	30
2.2.2 Single Degree of Freedom Modules.....	32
2.2.3 Multi-Degree of Freedom Modules.....	40
2.2.4 Analysis of Existing Mechanisms	52
2.3 Alternate Robot Mechanisms.....	59
2.3.1 Novel Soft Actuator for the Musculoskeletal System	59
2.3.2 LIMS2 AMBIDEX - Quaternion Mechanism.....	60
2.3.3 Hydraulic Wrist Mechanism for Hand Prosthesis	62
2.3.4 Alternate Mechanisms Conclusions	63
2.4 Summary	64
3 SASAR MECHANISM DEVELOPMENT	67
3.1 Introduction	67
3.2 Initial Design Calculations.....	67
3.2.1 Module Size	67
3.2.2 Torque Requirement	70
3.2.3 Design Requirements Discussion	72

3.3 SASAR Initial Prototype	73
3.3.1 Overview	73
3.3.2 Mechanical Design	74
3.3.3 Electrical Architecture	75
3.3.4 SASAR 1 Control and Motion	76
3.3.5 SASAR 1 Findings and Analysis	76
3.4 SASAR Design Configuration Choices	80
3.4.1 New Design Criteria	80
3.4.2 Modular Architecture	82
3.5 Joint Module Concepts	84
3.5.1 Pitch-Yaw	84
3.5.2 Pitch-Roll	86
3.5.3 Quaternion	88
3.5.4 Spherical	90
3.5.5 Motion Range Analysis	91
3.5.6 Final Analysis and Conclusion	93
3.6 Quaternion Mechanisms Design	95
3.6.1 Geometry Selection	95
3.6.2 Analysing Quaternion Motion in 3-Dimensions	102
3.6.3 Quaternion Joint Position and Torque Estimation	103
3.7 Summary	107
4 SASAR AUTONOMY AND CONTROL	109
4.1 Introduction	109
4.2 Definition of Autonomy	109
4.2.1 Levels of Autonomy and Application Scenarios	110
4.2.2 Existing Autonomy Definitions and Standards	111
4.2.3 SASAR Definition of Autonomy	117
4.2.4 Definition of Autonomy Conclusion	120
4.3 SASAR 2 Control	121
4.3.1 2D Model Structure	121
4.3.2 Methods of Control for SASAR 2	122
4.3.3 N-Trailer for SASAR	123
4.4 Modelling, Simulation and Analysis	126
4.4.1 Kinematic Model	126
4.4.2 Simulation Results	129
4.4.3 Simulation Discussion	131
4.5 Summary	132
5 SASAR 2: DESIGN AND IMPEMNTATION	133
5.1 Introduction	133
5.2 SASAR 2	133
5.3 Joint Module	134
5.3.1 Design Overview	134

5.3.2 Electrical Architecture and Component Selection	136
5.3.3 Mechanical Design.....	139
5.3.4 Firmware and Control System.....	143
5.3.5 Module Cost and Manufacturing Time.....	144
5.4 Supplementary Modules	144
5.4.1 Drive Module	144
5.4.2 Tail Module.....	146
5.4.3 Head Module.....	146
5.5 Joint Module Analysis	147
5.5.1 Torque Evaluation	147
5.5.2 Results and Discussion	149
5.5.3 Response Evaluation	151
5.6 Comparison To Similar Mechanisms	157
5.7 Summary	158
6 RESEARCH ACHIEVEMENTS AND FUTURE RECOMMENDATIONS	161
6.1 Introduction	161
6.2 Research Findings.....	161
6.3 SASAR 2 Summary and Future Considerations	164
6.3.1 SASAR Summary.....	164
6.3.2 Future Considerations.....	165
6.4 Closing Thoughts.....	167
REFERENCES.....	169
APPENDICES	183
Appendix A Customer Provided Information	183
Appendix B Prototype Mechanisms	184
Appendix C Estimation of Quaternion Joint Performance	186
Appendix D SASAR 2 Costings	188

LIST OF FIGURES

Figure 1.1. Global Distribution of Snakes [2]	16
Figure 1.2. Comparison of Number of Published Articles containing "Robot" to "Snake-Like Robot"	17
Figure 1.3. Snake-like Robot Examples.	19
Figure 1.4. The Four basic Gaits of Snake Motion [19]	22
Figure 2.1. ISO 8378 Robotic Wrist Coordinate System [40] (left) and our system (right)	30
Figure 2.2. Base Mechanism Structures	31
Figure 2.3. FLSUN Kossel Delta-style 3D Printer [41].....	32
Figure 2.4. Amphibot II (top) and Litte Red (bottom) [44,45]	33
Figure 2.5. ACM-R3 (a), ACM-R4 (b), ACM-R4.1 (c), ACM-R4.2 (d) [46–49] ..	34
Figure 2.6. Unified Modular Snake Robot [49] (left) and Series Elastic Snake Robot [50] (right).....	36
Figure 2.7. Mamba (right), Prototype Snake Robot (right) and Serpens (bottom) [51–53].....	37
Figure 2.8. T2 Snake-4 (left) and Screw-drive Snake Robot (right), and Screw-drive Snake robot joint mechanism (bottom) [54,55]	41
Figure 2.9. NTNU Snake Like Robots PIKo (top) and Kulko (bottom) [56,57] ..	42
Figure 2.10. HITSZ-Snakebot II [58].....	43
Figure 2.11. ARCSnake (left) and “U-Joint” Mechanism Schematic (right) [60]	44
Figure 2.12. Woodstock (right), Second Generation HRM Module (centre) and the TSD Snake (right) [61,62]	44
Figure 2.13. Medusa (left) and Joint Schematic (right) [63].....	45
Figure 2.14. KAIRO 3 Snake Like Robot [69].....	45
Figure 2.15. Elephant Trunk Robot (left) and Mechanism Schematic (right) [70]	46
Figure 2.16. ACM-R7 Linkage Mechanism Diagram [72]	47
Figure 2.17. ACM-R7 (left) and ACM-R8 (right) [72,73]	48
Figure 2.18: ACM-R5 (Top), ACM-S1 (Left), ACM-S1 Joint Mechanism (right) [74,75].....	49
Figure 2.19: OmniTread OT-8 (left) and OT-4 (right) [35,77].....	49

Figure 2.20. Module Length Comparison	55
Figure 2.21. Module Aspect Ratio Comparison	56
Figure 2.22. Module Mass Comparison.....	57
Figure 2.23. Joint Angle Range Comparison.....	57
Figure 2.24. Module Torque Comparison.....	58
Figure 2.25. Novel Soft Actuator, Model (left) and Prototype (Right) [82].....	60
Figure 2.26. Quaternion Wrist Mechanism and LIMS2-AMBIDEX [83].....	61
Figure 2.27. Hydraulic Wrist Prosthesis Renders [84]	62
Figure 3.1. Module size and Turn Radius Calculation	68
Figure 3.2. Aspect Ratio Examples	69
Figure 3.3. Module Area Estimates	70
Figure 3.4. Motor Torque Free-body Diagram	70
Figure 3.5. Module Size Estimations for $W = 35\text{ mm}$	72
Figure 3.6. SASAR 1	73
Figure 3.7. SASAR 1 Module CAD Model (left) and CAD model of Modular Snake Robot module [86] (Right)	75
Figure 3.8. Prototype SASAR 1 Hinge Bracket Showing Layer Failure	78
Figure 3.9. Pitch-Yaw Concept CAD Model and Prototype	85
Figure 3.10. Cross Section CAD Model of New Pitch-Yaw Module sub-section	85
Figure 3.11. Pitch-Roll Joint Prototype and CAD Model showing X-Section	87
Figure 3.12. Quaternion Prototype and Model.....	89
Figure 3.13. Spherical Joint Design Prototype and CAD Model	90
Figure 3.14. Mechanism Motion Analysis Models. (a) Pitch-Yaw, (b) Pitch-Roll, (c) Quaternion, (d) Spherical	91
Figure 3.15. Simulated Rolling Contact of two hemispheres using virtual pivot point.....	92
Figure 3.16. Available Workspace for four concept joint mechanisms	93
Figure 3.17. Quaternion Mechanism Geometry and Key Measures. 3D Model (left), 2D Geometry on 3D CAD Model (centre) and 2D Structure showing variable radius (right).	96

Figure 3.18. Graph showing the variance of the theoretical radius for a Quaternion mechanism with values $hc = 30, ho = 5, wc = 25$	97
Figure 3.19. Graphs showing Δr estimates over a range of values for Arm Length (hc) [Top], Joint Length (ho) [Middle] and Arm Width (wc) [Bottom]	98
Figure 3.20. Graph showing Δr estimates for $hc = 30, wc = 25$ and $-25 \leq ho \leq 25$	99
Figure 3.21. Concept Quaternion Mechanism with negative ho value.....	99
Figure 3.22. Geometry Scan Results for Pipe Diameters in 3D and 2D orientations (a: 40mm, b: 60mm, c: 80mm)	101
Figure 3.23. Prototype SASAR Quaternion Joint, showing Interfering limbs ..	103
Figure 3.24. 2D Quaternion Mechanism Wire and Position Changes	104
Figure 3.25. 3D CAD model of a Quaternion mechanism, showing the bending plane angle.	105
Figure 3.26. Static Quaternion Force Model.....	106
Figure 4.1. The 6 Levels of Automation defined in J3016 [91].....	112
Figure 4.2. Diagram of the Boyd OODA Loop [94]	114
Figure 4.3 Level of Autonomy Assessment [95]	116
Figure 4.4. 2D Serpentine Robot Model	121
Figure 4.5. Schematic of SASAR for Use with N-Trailer Variables.....	124
Figure 4.6. Kinematic Model System.....	127
Figure 4.7. Kinematic Model Subsystem Architecture	128
Figure 4.8. Primary SASAR Path Following Results, Showing the position of the head module relative to the desired path, plus the position of the robot at 30 second intervals.....	130
Figure 4.9. Graph showing the total cross-track error of the N-Trailer Model Vs. the SASAR version. Note: only the results for $t \geq 30$ are shown.....	130
Figure 4.10. Simulated Robot in the Webots Simulation Environment	131
Figure 5.1. Render of SASAR 2	134
Figure 5.2: Joint Module CAD Model and Prototype	134
Figure 5.3. Predicted Tensile Load of chosen Geometry	137
Figure 5.4: Basic Electrical Architecture (Lanes required).....	138
Figure 5.5: SASAR Joint Module PCB's	139

Figure 5.6: Joint Module Partial X-Section showing tensile wire path.	139
Figure 5.7: Housing Assembly Process.....	140
Figure 5.8. 3D Printed Main Housing and SLA Printed Housing Collars Ready for Post-Processing	141
Figure 5.9. Predicted Performance of selected Quaternion Geometry	142
Figure 5.10. SASAR Quaternion Mechanism CAD Model showing Assembly	142
Figure 5.11. Iterations of SASAR Quaternion Joint [Bottom]	143
Figure 5.12: CAD model of SASAR 2 Drive Module.....	145
Figure 5.13: Drive Module Prototype	146
Figure 5.14: SASAR 2 Tail Module.....	146
Figure 5.15: SASAR 2 Head Module	147
Figure 5.16. Torque testing set up for Tension mode where $\phi = 30, \beta = 30$, and Compression Mode where $\phi = 0, \beta = 0$	149
Figure 5.17. Response Testing Rig CAD Model (Top) and Rig in Testing Process (Bottom).....	152
Figure 5.18. Joint Module Response to 90° Step Input	153
Figure 5.19. Ramp Input Response Graphs for Joint Module (a) 90 deg/s, (b) 75 deg/s, (c) 60 deg/s, (d) 45 deg/s, (e) 30 deg/s, (f) 15 deg/s.....	155
Figure 5.20. SASAR Joint Model in pose $\mathbf{0}, \mathbf{0}$ [left] and in pose $\pi, \pi/2$ [right], showing how the tension in the cabling is loosened.	156
Figure 6.1. SASAR 2 Joint Module Showing Connection Interfaced	162
Figure 6.2. Example Quaternion Mechanism Showing Limb Failure	164
Figure 6.3. A disassembled Savox SH-0256 Servomotor (left) and the Walfront motors used in SASAR (right).....	166

LIST OF TABLES

Table 2-1. Single DoF Robot Summary.....	39
Table 2-2. Multi-DoF Robot Summary.....	51
Table 2-3. SLR and HRM Mechanisms Physical Properties.....	53
Table 3-1. Module Torque Estimations.....	71
Table 3-2. SASAR 1 Joint Module Properties.....	77
Table 3-3. Total Number of Degrees of Freedom for Non-Wheeled, multi-plane Snake Robots.	79
Table 3-4. Summary of Prototype Joints	94
Table 3-5. Joint Simulation Limits.....	100
Table 3-6. Quaternion Geometric Limits for 3 Pipe Sizes.....	100
Table 4-1. SASAR Autonomy Mode Features (X = Feature Active)	111
Table 4-2. Sheridan's Scale of Degrees of Automation	113
Table 4-3. Definition of SASAR Level 1.....	117
Table 4-4. Definition of SASAR Level 2.....	118
Table 4-5. Definition of SASAR Level 3.....	119
Table 4-6. Definition of SASAR Level 4.....	120
Table 5-1. SASAR 2 Joint Module Properties Comparison	136
Table 5-2. Compression Torque Testing Results in Nm.....	150
Table 5-3. Tension Torque Testing Results in Nm	150
Table 5-4. Average Compression/Tension Torque Testing Results, in Nm	150
Table 5-5. Average Steady-State Errors for a given Input Target.....	154
Table 5-6. SASAR 1 and 2 Performance Comparison verses Similar Snake-Like Robots.	158
Table A-1. Customer Design Specification	183
Table D-1. SASAR 2 Joint Module Costs	188

LIST OF EQUATIONS

(3-1)..... 68

(3-2)..... 68

(3-3)..... 68

(3-4)..... 71

(3-5)..... 92

(3-6)..... 96

(3-7)..... 104

(3-8)..... 105

(3-9)..... 106

(3-10)..... 106

(3-11)..... 107

(3-12)..... 107

(4-1)..... 122

(4-2)..... 122

(4-3)..... 124

(4-4)..... 124

(4-5)..... 124

(4-6)..... 125

(4-7)..... 125

(4-8)..... 125

(4-9)..... 125

(4-10)..... 125

(4-11)..... 125

(4-12)..... 126

(4-13)..... 126

(4-14)..... 126

(4-15)..... 126

(4-16)..... 127

(5-1).....	156
(5-2).....	157

LIST OF ABBREVIATIONS

ACM	Active Cord Mechanism
AFM	Autonomous Flight Management
CAD	Computer Aided Design
CAN	Controller Area Network
CMU	Carnegie Mellon University
COTS	Commercial off-the-shelf
DC	Direct Current
DoF(s)	Degree(s) of Freedom
FDM	Fused Deposition Modelling
FTL	Follow-the-Leader
FZI	Forschungszentrum Informatik
HRM(s)	Hyper-redundant manipulator(s)
ID	Identification
LoA	Level of Autonomy
NTNU	Norwegian University of Science and Technology
PCB(s)	Printed Circuit Board(s)
PID	Proportional-Integral-Derivative
PSR	Prototype Snake Robot
PSU	Power Supply Unit
PWM	Pulse Width Modulation
SAE	Society of Automotive Engineers
SASAR	The Smart Snake Reconnaissance System
SLA	Stereolithography
SLR(s)	Snake-Like Robot(s)
SMA	Shape Memory Alloy
SoC(s)	System(s) on Chip
TCP	Twisted and Coiled Polymer
UART	Universal asynchronous receiver/transmitter
UCSD	University California San Diego
UGV	Unmanned Ground Vehicle
USB	Universal Serial Bus

1 INTRODUCTION

1.1 Snake-Like Robots

1.1.1 Context

Snakes, or the suborder *Serpents* of the *Reptilia* class, are one of, if not the most distinctive creatures in the animal kingdom. Not only due to their unique locomotion methods, but more significantly how they, as a species, have adapted to almost all environments in the world. At the most recent count there are 3921 individual species of snake [1], with snakes found in almost all corners of the globe (Figure 1.1). What is most incredible is all those nearly four thousand species all utilise the same body shape, no matter what environment they find themselves in.

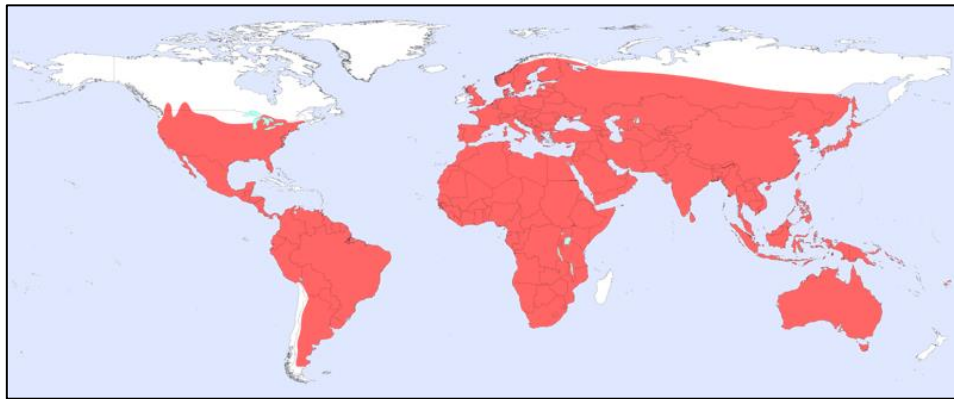


Figure 1.1. Global Distribution of Snakes [2]

This ability to adapt their singular body shape to overcome the challenges of moving through almost any terrain is the main feature which researchers have been looking to recreate through advanced robotics, and devices we call Snake-like Robots (SLRs).

As with all robots, development of SLRs has significantly increased since the turn of the millennium. While this could be attributed to the acceleration of available computing power, there was a sharp rise in specific “Snake Robot” related articles in the early years of the millennium, as shown in Figure 1.2. This rise assumedly follows the tragic events of the 11th of September 2001 where a requirement for a device able to easily delve into complex unstructured voids was discovered.

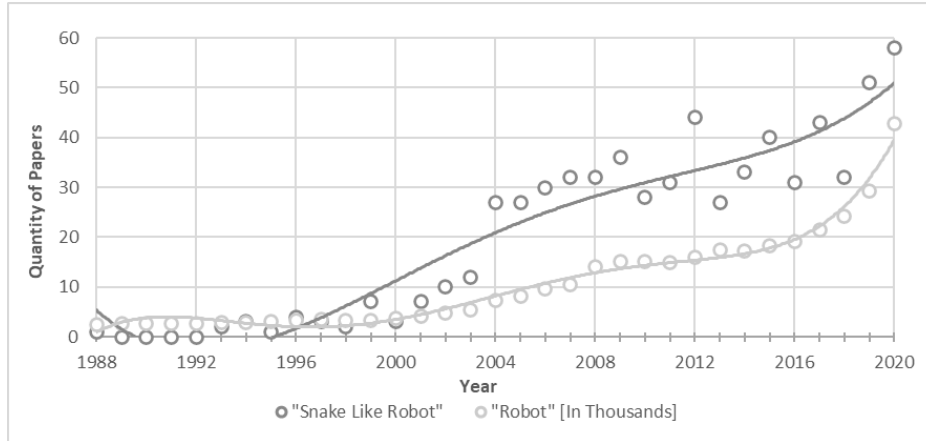


Figure 1.2. Comparison of Number of Published Articles containing "Robot" to "Snake-Like Robot"

The advantages of this robot are clear to see: Where a traditional wheeled or legged robot could be trapped by unclimbable rocks, loose ground, or a sudden transition from land to water, a snake could slither, adjusting its gait to solve each terrain, as necessary. Where a quadcopter or drone would be blocked by a small opening, enclosed space, or narrow pipe, the small cross section of a snake allows it to delve deeper into confined spaces without fear of becoming trapped or losing a propellor to an unseen obstacle.

1.1.2 Snake-Like Robot Designs

Shigeo Hirose performed the initial research and development of SLRs with his Active Cord Mechanism (ACM) series, which has produced near a dozen robots, all with differing capabilities [3–12]. Other work such as that by Pettersen et al of the Norwegian University of Science and Technology (NTNU) have focused on the modelling and control of SLRs, alongside producing their own prototypes [13–22]. Additionally, Crespi et al [23,24] have focused on amphibious robots, whereas researchers from Carnegie Mellon University (CMU) have created novel joint architectures and completed trials using their robots in disaster situations [25–33]. While all these robots are designed and built for different purposes, they keep the basic architecture of the robot consistent.

But how do we define a snake-like robot? The phrase "*Snake-Like Robot*" conjures images of chromed, robotic pythons, with glowing eyes and sharp fangs.

“Snake-Like Robot” is a vague definition and covers a broad array of devices, without explicitly defining what they are. Many authors do have their own definitions for describing the different types of SLR. One such definition comes from Granosik & Borenstein, 2005, who define two types of snake-like robot, the “snake robot” and the “serpentine robot”.

“A ‘snake robot’ ... is a multi-segment mechanism that derives propulsion from the relative motion of the joints only; that is, it uses no wheels, legs, or tracks for propulsion.”

“...a ‘serpentine robot’ is a multisegmented mechanism that derives propulsion from wheels, legs, or tracks. Joints connecting the segments may be either powered or unpowered.”

GRANOSIK & BORENSTEIN, 2005 [35]

Using these definitions, we list a number of design features which are typical to snake and serpentine robots.

Serial Structure. Perhaps the most obvious feature which differentiates all snake-like robots from a traditionally remote controlled UGV (unmanned ground vehicle) is the composition of multiple sections, linked in a sequential chain. The linkages in the chain need not be identical, though commonly they are. These linkages typically contain all the required electro-mechanical, and control systems for control of the robot. While not necessarily defined, a typical snake like robot is made from between 3 and 20 of these actuated sections, usually termed linkages, or modules. Snake robots generally require more linkages to achieve their desired motions, compared to the actively driven serpentine robots.

Joint Actuation. The joint in these linkages is a crucial part of the design, especially in the case of true snake robots, who must produce all their motion using only the forces provided by these actuated joints. For serpentine robots, these joints can be passive pivots, though are typically also controlled. The mechanisms used can vary wildly depending on the specific robot. An electric motor of some kind is most common, but the use of pneumatics and other

transducer systems is also possible. The intention remains: to create a rotation between two of the robots' linkages.

Elongated Aspect Ratio. As with snakes, SLRs have a small cross-sectional area in comparison to their length. Most serpentine robots will tend to have a larger cross-section to allow for the inclusion of the drive mechanism, typically wheels or tracks.

Sensors and Tools. While some are designed simply as developmental platforms, many snake-like robots are created with a purpose in mind. A purpose typically means some form of tool, or more commonly, a sensor is incorporated, and in almost every case the sensor in question is a camera mounted to the front on the robot, giving the operator a first-person point of view. Cameras are by no means the only sensor it is possible to use. Air quality, temperature, and radiation sensing could all easily be included. As for tools, the unique body of snakes means the structure of the robot itself could become the tool for simple tasks, but more complex tools may be included.



Figure 1.3. Snake-like Robot Examples.

A: ARCSnake, B: Mamba, C: ACM-R8, D: OmniTred OT-4 [10,18,34,35]

1.1.3 Snake Robot Motion

The motion of SLRs is evidently geared to imitating natural snakes. First discussed by Gray in 1946 [36], snake motion can be broken down into four distinct motions, each with its own purpose. While these motions have been improved upon for use in SLRs, the basic concept remains the same. The four motions styles are: Lateral undulation, Concertina Locomotion, Rectilinear Crawling, and Sidewinding. While most natural snakes will utilise two, or possibly three of these gaits, a snake-like robot should be able to achieve all four, and transition between them easily. This great versatility is one of the main benefits of the SLR over other conventional mobile robot designs.

Lateral Undulation. This form of motion is also referred to as serpentine motion and is the fastest and most common form of motion in snakes. This motion comes from the snake propagating waves backward along its body from its head towards the tail. The scales of the snake's body grip on imperfections in the ground and provide the propulsion using a reactive force. As this motion is dependent upon the friction between the surface and the snake it may become ineffective on slippery or loose surfaces.

As the creature progresses, every section of its body will cross the same patch of ground, meaning there will be no static contact between the ground and the snake. This constant motion means the snake only needs to overcome static friction once, at the moment it begins the action, and instead is restricted by the easier kinematic friction. Nevertheless, a snake will typically lift its body at the peaks of the curvature to reduce the kinematic friction also.

Additionally, this motion is viable as a form of movement whilst submerged, where the resistance of the water replaces that of the ground.

Concertina Motion. This motion is often employed when the snake is traversing a narrow path. The snake will form curves in its rear section, pushing the peaks of these curves against the walls of the path, creating anchor points. The snake then extends its front section, using the anchor points to push from. Once the front is extended, the snake will form curves using the front section and create

new anchor points with its front. The rear of the snake is then pulled forwards, and the process repeats. Concertina motion is a very slow gait, and inefficient in terms of energy used per unit distance covered, but it can be used to move through tight spaces, where lateral undulation cannot. This gait has also been witnessed allowing a snake to climb vertically up the trunk of a tree.

Rectilinear Crawling. This motion is also very slow and often used by much larger snakes when hunting. This is due to its usually silent and very controllable motion. The snake moves in a straight line, by extending its body forward and creating anchor points on the ground with the scales on its underside. The extended segment then contracts and pulls the snake forward. By having multiple alternating segments of extending and contracting the snake can create a smooth motion, not dissimilar to that of the millipede.

Sidewinding. While there are a variety of species which use all the previous three forms of motion, sidewinding is unique to snakes which make areas with dry loose terrain their homes, such as sand. As discussed above, lateral undulation is difficult on loose surfaces, thus snakes which live in such environments have developed this sidewinding motion to overcome their terrain.

The sidewinding motion has parallels with the concertina locomotion, where half the snake acts as an anchor, while the other half moves. However, *Crotalus Cerastes*, more commonly known as the Sidewinder snake, has been measured moving at nearly 20 mph [37].

A typical motion cycle for sidewinding starts with the snake throwing its head, and the front portion of its body, sideways using the rear of its body as the anchor to push off from. Typically, the direction of motion is about 45 degrees from the direction the head is thrown in. The head, and front portion of the body then act as anchor to allow the rear of the body to be pulled forward, and the process repeats. The motion tends to leave a very characteristic track on the ground.

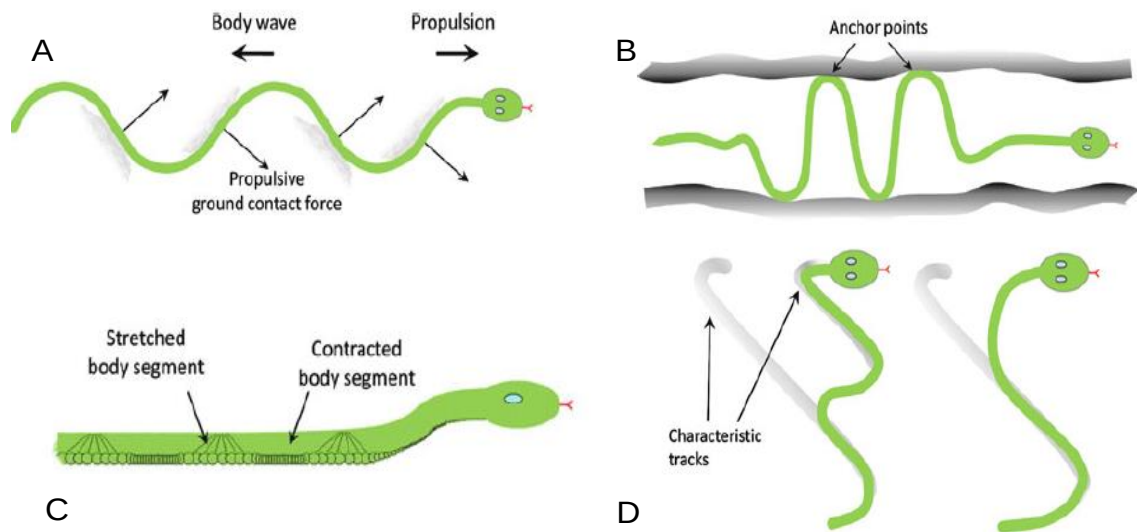


Figure 1.4. The Four basic Gaits of Snake Motion [19]

A - Lateral Undulation, B - Concertina, C - Rectilinear Crawling, D - Sidewinding

1.2 Potential Benefits of Snake-Like Robots

1.2.1 Advantages of Snake-like Robots

There are numerous advantages of snake-like robots. Several of these may be comparable to specifically designed wheeled robots or drones, but the greatest advantage of snake-like robots is their ability to adapt, just like their biological cousins, utilising their body in numerous ways to complete the present task.

This versatility can be broken down into a number of key aspects. First of all, the multiple gaits and other motion techniques offer huge adaptability in terms of the environments a SLR can traverse. More specifically, their ability to transition from one environment to another with only a change in the motion gait. For example, to climb an incline of loose rocks and sand, a typical wheeled robot would rely upon large tyres or tracks to find traction. However, if the next obstacle is a narrow pipe structure, the large tyres and tracks would instantly become a hinderance. Similarly, a small, wheeled robot for moving through narrow pipes: once the device reaches the end of the pipe and must climb a vertical step it would become stuck, whereas a SLR can similarly configure itself to climb up the step using its long body and tail to support itself. Finally, many SLRs have been engineered to be amphibious. The lack of external propulsion mechanisms makes the task of

protecting the internal components from the environment a much easier task compared to wheeled or propelled vehicles.

The second stage of this versatility is the operational flexibility of the devices. This is partly an evolution of the terrain adaptability, but also a feature of the typical SLR mechanical structure. The ability to cross multiple terrains means the user of a SLR can invest in fewer devices. Take the case above of a robot having to navigate through an incline of loose rocks and climb through a small pipe. Rather than requiring several distinct robots, each designed for a particular environment, the ideal snake robot would be capable of performing both tasks appropriately.

The long serial chain of a typical SLR also opens them up to a modular approach to design, with the operator able to select the appropriate configuration for the robot before it commences the task. This could be the addition of extra sensors, or a specific tool.

1.2.2 Applications

Major questions to consider are: within which situations would a snake-like robot be useful? And why would a snake robot be preferred over some other system? We see there being several uses for SLRs, some which are even in use today. Typically, these use-cases revolve around getting either a sensor or small tool into a location a medium distance away, which would otherwise be inaccessible due to obstacles or the environment.

Search and Rescue. One of the greatest advantages of the snake like robot over traditional exploration robots is this ability to crawl through incredibly tiny spaces, relative to the size of the robot. Match this with the capability to delve deep into structures and you have the almost perfect robot for search and rescue operations. The robot can crawl through a wreckage and use a range of sensors to find any trapped individuals, keeping the rescue crew better informed during their operations.

Exploration and Hazardous Environments. The properties which make SLRs perfect for search and rescue will also be vital in general exploration. Not only are the physical properties excellent for getting into difficult terrain, but the different

movement gaits available give additional variety. In loose terrain, such as deep sand, many wheeled robots would find themselves beached, and leg robots would be unable to gain purchase. Additionally, recent advances in machine learning could be transposed to the motion systems of snake-like robots, making navigating through unknown environments easier. As for moving through hazardous environments, any robot, be it snake-like or otherwise, will always be a suitable option above risking human life. A good example case for this would be sending a snake-like robot equipped with a Geiger counter into an area with possible radiation. The robot can then explore and note any areas of dangerous radiation before endangering the operator.

Inspection and Maintenance. Again, using a SLR's ability to manoeuvre through tight spaces could be used in maintenance applications. In many buildings or large vehicles there can be many places which are either hard to access or have to be specially designed to be accessed, for example large pipes, or a wing section which could be prone to rust or fatigue. A device capable of easily traversing these areas could save substantial time during mandatory inspections. Furthermore, specific sensors or tools could be appended to a snake-like robot for a more specialised function. For example, using infrared light and sensors for measuring material fatigue and defects, or a small cable spool for running wires through a complex path.

Reconnaissance and Subterfuge. There is of course a possible military intelligence application for this technology. By their very nature, a small, remotely controllable device, with a multitude of sensors would be an excellent tool for covert surveillance. The ability to mimic a real snake gives an additional layer of camouflage. Likewise, a snake-like robot could be modified to carry a small payload for sabotage operations.

Photography. The final possible application is possibly the least advantageous use of the snake-robots' unique abilities, but there still may be some benefit gained. The vast majority of drones sold to date are for photography. While a snakes-eye-view may not grant the most beautiful perspectives, a suitably

designed snake-like robot could use its own body as a tripod, or clamp to a tree to give a better perspective.

1.2.3 Disadvantages of Snake-like Robots

The intricacy of snake-like robots is, however, a hindrance. Modern computing power and manufacturing techniques may have reduced the cost of creating a snake-like robot, but nevertheless they are still incredibly complex devices with many degrees of freedom which must be controlled in a specific fashion to achieve the desired motion. While the complexity of control of snake-like robots has been accelerating, we are still not quite at the level of natural snakes.

Secondly, there are additionally some weaknesses intrinsic to the design of these robots. The small potential size is one the greatest strengths of the SLR, but it comes with its own drawbacks. As with any mechanical device, reducing the size will restrict the usable components, and lead to compromises in design. In particular, the limit of available torque and speed are a concern.

Another weakness is overall traversing velocity, or the speed at which snakes move. Relative to other creatures, snake motion is energy efficient, but the typical serpentine motion is slow. In nature this does not hamper snakes, as they typically rely on stealth to hunt, but it could prove to be an issue for a robotic device which must complete a task in a given timeframe or with other constraining factors.

1.3 Research Aims and Objectives

The overall aim of this project is to design and build a novel snake-like robot. The robot will be designed to be tasked for routine maintenance and inspection applications, thus must be easy to operate, and have an autonomous mode if possible. This is as desired by our customer, who have provided a full design specification (Table A-1). Additionally, the robot should be able to be used in other scenarios, which are not yet defined. Thus, there must be an allowance for futureproofing and customisability.

We target three goals to achieve this aim, each with their own objectives. They are Modularity, Semi-Autonomy and Novelty.

1.3.1 Modularity

This robot should be able to be easily reconfigured with provisions to be able to include additional functionality at a later date. To achieve this a number of smaller objectives are targeted. They are:

- A common mechanical and electrical interface between the modules of the robot.
- A robust central control system which can be adapted to suit any configuration.
- A universal connection interface between the operator and the robot, to allow for any additional tool or sensor to be added.

1.3.2 Semi-Autonomy

A snake-like robot is a complex robotic device, with typically a large number of actuators and degrees of freedom. In order to reduce the complexity of control, a semi-autonomous system is targeted which would require a low number of operator inputs for a high degree of robot control. We target a number of objectives to achieve this:

- Multiple, well defined control modes, with a range of autonomy levels.
- The creation of a kinematic or dynamic model of the robot, to allow for simulation before manufacture.
- Investigate machine learning or artificial intelligence for the purpose of control and obstacle detection.

1.3.3 Novelty

As the target aims will be difficult to complete through conventional means, we aim to create a design unlike any existing snake-like robot currently existing or in development. The small size target specifically will require us to investigate new or novel joint mechanisms or other emergent technologies. We therefore have a small number of additional objectives for achieving this goal.

- Research and design a new snake-robot actuation mechanism which can be scaled to be both smaller and larger easily. This could be our own new

mechanism design, or an actuator repurposed from a different field of robotics or engineering.

- Use additive manufacturing or other 3D printing technologies to aid in the creation of a novel design. This will also ensure the design is cost and time-efficient and can easily be repaired or defective components replaced. This includes traditional filament based fused deposition modelling (FDM), and resin based Stereolithography (SLA) printing.

1.4 Thesis Structure

In Chapter 1 we have introduced the background context of the project, including a brief look into the history of snake-like robots, why they were created, and what future purposes they could be used for. We also described the common features of snake-like robots, and what advantages these give to SLRs over traditional robots. Finally, we have described the goals for this particular research, and what we hope to achieve with it.

In Chapter 2 we review a selection of existing snake-like robots, plus their more well-defined cousins, the Hyper-redundant manipulator (HRM). HRMs are a type of robotic arm, similar in structure to SLRs but with a fixed base and single free end. We compare and contrast the mechanisms which control these devices and determine a number of different options which could be used in future snake-like robot designs. Finally, we look at mechanisms used in different areas of robots, including anthropomorphic and other bio-inspired robotics, seeking inspiration for new possible mechanism types.

Chapter 3 focuses on the mechanical research for a new snake-like robot. We perform a number of design calculations to define the requirements more accurately for a new snake-like robot. We then present SASAR 1, our first prototype robot, which is used to test the feasibility of reaching our desired goals. Next, we take the research performed in Chapter 2 and create four prototype joint mechanisms for use with a new snake-like robot. These four prototypes are analysed and the one shown to be most suitable is selected. We then apply the

design calculations from the beginning of this chapter to the new mechanism in order to select suitable geometry.

The focus of Chapter 4 is the control and motion for the new prototype robot. We begin by investigating the term “Semi-Autonomous,” producing a list of criteria a snake-like robot could have to be considered semi-autonomous. We create four operational modes for a new snake-like robot, and fully define the criteria required to fully these modes. The modes range from pure manual control to a fully autonomous system. Next, we compare two existing styles of control (Follow-the-leader and N-trailer) and selected the most appropriate for our purposes. This method is then slightly altered from the original design to suit our new robot design. We then simulate the new method using a 2-dimension mathematic model before trialling it into a 3-dimensions physics testing environment.

In Chapter 5 we present SASAR 2, our new prototype snake-like robot. We discuss the mechanical design of the four new robotic modules, focusing on the Joint Module. The Joint Module is then physically tested, and its performance compared to that of the original SASAR 1 robot, plus a selected number of existing snake-like robots which have similar purposes or designs.

Finally, in Chapter 6 we assess the progress of the project against our initial aims and objectives. We look at what possible improvements could be made to the current robot and suggest new options for future research of the robot.

2 EXISTING SNAKE-LIKE ROBOT DESIGNS

2.1 Introduction

In this section we discuss a number of snake-like robots which have been developed for a variety of use-cases, and by a range of institutes. We can categorise snake-like robots in a number of different ways and, as previously mentioned, Granosik and Borenstien [38] use the snake- and serpentine- robot categorisation. This can be divided further by defining two types of snake robot, those which are ‘non’ wheeled and those with ‘passive’ wheels, in addition to the serpentine robots which are considered to have ‘active’ wheels. However, in this section we somewhat disregard these classifications, and focus our attention solely to the mechanisms used to actuate the spine of the robot. We compare how different mechanisms function and investigate how these physical properties of the system are related to this function.

Furthermore, we consider other areas of robotics, primarily hyper-redundant manipulators. Many hyper-redundant manipulators have been produced in the maintenance and inspection sectors. One of the earliest examples comes from the Jet Propulsion Laboratory and is somewhat confusingly named the JPL Serpentine Robot [39]. These HRMs are characterised by their high number of total degrees-of-freedom (DoFs), and are usually formed of a chain of many, characteristically identical, robotic links. Hence the similarity to snake-like robots.

Mounted on a platform of some kind (both stationary and mobile), the end of a HRM generally holds a camera, although some with other sensors and even activatable tools do exist. They are typically used for inspecting enclosed and intricate spaces, where the path to the target location would be inaccessible for a typical 6 DoF robotic arm. The high number of degrees of freedom allows the robotic arm to bend into complex shapes to avoid any obstacles and reach its desired location, but in free space these additional degrees of freedom would be considered redundant, hence the name. While snake-like robots are technically “Hyper-redundant” as they have a much larger configuration space than their task

space, for this project we refer to HRM as overly redundant robotic arms with one fixed.

The final area of robotics we consider are a small range of bio-inspired mechanical joints. These joints are designed to either replace or simulate a natural skeletal joint for the purpose of robotics or prosthetics. We look to see if any of these mechanisms could be used in a snake-like robot to improve the mechanical design of the joint mechanism.

2.2 Conventional Snake-like Robot Mechanisms

2.2.1 Snake-Like Robot Coordinate System

In order to more clearly identify and separate different mechanisms used in snake-like robots and hyper-redundant manipulators we define a generic fundamental mechanism. The robot is described as a chain of these mechanisms, which are serially linked. A local body coordinate frame is connected to each module and is used to assess the variations in designs, with its origin about the centre of mass of the mechanism. We use a system similar to the ISO 8373 Industrial robot configuration and basic parameters for a robot wrist [40], with the addition of an additional prismatic axis in the longitudinal x-axis direction.

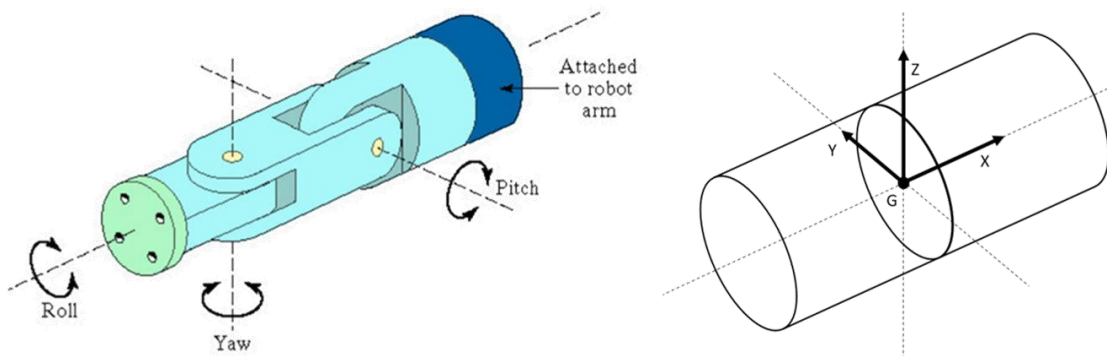


Figure 2.1. ISO 8378 Robotic Wrist Coordinate System [40] (left) and our system (right)

We use a specific nomenclature for describing the complex structures for some of these joint mechanisms. Each joint is described overall in terms of its number of degrees of freedom, and each degree is given a direction of Pitch, Yaw, Roll

or Longitudinal as described by Figure 2.1. Finally, we consider the composition of the joint, in terms of its basic structure. Almost all mechanical joints can be reduced down to a set of basic mechanisms. These basic joints are Revolute (R), Prismatic (P), Universal (U) and Spherical (S) mechanisms (Figure 2.2).

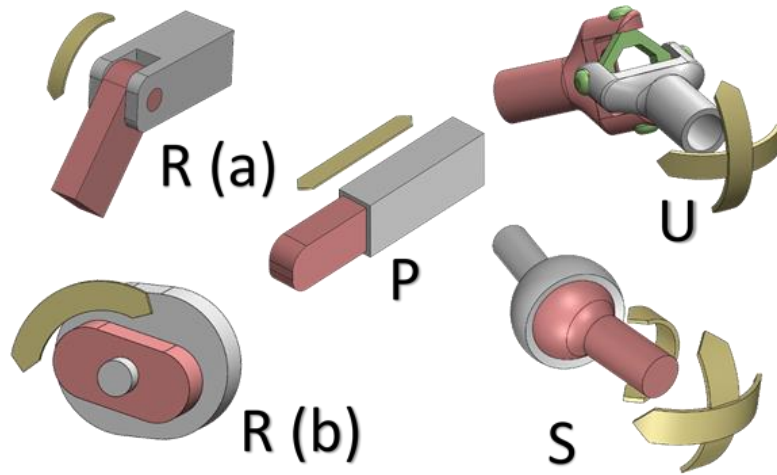


Figure 2.2. Base Mechanism Structures

The robotic joints presented here will either be a single one of these basic mechanisms, or a combination of them in a parallel structure. A parallel linkage is described in terms of the number of identical links, followed by the structure. An example parallel structure would be a delta-style 3D printer (Figure 2.3). The print head can be controlled positionally in X, Y and Z by adjusting the vertical height of the three linear carriages. The carriages are connected to the print head by a ridged bar with spherical ball connectors at each end. This configuration would be considered a **3PSS** system, with the underlined 'P' distinguishing the actuated component of the structure. Any additional structures affixed to the mechanism will be appended in a similar fashion to the description. Should the printer have a rotary motor on the print head to allow the nozzle to rotate in the vertical axis, it would be considered a **3PSS-R** structure.



Figure 2.3. FLSUN Kossel Delta-style 3D Printer [41]

Throughout the remainder of this chapter, we will be exploring how multi-DoF modules compare to single-DoF modules and investigating which mechanisms we should use for our prototype snake-like robot. We take special note of the physical properties of the robot modules, specifically looking at the torque, rotational speed, joint actuation range (or workspace), mass and size. Each of the robots discussed is summarised in a table at the end of each sub-section, and the physical properties are collated in Table 2-3 at the end of this section.

2.2.2 Single Degree of Freedom Modules

A common theme in snake-like robot design is a modular design of single DoF modules, which rotate in either the pitch or yaw direction. These modules are then arranged such that the axis of one module is perpendicular to its neighbours, meaning a pair of these modules would create a 2-DoF, pitch-yaw, mechanism. However, these 1-DoF modules are distinctly separate components, therefore we initially consider them independently.

Single Plane Robots

While the majority of single-DoF module snake-like robots do follow the pattern of alternating pitch-yaw mechanisms, it should be noted that there are a small number which do not. These robots have been designed to function on a single plane only, thus only yaw actuation is required. For example, the ACM-R1 [42], the AmphiBot series [43,44] and Little Red [45] from Hirose, EPFL and Nankai University, respectively. All these robots use a similar mechanical structure for their mechanisms, utilising a combination of a geared electric motor to directly drive the joint, and a potentiometer to control the position. Note that Little Red uses a commercially available “off-the-shelf” (COTS) servomotor (Figure 2.4).

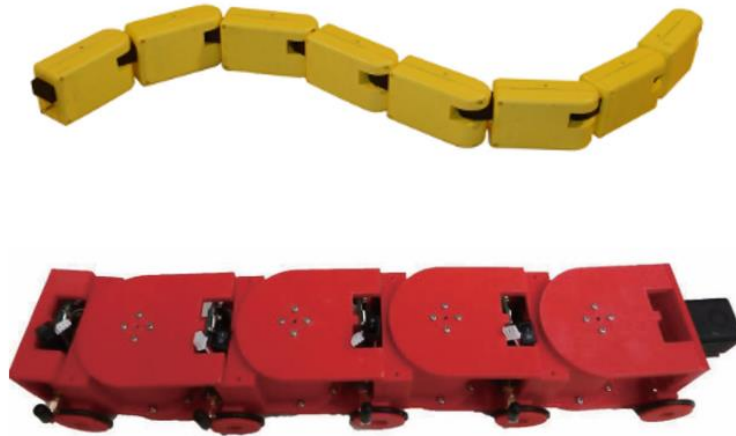


Figure 2.4. Amphibot II (top) and Little Red (bottom) [44,45]

Multi-Plane Robots

ACM-R3 [8] is the first of the Hirose snake robots comprised of identical single-DoF joints but used in multiplane motion. The R3 features large distinctive outer passive wheels on both the pitch, and yaw-oriented modules, allowing the R3 to utilise a variety of serpentine gaits. The R3 is a more function-focused robot with a reduced size and mass compared to its predecessors. It has a length of 155 mm per module, a mass of 650 grams, with a maximum torque output of 19.1 Nm. However, the large wheels limited the available workspace to only ± 62.5 degrees, though the mechanism could reach ± 90 degrees when the wheels are removed. The internal mechanism of these pitch/yaw modules was a geared electric motor,

which would rotate one half of the housing in relation to the other. This mechanism was designed to rotate in the same axis as the wheels, as is the case with the subsequent three robots.

Those three follow up robots are the ACM-R4, and its two evolutions, the ACM-R4.1 and R4.2. These three all share similarities in the design of ACM-R3 but are all actively driven serpentine robots (Figure 2.5). The ACM-R4 [46] resembles a slimmed down version of the R3, with the mass slightly increased per module due to the added drive motor, totalling just under a kilo.

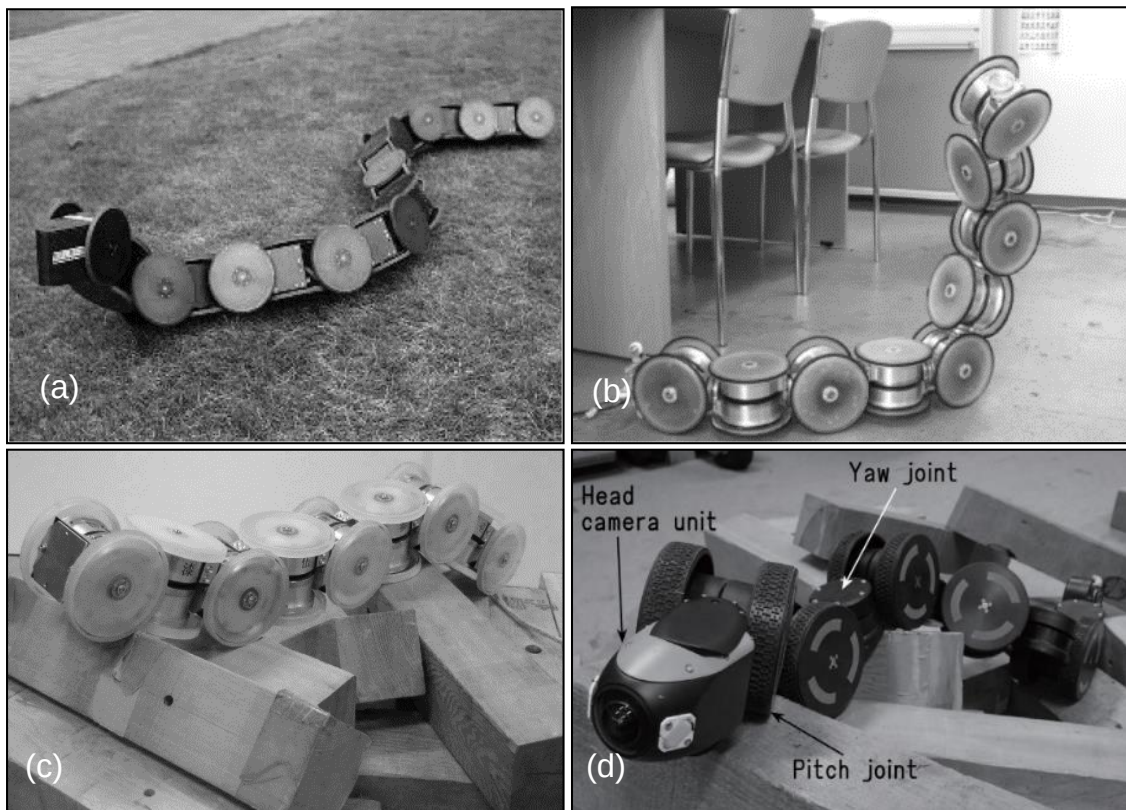


Figure 2.5. ACM-R3 (a), ACM-R4 (b), ACM-R4.1 (c), ACM-R4.2 (d) [46–49]

ACM-R4.1 [47] was an improved version of this, again reducing size and mass. The torque output of the R4.1 was also reduced down to 6.7 Nm from 14.1 Nm in the R4. One notable upgrade in the R4.1 was the addition of a torque sensor and limiter in the joint motor. Head and tail camera units were later added to the R4.1.

For the R4.2 [48] the active wheels on the yaw modules were removed in order to increase the pitch axis operational range from ± 55 degrees in the R4.1, up to

an impressive ± 117.5 degrees in the pitch direction for R4.2. The remaining active wheels on the pitch axis were also increased in size to aid practicability. This robot features many modern upgrades, included a housing made from a vacuum injected carbon-nylon resin, vastly improving the strength-to-weight ratio. This is shown in the comparison to R4.1, where the total of one pitch and one yaw module is 613 grams, compared to the ACM-R4.2's total of only 288 grams for two modules. The mechanism style is much the same, with the same output torque and torque sensing features as the R4.1.

Many snake robots and HRMs have been developed at Carnegie Mellon University over the previous three decades. Here we look at two recent versions: The Unified Snake Robot, and the Series Elastic Snake robot (Figure 2.6). Both follow a similar design principle of a single 1-DoF mechanical joint, though with some slightly different properties.

The Unified Snake Robot [49], also known as Uncle Sam, is a modular snake robot with a relatively small size, each module is 50.8 mm (2 inches) in diameter, and the length between axes is also 50.8 mm. These modules feature a parking brake in addition to the normal motorised mechanism. This brake is designed to lock the mechanism in position without using any energy, instead requiring energy to disengage. This brake increases the holding torque of the mechanism from 2.71 Nm to 6.8 Nm and is controlled using a shape-memory alloy (SMA) to actuate the locking brake. The mechanical power for each module is provided by a geared DC electric motor which is directly connected into the next module in the chain. This linkage also conceals the electrical connection to the next module.

The follow up to Uncle Sam is the Series Elastic Snake Robot, or SEA Snake [50]. The SEA Snake is a slightly upgraded version of the Unified Snake Robot, though the parking brake mechanism has been removed in favour of a more effective motor. The SEA Snake is slightly longer at 64 mm per module, and more powerful, with an expected torque output of 7 Nm. This robot, features an elastic spring element in the output of the joint mechanism, adding a level of passive compliance. There is a slight change in the mechanism, with the mechanical joint now being separate from the electrical connection between the modules.



Figure 2.6. Unified Modular Snake Robot [49] (left) and Series Elastic Snake Robot [50] (right)

Finally, we look at three 1-DoF modules with similar square-profile designs. These are “Mamba” from NTNU [51], “Serpens” from the University of South-Eastern Norway and the “Prototype Snake Robot” from Harbin Institute of Technology, China [52] (Figure 2.7).

Mamba is one of several snake robots created at NTNU. This version uses a powerful COTS servomotor as the actuating element of the mechanism. Interestingly, this servomotor is mounted directly on a series of force sensors, allowing Mamba to measure the contact forces between itself and the environment. The joint modules are also sealed using O-rings, making Mamba completely waterproof. One other design aspect of Mamba which is of interest to us is its modularity. Mamba has several different design configurations, using multiple gaits, including actively driven wheels and fins, passive wheels, and the non-wheels variant shown in Figure 2.7.

Serpens is another serpentine robot, featuring a similar elastic actuator as part of its mechanism. Serpens is designed specifically to be low-cost and uses only

COTS components, in combination with a 3D-Printed body. Serpens has roughly the same profile shape as Mamba, 75 mm compared to 70 mm square, but is over twice the length at 200 mm between axes, compared to 89 mm in Mamba. This is due to the extra gearing that is required to include the elastic actuation element. Additionally, a battery is included in each module, plus Serpens has been designed to have a screwless assembly, thus additional space is required for this mechanism. Each of these modules is independent, and due to the on-board power supplies within each module, there is no need for any electrical connection between modules within the robot. Serpens can be configured in both a single-plane, and multi-plane style.



Figure 2.7. Mamba (right), Prototype Snake Robot (right) and Serpens (bottom)
[51–53]

Finally, we have the Prototype Snake robot (PSR), which is significantly smaller than the other two, down to a 40 mm square, though it is roughly the same module length as Mamba (88mm to 89mm). Where Mamba uses a COTS Servomotor as the main driver for its modules, this PSR uses a small DC electric motor with a

harmonic reduction gear, which is equipped with a torque sensor. The gearing from the reducer gives the PSR 3 Nm of torque, equal with Serpens, compared to 2.3 Nm for Mamba. Despite its metallic construction, the PSR has the lowest mass at only 200g per module, with Mamba and Serpens at 310g and 500g respectively, the included battery pack in each module greatly effecting Serpens in this measure.

Single DoF Summary

The single DoF module is the most common design when it comes to snake like robots and is typically used in a pair with another single DoF module. There are few which do not conform to this, which are single plane robots. These are typically very early prototypes, such as the ACM-R1, or are used for specific research goals, such as the Amphibot or Serpens.

For snake-like robots which move three-dimensionally there are two distinct designs: either the modules are designed to be identical, and the connection point between the modules rotates 90 degrees, or there are two variants of the module, one for each axis of motion. As shown here, there are benefits of both methods, for example having a common interface between all modules when the two are identical. Each of the robots presented here have their own design goals and achieve them in a variety of ways. Table 2-1 gives a summary of each of these robots.

Table 2-1. Single DoF Robot Summary.

Robot Name	Summary
ACM-R1	One of the first Snake-like robots. Uses large motors to power each of the 16 joints. Limited workspace, but each joint has a small spring to allow for some movement out of plane.
AmphiBot,	Fully waterproof single-plane robot designed for amphibious operation. Modular design, with each module containing a small battery, plus a communications bus which runs the length of the robot.
AmphiBot II	An evolution of AmphiBot with upgraded torque output and wireless communications added.
Little Red	Modular 3D Printed construction using servomotors as the actuator. A bespoke PCB controls each module, integrating the power, communication, and sensing for each module.
ACM-R3	Powered by a modified servomotor, each of the 21 modules has distinctive passive wheels for serpentine motion. The wheels limited the actuation range thus were removed from the vertical modules in future iterations of the R3. The joints rotate around the same axis as the wheels.
ACM-R4	Actively driven and upgraded version of the R3. Less torque output, but lighter modules than its predecessor.
ACM-R4.1	A similar upgrade as from the R3 to R4, with less torque, but lighter modules. This version added tactile sensing to each joint to aid in control of the robot.
ACM-R4.2	The next evolution of ACM-R4 which removes the yaw wheels and increases the width of the tyres on the pitch wheels for increased traction. It is much lighter thanks to injection moulded polymer housings. The pitch and yaw modules are functionally identical except for the removed wheels granting a larger range of motion.
Unified Snake Robot	Single DoF Module with geared motor rotated to connect to the next module. The electrical connections are cleverly passed through the axle. Additionally, a breaking mechanism is included to increase the holding torque of the robot.
Series Elastic Snake Robot	Slightly upgraded version of the Unified Snake Robot without the breaking mechanism, but with the addition of a rubber elastic element, adding some compliance to the mechanism.
Mamba	Modular tactile sensing robot with full waterproofing. Additional modules have been designed to allow for various types of manoeuvring.

Serpens	3D Printed construction using COTS drive motor with small elastic actuator. Relatively heavy modules due to included batteries in each module. Designed to not require traditional fixings to connect one module to the next.
Prototype Snake Robot	Metallic construction with bespoke harmonic drive reduction.

2.2.3 Multi-Degree of Freedom Modules

For multi-DoF modules, there are several approaches which have been taken for previous designs. Typically, this depends upon what the designers of each project are setting out to achieve. Here we split these into three further sub-categories: robots which use commercially available motors directly, robots with bespoke revolute mechanisms, and robots with bespoke parallel mechanisms. Here we define bespoke revolute mechanism robots as any multi-DoF joint robot where the degrees-of-freedom are rotational, and not directly driven by a commercial motor. A commercial motor reconfigured or driving a secondary geartrain is considered bespoke. Parallel mechanism refers to the system which powers the joint, as opposed to the motion of the joint itself. The output degrees of freedom from all snake-like robotic modules must be revolute from one module to the next to create the desired motion. These parallel joints simply utilise prismatic mechanisms to achieve this.

COTS Servomotor Joint Mechanisms

Two examples of robots which directly use commercial servomotors as their main actuator are the T² Snake-4 [54] and the Screw-drive snake robot from Kyoto University [55] (Figure 2.8). Both robots use Dynamixel servomotors connected in series with a bracket to rotate each motor in relation to the previous, to creating the 2-DoF joint mechanism required. This is effectively the same as using two single-DoF as before, although there is a distinction. The key difference between multiple single-DoF modules and a single multi-DoF module is the location at which the sections can be easily disconnected. For these two robots, each distinct section contains two motors.

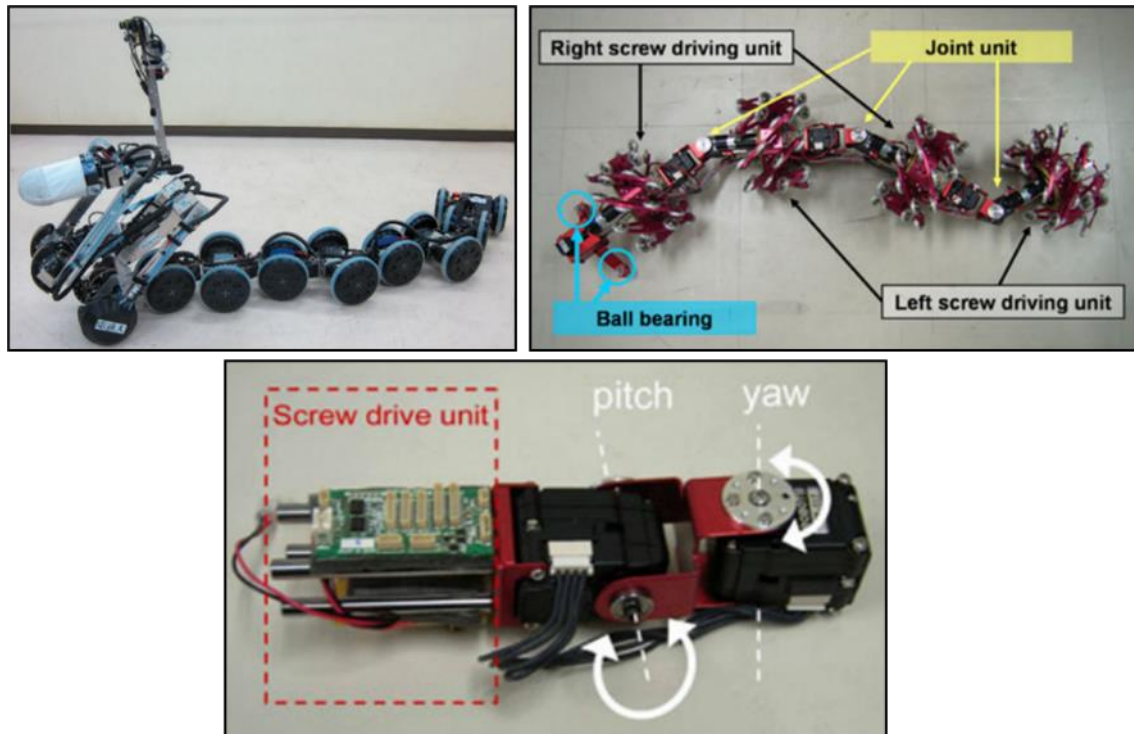


Figure 2.8. T2 Snake-4 (left) and Screw-drive Snake Robot (right), and Screw-drive Snake robot joint mechanism (bottom) [54,55]

Whilst using COTS servomotors as the whole joint mechanism can be useful, it also limits the design in certain ways. The benefits are typically cost, and ease of controllability but the designers lose the ability to incorporate the joint mechanism into the function of the robot. For example, the Dynamixel servomotors used in the T2 snake 4 can output 10.6 Nm of torque and can be accurately controlled to 0.0879 of a degree, which are very impressive values. However, as we will see later, the drawback is the module lengths are typically significantly longer, as the two servomotors must be assembled sequentially. It is no surprise then that this type of design is typically used upon robots where the actuation is not the main emphasis of the robot, where the designers are focusing upon developing other features, such as the Screw-drive drive mechanism here.

Bespoke Revolute Mechanisms

Before Mamba, NTNU had also produced two other snake-like robots, named PIKo and Kulku [56,57] (Figure 2.9). PIKo is a small snake-like robot formed of only five modules, with the key aim of being able to crawl up vertical pipes for the purpose of inspection. The main distinguishing feature of the mechanical design

of PIKo is its lockable wheels. The rubber wheels on the up and lower of this robot are typically passive, but through a mechanism can have their rotation locked, allowing vertical travel inside a pipe network. The joint mechanism is driven by a pair of chain-drive geared electric motors which can output 11.5 Nm, with a maximum travel range of ± 75 degrees.

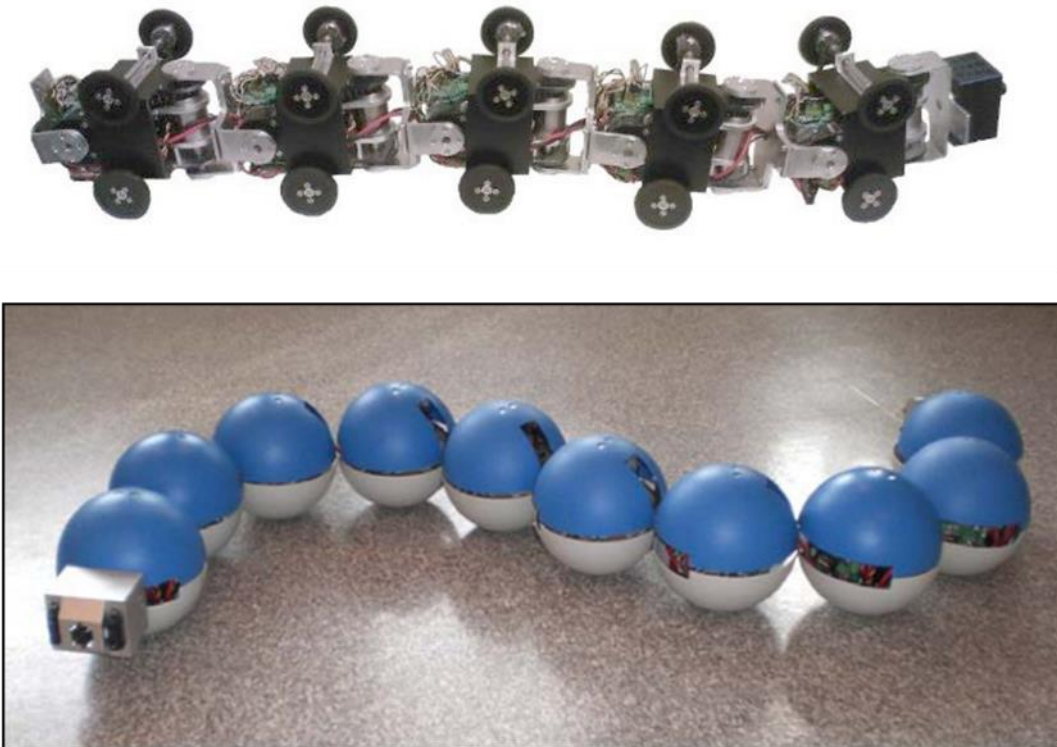


Figure 2.9. NTNU Snake Like Robots PIKo (top) and Kulko (bottom) [56,57]

Kulko is powered by a pair of geared down Hitec servomotors, allowing each module to pivot in both pitch and yaw from inside the outer spherical shell. This shell is the key feature of Kulko, as it is connected to the internal chassis using a set of eight force-sensing resistors. These sensors allow the robot to measure its contact force with the environment and compensate accordingly. The joint mechanism of **Kulko** is a little less powerful, and only 4.5 Nm, and its range is more limited at ± 45 degrees. The arrangement of the mechanism creates a universal style mechanism as the axes for pitch and yaw overlap with each other in the centre of the spheres which make up Kulko. This can lead to advantages such as simplifying the control system, as the separation between rotation axes will be equal for both the pitch and yaw motions.

Similar to Kulko, we have HITSZ-Snakebot II, from Harbin Institute of Technology, China [58] (Figure 2.10). This robot uses a 3D printed spherical structure powered by two servomotors to create the same universal mechanism inside a sphere-shaped shell. The advantages of 3D printing are also used to include passive wheels directly into the spherical structure. The reduced mass of using a 3D printed structure is crucial in this instance, as this robot is intended for use aboard space stations. Using a 3D printed structure means it is not only lightweight, reducing the fuel costs of getting the device to a space station, but can be easily repaired, or even constructed, once onboard by simply printing the required components.

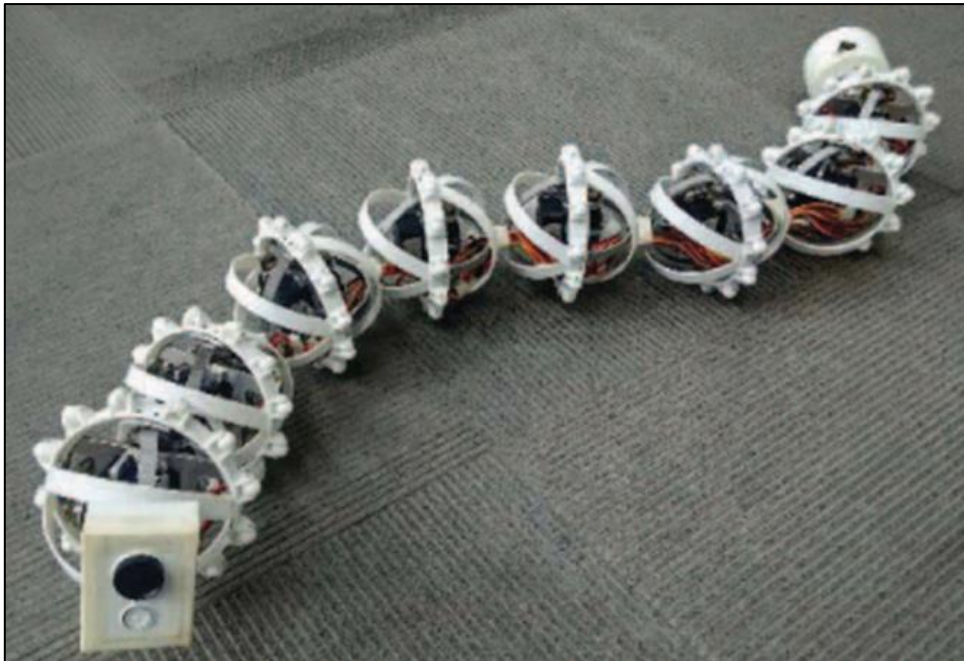


Figure 2.10. HITSZ-Snakebot II [58]

Two other robots which employ a universal style mechanism are the snake-like robot ARCSnake [59] and Woodstock, a hyper-redundant manipulator [61]. The ARC Snake, created at UCSD, is a recent addition to the world of snake-like robots. It is a serpentine robot which features an innovative Archimedean screw-drive mechanism as to provide the longitudinal motion (Figure 2.11). These “Screw-drive” modules are separated by “U-Joint” modules, a 2-DoF mechanism which provides the pitch and yaw motion for the snake. This mechanism is driven by two electric motors which use timing belts to move the universal mechanism.

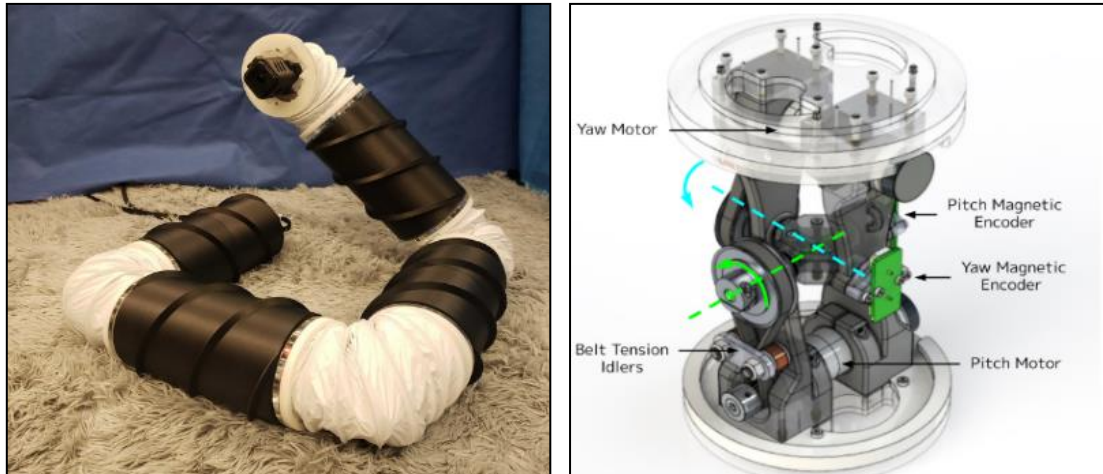


Figure 2.11. ARCSnake (left) and “U-Joint” Mechanism Schematic (right) [60]

Woodstock, also known as the Second Generation Hyper-Redundant Mechanism, is another device developed at Carnegie Mellon University. While initially mounted on a wheeled mobile base, this robot has also been used as the foundation of the “Toroidal skin drive” snake robot [62], a serpentine robot which, as the name suggests, creates forward motion by actively driving its surrounding skin backwards. This robot achieves the same universal style mechanism using an innovative “cross-gear” design (Figure 2.12). This compact design is only 40 mm in diameter and has a length of 101 mm between axes yet can still produce 9.69 Nm of torque. However, it is limited to an operational range of only ± 50 degrees.

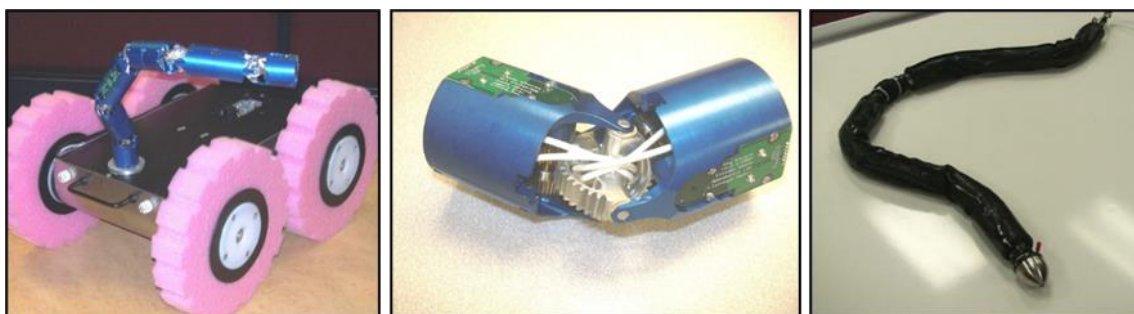


Figure 2.12. Woodstock (right), Second Generation HRM Module (centre) and the TSD Snake (right) [61,62]

The next mechanism is a HRM with a 3-DoF joint from CMU. Known as Medusa, this Hyper-redundant robot uses a bevel style mechanism with three rotational gears (Figure 2.13). This extra rotational degree of freedom allows the

manipulator arm to reach a motion range of up to ± 120 degrees [63]. This robot is one of few here where each module has greater than two degrees of freedom, with a total of three available.

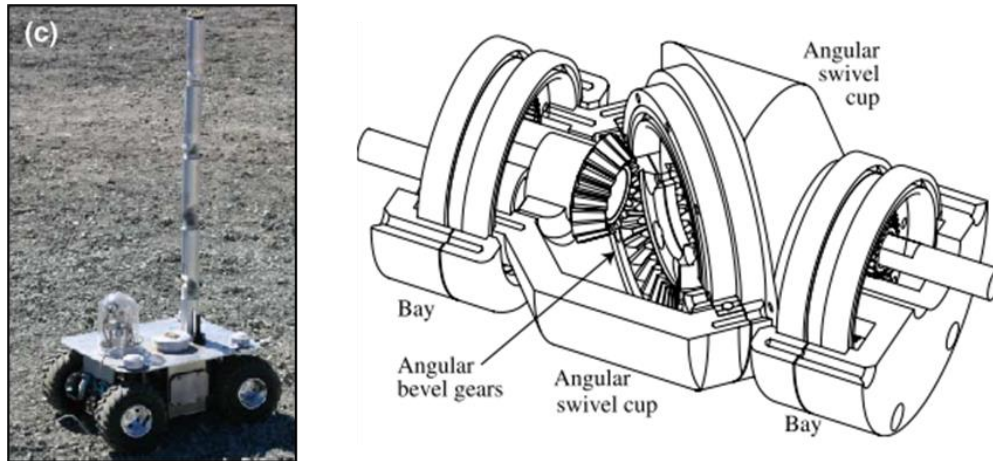


Figure 2.13. Medusa (left) and Joint Schematic (right) [63]

This bevel style mechanism has also been used for a small number of snake-like robots developed by FZI (Forschungszentrum Informatik) in Karlsruhe, Germany. This began in the late 1990s, with a range of actively driven serpentine robots designed for exploring sewage systems, named KAIRO and MAKRO [64–66]. The orthogonal bevel mechanisms and additional degree of freedom allow these robots to rotate sections of their bodies axially, ensuring the driven wheels are always in contact with the surface it is attempting to traverse. The most recent iteration of this series is KAIRO III [67,68]. These robots are significantly larger than the majority discussed in this section, with the early models weighing in at around 30 kg each, and the typical KAIRO-III configuration shown in Figure 2.14 coming to a total of 47 kg.



Figure 2.14. KAIRO 3 Snake Like Robot [69]

The designers describe KAIRO in terms of modules, with the wheeled drive modules separated by the actuating joint modules. The drive modules not only contain the active drive system, but also can be specialised to suit other requirements of the robot, for example as a housing for the embedded PC controller or battery enclosure. The joint module specifically has a mass of 4 kg, and length of 148 mm and a diameter of 160 mm. To actuate a set of electric motors are geared down using a harmonic drive, giving an output torque of 4.8 Nm, and actively controlled using optical encoders to measure the true position.

Robots with parallel Mechanisms

Woodstock was the follow up to another hyper-redundant arm from CMU, known as the elephant trunk robot [70]. This system used a pair of parallel ball screws to provide linear motion into a hinged “cross-joint,” giving a **2PSR** structure (Figure 2.15). The lead screws and corresponding motors are housed together in pairs, with one motor powering the pitch axis rotation of the next cross-joint in the chain, and the other controlling the yaw axis of the previous joint. The lead screws have an actuation range of 22.4 mm, which translates to a rotational workspace of ± 55 degrees, which the mechanism can travel in 5 seconds (11 degrees/second). Various gearings from the motor to the lead screw and cross-joint mechanism ultimately result in a torque output of 4.5 Nm for each axis.

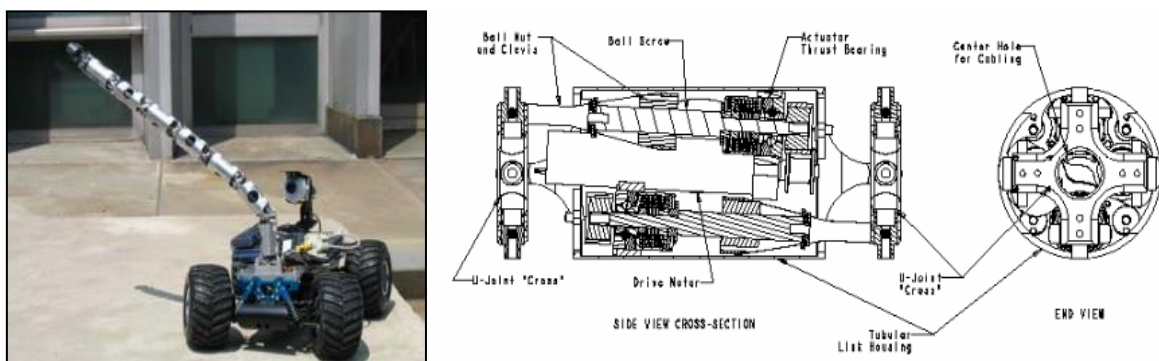


Figure 2.15. Elephant Trunk Robot (left) and Mechanism Schematic (right) [70]

ACM-R2 was the first truly multi-plane robot from Hirose [71]. This mechanism used a system of tensile wires and pulleys powered by two large 20-Watt motors. The pulley system was designed such that both motor axes were parallel and

used an inventive arrangement where both the pitch and yaw degrees of freedom are controlled by both motors. Rotating in the same direction actuates the pitch angle and rotating in opposite directions causes the yaw of the joint to actuate. These modules had a mass of 2.5 kg and were able to output 40 Nm of torque, with the available workspace now increased to ± 60 degrees.

Later, Hirose developed the ACM-R7 and ACM-R8 [72,73] which actuate their joint modules using a similar method. Where ACM-R2 used a cable and pulley system, these two both use lever arms to transmit the motion. In both instances, a pair of rotary motors control two linkage arms, which are connected via ball joints to a secondary frame, hinged in the yaw axis to the linkage arms. This gives a **2RSS-RR** arrangement. Rotating both linkage arms in the same direction alters the pitch angle of the mechanism, while rotating the motors in opposite directions can change the yaw angle (Figure 2.16).

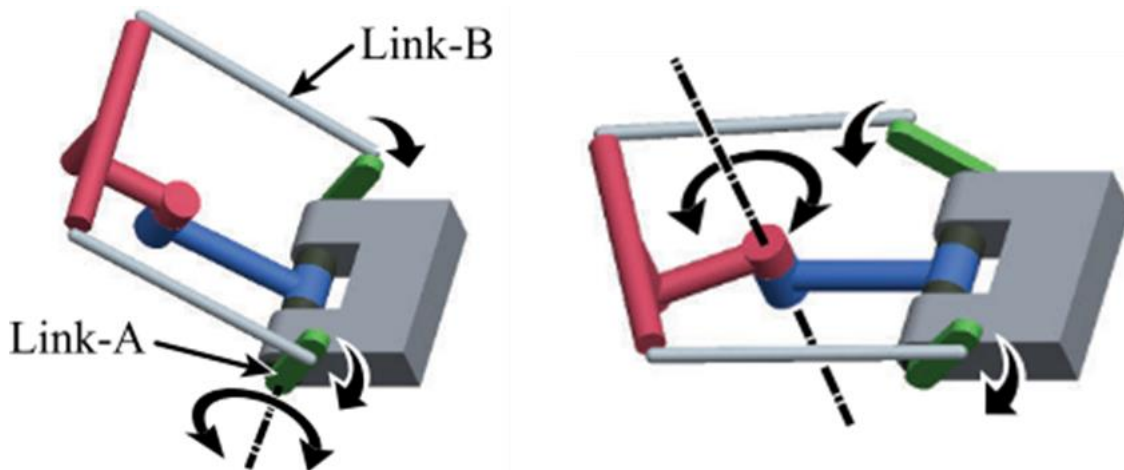


Figure 2.16. ACM-R7 Linkage Mechanism Diagram [72]

However, the two robots could not be much more different in function. R7 is a relatively small development platform (modules size 74 x 140 x 88 mm, 0.65 kg), designed to form itself into a continuous loop, using what is referred to as a “Serpenoid Oval” to roll on the environment. Despite its smaller size, the two 6-Watt motors in the R7 are still able to produce a torque of 16 Nm at the joint interface (Figure 2.17).

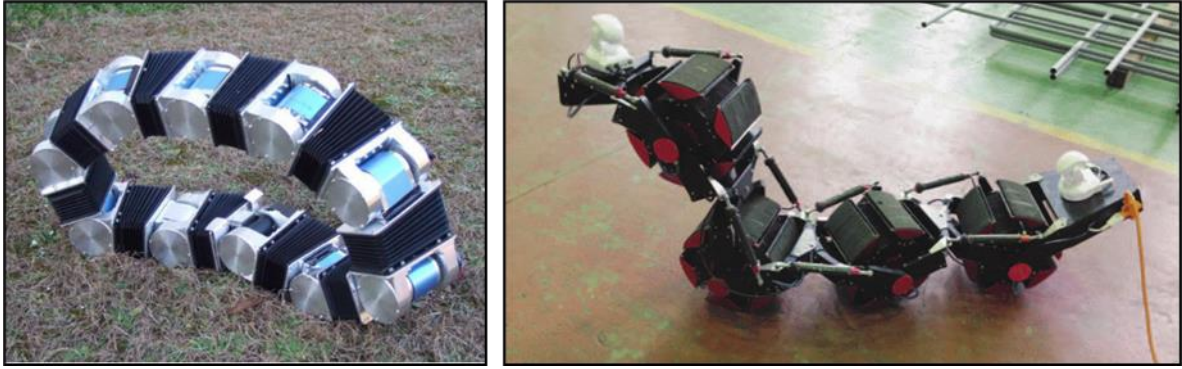


Figure 2.17. ACM-R7 (left) and ACM-R8 (right) [72,73]

Conversely, the R8 is a much larger platform, with each module being 300 mm by 360 mm by 440 mm (Height by Width by Length) with a mass of 8.5 kg. This robot is actively driven by a single large mono-wheel on each module. One minor change in the joint mechanism is the introduction of force sensors in the two linkage arms as a feedback mechanism to the central robot controller.

While not necessary for motion of a snake robot, the addition of a third degree of freedom can still increase the utility of a robot. The first, and thus far only, 3-DoF-module robot from the Active Cord Mechanism series is the ACM-S1 [74]. This robot is a slimline version of the Amphibious ACM-R5 [75] (Figure 2.18). It uses a system of three prismatic slide-screws, with flexible ends, to actuate the robot in pitch, yaw, and longitudinally. This would be classed as a **3P** arrangement. The mechanism inside ACM-S1 can extend and contract through a range of 100 mm, meaning it can almost double the length of each module. For a snake-like robot, this additional longitudinal DoF can allow for more serpentine gaits to be used, such as rectilinear crawling, or could be used for extending over large gaps.



Figure 2.18: ACM-R5 (Top), ACM-S1 (Left), ACM-S1 Joint Mechanism (right)
[74,75]

The final 2-DoF parallel mechanism we consider are the pneumatic bellows of the OmniTread robots, OT-8, and its successor OT-4 [35,76,77]. Their distinctive feature is the square-profile modules, with each of the four outer faces featuring large, powered tracks (Figure 2.19).



Figure 2.19: OmniTread OT-8 (left) and OT-4 (right) [35,77]

Both use a system of pneumatic bellows to actuate the 2-DoF joints between each module. Each module is separated by four of these bellows, which can be inflated with compressed air to extend if required. For OT-8, an additionally universal joint mechanism also runs through the centre of these bellows, restricting the motion to 2-DoF, and doubling as the drive train for the outer tracks, resulting in a **4P-U** structure. These devices are relatively large, OT-4 being 186 mm on a side, and a length of over 250 mm, while OT-8 is around half the size, 82 mm on a side with a length of 145 mm. Due to the limitations of this mechanism, the OT-4 can only manage a maximum working range of ± 41 degrees, reduced to only ± 33 degrees in non-cardinal directions.

One important detail about these robots is they are not repeating modules of the same design. Internally each module serves a purpose, and the robots could not function at all if one of these were missing. For example, the OT-4 is comprised of 7 modules. The centre module contains a drive motor which powers the external tracks, and the two modules either side each contain an air compressor, which feeds the pressurised air to the other module when needed. The final two end modules are both designed to contain a payload if required and can be opened upon command.

Multi Degree-of-Freedom Summary

In most cases, the design of multi-DoF modules has led to the development of innovative designs, creating complex mechanisms to provide the necessary degrees of freedom. For the 2-DoF pitch-yaw snake-like robots and hyper-redundant manipulators an effort is made to overlap the two axes. The exceptions to this are the two robots using COTS Dynamixel servomotors to directly control the joint. It could be argued these are more akin to the single-DoF style mechanisms, as the unit which directly actuates the joint is a separate component, with one degree of freedom.

The ACM-5 and KAIRO 3 robots do highlight some additional features which can be granted from adding a third degree-of-freedom, which in both cases improve the forwards motion of the robot by introducing different available gaits. Finally, there are a number of devices which use parallel mechanisms. These devices

are all very distinct, using different ways to create the prismatic joint, and many more mechanisms to apply it. A summary of all the robots listed in this section can be found in Table 2-2.

Table 2-2. Multi-DoF Robot Summary

Robot Name	Summary
T ² Snake-4	Uses Dynamixel Servos to control the joints and drivetrain. Incorporates many sensors and includes an Omni-Gripper on the arm. Wireless communications to main controller.
Screw-drive Snake Robot	Joint mechanism is two Dynamixel servos. Forward drive is provided by the large screw-drive rings, each with passive wheels on the circumference.
PIKo	Relatively large modules powered by custom joint system of chain driven geared motors. Designed for climbing inside pipes, thus the passive wheels can be locked in place to provide extra grip.
Kulko	Spherical shaped robot with modified servomotors to control the joints. The outer shell of the robot is mounted on force sensors, giving Kulko Tactile sensing.
HITSZ-Snakebot II	3D printed body with a ring of passive wheels. This robot uses servomotors to power the joint actuation.
ARCSnake	Modular robot consisting of two types of modules, one 2 DOF mechanical joint, and one powered Archimedean screw module to drive the forward motion of the robot. The ARCSnake uses a pair of geared motors to drive the joint, controlled using magnetic position sensors.
Woodstock	HRM formed of a chain of 2-DoF modules. The joint is actuated using reconfigured servomotors and a custom PCB, which drive a unique 'cross-gear.' This design has been mounted on a mobile base and used for the TSD skin drive snake robot.
Medusa	A 3-DoF HRM, which uses orthogonal bevel gears to create its motion. It has a large actuation range of 120 degrees. Typically mounted on a mobile base, similar to Woodstock.
MAKRO	Large robot designed for sewer inspection. Each section contained three revolute actuators, two axially and 1 angled to give a bevel-gear style arrangement.
KAIRO 3	Evolution from the same team that created MAKRO, this sewer inspection serpentine robot uses the same bevel-gear style arrangement. It has a modular interface allowing for multiple configurations to be used.

Elephant Trunk Robot	HRM which uses two ball screws, attached to pivot. Each module drives 1 DoF for the next connection, and 1 for the previous. The ball screw help gear the mechanism to give it good torque and workspace.
ACM-R2	Pseudo-parallel linkage which uses tensile cables, driven by a pair of axially aligned motors. This was a very early prototype from Hirose which eventually developed into ACM-R7 and R8.
ACM-R7	Unique parallel linkage using two co-axial motors to provide 2 DoF motion. This modular robot can form itself into a loop and move using a unique rolling motion.
ACM-R8	Uses the same parallel linkage as R7 on a much larger scale. Two of the parallel members contain force sensors to allow for tactile sensing.
OmniTread OT-8	Pneumatically actuated robot which uses four bellows for moving the joints. An axle runs through the robot to drive the tracks on the outside of each module.
OmniTread OT-4	Smaller version of OT-8 which runs on a similar principle. Each module serves its own purpose for the robot, with the end two modules containing an empty space for a payload.
ACM-R5	First multi-DoF module. Similar design to ACM-R5. An amphibious robot which uses three flexible lead screws to get an additional third degree of freedom in the longitudinal direction.

2.2.4 Analysis of Existing Mechanisms

The physical characteristics of each of the mechanisms are listed in Table 2-3. The measurements taken are for one single joint of each robot only, and in the instance where a robot is modular, we only look at the details of the mechanism which provides the joint. For each DoF the mechanism has a corresponding Range and Torque, and they will be presented in the order given in the DoF description column, separated by a comma (,). If the authors do not give a certain value, it will be represented with a hyphen (-). The units may change depending on the mechanism, different units are described in the column heading and represented in italics. If only the mass or length of the whole robot was given, the value per module is estimated by averaging the total by the number of modules. The module size will be given in either diameter by length, or width by height by length, where the length being the distance between two rotational axes.

Table 2-3. SLR and HRM Mechanisms Physical Properties

Robot Name	Degrees of Freedom	DoF Range ($\pm$Deg or mm)	Torque/ Force (Nm or N)	Module Size (mm)	Module Mass (g)
ACM-R2 [71]	2 [Pitch, Yaw]	60.0, 60.0	40.20, 40.20	200.0 x 190.0 x 180.0	2500
ACM-R3 [47]	1 [Pitch/Yaw]	62.5	16.00	140.0 x 74.0 x 143.0	605
ACM-R4 [46]	1 [Pitch/Yaw]	100.0	14.10	130 x 130 x 122.2	956
ACM-R4.1 [48]	1 [Pitch/Yaw]	65.0	6.70	104.0 x 104.0 x 108.8	613
ACM-R4.2 [49]	1 [Pitch]	117.5	6.70	102 x 69 x 98.3	288 ^A
	1 [Yaw]	68.0	6.70	102 x 69 x 98.3	
ACM-R7 [72]	2 [Pitch, Yaw]	90.0, 90.0	16.00, 16.00	140.0 x 74.0 x 143.0	650
ACM-R8 [73]	2 [Pitch, Yaw]	90.0, 45.0	100.00. 100.00	360.0 x 300.0 x 440.0	8500
ACM-S1 [74]	3 [Pitch, Yaw, long]	60.0, 60.0, 100.0	-, -	80.0 x 268.0 ^B	-
ETR [70]	2 [Pitch, Yaw]	55.0, 55.0	4.50, 4.50	-, -	-
Woodstock [61]	2 [Pitch, Yaw]	50.0, 50.0	9.69, 9.69	40.6 x 101.0	177
Uncle Sam [50]	1 [Pitch/Yaw]	90.0	1.30	50.8 x 50.8	160
SEA Snake [51]	1 [Pitch/Yaw]	90.0	7.00	51.0 x 64.0	205
PSR [53]	1 [Pitch/Yaw]	90	3.00	40.0 x 40.0 x 88.0	200
KAIRO III [67,68]	3 [Pitch, Yaw, Roll]	90.0, 90.0, 360.0 ^C	4.80, 4.80, 4.80	160.0 X 148.0	4000
Medusa [63]	3 [Pitch, Yaw, Roll]	120.0, 120.0, 360 ^C	-, -, -	46.0 x 185.6	450

PIKo [57]	2 [Pitch, Yaw]	75.0, 75.0	11.50, 11.50	130.0 x 140.0 x 150.0	1252
Kulko [58]	2 [Pitch, Yaw]	45.0, 45.0	4.50, 4.50	130.0 x 130.0	960
Mamba [52]	1 [Pitch/Yaw]	90.0	2.30	70.0 x 70.0 x 89.0	310
Serpens [54,78]	1 [Pitch/Yaw]	90.0	3.00	75.0 x 75.0 x 200.0	500
T ² Snake-4 [55]	2 [Pitch, Yaw]	113.0, 65.0	10.60, 10.60	218 x 250 x 181	1022
ScrewDrive Robot [56]	2 [Pitch, Yaw]	90.0, 90.0	3.70, 3.70	180 x 226.0	-
ARCSnake [60]	2 [Pitch, Yaw]	90.0, 90.0	2.10, 2.10	110 x 168	880
HITSZ- Snakebot II [59]	2 [Pitch, Yaw]	45.0, 45.0	1.77, 1.77	103.0 x 126.0	330
MAKRO [79]	3 [Pitch, Yaw, Roll]	90.0, 90.0, 360.0	-, -, -	160.0 x 150.0	6000
OmniTread OT-8 [80]	2 [Pitch, Yaw]	-, -	-, -	186.0 x 186.0 x 268.0	2720
OmniTread OT-4 [81]	2 [Pitch, Yaw]	41.0, 33.0	-, -	82.0 x 82.0 x 145.0	357
A. The mass is given for two 1-DoF modules. One each of Pitch and Yaw B. This is the maximum length ACM-S1 can extend to C. It is unclear from text if 360-degree rotation implies an unlimited rotation or a total of 360.					

Many of the hyper-redundant mechanisms shown here share similar qualities, but equally, in many cases it is hard to compare one robot to the next, as they are almost all independently designed for a particular purpose. Additionally, when comparing snake-like robots we must consider the chosen locomotion method of the robot, is it considered a snake robot or a serpentine one? We must also consider the use-case and year of development of these robots, if we wish to accurately compare the mechanisms used. For example, ACM-R2, ACM-R8, KAIRO-3 and OmniTread OT-8 are all significantly larger than the rest of the robots considered here, as their use-cases demanded they be. Additionally, not

all of the information required to truly analyse these mechanisms against each was available, therefore we have had to remove these from our evaluation.

For the 18 remaining robots for which we have the required information, we begin by comparing the difference between single and multi-DoF modules. We examine three physical properties: the total module length, the operational range, and the torque-mass ratio of the robots. We then evaluate them in their own category and consider how the averages for single and multi-DoF modules compare.

Module Size Analysis

In order to make the comparisons fairer and more accurate, we compare the physical properties of two single-DoF modules to those of a single multi-DoF module, beginning with the size. In terms of length, there is a clear trend of two-single DoF module to be larger. Figure 2.20 shows a comparison of the relative lengths of each of the modules, along with an average for the two types. Of the selected modules, the single-DoF modules average at a total length of 205.5 mm compared to 150.7 mm for the multi-DoF modules. There is clearly an outlier here in the robot *Serpens*, which when removed from the calculation brings the single-DoF average down to 177.7, still over an inch difference on average. This clearly suggests that if the aim is to reduce length, then a multi-DoF module should be used.

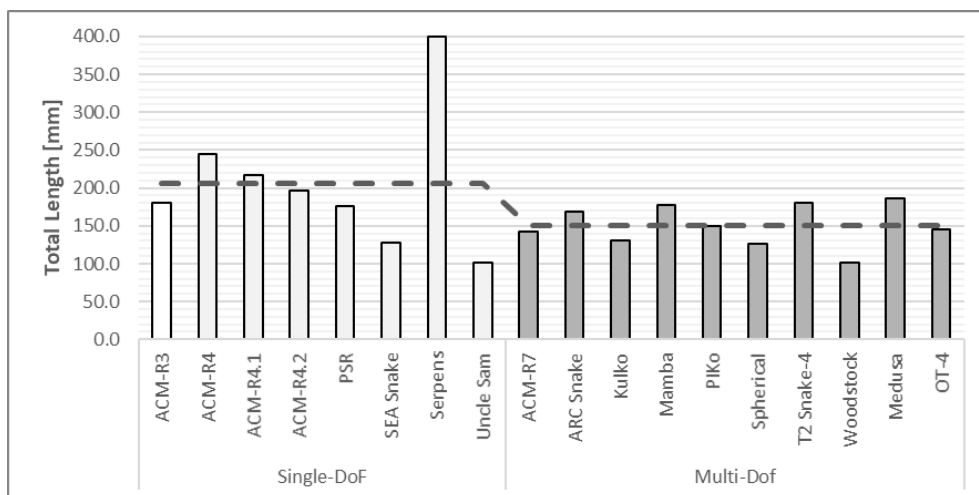


Figure 2.20. Module Length Comparison

As we wish to inspect other aspects of the module size, we also consider the maximum widths of the mechanisms. However, if we inspect the width values, we can see that they cover a broad range, fluctuating from as low as 40 mm for robots such as Woodstock, to as high as 150 mm with the T2 Snake-4, though this does include the driven wheels of the Snake-4. Instead, we consider the aspect ratio of the mechanisms, as a ratio of length to width.

Figure 2.21 shows the results of this analysis, which displays that the multi-DoF modules are typically wider, but this gives them a squarer profile, with many coming close to, or equalling, a ratio of 1:1, length: width. The average values for each are 2.72 and 1.79 for the single- and multi-DoF modules, respectively. From this we can discern that width is a more crucial aspect of design for the single-DoF modules, and if we wish to increase the width of a single DoF module, the length may have to increase more compared to a proportional width increase in a multi-DoF module.

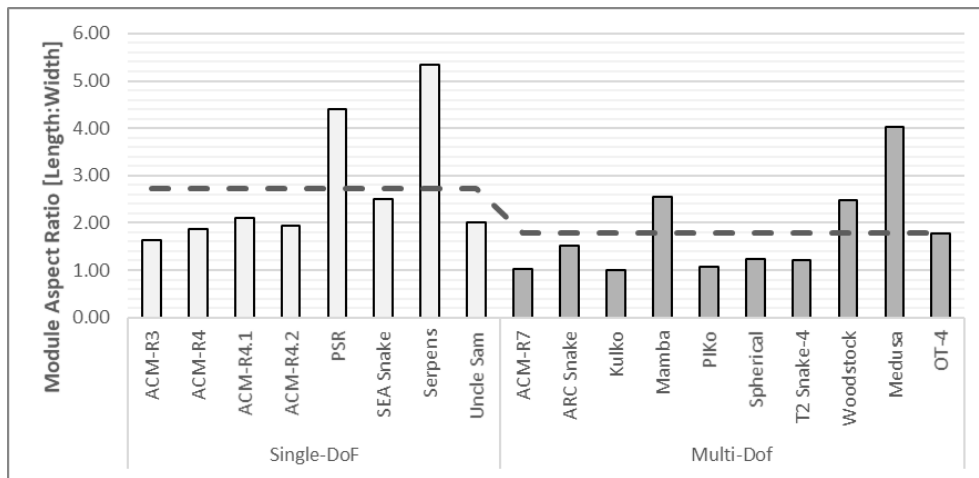


Figure 2.21. Module Aspect Ratio Comparison

The final physical property we consider is mass. This is a more difficult factor to analyse, as we must consider the manufacture of the robot, as well as the era in which it was manufactured. From Figure 2.22 we could easily conclude the single-DoF are on average heavier than their multi-DoF counterparts. This statement would seem correct, as we have just analysed the lengths and seen the single-DoF modules are also longer, thus the additional mass is unsurprising. However,

as can be seen between ACM-R4.1 and R4.2, a single evolution in design can reduce mass considerably once modern materials are used.

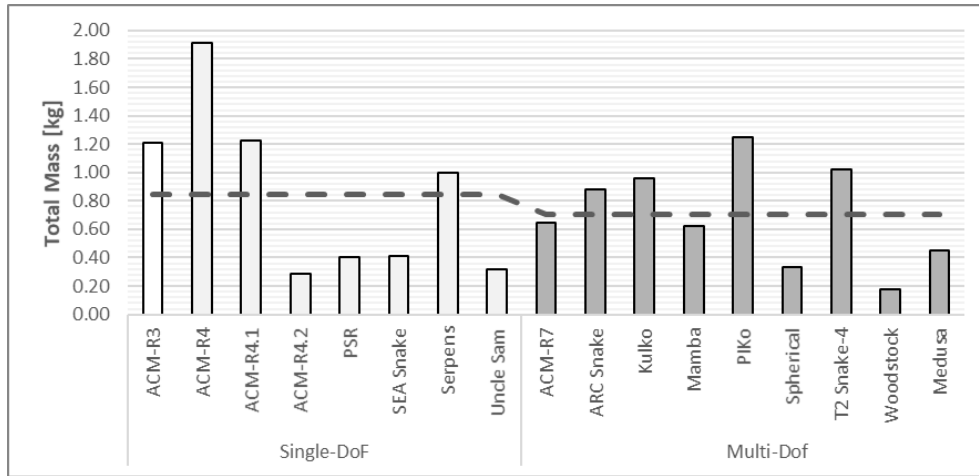


Figure 2.22. Module Mass Comparison

Module Performance

Next, we consider the performance characteristics of the modules, beginning with the operational range. Here we see a reverse in results in comparison to the size analysis, with the single-DoF modules having a better range more consistently, although the majority of the mechanisms do reach ± 90 degrees, which is a common target workspace (Figure 2.23).

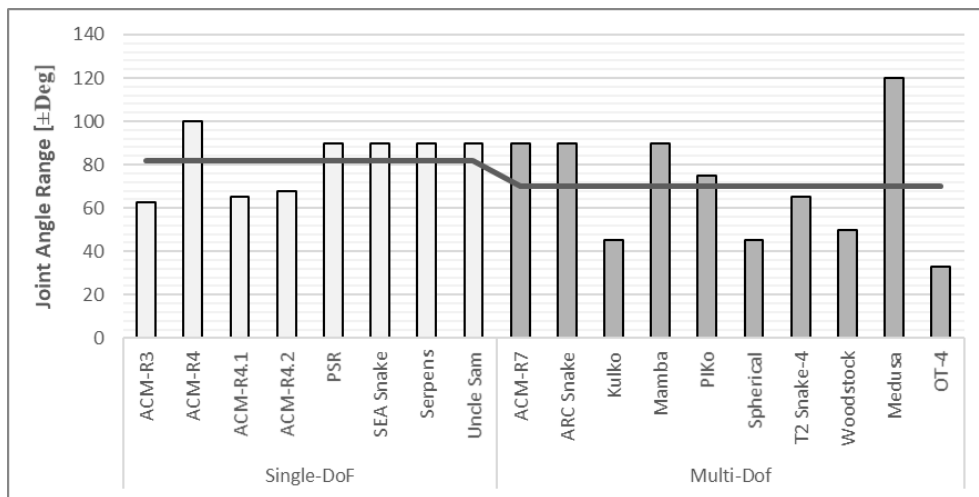


Figure 2.23. Joint Angle Range Comparison

Finally, we consider the module torque output. This value is difficult to compare across modules as there is very little relation between torque and if the module

is multi-DoF or Single-DoF; though as Figure 2.24 shows, single-DoF modules do tend to have marginally higher torque, other than a few outliers. There is a minor trend between torque and the module mass, and torque and module volume (Figure 2.24). This is understandable as common sense would dictate more mass and more space mean more powerful motors can be included. We suspect the main deciding factor of torque output will be the module speed. This will have an inversely proportional relationship, depending upon the gear ratios used in the drive train. Unfortunately, the rotation speed of most systems is not provided by many of the authors.

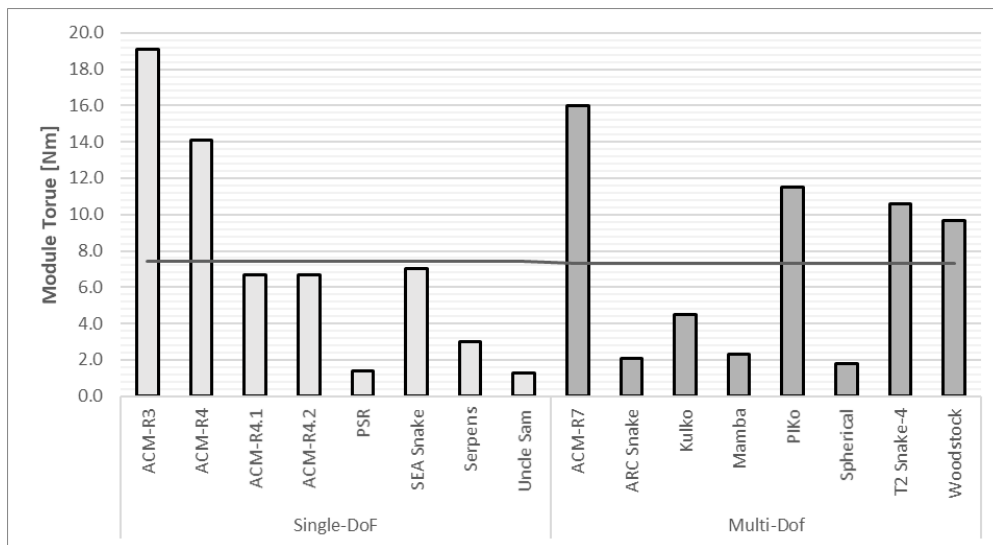


Figure 2.24. Module Torque Comparison

Analysis Conclusions

Snake-like robots and their counterparts, Hyper-redundant mechanisms, share many physical properties. However, both also use a variety of different joint structures to achieve their required motion. For snake-like robots, this joint structure is typically a simple use of COTS servomotors, although in many cases a bespoke mechanism designed to enhance the locomotion style. The mechanism designs differ considerably, depending on the chosen style of the robot. For example, serpentine robots may have more powerful motors to ensure they can lift the extra mass added by the secondary drive system, whereas snake robots may need additional power as they rely solely upon the actuation of their joint mechanism to move.

For many Hyper-redundant manipulators, the project goals drive the design of the mechanisms. For example, the robot *Medusa* was developed to maximise the usable workspace of each module, hence a novel mechanism was created which increased the range of motion.

Analysis of a set of 18 robots, both consisting of two single-DoF modules and a single multi-DoF module has shown that both types have their benefits. Single-DoF modules are typically able to reach greater joint angles, and are narrower, but at the cost of increased mass and crucially a large increase in module length. However, these differences are minimal, and could possibly be explained by the additional mass and space required for the connection between the two single-DoF modules.

A critique of this analysis would suggest the sample size is too small to draw conclusions from, and the complexity of designing a robotic system can give the creator greater freedom to alter their design. For example, reducing the mechanism speed to increase the torque output. However, it is possible to theorise that producing a tailored multi-DoF mechanism does allow for a smaller mechanism, which can incorporate other components. Nevertheless, this leads to a wider mechanism with a shorter range of motion.

2.3 Alternate Robot Mechanisms

To further aid our investigations we look to other areas of robotics, searching for multi-DoF mechanisms which could be used to create a new snake-like robot. We look for new technologies and unique mechanical designs, which can be shown to improve upon the traditional geared electric motor used in the vast majority of snake-like robot models.

2.3.1 Novel Soft Actuator for the Musculoskeletal System

This device, created at the University of Texas, Dallas, is a close approximation of the spinal column of a human. This system uses a Twisted and Coiled Polymer (TCP) muscle, fabricated from a fishing wire [82]. The application of a current across the TCP causes it to contract, and when used in antagonistic pairs can mimic muscle fibres. This is then combined in a silicone mould with a 3D printed

structure acting as the ball and socket joint (Figure 2.25). The resulting structure is similar to the OmniTread, with a **4P-S** structure.

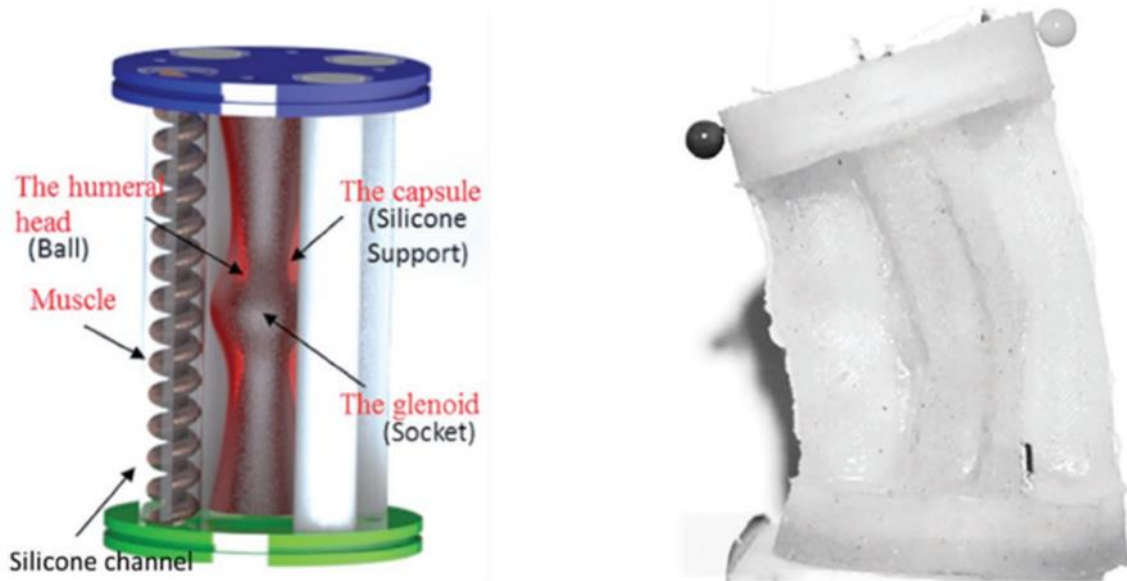


Figure 2.25. Novel Soft Actuator, Model (left) and Prototype (Right) [82]

A line of multiple of these mechanisms could hypothetically create robotic imitation of a natural snake, replicating the vertebrae. Unfortunately, this system lacks in terms of the mechanical properties when compared to some of the systems previously considered. Each of the TCP muscles can supply around 0.8 N of force at most, and the Novel Soft Actuator structure can only reach a maximum of 25 degrees bending angle. In terms of speed, the Novel Soft Actuator can move to an angle of 20 degrees in around 1 second. There are also other limitations to the technology. As heat is the triggering effect causing the fibres to contract, there can be some degradation in performance in higher temperature environments, and, although the use of antagonistic pairs of muscles can help, the fibres must be cooled to return to their original positions.

2.3.2 LIMS2 AMBIDEX - Quaternion Mechanism

The next mechanism we study is another anthropomorphic design, replacing the wrist joint of a human as opposed to the spine. This mechanism is named the Quaternion mechanism by its creators, and uses a three-bar parallel linkage, in a **3RRRR** configuration to generate a two degree-of-freedom mechanism. The

mechanism simulates the rolling contact of two hemispheres of equal radius [83]. This means there is always a path through the centre of the linkage of constant length, irrespective of the pose of the mechanism. This gives the option, which LIMS2 uses, of creating a third degree-of-freedom by passing a mechanical linkage through this passage, adding a roll axis to the wrist.

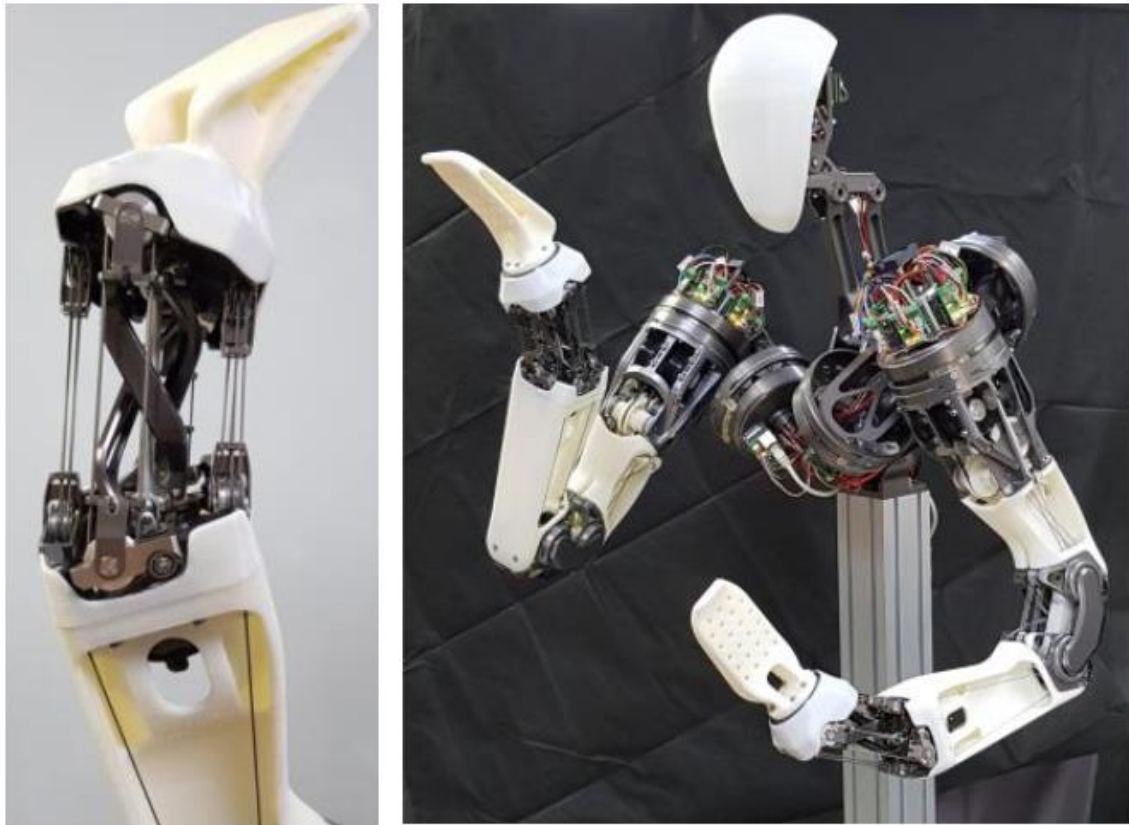


Figure 2.26. Quaternion Wrist Mechanism and LIMS2-AMBIDEX [83]

The Quaternion is however a passive mechanism, reliant on external forces to actuate it, hence no underlined component of the structure description. In the case of LIMS2, a set of two tensile wire pairs is used, in combination with a set of pulleys on each side of the mechanism. These wires are then fed back through the rest of the robotic structure to the central body, where the motors are. In the case of LIMS2 this helps remove the high-density components such as motors and gears away from the extremities of the robot, improving the stiffness of the whole arm mechanism.

This mechanism has excellent mechanical properties in comparison to many of the other mechanisms. It can output a peak torque of 34.7 Nm and reach an angle

of 90 degrees across all bending planes. Most impressively is the rotational speed of a claimed 1179 degrees per second, meaning it can perform a full swing from +90 degrees to -90 degrees in little over 150 milliseconds. The structure itself has a mass of 446g and, while its size is not mentioned, the crucial geometric values of the mechanism axes are given and would suggest the mechanism is a little over 30 mm in diameter, and around 100 mm in length. Note, this is without the pulley and motors, however.

2.3.3 Hydraulic Wrist Mechanism for Hand Prosthesis

This mechanism again is designed to mimic the human wrist, though in this instance as a wearable prosthesis [84]. This gives a new set of design parameters which a conventional robotic mechanism would not need to obey. For example, the aim of this project was to create a device capable of replacing a human wrist exactly, meaning the target rotational ranges are comparatively low, and differ depending on the axis.

Nevertheless, we examine the mechanism, as it is a small yet powerful device, and is powered by a system not yet considered: Hydraulics. This device uses a pair of small veins (visible in Figure 2.27) to actuate the position of the mechanism.

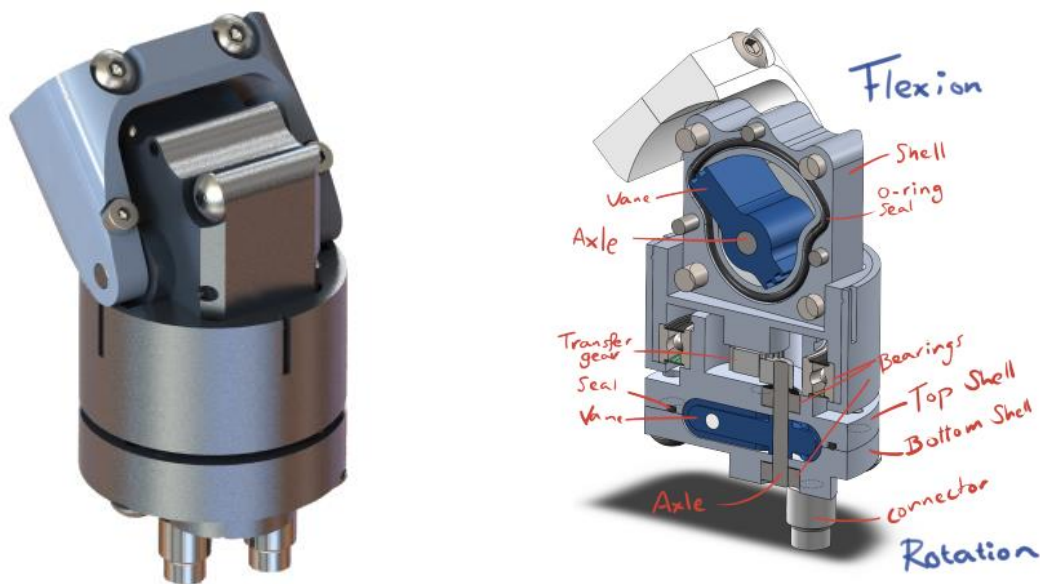


Figure 2.27. Hydraulic Wrist Prosthesis Renders [84]

Similar to LIMS2, the use of hydraulics allows the mechanism to displace the heavy mechanical components (a compressor for example) away from the active mechanism, thus reduce the mass, and size of the mechanism significantly. In this instance the final mechanism, while only theorised, was estimated to have a mass of only 72.8 g, and able to move through the full range of ± 45 degrees for both the rotation and flexion (pitch) directions. Unfortunately for the author, discrepancies in the manufacture caused the compressed fluid to leak, thus no meaningful torque measurement can be found.

2.3.4 Alternate Mechanisms Conclusions

Here we have considered three very distinct mechanisms, albeit from similar areas of robotics. Each has their own benefits, and while some of the designs are more complex than others, there is a clear front-runner in terms of performance metrics.

The Novel Soft Actuator is a fascinating design, and a strange combination of robotics and pseudo-bionics. It is easy to see how this technology could one day evolve into a more accurate bio-robotic vertebrae for powering a snake-like robot. Unfortunately, in its current state it simply does not have enough benefits mechanically, lacking in speed, range, and torque. While the range could be compensated for by using multiple short, smaller range joints, as a natural spine does, the torque output and general complexity to manufacture limit this as an option for this project.

The hydraulic wrist suffers similarly. While impressive in its compact size and low mass, the complexity of manufacturing such a small device for use with pressurised fluid would require a very high level of precision during manufacture. Secondly, we have the issue of locating the pressurising system within the device. For a prosthetic, a small pack could easily be worn on the user to supply the pressurisation, but for a small snake-like robot, this would be more difficult. While there are examples of this, for example the OmniTread series, which do use internal air compressors, using a remote compressor with hydraulic tubes tethered to the snake would be a possibility. However, this would significantly

increase the mass the robot has to drag behind it, and leaves the robot paralysed should one of the tubes become tangled or damaged while exploring.

Finally, the Quaternion mechanism used by LIMS2. While the mechanical properties of this mechanism are impressive, its complexity in design may be a factor in selection for a new snake-like robot joint. As discussed above, this mechanism replicates the rolling contact between two hemispheres with a constant radius. This is not entirely true, as the radius of the mechanism does change depending on the joint angle. However, careful selection of geometry using the equations derived for the design can result in the range of radius being negligible over the usable range. Similarly, to the hydraulic mechanism, the quaternion mechanism uses a power source which is remote to the mechanism itself. In contrast to the hydraulic system though, the required motors and drive train could easily be appended to the mechanism, if ones which are suitably small are used.

2.4 Summary

In this chapter we have presented a variety of different mechanisms which could be used to design a new snake-like robot. While the vast majority of existing snake- and serpentine-robots utilise a standard electric motor, many HRMs utilise novel mechanisms to achieve greater performance. We have analysed the different configurations of these robots, in terms of the arrangement of the degrees of freedom for the mechanism. While the differences are minimal, and the sample size small, it could be concluded that the arrangement of constructing a bespoke device, with multiple degrees of freedom, does result in a shorter, lighter, mechanism. However, the working range of the mechanism will most likely be lower, and the mechanism will most likely be wider, possibly restricting the use-cases for the robot.

From this learning we can better inform our future decisions when it comes to development of a new mechanism. For example, if module length is of a concern, then utilising a multi-DoF module would be more advisable. Conversely if there is a fear the module will be too wide, then an array of single-DoF modules should be used. For each type of structure, we have also shown numerous examples.

Typically, the single-DoF modules will use COTS motors, though there are examples, such as Uncle Sam and the SEA snake, which use bespoke designs. For the multi-DoF modules the opposite is true. The majority utilise custom designed mechanisms, although a few still rely on COTS motors and actuators.

We have also investigated some alternate mechanisms which could be used. These are actuated using methods which are not typically seen in SLRs or HRMs. The TCP soft actuator system imitates the spinal column and muscular structure found in biological creatures. Despite the small operational range of the mechanism, a number of these joints in series could form a working snake-like robot which is much closer to the motion of actual snakes. However, the technology is still immature, the slow response time, plus the possible interference caused by moving through high-temperature environments mean it is currently not a viable solution.

The final two mechanisms examined both replace a human wrist with a two-DoF structure. Both mechanisms also aim to reduce the mass of the mechanism by moving the actuating power source away from the moving components, one with the use of hydraulics, and one with a cable driven mechanism. The hydraulic wrist unfortunately shows the weakness of the technology, as a small manufacturing defect caused the mechanism to fail. However, the “Quaternion” mechanism from the LIMS2-AMBIDEX robot is very impressive. It has a relatively small size, and excellent physical characteristics.

3 SASAR MECHANISM DEVELOPMENT

3.1 Introduction

This chapter develops the required theoretical work related to the design of a snake-like robot. First, we perform design analysis and calculations to estimate the minimum required torque and maximum size for the new prototype robot. This information is then used to produce SASAR 1, our first prototype snake-like robot. We examine the design of SASAR 1, critiquing its weaknesses and formulating a list of checks which can be used to improve future designs. This list is used alongside the previously calculated physical requirements to create a concept for the next SASAR prototype. This includes deciding upon if the robot will be a snake- or serpentine- robot, and which of the mechanisms previously researched should be used. We ultimately decide upon a serpentine robot utilising a bespoke Quaternion mechanism. We then discuss the theory relating to the design of a suitable Quaternion mechanism, analysing the geometry of the joint and estimating how the required torque of the joint translates to the motor powering the joint.

3.2 Initial Design Calculations

3.2.1 Module Size

Using the initial design specification, several design targets can be calculated. Firstly, we look at the module size, and create a relationship between the desired length (L), diameter (d) and joint angle (ϕ) range of the modules by considering the tube width (W) and turn radius (R_M). In Figure 3.1 we consider a three-module robot which is at its maximum possible size to fit through a pipe, diameter W , turning through a radius R_M .

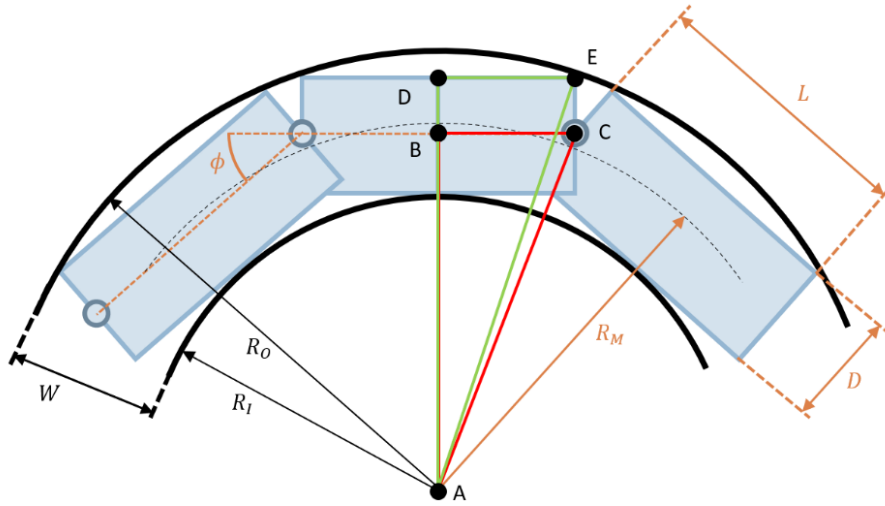


Figure 3.1. Module size and Turn Radius Calculation

The two triangles, ABC and ADE can be created, and the following Pythagorean and trigonometric relationships made. Note that Angle $B\hat{A}C$ is equal to $\frac{\phi}{2}$.

$$R_I + \frac{d}{2} = \frac{L}{2} \tan\left(\frac{\phi}{2}\right) \quad (3-1)$$

$$R_O^2 = \left(R_I + \frac{d}{2}\right)^2 + \left(\frac{L}{2}\right)^2 \quad (3-2)$$

By substituting R_I and R_O for $R_M \pm \frac{W}{2}$, and solving these two equations, an estimate for d can be calculated as a function of ϕ , R_M and W .

$$d = \frac{1}{\tan^2\left(\frac{\phi}{2}\right) + 4} \left[\pm \sqrt{2} \left(4R_M^2 \sec^2\left(\frac{\phi}{2}\right) + 4R_M^2 \cos(\phi) \sec^2\left(\frac{\phi}{2}\right) \right. \right. \\ + 6R_M W \sec^2\left(\frac{\phi}{2}\right) + 2R_M W \cos(\phi) \sec^2\left(\frac{\phi}{2}\right) \\ + W^2 \sec^2\left(\frac{\phi}{2}\right) + W^2 \cos(\phi) \sec^2\left(\frac{\phi}{2}\right) \left. \right)^{\frac{1}{2}} - 2R_M \tan^2\left(\frac{\phi}{2}\right) \\ \left. - 4R_M + W \tan^2\left(\frac{\phi}{2}\right) + 2W \right] \quad (3-3)$$

Thus L can be equally estimated by inputting the calculated d estimate into either equation (3-1) or (3-2).

As a general target, we wish to maximise the total available volume of the module, and thus the area estimated here. A larger volume gives more space to include the required components, making the mechanical design task easier. One other aspect to consider is the aspect ratio of the module. The three examples shown in Figure 3.2 exhibit how the aspect ratio of the mechanism may affect the design of the joint module.

As the module length increases, the diameter of the module must decrease, and the required joint angle range increases. The most suitable aspect ratio will be dependent on the design of the mechanism.

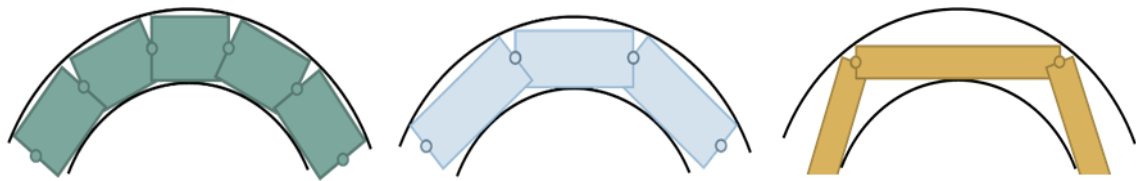


Figure 3.2. Aspect Ratio Examples

With this knowledge we make three calculations to predict what the most efficient use of space will be for the three desired pipe diameters considered in the initial design specification: 40, 60 and 80 mm. Figure 3.3 displays the results of calculating area over a range of values for ϕ . We estimate the maximum area a module could be, to fit through an 40mm diameter pipe, would be a length of 97.6mm and a diameter of 27.3mm. This module would also have to be able to reach a joint angle of at least 67 degrees. For the 60mm pipe we estimate 122.7mm length by 41.3mm diameter, joint angle of 82 degrees, and for the 80mm pipe a length of 145.2 by 55.5 diameter but having to reach a joint angle of 93.9 degrees. This is due to the high curvature of the large pipe diameter, meaning the joint will fold back on itself. If we limit this to 90-degree angle, we get a length of 138.1, and diameter of 58.2.

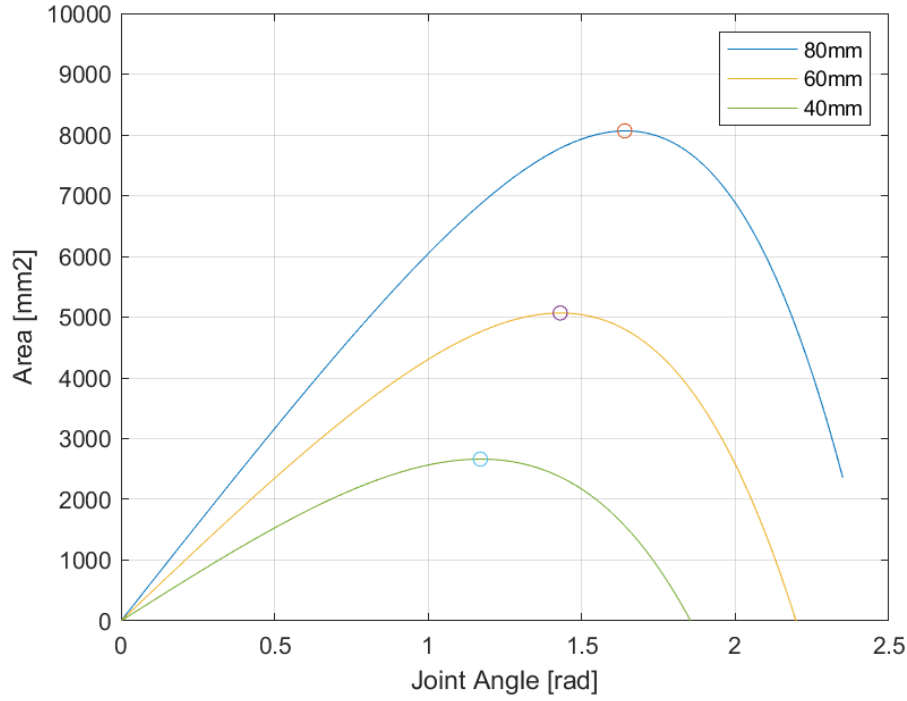


Figure 3.3. Module Area Estimates

3.2.2 Torque Requirement

A basic torque estimate can be found by treating the robot as a simple cantilever beam. We assume the robot is fixed at one end and pivots about a mechanical pin joint, with a number of modules extending past this joint (Figure 3.4). The free-body diagram of this can be modelled as a cantilever beam.

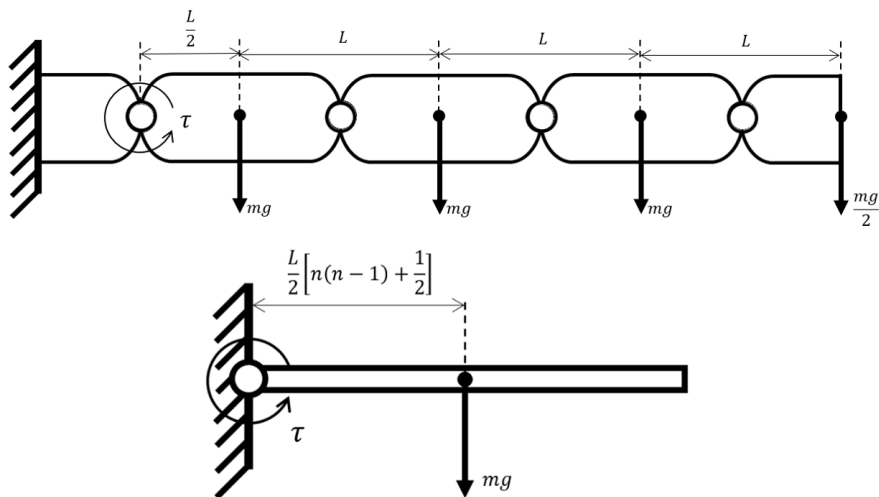


Figure 3.4. Motor Torque Free-body Diagram

The estimated torque, τ , is a factor of the number of modules, n , the module mass, m , the module length, L , and the gravitational acceleration constant, g . As first seen in [85], the general formula can be shown as:

$$\tau = \frac{mgL}{2} \left[n(n-1) + \frac{1}{2} \right] \quad (3-4)$$

We look to calculate estimates for our three possible pipe sizes found above, however, a value for the mass of the module is required. We offer two assumptions, one for a module mostly consisting of 3D printed components, similar to Mamba [18], and one for a module consisting of mostly metallic components, similar to Woodstock [85]. We use these examples as they are similar in size to the three example sizes above. As these two robots are both of known size and mass, we can calculate a module density. This is used to estimate the mass of our three new modules. We assume our new modules are solid cylinders for the purpose of this calculation. Mamba was estimated to have a module density of 7.11×10^{-4} , and Woodstock of 1.35×10^{-3} (both in $\frac{g}{mm^3}$).

For each case we assume the module will need to support at least three more modules to allow the robot to step over a height of 80 mm ($n = 3$). If a shorter module is designed, this value may need to be increased. We also include an additional payload of 50 grams within each module. Table 3-1 displays the estimated module masses and therefore required torques for each of the three modules sizes, for both the polymer and metallic construction modules.

Table 3-1. Module Torque Estimations

Material	Module Estimate	Diameter (mm)	Length (mm)	Module Mass (g)	Required Torque (Nm)
Polymer	40mm Pipe Estimate	27.3	97.6	90.61	0.28
	60mm Pipe Estimate	41.3	122.7	166.85	0.65
	80mm Pipe Estimate	58.2	138.1	311.16	1.37
Metallic	40mm Pipe Estimate	27.3	97.6	127.33	0.40
	60mm Pipe Estimate	41.3	122.7	272.51	1.07
	80mm Pipe Estimate	58.2	138.1	547.32	2.41

3.2.3 Design Requirements Discussion

From the calculations completed here, we created estimations for the size, and thus torque, for the three possible pipe options, depending on the material the module is eventually manufactured from. It should be noted, these are not intended as precise targets to which a robot must conform, only a guide for future design and development. As such, the targets determined in this section should be revisited throughout the development of a snake-like robot, once more information is known.

For example, if we assume we can design a module with a diameter of 35 mm, we can use the above information to calculate targets for module length, joint angle and, eventually, torque. Figure 3.5 shows how the diameter and length of a module designed for a 44 mm width pipe changes with the required joint angle. At the point where the diameter is 35 mm, the corresponding length and joint angle are 63mm and 44 degrees (0.77 rad) respectively. Using this information, we can discern that a module of diameter 35 mm should be at most 63 mm in length, and able to rotate at least 44 degrees.

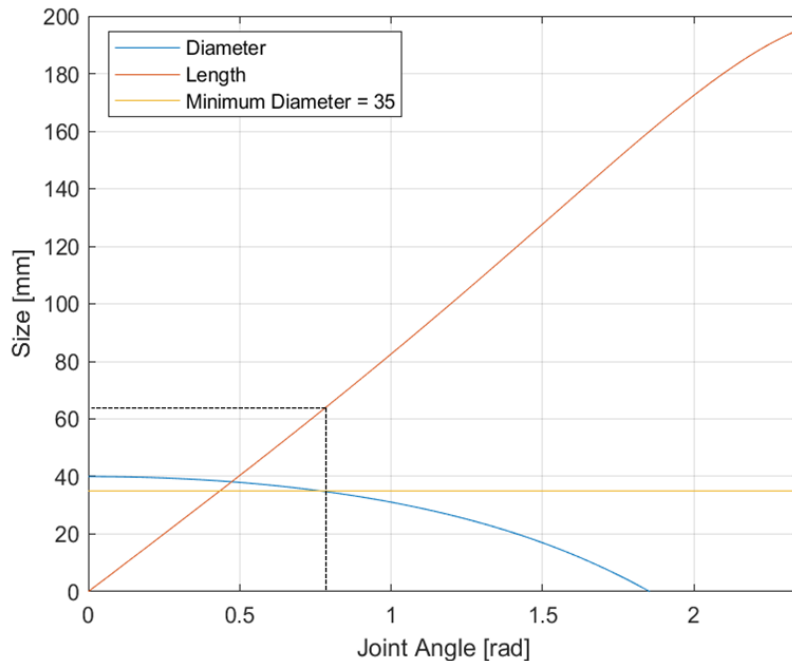


Figure 3.5. Module Size Estimations for $W = 35 \text{ mm}$

This information could then be used to help predict the required torque. In the example above, a 35 mm diameter 63 mm long module, when lifting four other modules, would require 0.34 Nm of torque if the modules were plastic, and 0.51 Nm if they were manufactured from metal. These calculations can be used to inform decisions when selecting components and then refined further as the design process continues.

3.3 SASAR Initial Prototype

3.3.1 Overview

In this sub-section we present the prototype snake robot SASAR 1. This robot is a simple construction of mostly COTS components with a 3D printed structure. It is created with a view to examine the feasibility of designing and manufacturing a small, modular, snake-like robot using inexpensive methods. We then use the experience gained from creating SASAR 1 to inform design decisions for future prototypes. The overall design of SASAR 1 consists of a serialised chain of four modules, termed “Joint Modules”. These modules each provide two degrees of freedom, controlling the pitch and yaw angle of the module. This gives a total of eight DoFs across the entire robot. Finally, two large, rubberised feet are attached to the front and rear of the robot (Figure 3.6). As described later below, these provide friction to the robot, and help with stabilising the robot.



Figure 3.6. SASAR 1

3.3.2 Mechanical Design

For component selection we begin by utilising Table 3-1 above, and targeting a mostly polymer-made robot with the ability to fit through a 40mm pipe. We therefore select the Savox SH-0256 servomotors. These motors have an output torque of 0.45 Nm, with a height of 29.4 mm. While this is slightly larger than desired, Figure 3.5 shows a maximum diameter of 35 mm would be acceptable, providing the length is reduced to 63 mm.

The mechanical 3D printed structure takes inspiration from the Modular Snake robot by Wright et al [86]. This simple design consists of two main components, termed the “U-Case” and “Servo Back” by Wright, although we term them the “Hinge bracket” and “Servo Casing” respectively. Functionally, these components achieve the same purpose, although the arrangement for SASAR 1 is altered. The Servo Casing replaces the bottom section of a standard servomotor housing, giving further functionality to the motor. Primarily this component contains an additional pivot point, axially colinear to the servomotor output shaft. This helps to improve the stability of the mechanism. It also includes a perpendicular bracket, acting as the mechanical interface between modules. The Hinge Bracket is again similar in function to the U-bracket, again with an additional purpose. This bracket interfaces with the servomotor output shaft and the additional pivot point to create the 1-DoF mechanism. Figure 3.7 displays the designs of the two modules for comparison, modelled using computer aided design (CAD).

There are a number of differences between the Modular Snake robot and SASAR 1, with the primary distinction being the number of degrees of freedom per module. Each SASAR 1 Joint Module is an assembly of two of the single-DoF mechanisms described above. By doing this we can halve the number of microcontrollers needed to control each module, as well as ensuring the mechanical and electrical interface between two modules is consistent. Additionally, where the Modular Snake robot uses a specialised ball-bearing assembly to support the underside of the U-bracket, the tight space constraints of SASAR 1 mean only a simple pin and bushing are used, with a circlip to secure the mechanism.

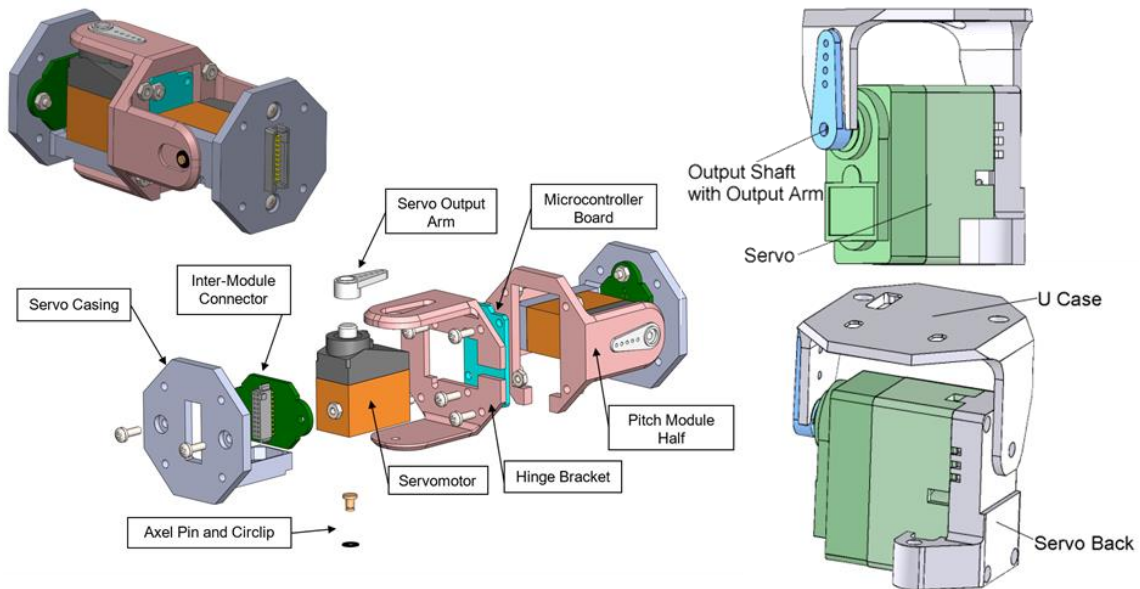


Figure 3.7. SASAR 1 Module CAD Model (left) and CAD model of Modular Snake Robot module [86] (Right)

3.3.3 Electrical Architecture

Electrically there are also differences between the two systems. To reduce the cost, and keep the majority of components COTS, there are only two simple bespoke printed circuit boards (PCBs) within a SASAR 1 Joint module. These are the two modular connector boards which are used to support a 20-pin Samtec connector. The PCB contains tabs to which a wire loom can be soldered. Whereas the Modular Snake robot has a completely bespoke PCB design to replace the existing control board within the servomotor.

The electrical architecture of SASAR 1 is uncomplicated. Four tethering wires are connected to the tail end of the robot. These are the 5-volt Direct Current (DC) power, ground and two serial data lines (Receive and Transmit). All four are output from a laptop computer using a CP2102 USB to UART adapter. The wires then connect directly to the microcontroller board in each module, via the connector interfaces. Each microcontroller then passes the power, ground, and

a pulse width modulation (PWM) signal to its two servomotors. A capacitor is used on the front of the snake to reduce the noise in the serial bus.

3.3.4 SASAR 1 Control and Motion

Internally, each module runs a simple code structure. The microcontroller first connects to the serial bus and centres the motors to their zero position. It then pauses, listening to the serial bus until a predefined module identification (ID) number is called at the start of a message. The microcontroller then records the information following the ID, which consists of the two new angles to move each of its motors to, plus a relative speed value. By interpolating between the current position, and the new required position, the modules move in a smooth motion. The speed value determines how quickly this is achieved by defining the number of intermediate steps. The servomotors themselves have a built-in positional feedback system, thus no additional controller is required.

In motion, the robot crawls much like an inchworm, rather than a snake. This is done in four stages. Firstly, the robot lifts the rear foot and ensures the front foot is in contact with the ground. It then contracts the rear by lifting it towards the front of the robot, before lowering the rear foot. The front foot is then lifted and extended before the cycle is repeated. This was achieved by manually walking the robot through the desired motion and recording the joint angles at each stage.

3.3.5 SASAR 1 Findings and Analysis

As discussed above, the main aim of creating SASAR 1 was to discover possible issues which may arise when building a snake-like robot, and research the overall feasibility of creating a small, modular, mostly COTS robot. In this section we discuss the final performance of SASAR 1 and detail the issues which arose with a view to creating a list of checks which can be used to ensure greater success with any future prototypes.

SASAR 1 Performance

Table 3-2 summarises the physical properties of the SASAR 1 Joint Module. The module has an octagonal profile with a width of 35 mm. Unfortunately, we were

unable to meet the target of a 63 mm total length, with the module being 75 mm. However, the mass is only 50.4 grams in comparison to the estimated 90 grams.

Table 3-2. SASAR 1 Joint Module Properties

Property	Value
Degrees of Freedom	2 Per Module, [Pitch, Yaw]
Module Size (H x W x L)	35 mm x 35 mm x 75 mm
Module Mass	50.4 grams
Output Torque*	0.45 Nm
Operational Range*	±45 degrees
Rotational Speed*	375 Degrees per Second (6.54 rad/s)

**Attributes are taken directly from the Savox Servomotor Datasheet [87]*

Additionally, the torque and operational range of the modules is worse than expected. SASAR 1 was found to be unable to lift the remaining three modules when one end was clamped. Using equation (3-4), and the values presented in Table 3-2 we can estimate a required torque of 0.23 Nm. Therefore, the servomotors must be underperforming.

The control and communication systems allow for each module, and each degree-of-freedom within each module, to be independently positioned. Furthermore, as the modules use a standard serial communications bus, the central controller can be replaced with something more suitable if required.

The inchworm-style motion is a result of an unsuccessful attempt to implement a serpentine motion into the robot. While it was possible to propagate a simple sine wave pattern down the snake (see [88]) the motion was unsuitable and provided no forward motion. Thus, the front and rear feet were added, and the inchworm gait was implemented.

SASAR 1 Analysis

There were several issues which caused the length to be increased. Firstly, while the servomotors are only specified as 22.8 mm in length there must be additional clearance for the hinge bracket to pass without clashing with the corner of the motor. The placement of the microcontroller board also increased the overall

length of the module. Finally, the thickness of the components such as the servo casing rear bracket had to be a specified thickness to work with the selected connectors. There were also issues with attempts to reduce the geometry of the two 3D printed parts, causing the components to become too weak, particularly in the plane the component is printed in. Figure 3.8 shows an example of this type of layer failure.



Figure 3.8. Prototype SASAR 1 Hinge Bracket Showing Layer Failure

To improve upon SASAR 1, a number of design goals and checks are put in place. Firstly, for the 3D printed structure several aspects should be targeted. To reduce the possible failure modes due to out-of-plane forces and layer separation, parts should be designed with FDM manufacture in mind. This means reducing overhangs and ensuring load-bearing parts are printed in the strongest plane. Alternative additive manufacturing techniques may also be explored, along with methods for improving the layer adhesion of 3D printed FDM components.

With regards to the poor motor performance, this was found to be an electrical issue. The cause being a combination of the control laptop USB port being unable to provide sufficient current, and the poor quality of wiring causing interference and a large voltage drop. The voltage drop also causes the control system within the servomotors to become unstable, as the potentiometer suffers from an unexpected low voltage. This restricted the maximum range each motor can be turned to and the motor would oscillate randomly towards the extreme positions.

To solve this issue, it should be ensured that all electrical interfaces are rated for the required power, plus the power supply unit (PSU) should be of suitable power, as the USB output from a laptop will not suffice. Additionally, including some form of power storage inside the modules will help prevent unintended shutdowns of the microprocessor. This could be in the form of a small independent battery or capacitor array.

While the serial communications bus was satisfactory, there was clear degradation in the signal over the length of the robot. This is typical for the serial bus, which is not intended for use of long distances. While it did not cause any issues for SASAR 1, alternate communication busses and standards should be investigated for future prototypes.

The reasons for being unable to recreate serpentine motion are easy to understand when SASAR 1 is compared with any of the snake robots presented in Chapter 2. The most evident is the total number of degrees of freedom. SASAR 1 has a total of eight DoF, coming from four- two degree-of-freedom modules. It is a snake robot and has no other means of forward motion other than the body itself, classifying it as a non-wheeled robot. As shown in Table 3-3, non-wheeled robots will typically have between 10 and 20 total degrees of freedom, and an average of 16 when rounded to the nearest whole unit.

Table 3-3. Total Number of Degrees of Freedom for Non-Wheeled, multi-plane Snake Robots.

Robot	Total DoFs
ACM-R7	20
Kulko	20
Mamba	12
SEA Snake	16
Serpens	10
Uncle Sam	16
<i>Average</i>	<i>15.67</i>

For future SASAR prototypes, we suggest looking to vastly increase the number of modules or change to a different style of motion. For example, the average number of DoF for an actively driven robot is only 10. This would be the same as adding only a single extra module to SASAR 1.

The fatiguing of the electrical wires is the most common reason for failures in SASAR 1. As each of the motors turn, the electrical wires become strained, then compressed as the motor returns to its neutral position. This causes fatigue on both the wire and any soldered joints. As it is likely electrical connection will need to pass through a joint mechanism appropriate hardware should be used, such as flexible PCBs or wired connectors with suitable usage rating. Reducing the number of locations per module where the electrical connections cross a movement axis would be preferable, for example using a multi-DoF module as opposed to a single, or combination of singles as used in SASAR 1.

To summarise, we have established four areas in which the design of a future SASAR prototype, or another snake-like robot could be improved. Firstly, structural components which are 3D printed should be specifically designed to avoid the disadvantages of additive manufacturing, and other techniques should be investigated if this is not possible. Secondly, the electrical connections and power supplies should be rated to deal with the power consumption of the whole robot, and any locations in which electrical connections cross the motion axis of the robot should be reinforced and use correctly rated connectors. We also suggest the use of a different communications bus, specifically suited to longer distances. Finally, we suggest the use-case of the robot is closely examined, and the desired drive-style, and thus total number of required modules chosen based upon this examination.

3.4 SASAR Design Configuration Choices

3.4.1 New Design Criteria

Before commencing the design of the next SASAR prototype, we must decide upon the configuration of the robot. Using the experience gained from the design of SASAR 1, in combination with the research performed above, a number of new

design criteria are created. The aim of these criteria is to ensure the new prototype snake robot will be successful, both in its final function, and the process of creating it.

The original size target of a 40mm tube will be overlooked, in favour of ensuring a working mechanism can be produced.

Many of the issues with SASAR 1 are related to the limited diameter. This restriction forced the use of the Savox servomotors, as no other suitable motors with the required torque and size could be sourced. By lifting this constraint more components become available for selection, thus more varieties of design can be considered. Additionally, the time and cost of manufacture will most likely be reduced, as the exceptionally small scale would require a complex design, which would be costly to produce. As a compromise, scalability analysis will be performed upon any new mechanism design, to ensure the design could feasibly work in the smaller pipe, if necessary.

The new joint module should be a multi-DoF mechanism, with a minimum two degrees of freedom.

The increase in the width allowance means the length of the module will be the next constraining factor. As has been shown, a multi-DoF module will typically be shorter in length than the combination of multiple single-DoF modules. If possible, the design should have rotational axes which overlap at a single point. This would reduce the strain placed upon the connecting wires within the modules, one of the main causes of failures in SASAR 1. A multi-DoF module may typically reduce the available range of the robot, but this is unlikely to be a limiting factor.

The robot will be an actively driven serpentine robot, with the capability to become a snake robot if possible.

Of the serpentine gaits discussed in Chapter 1, only one or two will be possible for a snake robot to use when crawling through pipes. These gaits would be concertina motion, and rectilinear crawling. A form using both of these could be possible in a snake-like robot, but due to the relative length of each actuating section in comparison to a natural snake, it would be difficult to achieve within the

confines of a small pipe. Additionally, rectilinear crawling is the slowest and most energy inefficient snake motion gait. It also relies on very small movements within the structure of the snake (see [35] for further detail). Therefore, the decision has been made to transition the robot to have actively driven wheels. These wheels should be able to be removed, or replaced with a passive version, thus a modular structure of “Joint” and “Drive” modules will be used.

The robot will be remotely operated use a central controller, to which the robot is tethered to.

This was the case with SASAR 1, which was controlled and powered using a laptop computer. Even with the strict size constraint lifted, an appropriately powerful microprocessor will not be able to fit within the unit. Another issue is the cooling of the processor. Modern SoCs (Systems on Chip) such as the Broadcom BCM2711 used in a Raspberry Pi 4 require a power source of at least 15 Watts. This energy must be dissipated somehow, and snakes are very poor at thermal control, due to their body volume to surface area ratio [89]. While wireless communications would be possible, the high-frequency radio waves used would be unable to communicate within tight metallic structures, such as the pipes SASAR is intended for.

Therefore, any future prototypes should be remotely controlled via a tethering cable. Tethering in this way also removes the need for a large onboard battery, as the bulk of electrical energy can be supplied using the tether. Finally, the tether could be used to anchor the robot, and recover it, if necessary.

3.4.2 Modular Architecture

We must now consider the modularity of the robot, as desired by the project aims. With the transition to an actively driven serpentine robot, the mechanism which provides the forward motion should be detachable from the mechanisms which provided the joint actuation. We must also consider what sensors are required for operating the robot, and how the robot will connect to the central controller. Thus, four basic module concepts are created.

Joint Module. Similar to SASAR 1 this will be the basis of the robot mechanism, and the crucial component in actuating the motion of the body of the robot. It is in effect the spine of the snake. It must have at least two degrees of freedom and able to rotate horizontally (yaw) and vertically (pitch). These degrees of freedom must both be accurately controllable, suitably powerful, and appropriately responsive.

Drive Module. If the Joint Module represents the spine, then the Drive Module represents the legs of the serpentine robot. This module must have at least one degree of freedom, although it could have more if independent control of each wheel is required.

Head Module. The head module is the only module which is not crucial to the basic operation of the robot; however, using the robot for its intended operations would depend significantly upon the Head Module. In its simplest form a head module should contain a camera, which can be viewed by the operator to aid with the control of the snake. To aid with autonomy additional sensors may be required. The head module should be designed such that it can be easily interchanged with replacements featuring additional or different sensors, as the mission requires.

Tail Module. The tail module has a simple task of converting the incoming tether from the central controller to the modular interface used by the remainder of the robot. While mechanically and electrically simple, the module should be designed with additional strength or locking features for the tethering cable, as losing the tether mid-operation would be disastrous.

Each of these concepts are extremely basic and cover only the fundamental features which will be necessary. The key benefit of modular design, when implemented correctly, is the rapidity in which new modules can be created and integrated into the existing structure. The example of swapping Head Modules to included different sensors is understandable, but other less initially obvious changes and improvements could be made. For example, replacing the Tail Module with a new design featuring wireless communications and a battery. If additional battery life is required, a new module could be designed to be inserted

between any two modules which contains additional cells. The drive wheels could be replaced with fins and propellers if the robot was made to be amphibious, or replaced with a module with larger wheels, or even tracks for moving over extreme terrain.

3.5 Joint Module Concepts

From these new criteria, the original design specification plus the research into existing joint mechanisms, three new joint concepts are proposed. These are a Pitch-Roll mechanism, a Quaternion Style mechanism, plus a Spherical joint. Each new joint is compared to the others, plus a Pitch-Yaw joint, similar to the one used in SASAR 1. For each of the four designs, a 3D printed prototype was created. Each prototype was of similar diameter and length.

We only look to compare certain aspects of the design, not their overall performance. Therefore, physical properties, such as torque and rotational speed are not necessarily considered, as these are highly dependent on the chosen motors. We do consider the relationship between the motor and the overall movement of the joint. The other main aspects we consider are the ease of design, the adaptability of the structure and the manufacturing processes required.

3.5.1 Pitch-Yaw

As mentioned above, the pitch-yaw design is similar to the original SASAR 1 joint module design, although with a number of small improvements. Most of these improvements prioritise the aim of reducing the number of failures compared to SASAR 1, rather than improving the overall structure of the robot. One structural change is to reverse the position of the pitch motor at the front of the joint. This results in the connection points at the rear of the servomotor being closer to the back of the joint, the new location of the control board. This means the two sub-sections of the module are also identical, making manufacturing much simpler (Figure 3.9).

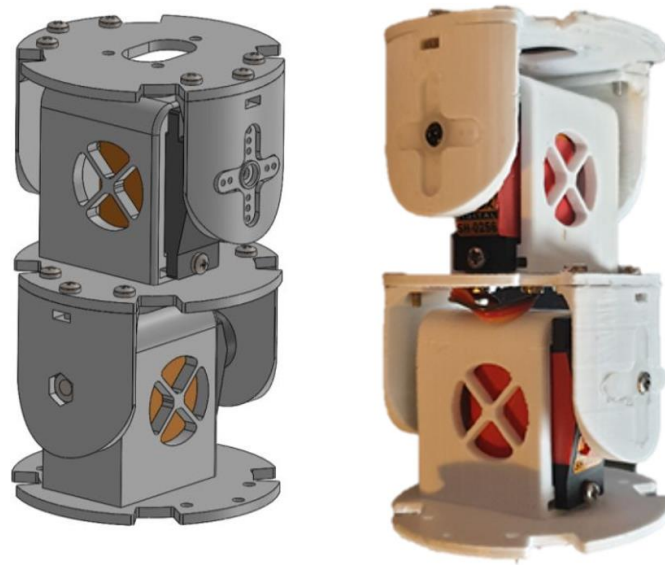


Figure 3.9. Pitch-Yaw Concept CAD Model and Prototype

The rescinded size restraint allows for a larger servo-casing to be introduced, which includes a small ball-bearing to act as the secondary pivot point. This also means the large X-shaped servomotor arm can be used, creating a better interface between the motor and the hinge bracket. Finally, the hinge bracket design has been reworked, and separated into three distinct components: the attachment plate plus two arms, one for the servomotor and one for the bearing. Similarly, the servo casing is removed from the vertical plate it would use. This allows all the components to be printed in the orientation which would provide the most strength. The components are then bolted together (Figure 3.10).

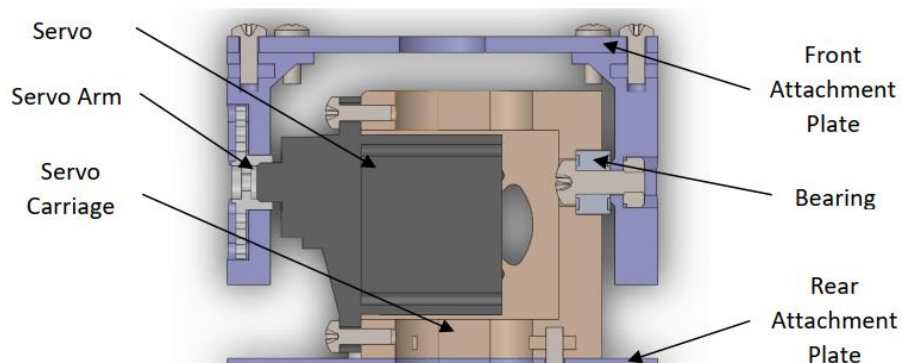


Figure 3.10. Cross Section CAD Model of New Pitch-Yaw Module sub-section

While some improvements have been made, there are still some downsides which are a result of the structure. Firstly, the output of the joint is directly connected to the motor. While this is not initially a problem, it can become one if suitable motors cannot be sourced, as any additional gearing or other connection allows for the torque and speed of the device to be more tuned to the exact needs of the joint. We also have the same issue of using COTS servomotors as before. They typically have a limited range and torque, plus their on-board controllers may also not be suitable for our needs. Additionally, we have similar issues with the wiring path becoming stretched as the joint bends. If we leave too much slack in these wires, they become entangled within the mechanism, and if they are taut then the additional strain causes fatigue over time.

3.5.2 Pitch-Roll

This joint idea uses a typical 1-DoF mechanism with an additional degree of freedom allowing the axis of this 1-DoF joint to be positioned in any orientation. Based on the designs of the FZI robots, KAIRO and MAKRO [64,65,67–69], the axial rotation about the length of the snake lets the mechanism position itself with drive wheels in the best alignment. It also means the main actuation axes (the 1-DoF joint) can be adjusted to suit certain environments. For example, if the robot is only required to move along a flat plane, all of the axes could be rotated to become “yaw” joints. This results in twice as many degrees of freedom in this plane, in comparison to the Pitch-Yaw mechanism. This prototype uses the same “pitch” mechanism and servomotor from the Pitch-Yaw design. This is mounted on a plate set inside a large ball bearing with an internal gear, meshing with a continuous servomotor, which can rotate freely (Figure 3.11).

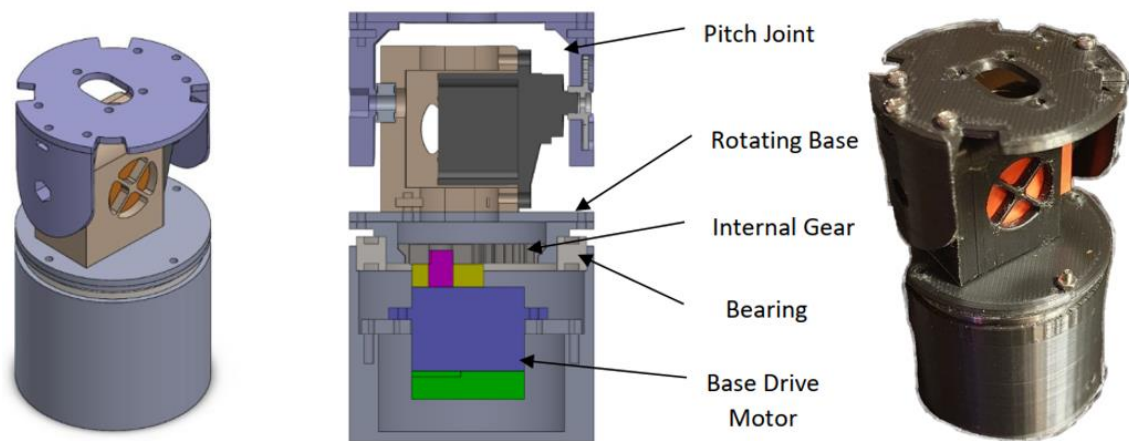


Figure 3.11. Pitch-Roll Joint Prototype and CAD Model showing X-Section

While this mechanism structure does offer gear advantages, in practice it may be hard to construct successfully. Firstly, the 1-DoF “Pitch” subsection will suffer the same disadvantages as the Pitch-Yaw module. Those being the continued issue with straining the free wires, the direct connection to the servomotor output and the use of servomotors. Although, as above, these last two points are void if suitable servomotors can be found.

The axial “Roll” rotation will also be complex to design, both mechanically and electrically. To be fully useable, it would require at least 360-degree rotation, with continuous rotation preferable, which can be fully position-controlled. It will also require a large holding torque, to ensure the mechanism does not slip out of position when performing complex motions. This could be achieved through the addition of a secondary braking system, such as in the snake robot Uncle Sam, but this again increases the complexity of design.

One issue with the structure could be the difficulty in passing a successful electrical connection through the “Roll” mechanism. The 3D printed prototype shown in Figure 3.11 had numerous problems with the control wires for the “Pitch” servomotor becoming tangled within the gears connecting the base drive motor to the rotating base.

Another issue is the relative strength of the Roll axis. The majority of the load on this joint will be perpendicular to its axis. This could result in the bending of the axis if the rotation mechanism is not suitably strong.

Finally, if we consider the robots this idea is based upon, the KAIRO and MAKRO series, we can see this concept has one fewer degree of freedom. The third degree of freedom in KAIRO and MAKRO ensures the drive wheels can always be in the ideal orientation. This result may be possible to achieve with the two-DoF module concept here, though it would rely on two of these modules working in tandem with a single drive module. This would result in each combination of drive and joint modules effectively becoming a four-DoF module.

3.5.3 Quaternion

The Quaternion mechanism presented here is a development of the wrist mechanism created for LIMS2-AMBIDEX [83]. It uses the same **3RRRR** structure, with a pair of tensile cables to drive the motion. The main difference in this design compared to LIMS2, is the relative positions of the motors which drive these cables. LIMS2 uses the cable driven nature of the design to remove the heavy motors away from the end effector, increasing the stiffness of the mechanism. For SASAR, we mount these motors at either end of the mechanism, with one motor controlling the vertical (Pitch) motion and one to control the horizontal (Yaw) motion. For the purpose of the prototype a pair of continuous SG90 servo motors are used. These motors can be velocity controlled, but not positionally. The motors are mounted perpendicularly to each other, with a large drum connected to the output of the SG90 motors. A single Nylon tensile wire is wound around the drum, with half wound clockwise into the upper section, and the remainder wound anti-clockwise in the lower section. The two ends are then fed through guides and passed into the Quaternion mechanism through openings in the base of the mechanism (the orange sections in Figure 3.12). The wires then run the length of the mechanism where they are fixed in place in the corresponding holes in the opposite mechanism base. Rotating the servomotors in either direction causes half the connected wire to be wound in, contracting the length between the mechanism bases, curving the joint. This arrangement also causes the same wire to be loosened on the opposite side. Thus, as long as both sides of the wire are properly tensed before actuation, the mechanism will remain stable.

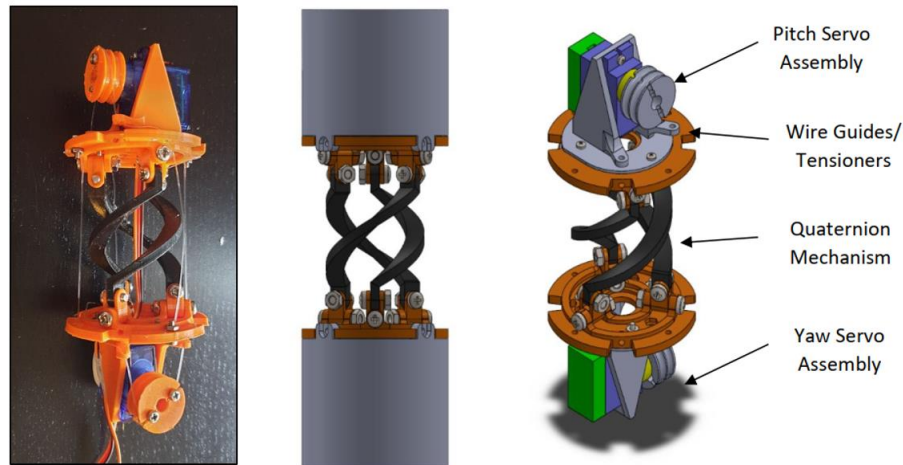


Figure 3.12. Quaternion Prototype and Model

This mechanism offers numerous ways to adjust its mechanical properties. For example, the diameter of the drum can be adjusted to compensate for a lack of torque, with a smaller drum requiring a less powerful motor, although the speed of the joint will suffer. Similarly, as is the case in LIMS2, the wires can be connected using a pulley system, again increasing the torque, and the stiffness of the mechanisms, at the cost of speed.

One of the crucial aspects of this mechanism is the geometry of the three linkages which make up the main structure. If an unsuitable set of geometry is chosen, the performance of the joint can be reduced. Additionally, incorrect geometry in the design and shape of the main limbs which make up the linkage can result in collisions between these limbs, preventing the mechanisms from reaching the extreme angles. However, this linkage design, in combination with the motion, means there is always a void of equal length running through the centre of the linkage. In LIMS2 this is used to pass a mechanical transmission shaft through, giving the wrist a third degree of freedom. For our needs, this void provides an excellent path for the connecting wires.

The unique motion of the Quaternion mechanism accurately approximates the rolling contact between two hemispheres. This means, unlike the four other designs there is not one singular point of rotational for each degree of freedom. This may affect the complexity of the control systems employed for controlling the robot.

3.5.4 Spherical

The final prototype is a spherical style mechanism. This joint uses a ball and socket structure, akin to the spinal column of a natural snake. We utilise the slightly flexible nature of the filament used in 3D printing to gain a tight, but smooth, connection at the ball-socket interface. This mechanism is powered in the same manner as the Quaternion mechanisms, using two continuous servomotors in combination with a drum and tensile-wire system to actuate the robot. Therefore, many of the positives of the Quaternion mechanism can be translated to this structure. The exception being there is no clear path for the electronic cables to pass through the module.

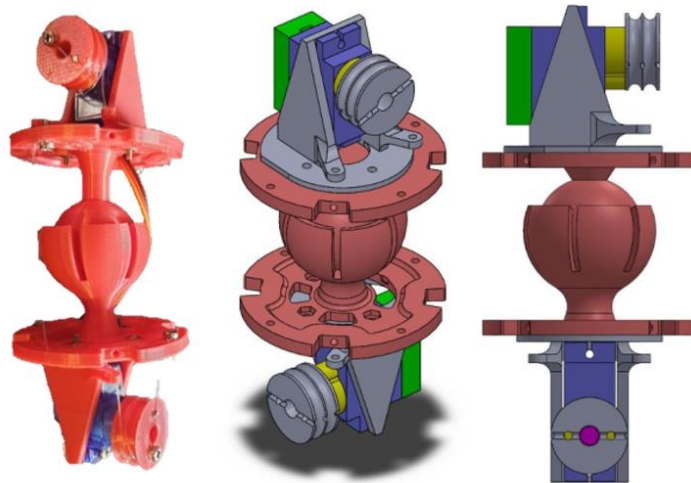


Figure 3.13. Spherical Joint Design Prototype and CAD Model

Unfortunately, this mechanism has numerous other deficiencies, the most major of which is the unconstrained additional degree of freedom. As shown in Figure 2.2, spherical type joints have three degrees-of-freedom, vertical, horizontal, and rotational (Pitch, Yaw, Roll). While the tensile cables constrain two of these, the rotational degree of freedom is unconstrained. This leads to the system becoming unstable, and unable to be accurately positioned as it rotates out of line with the cables. Secondly, in order to restrain the ball laterally, the socket must encompass the majority of it. This harshly restricts the maximum possible joint angle of the mechanism.

3.5.5 Motion Range Analysis

We additionally consider the relative motion ranges for each of these structures. As the final geometry could not yet be known, we instead consider the four prototypes to be equal in length, with any movement axes and component interfaces equally separated. Figure 3.14 displays a representation of the sizes of the models, with each being a unit length in its zero position.

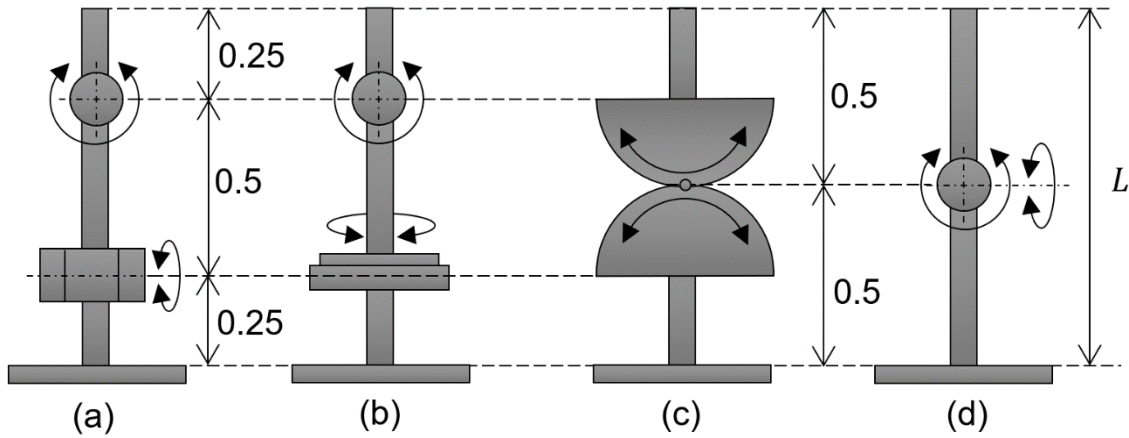


Figure 3.14. Mechanism Motion Analysis Models. (a) Pitch-Yaw, (b) Pitch-Roll, (c) Quaternion, (d) Spherical

We aim to estimate all the possible locations the end of the mechanisms can reach. This is done by compounding the required rotation and translation matrices, resulting in equations for estimating the Cartesian position of the end effector in terms of the joint angles (ϕ) and bending plane angle (β) if applicable. A number of limits are imposed upon the simulated mechanisms. We assume a maximum joint angle of $\frac{\pi}{2}$ for all but the Spherical joint, which is limited to $\frac{\pi}{3}$ due to the restriction of the socket on the mechanism. It is assumed the Pitch-Roll, Quaternion and Spherical joints can rotate a full 360-degrees.

The Quaternion mechanism has an additional variable. Rather than simulate the rolling contact, we simplify the structure as a perfect, unconstrained, spherical joint at a virtual pivot point equidistant from the base of the hemispheres. Figure 3.15 displays how the length from the base to the virtual pivot point increases as the joint angle increases.

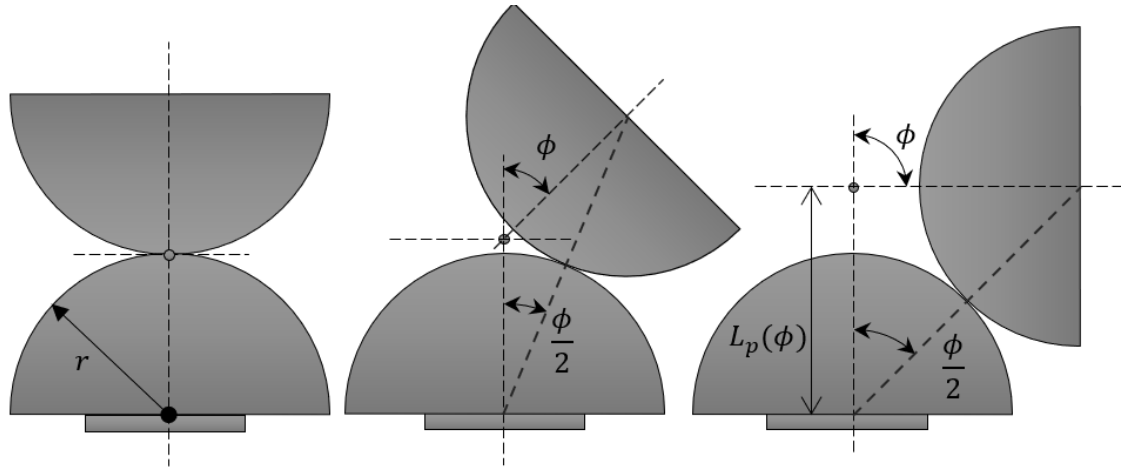


Figure 3.15. Simulated Rolling Contact of two hemispheres using virtual pivot point

The relationship between joint angle and pivot length is:

$$L_p(\phi) = \frac{r}{\cos\left(\frac{\phi}{2}\right)} \quad (3-5)$$

Where $r(\phi)$ is the new pivot length, r is the original radius of the hemisphere, which is half the overall mechanism length, and ϕ is the joint angle.

The results of these simulations (Figure 3.16) clearly show the advantages of the three proposed new mechanisms over the original Pitch-Yaw design. All three have a much more consistent range, equal in all positions. The Pitch-Roll and Spherical mechanisms form more spherical surfaces whereas the Quaternion is distorted by the rolling contact simulation. See Appendix B for further explanation.

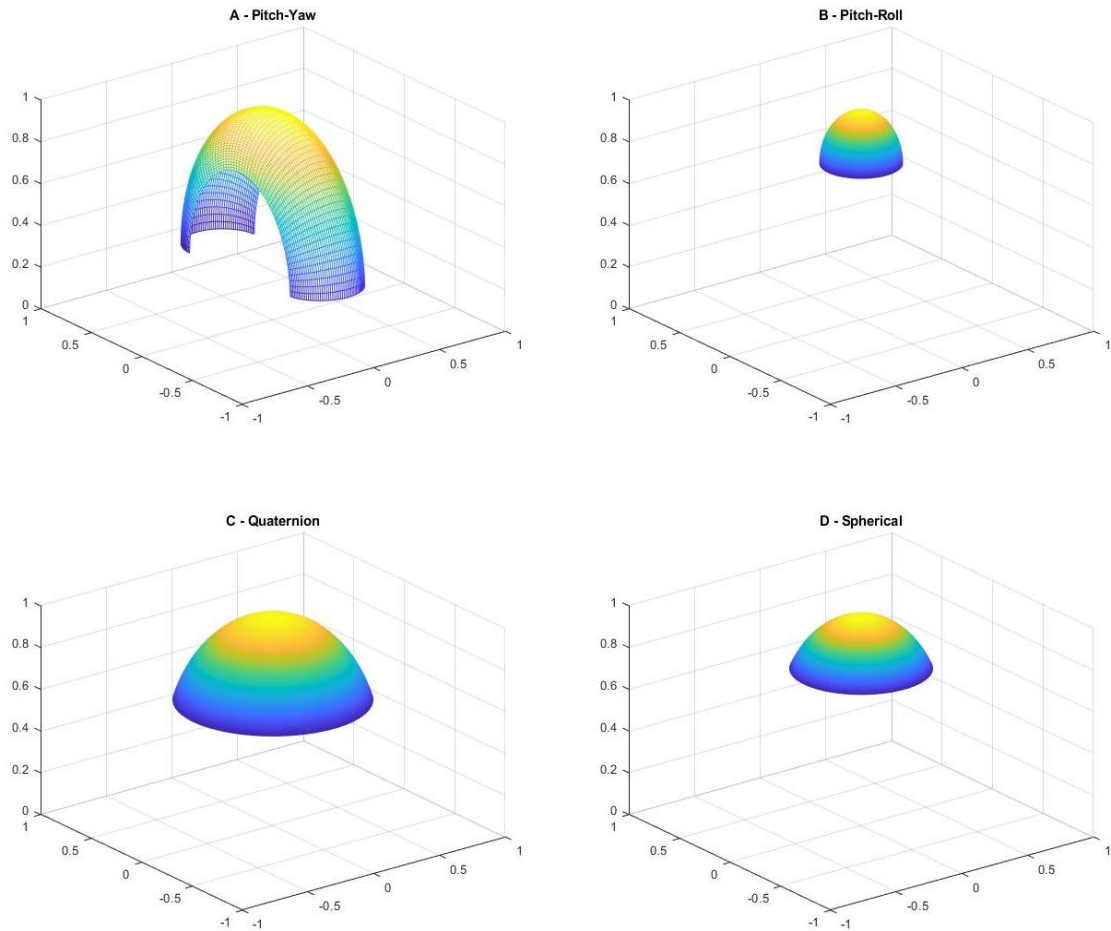


Figure 3.16. Available Workspace for four concept joint mechanisms

3.5.6 Final Analysis and Conclusion

The selection of a new joint mechanism design must be based upon all aspects of design, not only the theoretical mechanical properties. The mechanism complexity, manufacturing costs and intricacy of assembly should all be considered. Another factor we should consider is novelty. Innovation and uniqueness are two of the minor desirable features for this project. The pitch-yaw and, to some extent, the pitch-roll configurations have both been used in the design of snake-like robots before. The use of a spherical or quaternion mechanism mimicking the spine of a natural snake have not. This is an opportunity to include some innovation within the project. Table 3-4 shows a summary of the positive and negative aspects of each of the four prototype structures.

Table 3-4. Summary of Prototype Joints

Mechanism	Advantages	Disadvantages
Pitch-Yaw	<ul style="list-style-type: none"> • Simple Construction • Can use mostly COTS components • Control Systems already exist 	<ul style="list-style-type: none"> • Poor workspace consistency • Direct connection to motor limits customisability • Limited by available motors • Fatigue of electrical connections likely
Pitch-Roll	<ul style="list-style-type: none"> • Control Systems already exist for similar systems. • Half the mechanism uses COTS components, reducing complexity in part. • Constant and full workspace. • Somewhat unique 	<ul style="list-style-type: none"> • Direct connection to motor limits customisability • Limited by available motors • Fatigue of electrical connections likely and complicated by the longitudinal rotation. • Possibly requires an additional rotational degree of freedom, increasing complexity again.
Quaternion	<ul style="list-style-type: none"> • Full access to the workspace. • Selection of geometry and components can be used to tune the joint properties easily. • Excellent electrical cable protection • Fully unique, never been used in a snake-like robot. 	<ul style="list-style-type: none"> • Inconsistent workspace but it can be estimated. • No existing controllers. • Cable driven means there is no direct measure of location in the mechanism.
Spherical	<ul style="list-style-type: none"> • Most life-like representation of a snake. • Consistent workspace. • Selection of geometry and components can be used to tune the joint properties easily. • Fully unique, never been used in a snake-like robot. • Likely to be easy to adjust existing controllers 	<ul style="list-style-type: none"> • Small workspace • Fatigue of electrical connections likely • Cable driven means there is no direct measure of location in the mechanism. • Additional degree of freedom makes control almost impossible.

The Pitch-Yaw and Spherical mechanisms both have major flaws, in the poor workspace range and uncontrollability, respectively. Therefore, the Pitch-Roll and Quaternion are the two most viable joints. We choose the Quaternion as the basis, as we feel the Pitch-Roll would need to be reworked to include an additional degree of freedom. While this makes the task of designing a control system more challenging, the uniqueness of the design, along with its other benefits make it an excellent structure for the mechanism within a snake-like robot.

3.6 Quaternion Mechanisms Design

3.6.1 Geometry Selection

Previously we have described the 3-bar parallel Quaternion mechanisms as mimicking the rotation of two hemispheres. This is not fully accurate as the motion is more akin to that of two ellipsoids. The radius of this “virtual ellipsoid” is not smooth and can vary significantly both increasing and decreasing depending upon the joint angle. Fortunately, it is possible to predict this deviation by calculating the position of the contact point, and thus the new radius, over a range of joint angles. Using this method, we aim to select geometry with the least deviation, thus giving the most circular joint possible. Doing so will keep the mechanism consistent in all orientations.

Kim et al, 2018 [83], show the derivation for this equation, calculating $r(\psi)$ as a function of the joint geometry, h_o , h_c and w_c . Note ψ is used in place of joint angle ϕ , where ψ is the angle from the vertical axis to the point of contact between the two surfaces. As the two hemispheres are identical, this is equal to half the current joint angle, as shown in Figure 3.15. The locations of the other geometric constants are shown in Figure 3.17.

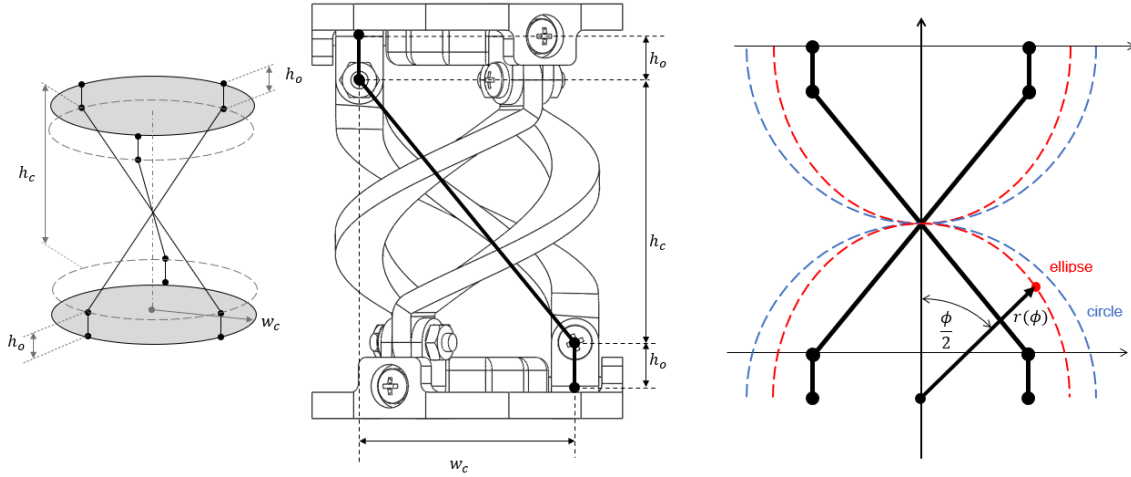


Figure 3.17. Quaternion Mechanism Geometry and Key Measures. 3D Model (left), 2D Geometry on 3D CAD Model (centre) and 2D Structure showing variable radius (right).

However, there appears to be a mistake in the derivation presented by Kim et al (see Appendix C.1 for clarification). We therefore derive the expression ourselves using the same method to find $r(\phi)$ as a function of h_o , h_c and l_c (3-6). This derivation is shown in Appendix C.2.

Note: length l_c is the length of the hypotenuse between h_c and w_c , thus $l_c^2 = h_c^2 + w_c^2$. This gives:

$$r(\phi) = \frac{h_o + \sqrt{h_o^2 + \left(1 + \tan^2 \phi \left(\frac{h_c}{l_c}\right)^2\right) \left(\left(\frac{h_c}{2}\right)^2 - h_o^2\right)}}{\cos \phi \left(1 + \tan^2 \phi \left(\frac{h_c}{l_c}\right)^2\right)} \quad (3-6)$$

We can then estimate the performance of the mechanism by calculating radius over a range of angles. A joint which has a low variance of r would be considered a better performing joint than one with a wide variance, as a wide variance would lead to difficulty in predicting the motion of the mechanism, plus increases the possibility of strain on any cables or connections passing through the mechanism. The example shown in Figure 3.18 has a maximum variance of 2.61 mm. This means that over the full range of motion, 0 to 90-degrees, the distance between the two ends of the mechanism will vary by over 5 mm.

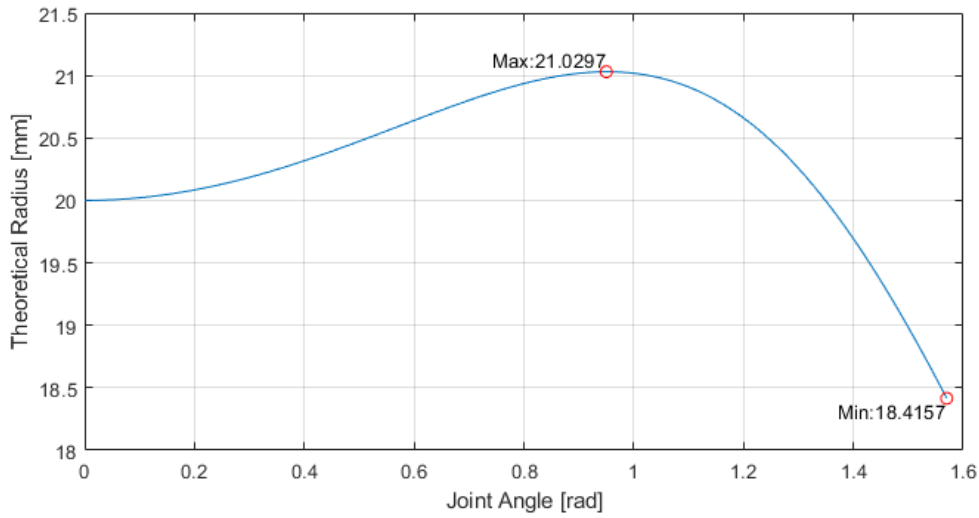


Figure 3.18. Graph showing the variance of the theoretical radius for a Quaternion mechanism with values $h_c = 30, h_o = 5, w_c = 25$.

To first analyse the effect of varying each of these values, we create the plots shown in Figure 3.19. These graphs plot maximum variance of the radius (Δr) over a wide range of values for each of the three key geometries (h_c, h_o, w_c). Note, the two values not varied remain constant at either $h_c = 30, h_o = 5$ or $w_c = 25$.

Analysing these graphs, a number of interesting points stand out. Firstly, in each instance there is a point where variance descends to near zero, before rising again. This means that there is likely to be a point at which the geometry is most effective for any combination of values. Therefore, if we are able to fix two of the three values as constants, we should be able to define the third value using this method and find the most effective geometry.

We can also tell from these graphs that there appears to be a maximum limit to the performance of the mechanism. After the lengths are increase so far, the rate of increase of the variance slowly decreases as the value tends to a fixed point. While not necessarily useful for our design task at this moment, it does show there is only a single point which provides the best performance for the mechanism.

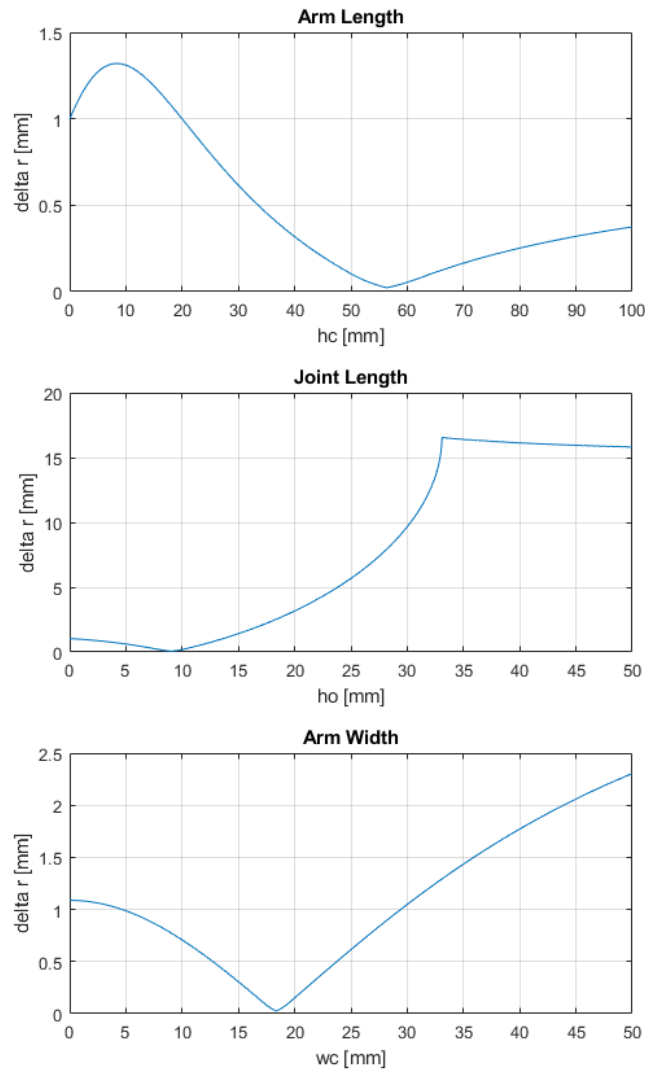


Figure 3.19. Graphs showing Δr estimates over a range of values for Arm Length (h_c) [Top], Joint Length (h_o) [Middle] and Arm Width (w_c) [Bottom]

Finally, while a value of $w_c = 0$ would be impossible to conceive, a value of $h_o = 0$ would effectively be replacing this linkage with a universal-joint style mechanism and is perfectly conceivable. The mechanism does not perform at its peak in this case, but it is better than an overly long mechanism. Out of curiosity, we also calculate the performance of a theoretical mechanism where $h_o < 0$. As shown in Figure 3.20, the performance of a mechanism with a negative h_o value is not so different to that with a positive one. Such a mechanism would be complex to manufacture, but not implausible, and would somewhat resemble the concept shown in Figure 3.21.

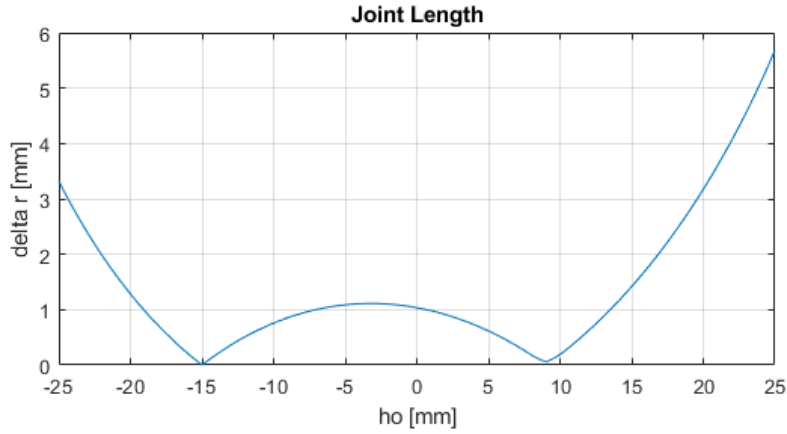


Figure 3.20. Graph showing Δr estimates for $h_c = 30$, $w_c = 25$ and $-25 \leq h_o \leq 25$

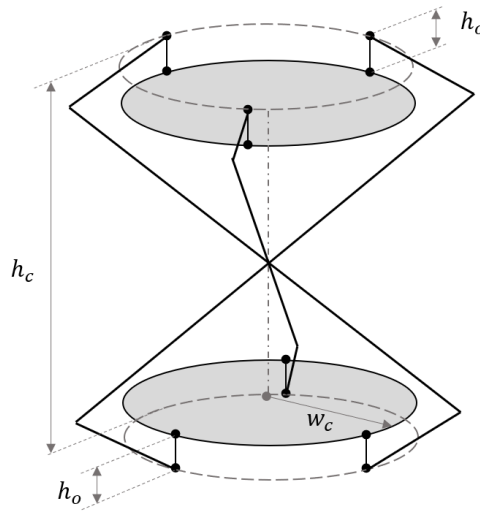


Figure 3.21. Concept Quaternion Mechanism with negative h_o value

Using this knowledge, we can begin to select geometry which will create the most suitable joint. We reuse the size estimates from Table 3-1 as a basis and predict the most effective geometry for each of the sizes.

Before calculating the output there are a number of assumptions which must be made. As shown above, typically reducing h_o usually improves the joint performance. Therefore, we set h_o to be 5 mm. This value is chosen as it results in the smallest possible joint with adequate strength, which we are currently able to produce using 3D printing. Once geometry for the other two lengths is found, we can check this value again.

Secondly, the values of length and diameter calculated in Table 3-1 are for the total diameter and length of the mechanism. As the prototype in Figure 3.12 shows, the Quaternion mechanism is only a portion of the joint. A section to house the motors and other electronics is required at either end, plus the additional length required for the drive unit. A section to house the motors and other electronics is required at either end, plus the additional length required for the drive unit. In terms of the diameter, the components have real width unlike the stick model in Figure 3.17. This also limits the minimum diameter. Finally, the tensile wires which actuate the mechanism must also protrude beyond this width.

Therefore, when creating estimates for the geometry we set the following limits:

Table 3-5. Joint Simulation Limits

$h_o = 5$	$h_{c,max} = (0.5 * L) - (2 * h_o)$ $h_{c,min} = 5$	$w_{c,max} = 0.8 * D$ $w_{c,min} = 15$
-----------	---	--

This gives the following values for the three pipe sizes under consideration. We still consider the 40 mm pipe for the time being. We can now calculate the best geometry for each pipe size. We do this by calculating Δr over the range $0 \leq \phi \leq \phi_{max}$ for each combination of geometry between the minimums and maximums. The true limits, plus the results for the minimum Δr calculated, and the geometry at the minimum Δr are shown in Table 3-6.

Table 3-6. Quaternion Geometric Limits for 3 Pipe Sizes

Pipe Dimension	$h_{c,max}$ (mm)	$w_{c,max}$ (mm)	ϕ_{max} (rad)	Δr_{min} (mm)	$w_{c,\Delta r_{min}}$ (mm)	$h_{c,\Delta r_{min}}$ (mm)
40mm	38.8	21.84	1.17	0.0243	38.55	20.75
60mm	51.35	33.04	1.43	0.0525	51.25	24.5
80mm	59.05	46.56	1.57	0.0756	58.65	26.6

The full results are shown in Figure 3.22, including the minimum location shown in red. It is clear to see from the graphs that maximising the arm length (h_c) will result in the best performing mechanism. Additionally, as shown in the 2D orientation, there is a clear curve of the minimum Δr . We can use this to select

our chosen geometry. Once a single dimension is chosen, (h_c or w_c) we can look to the curve on the corresponding graph and estimate the most appropriate value for the second dimension (w_c or h_c). This gives an excellent starting point from which the design can be refined.

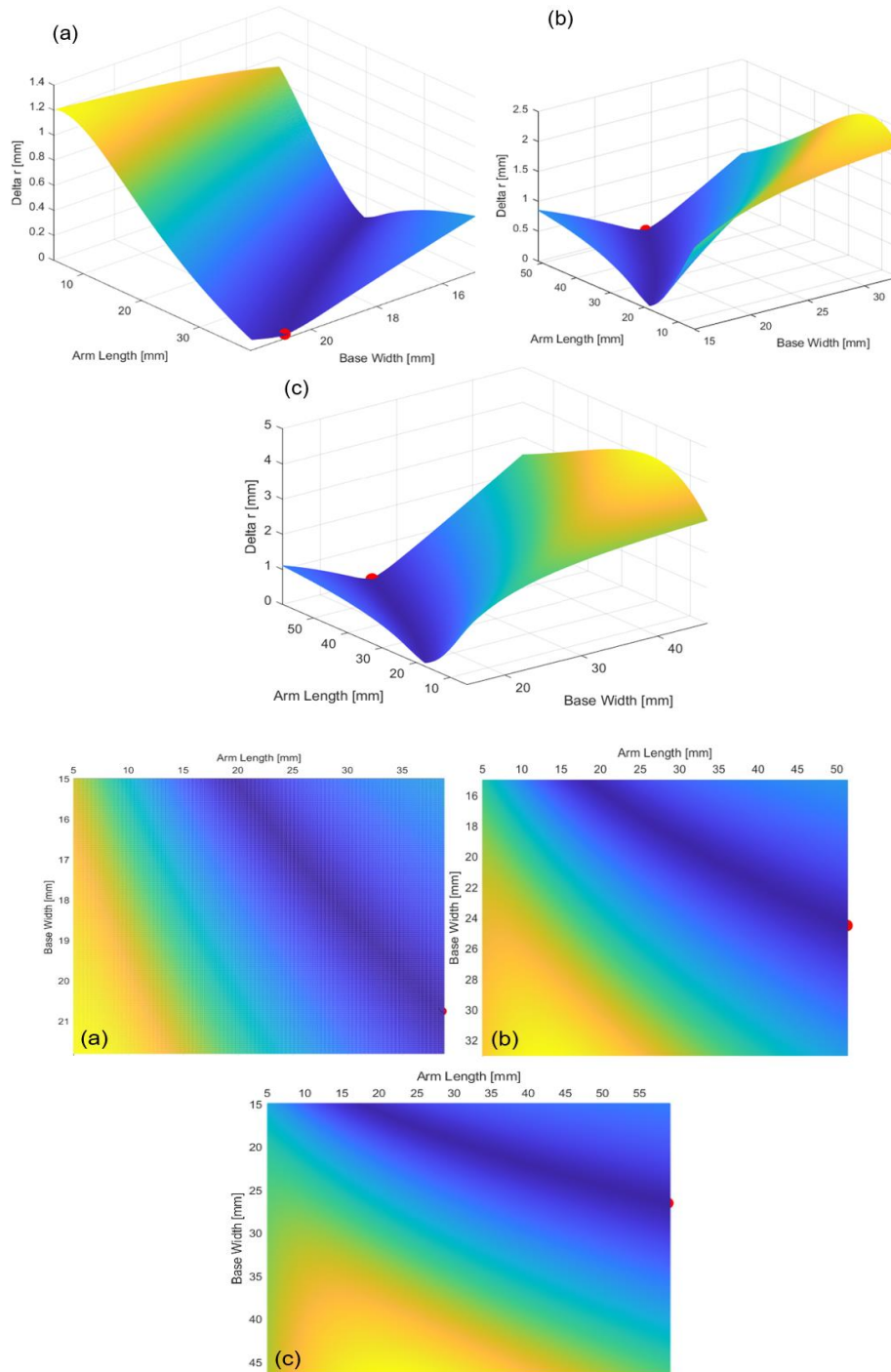


**Figure 3.22. Geometry Scan Results for Pipe Diameters in 3D and 2D orientations
(a: 40mm, b: 60mm, c: 80mm)**

3.6.2 Analysing Quaternion Motion in 3-Dimensions

Thus far we have only considered the motion of the quaternion mechanism in a 2-dimensional plane, with a single degree-of-freedom. We should also consider the motion in 3-dimensions. The reasoning for analysing the mechanism in 3-dimensions is to ensure the performance of the joint does not drastically alter when in different bending plane. This is however not a simple task.

In fact, if we investigate the origins of this mechanism, in Kim et al 2018, we can see that the authors opted not to explore this aspect of the structure. This is almost certainly due to the complexity of analysing the forward kinematics of parallel or “closed-loop” mechanisms. This is due to the wide variety of design possibilities with parallel manipulators. Additionally, the configuration space of these mechanisms, that is the list of positions the end-effector can access, are usually complicated, as they must satisfy a number of loop-closure equations. Furthermore, there are types of singularities within parallel mechanisms which simply do not exist with traditional open-loop manipulators and the choice of which joints to actuate, and which are left passive can affect which singularities are created. Whereas an open-chain robot would have all joints actuated. Overall, the main issue with analysing the forward kinematics of a parallel mechanism boil down to the fact that there can be multiple solutions to the problem.

Therefore, it is understandable that the authors chose not to attempt to describe the 3-dimensional motion of the Quaternion mechanism, and instead chose a different approach to prove the motion would be possible. They instead use a modified version of the Gruebler-Kutzbach criterion as described by Yu et al., 2012 [1]. Using this method, they analyse the position of the end-effector of the mechanism three times, once for each limb, over a range of motions, and compare locations. Due to the nature of the mechanism, the position will never be exactly the same as one would expect. However, the positions are relatively close and in real-world applications these discrepancies would be absorbed by flexibility of the linkages.

To examine the motion in 3-dimensions for SASAR, we rely on a practical approach. Once a set of values for the key geometries have been selected, a physical prototype is produced using additive manufacturing, such as the one shown in Figure 3.23. This particular example shows the major issue with selecting suitable geometry, which is the interference of the main linkage limbs with each other.



Figure 3.23. Prototype SASAR Quaternion Joint, showing Interfering limbs

Thankfully, due to the rapidity of 3D printing, we are able to adjust the geometry and quickly iterate on the design until suitable geometry is found.

3.6.3 Quaternion Joint Position and Torque Estimation

The final component of the Quaternion we investigate is the effect the circular motion has on the positional and torque requirements for the motor driving the joint. In particular, we require a calculation to estimate the joint pose in terms of motor position, and a method to compare torque applied to the joint to the tensile forces in the wires. An estimation of this allows us to specify the minimum required torque on the motor.

Position Estimation

We begin by looking at a 2D representation of the mechanism. Fortunately, the motion is well defined by Kim et al in [83]. As the mechanism rotates, the left wire

length l_{left} will contract by an equal amount by which the right side extends, as shown in Figure 3.24. This amount can be shown to be equal to $nw \sin\left(\frac{\phi}{2}\right)$, where n is the number of loops of wire used in the pulley assemblies. Of course, the change in wire length is directly linked to the rotational position (α), and radius (r_d) of the wire drum. Therefore, in a 2D plane, the relationship between drum angle and joint angle is:

$$\alpha = \frac{nw \sin\left(\frac{\phi}{2}\right)}{r_d} \quad (3-7)$$

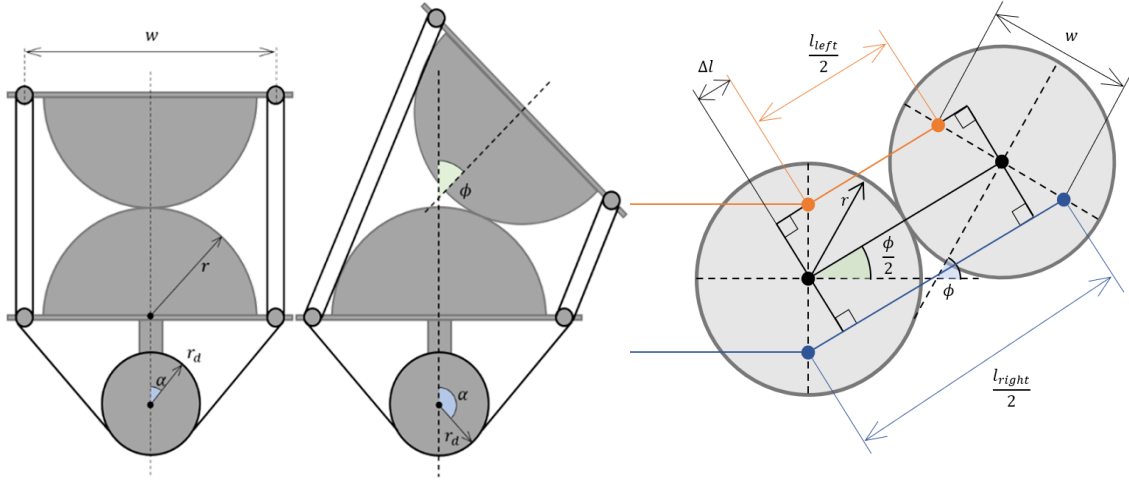


Figure 3.24. 2D Quaternion Mechanism Wire and Position Changes

For the 3-dimensional position, we introduce the concept of a bending plane. This is the dorsal plane in which the mechanism is bending, measured in the YZ plane, with reference to the Z axis (Figure 3.25). This replaces the traditional Pitch/Yaw directional descriptions for the quaternion mechanism, and we term this angle β . For example, a 30-degree joint angle in the “Pitch Up” direction would now consist of a pose of $\phi = 30, \beta = 0$, a 45-degree joint angle in the yaw-left direction would be $\phi = 45, \beta = 90$. However, this means the quaternion mechanism is not locked to moving in only the cardinal directions (up, down, left, right) and can effectively turn at any bending plane angle desired.

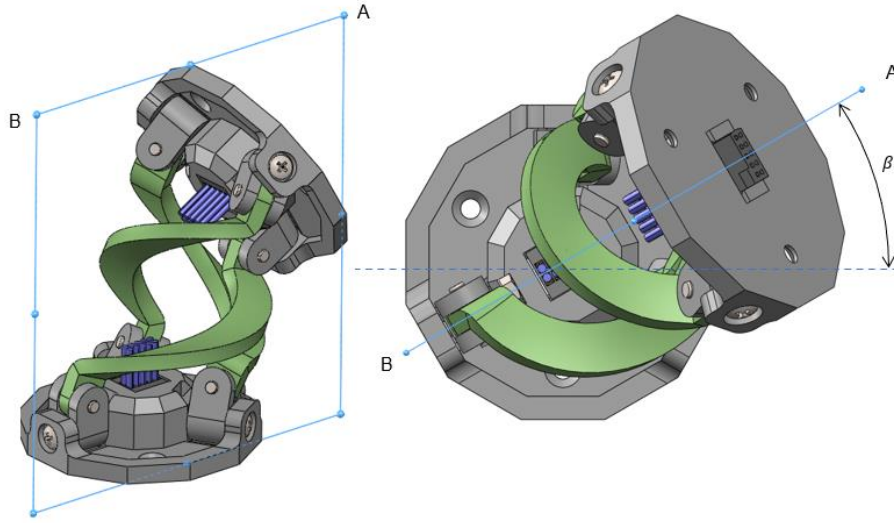


Figure 3.25. 3D CAD model of a Quaternion mechanism, showing the bending plane angle.

However, to use 3-D motion we require a second motor. We term these two motors the “Pitch” and “Yaw” motors, depending on which cardinal direction they primarily control. Again, Kim et al have shown the pose of a Quaternion mechanism (ϕ, β) can be defined in terms of the change in wire length of the two motors, and thus we can describe the angular positions of the two motor drums in terms of the desired pose of the robot.

$$\alpha_{Pitch} = \frac{nw \sin(\beta) \sin\left(\frac{\phi}{2}\right)}{r_d} \quad \alpha_{Yaw} = \frac{nw \cos(\beta) \sin\left(\frac{\phi}{2}\right)}{r_d} \quad (3-8)$$

Torque Calculation

We use a simple static body analysis, as only an estimate is required. We assume the familiar hemi-spherical rolling contact structure as before. The hemispheres have radius r , which we assume to be constant. A tensile wire connects the two halves of the module a distance $\frac{w}{2}$ from the central axis, under tension t_w . This wire is directly connected to a drum, radius r_d , which is coaxially attached to a motor with output torque τ_m . An identical wire connects the lower half but has no impact on the mechanism. A torque, t_j , is then applied to the mechanism, about

the pivot point. In the nominal pose, when the joint angle is zero, the tensile force on the wire will simply be:

$$t_w = \frac{2t_j}{w} \quad (3-9)$$

However, when $\phi \neq 0$ this will not be true. Instead, we envisage the pin joint structure in Figure 3.26. This is again a simple pin joint structure, however the location of the pivot, and thus the length of the members of the structure, change dependant on the joint angle. The pivot is the singular point of rolling contact between the two hemispheres.

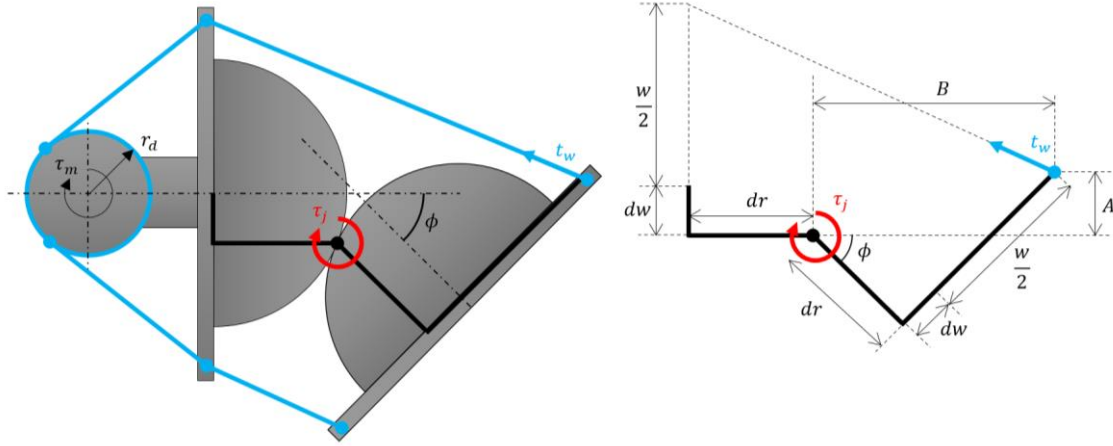


Figure 3.26. Static Quaternion Force Model

Where $dr = r \cos\left(\frac{\phi}{2}\right)$ and $dw = r \sin\left(\frac{\phi}{2}\right)$.

We can therefore create a moment balance equation about the pivot point.

$$\tau_j = A \left[t_w \sin\left(\frac{\phi}{2}\right) \right] + B \left[t_w \cos\left(\frac{\phi}{2}\right) \right] \quad (3-10)$$

Where:

$$A = [dw \cos(\phi)] - [dr \sin(\phi)] + \left[\frac{w}{2} \cos(\phi) \right]$$

$$B = [dr \cos(\phi)] + [dw \sin(\phi)] + \left[\frac{w}{2} \sin(\phi) \right]$$

We then solve for the tensile force in the wire. This simplifies down significantly and is interestingly not dependant on the radius of the quaternion mechanism.

$$t_w = \frac{2 \tau_j \sec\left(\frac{\phi}{2}\right)}{w} \quad (3-11)$$

Thus, by calculating t over the range $0 \leq \phi \leq \phi_{max}$, the maximum required torque $t_{w,max}$ can be found. The minimum required motor torque, $\tau_{m,min}$, can then be easily found using the drum radius r_d .

$$\tau_{m,min} = \frac{t_{w,max} \times r_d}{n_p + 1} \quad (3-12)$$

Where n_p is the number of pulley loops used ($n_p = 0$ for no pulleys).

3.7 Summary

In this chapter we have laid the foundations for the design of a new, novel, snake-like robot joint mechanism. We have used geometry to estimate the maximum size of the individual robot modules in the case of three different pipe structure, and then evolved this to estimate the required torque.

We have then used previous research to create three new prototype mechanisms, which have been compared against the original SASAR 1 Joint Module design. The designs were evaluated by a number of criteria, including the ease of manufacture, the complexity to design and the novelty of the structure. Finally, we studied the available workspace for each of the mechanism. From this, a Quaternion Joint mechanism was chosen.

We then investigated the design of the Quaternion mechanism, using the module size estimates to calculate estimates for the key geometric constants of the mechanism. Lastly, we used a simple pin-joint structure to create a formula for estimating the required torque of a motor, depending on the drum diameter, and how many pulley loops are used.

4 SASAR AUTONOMY AND CONTROL

4.1 Introduction

In this chapter we discuss the control of a snake-like robot. We begin by defining four operational modes for the robot, each with an increasing level of autonomy, and increasing functional requirements. To fully distinguish what we mean by the phrase “Autonomous” we look to existing descriptions used in industry, including those used by the Society of Automotive Engineers (SAE) for the design of autonomous vehicles, and those described by NASA (National Aeronautics and Space Administration) scientists for autonomous flight management (AFM) systems. We next investigate different control methods for our new snake-like robot, beginning by introducing our scheme for labelling a generic snake-like robot with driven wheels. We then present two methods for controlling the motion of a snake-like robot, primarily in a 2-D plane, selecting the algorithm which is most effective. We then investigate the changes which must be made to enable the algorithm to be compatible with a quaternion joint mechanism and use simulations to see how these changes affect the control of a robot. To analyse the algorithm, we compare it against an existing model using simulations, and then create a virtual 3D environment in which to test a simplified version of the robot.

4.2 Definition of Autonomy

In order to design a control system for SASAR we must first fully define what we expect the system to be capable of. To this end, a definitive explanation of the term “Autonomous” is required. The word “Autonomous” comes from the Greek *autonomos*, meaning living under one’s own laws [90]. Typically, the modern phrase ‘autonomous’ refers to a device or vehicle which can control itself without the need of a person to operate.

While this definition may be suitable in most situations, for our purposes a more specific description, detailing the expected functionality of an autonomous robot, such as SASAR, is required. With such a definition we will be able to critically analyse the performance of SASAR against its intended operation and allow any

future operator to understand what the device is capable of, and what autonomous mode would be suitable for their required purpose.

4.2.1 Levels of Autonomy and Application Scenarios

Four distinct levels of operation will be defined for SASAR, which have increasing levels of autonomy. They are:

- 1) Manual Mode
- 2) Semi-Autonomous Mode 1
- 3) Semi-Autonomous Mode 2
- 4) Fully Autonomous Mode

In manual mode, the human operator will use a handheld device to control the system, while monitoring the system using real-time images from the onboard camera. Therefore, the operator will use the images to determine the best route, and how to manoeuvre around any obstacles. All the controls are performed by the operator, such as choosing the type of serpentine movement, and in which direction to move. The Manual Mode aims to test solely the SASAR mechanical functions, such as joint motion, crawling, and obstacle avoidance capabilities. We should realise that the operator cannot directly observe the environment where SASAR may work.

In Mode 2, the first semi-autonomous mode, SASAR shall automatically detect obstacles and possible crawling paths and send the analysed results to the operator. The operator guides the SASAR motion, but it is the system which will interpret the operator's controls into a form of crawling. There will be no guidance or navigation functions in this mode: it is up to the operator to use the information provided by the system, such as live images and detected obstacles, to manoeuvre SASAR.

The second semi-autonomous mode (Mode 3) will have all the functionality of Mode 2, but with added guidance features. The system will be able to detect the obstacles and possible paths, like Mode 2, but will also be able to discern the most effective path according to its environment, and then move along this path. The human operator will be able to change the path of SASAR at any time but left alone the system will continue forward determining the best path available to

it, without need for intervention from the operator. In this mode, SASAR should have some fail-safe mechanisms implemented to ensure it does not continue into a hazardous position. This could be as simple as the system stopping if it detected no available safe path to move through and waiting for the operator to take control.

The final, fully autonomous, mode will have all the previous capabilities, with an additional navigation function. The system will use a pre-planned route or mission to guide itself to a target using onboard software and sensors. In this mode, there is no requirement for any human interaction with the system. SASAR will transmit all sensor data to an external device for data recording and monitoring. If an operator wishes, they can take command of SASAR at any time, by changing the operational mode to one of the previous three descriptions.

Table 4-1 condenses these descriptions down by showing the different available features of each mode.

Table 4-1. SASAR Autonomy Mode Features (X = Feature Active)

Mode	Live Sensors	Obstacle Detection	Path Following	Route Navigation
1	X			
2	X	X		
3	X	X	X	
4	X	X	X	X

4.2.2 Existing Autonomy Definitions and Standards

In order to define each level more accurately, existing definitions for autonomy are considered. The SAE has recently introduced a new standard, in which the levels of autonomy are defined for the design of a self-driving car [91]. This standard (J3016) defines 6 levels of automation, as shown in Figure 4.1. Similar to our definition, this standard differentiates levels of automation, based on whether certain features are controlled by the system or by the driver. The main distinction between autonomous and not autonomous occurs between levels 2

and 3, where the vehicle control system progresses from only performing part of the “*dynamic driving task*” to performing the entirety of it.

SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/Deceleration	Monitoring of Driving Environment	Fallback Performance of Dynamic Driving Task	System Capability (Driving Modes)
Human driver monitors the driving environment						
0	No Automation	the full-time performance by the <i>human driver</i> of all aspects of the <i>dynamic driving task</i> , even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	the <i>driving mode</i> -specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the <i>human driver</i> perform all remaining aspects of the <i>dynamic driving task</i>	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	the <i>driving mode</i> -specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the <i>human driver</i> perform all remaining aspects of the <i>dynamic driving task</i>	System	Human driver	Human driver	Some driving modes
Automated driving system (“system”) monitors the driving environment						
3	Conditional Automation	the <i>driving mode</i> -specific performance by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> with the expectation that the <i>human driver</i> will respond appropriately to a <i>request to intervene</i>	System	System	Human driver	Some driving modes
4	High Automation	the <i>driving mode</i> -specific performance by an automated driving system of all aspects of the <i>dynamic driving task</i> , even if a <i>human driver</i> does not respond appropriately to a <i>request to intervene</i>	System	System	System	Some driving modes
5	Full Automation	the full-time performance by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> under all roadway and environmental conditions that can be managed by a <i>human driver</i>	System	System	System	All driving modes

Figure 4.1. The 6 Levels of Automation defined in J3016 [91]

The “*Dynamic Driving Task*” is the operational and tactical aspects of driving, but not the strategic aspects. In this instance, J3016 defines the Operational aspects as the steering, braking, accelerating, and monitoring of vehicle and road, whereas the Tactical aspects are the response to events, for example, determining when to change lanes or turn. The final components are the Strategic aspects of driving, which include features such as choosing the intended destination or any waypoints.

A “Driving Mode” of an automated vehicle will typically include driving scenarios with characteristic dynamic driving task requirements. For example, merging to main road, cruising at high speed or being in low-speed traffic.

While these 6 definitions do represent automated driving systems very well, they have little transferability to the SASAR system, as all the 5 automated systems defined in J3016 remove the execution of the steering and other operational

control of the vehicle first, before increasing automation. The SASAR system is most closely related to J3016 level 5, but with the first column (Execution of Steering) being controlled by human driver and the system.

Another, simple autonomy scale is that used by Sheridan, first shown in 1992 [92]. This system defines 10 different levels of automation for computer control. Table 4-2 displays these 10 levels.

Table 4-2. Sheridan's Scale of Degrees of Automation

Level	Definition
1	The computer offers no assistance, the human must do it all.
2	The computer offers a complete set of action alternatives, and...
3	Narrows the selection down to a few, or...
4	Suggests one, and...
5	Executes that suggestion if the human approves, or...
6	Allows the human restricted time to veto before automatic execution, or...
7	Executes automatically, then necessarily informs the human, or...
8	Informs the human after execution, only if he asks, or...
9	Informs the human after execution if it, the computer, decides to.
10	The computer decides everything and acts autonomously

When broken down, this table can be simplified into 3 sections. Levels 1 and 10 are the appropriate extremes of full human or computer control, respectively. Levels 2, 3, and 4 define who is making the decision, the computer, or the human. The remaining levels (5 through 9) determine how to execute the decision.

As with the standard, this scale does not match with how the SASAR system is intended to perform, as these levels assume the system is only performing one overall task, not multiple, as is expected of SASAR. To look at this in more detail, a model first presented by Sheridan and Wickens, 2000 is considered [93]. This model splits the tasks any human or machine would be required to perform into four categories: Information Acquisition, Information Analysis, Decision and Action selection, and Action Implementation.

Information Acquisition is the act of sensing, monitoring, and bringing information to the attention of the next system, be that human or machine. Information Analysis is then the act of processing, predicting and otherwise analysing the data acquired. Decision and Action Selection uses the processed data to create a selection of possible responses to this information, and possibly narrowing the responses down to one. Finally, Action Implementation is the process of carrying out the chosen action, based on the decision.

A similar system, by Boyd initially in 1996, based solely on human decision making also exists [94]. This is known as Boyd's OODA loop, OODA being Observe, Orient, Decide and Act. Figure 4.2 shows a diagram of this loop.

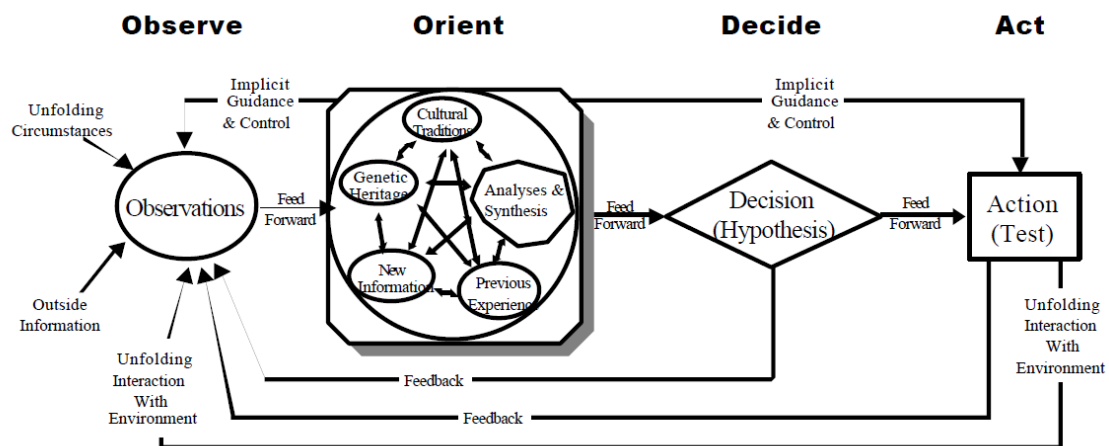


Figure 4.2. Diagram of the Boyd OODA Loop [94]

This system implements two key features over the previous models: feedback and implicit control. The feedback loops mean the decisions made during the Decide stage do not necessarily lead to the actions being carried out. They may

instead lead to new analysis or observation tasks. The Implicit control is a process which is running in the background of the process, for example a Guidance, Navigation, and Control system.

This system fits the SASAR model much better. An example process would be:

Observe. The system senses an increase in the incline of the surface that SASAR is moving across.

Orient. The system processes sensor data, such as multiple gyroscopes to determine the exact value of the incline, or that the measured stress on the actuators has increased because of the excess required power to continue up the incline.

Decide. Using the sensor data, the system calculates how much the power to the actuators must be increased by to maintain the same average velocity. The physical form of the snake may be altered, such as to increase the frictional coefficient of the snake.

Act. The additional power and/or physical alteration are implemented.

Feedback. The snake measures that the system is now moving faster than average intended speed, thus is being inefficient. The process repeats beginning with the Observe stage.

This may be a simplified version of the process, but it does correlate with the SASAR system better than the J3016 or the Sheridan scale.

Proud, Hart, & Mrozinski, 2003 [95] , took the Boyd OODA model, and combined it with a level scale similar to the Sheridan scale to create their Level of Autonomy Assessment (LoA) Scale. This 8-level scale has a definition of each element of the OODA model, which progresses each level (Figure 4.3).

Level	Observe	Orient	Decide	Act
8	The computer gathers, filters, and prioritizes data without displaying any information to the human.	The computer predicts, interprets, and integrates data into a result which is not displayed to the human.	The computer performs ranking tasks. The computer performs final ranking, but does not display results to the human.	Computer executes automatically and does not allow any human interaction.
7	The computer gathers, filters, and prioritizes data without displaying any information to the human. Though, a "program functioning" flag is displayed.	The computer analyzes, predicts, interprets, and integrates data into a result which is only displayed to the human if result fits programmed context (context dependant summaries).	The computer performs ranking tasks. The computer performs final ranking and displays a reduced set of ranked options without displaying "why" decisions were made to the human.	Computer executes automatically and only informs the human if required by context. It allows for override ability after execution. Human is shadow for contingencies.
6	The computer gathers, filters, and prioritizes information displayed to the human.	The computer overlays predictions with analysis and interprets the data. The human is shown all results.	The computer performs ranking tasks and displays a reduced set of ranked options while displaying "why" decisions were made to the human.	Computer executes automatically, informs the human, and allows for override ability after execution. Human is shadow for contingencies.
5	The computer is responsible for gathering the information for the human, but it only displays non-prioritized, filtered information.	The computer overlays predictions with analysis and interprets the data. The human shadows the interpretation for contingencies.	The computer performs ranking tasks. All results, including "why" decisions were made, are displayed to the human.	Computer allows the human a context-dependant restricted time to veto before execution. Human shadows for contingencies.
4	The computer is responsible for gathering the information for the human and for displaying all information, but it highlights the non-prioritized, relevant information for the user.	The computer analyzes the data and makes predictions, though the human is responsible for interpretation of the data.	Both human and computer perform ranking tasks, the results from the computer are considered prime.	Computer allows the human a pre-programmed restricted time to veto before execution. Human shadows for contingencies.
3	The computer is responsible for gathering and displaying unfiltered, unprioritized information for the human. The human still is the prime monitor for all information.	Computer is the prime source of analysis and predictions, with human shadow for contingencies. The human is responsible for interpretation of the data.	Both human and computer perform ranking tasks, the results from the human are considered prime.	Computer executes decision after human approval. Human shadows for contingencies.
2	Human is the prime source for gathering and monitoring all data, with computer shadow for emergencies.	Human is the prime source of analysis and predictions, with computer shadow for contingencies. The human is responsible for interpretation of the data.	The human performs all ranking tasks, but the computer can be used as a tool for assistance.	Human is the prime source of execution, with computer shadow for contingencies.
1	Human is the only source for gathering and monitoring (defined as filtering, prioritizing and understanding) all data.	Human is responsible for analyzing all data, making predictions, and interpretation of the data.	The computer does not assist in or perform ranking tasks. Human must do it all.	Human alone can execute decision.

Figure 4.3 Level of Autonomy Assessment [95]

This model starts at level 1 being complete human control, and concludes with level 8, complete computer control. In level 2 the human is still the primary, with the computer being used as a tool for emergency situations. For levels 3-5, the computer performs most operations, with decreasing levels of human interaction. Levels 6-8 the computer performs independently of the human, and the human has decreasing levels of access to the information the computer is producing.

To decide what LoA a current, or future process should be, Proud, Hart, & Mrozinski developed a questionnaire totalling 35 questions where each question would each receive a score out of 5. By a process of averaging and weighting the score for each question, the questionnaire would produce a score corresponding

to the level of autonomy described in Figure 4.3. However, without knowing what these weightings are for the questions, it is impossible to use.

4.2.3 SASAR Definition of Autonomy

The four levels defined in Table 4-1 can now be fully defined. By taking inspiration from J3016, the Sheridan scale, the OODA loop and primarily the Proud, Hart, & Mrozinski LoA Assessment, the following four definitions are created.

Level 1: Manual Control

The manual mode definitions are described by Table 4-3.

Table 4-3. Definition of SASAR Level 1

Category	Description	Reasoning/Example
Observe (Level 3)	The computer is responsible for gathering and displaying unfiltered, unprioritised information for the human. The human still is the prime monitor for all information.	In this mode the system will feed live images from the head of SASAR to the operator. Level 3 is the lowest level where the system is the prime source for collecting the data.
Orient (Level 1)	Human is responsible for analysing all data, making predictions, and interpretation of the data.	Operator looks at images and uses them to determine best route. This is the lowest level.
Decide (Level 1)	The computer does not assist in or perform ranking tasks. Human must do it all.	Again, the human operator is responsible for using the provided data to choose a route. Level 1 is the lowest level, as there is no system intervention.
Act (Level 1)	The human alone can execute decisions.	In this mode the system has no choice in the decisions.

Level 2: Semi-Autonomous Mode 1

The first semi-autonomous mode definitions are described by Table 4-4.

Table 4-4. Definition of SASAR Level 2

Category	Description	Reasoning/Example
Observe (Level 4)	The computer is responsible for gathering the information for the human and for displaying all information, but it highlights the non-prioritised, relevant information for the user.	The system only displays the video feed and will highlight the detected obstacles. This is the first level where any observations are highlighted and shown to the user.
Orient (Level 4)	The computer analyses the data and makes predictions, though the human is responsible for interpretation of the data.	The system detects obstacles and lists them for the controller. The controller determines how appropriate the information is.
Decide (Level 2)	The human performs all ranking tasks, but the computer can be used as a tool for assistance.	The operator looks at the images and highlighted obstacles and uses that information to form a path.
Act (Level 2)	Human is the prime source of execution, with computer shadow for contingencies.	The operator guides the system along the path they have decided upon. The system may choose to act if, for example, it is heading for a dangerous obstacle.

Level 3: Semi-Autonomous Mode 2

The second semi-autonomous mode definitions are described by Table 4-5.

Table 4-5. Definition of SASAR Level 3

Category	Description	Reasoning/Example
Observe (Level 6)	The computer gathers, filters, and prioritises information displayed to the human.	In this mode the controller will still need to see the data as it is being collected, to control the system. Level 6 is the highest level in which the data collected is still shown to the human.
Orient (Level 7)	The computer analyses, predicts, interprets, and integrates data into a result which is only displayed to the human if result fits programmed context (context dependant summaries).	The system uses the collected data to detect the possible paths and obstacles and uses this later. The data still needs to be available to the operator, thus level 7 is chosen as this is the highest level where this happens.
Decide (Level 7)	The computer performs ranking tasks and displays a reduced set of ranked options, without displaying why decisions were made.	The system uses the possible paths and selects the most appropriate. Other options are shown to the operator should they wish to change the action.
Act (Level 6)	Computer executes automatically, informs the human, and allows for override ability after execution. Human is shadow for contingencies.	This means the system will act immediately based on the above decision. However, this decision can be reversed by the user. Acting immediately means there is less downtime of the system while the human approves or disapproves the chosen action.

Level 4: Fully Autonomous

The fully autonomous definitions are described by Table 4-6.

Table 4-6. Definition of SASAR Level 4

Category	Description	Reasoning/Example
Observe (Level 7)	The computer gathers, filters, and prioritises data without displaying any information to the human. However, a "program functioning" flag is displayed.	In this mode the operator does not need to see the system working, but may wish to take control at various points, thus they need to know when it is operating.
Orient (Level 8)	The computer predicts, interprets, and integrates data into a result which is not displayed to the human	The operator does not need to know how or why the system is interpreting data.
Decide (Level 8)	The computer performs ranking tasks. The computer performs final ranking, but does not display results to the human.	As the system is following a pre-set path, it will have to make its own decisions about how to navigate the environment it is presented with
Act (Level 7)	Computer executes automatically, informs the human, and allows for override ability after execution. Human is shadow for contingencies.	As mentioned above, the human may wish to take control, hence level 7. In such a situation they would change the operational mode to a lower level.

4.2.4 Definition of Autonomy Conclusion

The results have come about from researching previous investigations in how to define the autonomy of a system. The result we choose comes mainly from the LoA Assessment created by [96]. While their scale contains no perfect description of SASAR, the individual 'observe, orient decide and act' descriptions have allowed for the formation of four distinct definitions of the autonomy for SASAR.

The four definitions are also an excellent way of measuring the progress of the project, as starting at the manual mode, each mode will increase in complexity compared to the previous. Manual mode will be the first mode implemented, eventually progressing through the lifetime of the project to have all four operational modes. The largest difference between modes will be between

modes 1 and 2, as this will require the implementation of a system able to detect and determine obstacles.

4.3 SASAR 2 Control

4.3.1 2D Model Structure

Initially we present the model of a generic snake-like robot in a 2D plane. As discussed in Chapter 3, the next iteration of SASAR will be an actively driven, Serpentine robot, thus we must include the drive wheels.

We model the serpentine robot as a chain of n linkages connected by $n - 1$ joints, as shown in Figure 4.4. The first linkage is signified as the front of the robot (the furthest from the tail). Note, we separate the description of linkage and module, as our modular interfaces are distinct from a link and pivot description. Each link has a pair of wheels, driven on an axis which passes perpendicularly through the link. For simplicity we assume this intersection is also the centre of gravity of the linkage and note it as P_i , which has coordinates (x_{p_i}, y_{p_i}) in the global coordinate frame. P_i is a distance l_{f_i} from the joint at the front of the module (termed J_i) and a distance l_{r_i} from the joint at the rear of the module (J_{i+1}). The total length (l_i) of the module is therefore the sum of the lengths l_{f_i} and l_{r_i} . Finally, we assume both wheels act simultaneously at point P_i , and have a radius l_w .

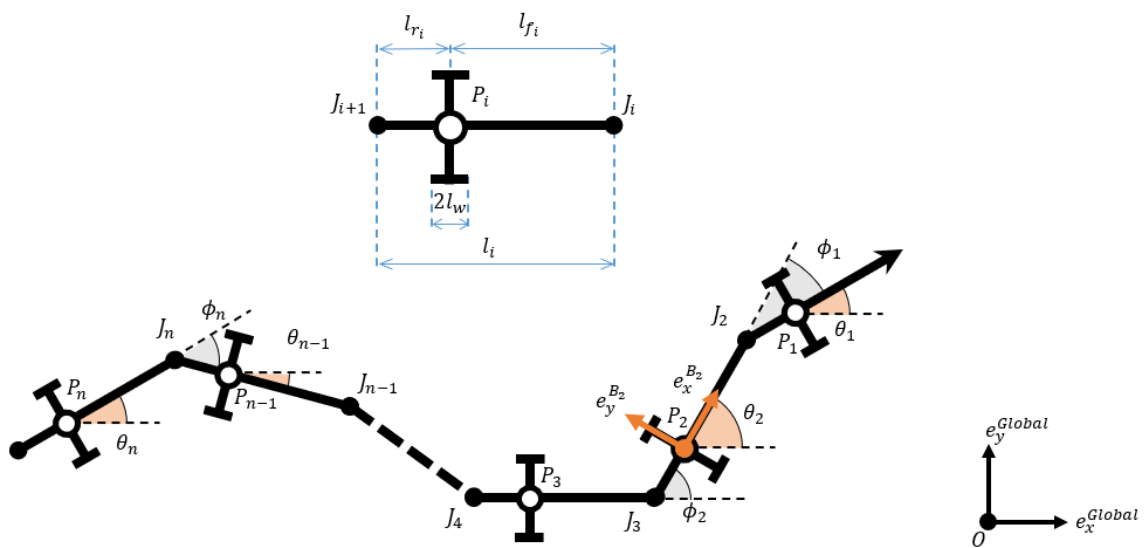


Figure 4.4. 2D Serpentine Robot Model

We next denote a global coordinate frame, with origin O , and axes $e_{x/y}^{global}$. The absolute position of each module is defined as the location of the centre of mass (P_i) relative to the global coordinate frame. A body coordinate frame is inserted with the origin at this point, with the x axis ($e_x^{B_i}$) running colinearly to the linkage, and y axis ($e_y^{B_i}$) perpendicularly to the linkage. The orientation of each linkage is measured as the linkage angle, θ_i , and is the angle from the global x axis to the body x axis. For all angles, the right-hand rule applies, meaning positive is in the anti-clockwise direction.

The final component required to define the pose of the linkage is the rotational position of the drive wheel. We label this θ_{w_i} . The pose, and thus velocities, of a linkage can be defined as $\mathbf{q}_i$ and $\mathbf{u}_i$ respectively.

$$\mathbf{q}_i = \begin{bmatrix} x_{p_i} \\ y_{p_i} \\ \theta_i \\ \theta_{w_i} \end{bmatrix} \quad \mathbf{u}_i = \begin{bmatrix} \dot{x}_{p_i} \\ \dot{y}_{p_i} \\ \dot{\theta}_i \\ \dot{\theta}_{w_i} \end{bmatrix} = \begin{bmatrix} \dot{x}_{p_i} \\ \dot{y}_{p_i} \\ \omega_i \\ \omega_{w_i} \end{bmatrix} \quad (4-1)$$

The pose and velocity vectors for an entire robot are then collated:

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \vdots \\ \mathbf{q}_n \end{bmatrix} \quad \mathbf{u}_i = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_n \end{bmatrix} \quad (4-2)$$

Therefore, we can describe the motion of any actively driven snake like robot.

4.3.2 Methods of Control for SASAR 2

The control of a snake robot can be separated into two aspects which must be controlled: the overall direction of the robot (heading and velocity) and the positioning of the robot body. Of course, these two interact, and typically the positioning of the body is crucial to controlling the direction of motion of the robot. We will focus on the heading control of SASAR later, but to begin with we look at the control of the body position.

Follow-the-Leader (FTL) and N-Trailer are two methods which have been used to control the body position of an actively driven serpentine robot. For both cases,

the task is simple: take a desired heading and velocity for the robot and calculate where the body of the robot should be in order to achieve the desired position.

FLT is the most commonly used method. It is used by Hirose in the ACM-R4.2 [7], MAKRO [65] and later T2 Snake-4 [97] along with many others. This method uses a process of shape fitting first proposed by Yamada et al, [98], with the key controllable aspect being the desired curvature of the robot. Once the curvature is selected by the operator, each module then attempts to follow the path of the module in front of it, hence the name, *“Follow-the-Leader.”*

However, an alternative method was developed by Murugendran et al, [22]. This method is based on the N-trailer system, a study of a single, powered vehicle, pulling an arbitrary number of articulated trailers (see [99,100]). Typically, this system is used to determine the positions and joint angles of the following trailers giving an input steering angle to the truck pulling them. However, Murugendran et al invert the problem, creating an algorithm which can iteratively determine the required trailer joint angles to result in the desired heading of the vehicle, even if there are no steering wheels. This results in a system into which the operator only needs to submit a desired heading angle, (δ_1) and velocity (v_{p1}), and the new joint angles and velocities for all following modules are calculated to achieve this motion. Murugendran et al showed when compared to FTL, this N-trailer method produces motion with significantly reduced cross-track error (XTE), keeping the body of the snake-like robot closer to the desired path. We therefore choose a modified version of the N-trailer method for SASAR, based on this research. Additionally, it is show that this approach reduces the required torque demanded by the linkage joints and wheels when compared to FTL. This could show a significant real-world benefit in the reduction of Power consumed by a robot.

4.3.3 N-Trailer for SASAR

The N-trailer method is a recursive kinematic model, which uses the current heading and speed of a linkage to determine the heading and speed of the next module, and so on. We utilise several different and new variables for this model, as shown in Figure 4.5, compared to the generic snake-like robot model. The

method we use here is similar to the method utilised by Murugendran et al, with one key difference. Due to the quaternion mechanisms, the linkage lengths are not constant and are replaced with the variables $L_{f,i}$ and $L_{r,i}$. These are functions of the linkage joint angle ϕ_i , the nominal quaternion radius, r , and the linkage length from the axle to the forward or rear quaternion mechanism ($L_{f/r}$).

$$L_{f,i} = L_f + \frac{r}{\cos(\phi_i)} \quad (4-3)$$

$$L_{r,i+1} = L_f + \frac{r}{\cos(\phi_i)}$$

For $i = 1 \dots n - 1$.

We also establish a new state vector and state vector derivative:

$$\mathbf{Q} = [x_{p1} \quad y_{p1} \quad \theta_1 \quad \phi_2 \quad \dots \quad \phi_n] \quad (4-4)$$

$$\dot{\mathbf{Q}} = [\dot{x}_{p1} \quad \dot{y}_{p1} \quad \dot{\theta}_1 \quad \omega_2 \quad \dots \quad \omega_n]$$

Where x_{p1} , y_{p1} , θ_1 is the pose of the head module, and ϕ_i is the joint angle for $i = 2 \dots n$. By knowing these values, the entire pose of the snake robot can be found using the simple relationship:

$$\phi_i = \theta_i - \theta_{i-1} \quad (4-5)$$

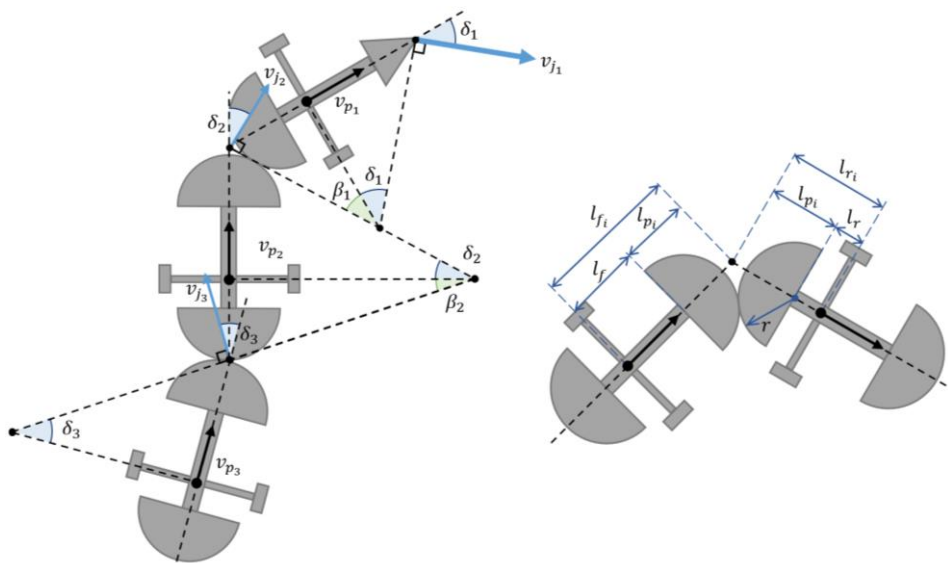


Figure 4.5. Schematic of SASAR for Use with N-Trailer Variables.

The aim of the model is to determine the required angular velocities of the module drive wheels, and angular velocities of the linkage joints, in order to achieve the heading and velocity desired by the controller. We therefore introduce a control vector:

$$\mathbf{U} = [\omega_{w,1} \quad \cdots \quad \omega_{w,n} \quad \dot{\phi}_2 \quad \cdots \quad \dot{\phi}_n] \quad (4-6)$$

Where $\omega_{w,i}$ is the rotational speed of the wheels on linkage i . Therefore, the speed of each link can be calculated as:

$$v_{P,i} = L_w * \omega_{w,i} \quad (4-7)$$

We begin the kinematic modelling with the geometric relationships between the velocities of connected links. Figure 4.5 displays the angles δ_i , β_i and ϕ_i , and the following relationships can be found geometrically:

$$-\frac{L_{r,i}}{\tan(\beta_i)} = \frac{L_{f,i}}{\tan(\delta_i)}$$

Therefore:

$$\beta_i = \tan^{-1} \left(-\frac{L_{r,i}}{L_{f,i}} \tan(\delta_i) \right) \quad (4-8)$$

$$\delta_i = \beta_{i-1} - \phi_i \quad (4-9)$$

$$v_{P,i} = v_{P,i-1} \frac{\cos(\delta_i)}{\cos(\beta_{i-1})} \quad (4-10)$$

We can therefore use equations (4-8) to (4-10), iteratively solving for $i = 2 \dots n$, using the known state vector, $\mathbf{Q}$, plus the two input variables, δ_1 and v_{P1} . This gives us the first half of the control vector.

To find the joint angle velocities, and complete the control vector, we begin by defining the velocities of a linkage:

$$\begin{aligned} \dot{x}_{P,i} &= v_{P,i} \cos(\theta_i) & \dot{y}_{P,i} &= v_{P,i} \sin(\theta_i) \\ \dot{\theta}_i &= \frac{v_{P,1}}{L_{f,i}} \tan(\delta_i) \end{aligned} \quad (4-11)$$

Next, we introduce the constraints, beginning with rigid-body constraints. These simply describe the position of the centre of a link with respect to the previous linkage position.

$$\begin{aligned}x_{P,i+1} &= x_{P,i} - L_{r,i} \cos(\theta_i) - L_{f,i-1} \cos(\theta_{i-1}) \\y_{P,i+1} &= y_{P,i} - L_{r,i} \sin(\theta_i) - L_{f,i-1} \sin(\theta_{i-1})\end{aligned}\tag{4-12}$$

The final constraint is the non-holonomic, zero-slip constraint. This simply states that there is no movement in the direction perpendicular to a linkage forwards motion.

$$\dot{x}_{P,i} \sin(\theta_i) - \dot{y}_{P,i} \cos(\theta_i) = 0\tag{4-13}$$

By using equations (4-11), (4-12) and (4-13), along with (4-5), we can calculate the angular velocities for the linkage joints.

$$\dot{\phi}_i = \frac{v_{Pi-1}}{L_{f,i-1}} \left[\sin(\phi_i) + \left(\frac{L_{r,i}}{L_{f,i-1}} \cos(\phi_i) + 1 \right) \tan(\delta_{i-1}) \right]\tag{4-14}$$

Again, solving iteratively allows us to complete the control vector. Note the state vector, and the calculated velocities are required to solve (4-14) for all $i = 2 \dots n$.

For the purposes of simulation, we estimate a new state vector using:

$$\mathbf{Q}_{new} = \mathbf{Q}_{Previous} + \dot{\mathbf{Q}}_{previous} * dt\tag{4-15}$$

Where dt is a known time-step. Note, the state vector derivative, $\dot{\mathbf{Q}}$, can be found using equations (4-11) and (4-14).

4.4 Modelling, Simulation and Analysis

4.4.1 Kinematic Model

In this section we discuss the modelling and simulation of our kinematic model. The structure of the system is a simple input-controller-plant system with a feedback loop. Figure 4.6 shows the basic layout of the system, produced in Matlab Simulink.

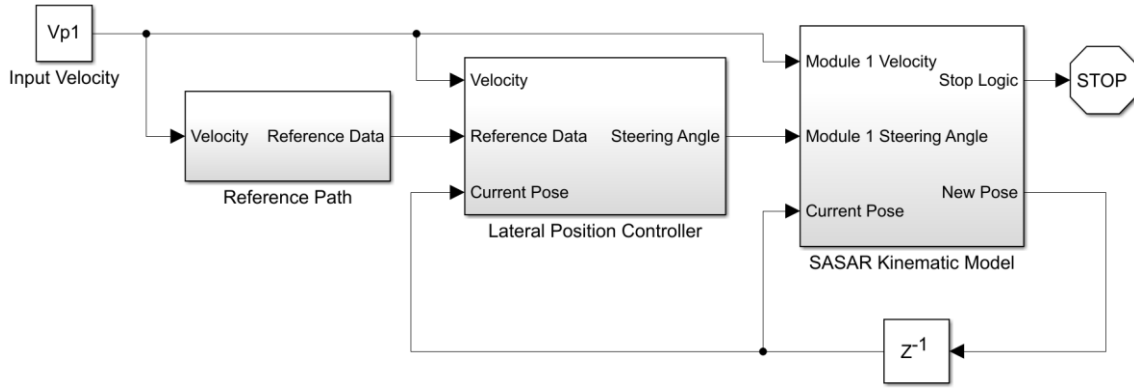


Figure 4.6. Kinematic Model System

Input: Velocity and Reference Path

There are two inputs into the system: A chosen velocity and a reference path. The path is simply a list of module poses, defined in terms of the distance the model has travelled. The pose is the desired head module position and orientation, $\mathbf{q}_1 = [x_{P,1} \ y_{P,1} \ \theta_1]$. By integrating the velocity of the head module, we can determine the current desired pose. For the purpose of our testing, we generate the path as a parametric equation. The final output of this Input subsystem is the direction value, either 1 or -1 depending on if the motion is forwards or reverse. This is required for the controller subsystem.

Controller: Stanley Lateral Position Controller

For a path following controller, we use the Stanley lateral position controller. First shown by Hoffman et al, [101], this controller is typically used for autonomous vehicles, but is well suited to the SASAR system, as it outputs a steering angle, which is needed for our kinematic model. A simple version of the Stanley control law can be described by the equation:

$$\delta(t) = \theta_e(t) + \tan^{-1} \left(\frac{\kappa e_{XT}(t)}{v_{p1}} \right) \quad (4-16)$$

Where θ_e is the orientation error, or the difference between the current linkage angle, and the desired pose angle, and e_{XT} is the cross-track error (XTE). Finally, κ is a gain parameter to be found and tuned. It is clear from inspecting equation

(4-16) that as the XTE increases, so to is the output steering angle, turning the vehicle more towards the path.

The XTE and orientation error can be found by comparing the current pose with the desired pose, as supplied by Input subsystem. The velocity of the front module is also required, as a faster module will require a less severe turning angle. The final input into the Stanley controller is a direction, as mentioned above. This due to the Stanley control law changing when operating in reverse, to account for the differences caused by the steering pair of wheels being at the rear of the moving vehicle. See [101–103] for more detail.

Plant: SASAR Kinematic Model

The final major component of our simulation is the kinematic model, the model for which is shown in Figure 4.7. This uses the equations presented in section 4.3.3 to calculate the new pose of the robot, considering the input steering angle from the Controller subsystem, plus the Input forward velocity. We initially calculate the new linkage lengths, before finding the δ_i and V_{pi} vectors for the current time step. The control vector can therefore be assembled, and thus find the new pose. Finally, we include logic to end the simulation when the maximum distance is reached.

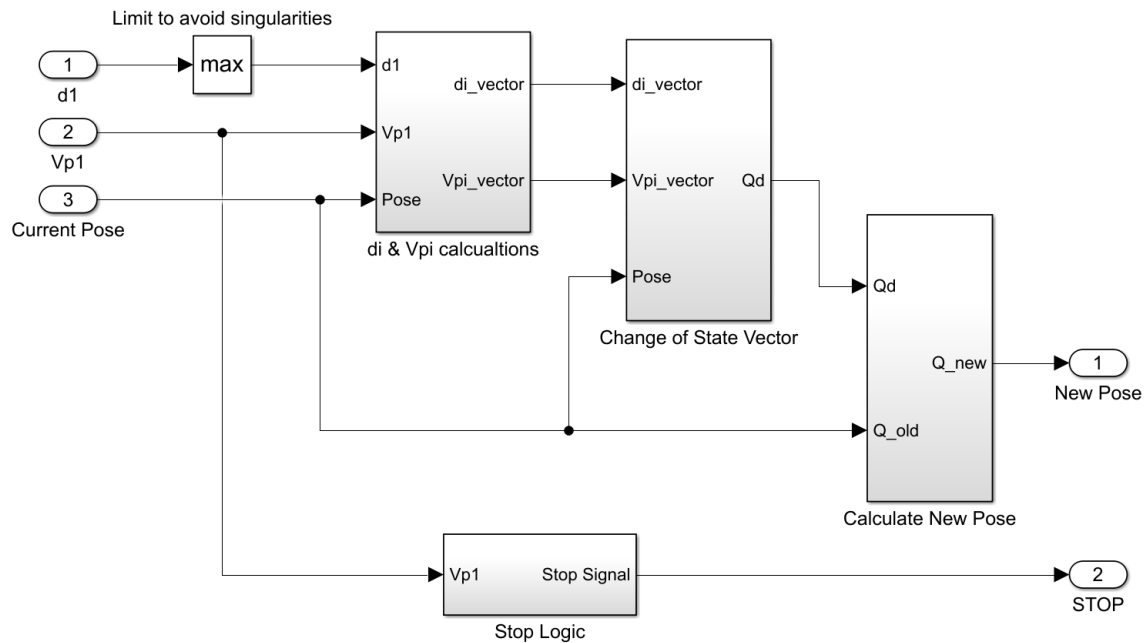


Figure 4.7. Kinematic Model Subsystem Architecture

4.4.2 Simulation Results

Simulation Parameters

For the purpose of these simulations, we attempt to keep the model as close to the dimensions of SASAR as possible. The robot is modelled with 5 modules ($n = 5$), with the lengths $L_f = L_r = 0.058 \text{ m}$ and $L_q = 0.02 \text{ m}$. SASAR moves at a constant speed of 0.2 m/s , and has a maximum joint angle of $\frac{\pi}{2}$. For modelling the N-Trailer version we assume $L_{PJ} = L_{JP} = L_{f/r} + L_q = 0.078 \text{ m}$.

Initial Path Following Simulation

The simulation is tested in two manners. Initially we simulate the motion of the robot following a complex path of multiple turns, beginning with an initial position offset. This serves both to test the kinematic model, and to ensure the Stanley controller is compatible with the system. To that end, we set the path to conform to the parametric equation $x(t) = \sin(2t) + 3 \sin(t)$, $y(t) = 2 \sin(3t)$. We also begin with the initial head pose of $\mathbf{q}_1 = [1 \ 0 \ \frac{\pi}{2}]$. The model completed successfully, conforming to, and following the path closely, as shown in Figure 4.8.

Cross Track Error Comparison

One of the benefits of the N-Trailer model is the reduction in total cross-track error. To ensure our results are consistent, we compare the total XTE of the linkage for both an N-trailer model, and our SASAR version of it. This means the sum of the XTE of each linkage to the intended path. The results, displayed in Figure 4.9, make an interesting discovery. The SASAR modifications, including a quaternion mechanism rather than a fixed pivot, have a consistently lower XTE, albeit minor. The average for each linkage, across the entire robot at time period is only 1.05 cm .

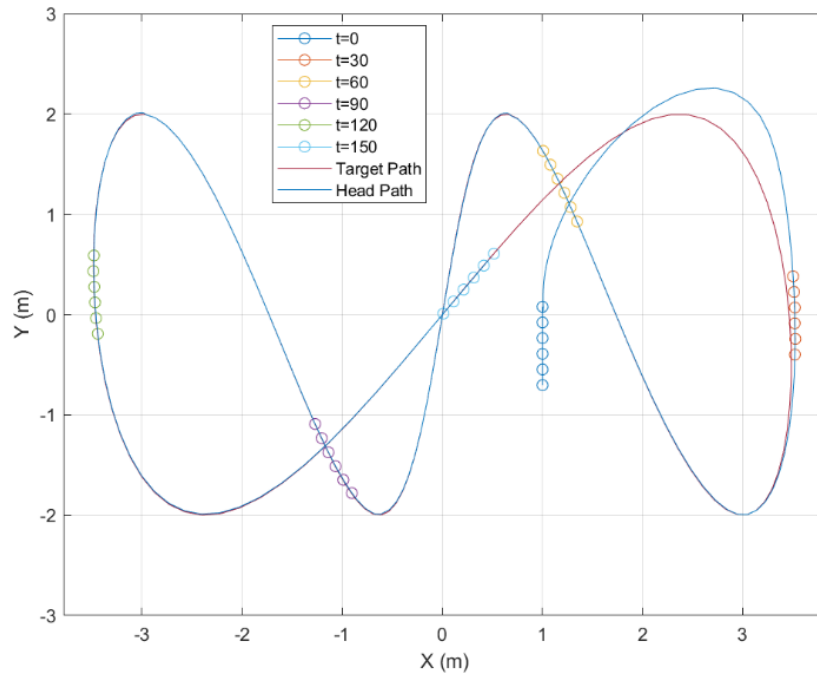


Figure 4.8. Primary SASAR Path Following Results, Showing the position of the head module relative to the desired path, plus the position of the robot at 30 second intervals.

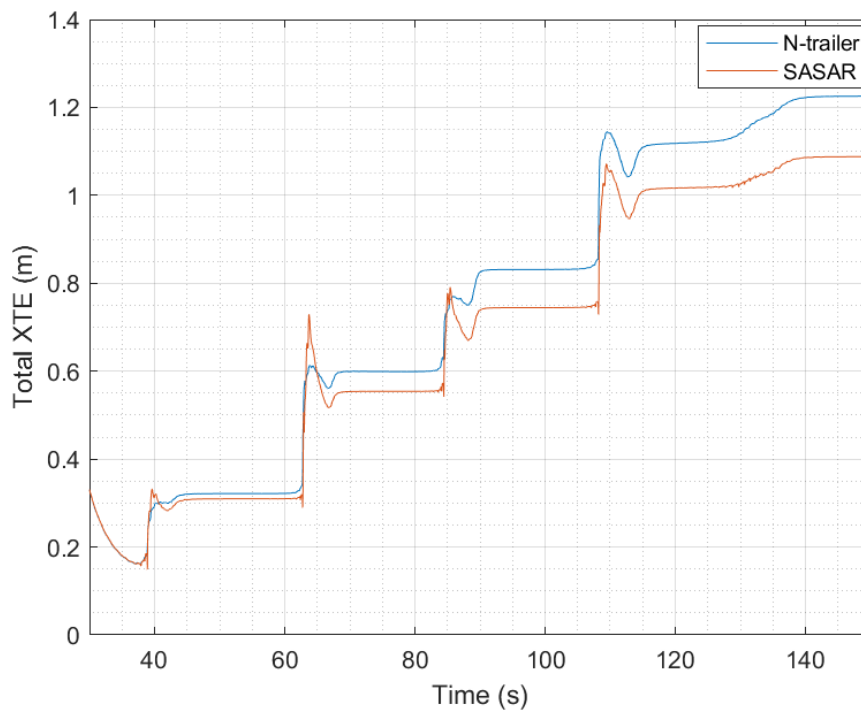


Figure 4.9. Graph showing the total cross-track error of the N-Trailer Model Vs. the SASAR version. Note: only the results for $t \geq 30$ are shown.

4.4.3 Simulation Discussion

The simulations performed show the changes implemented into the kinematic model will be suitable for our new robot design. The path-following experiment has shown that a simple lateral position controller can be introduced with relative ease, which will aid the future task of implementing autonomous navigation to the system. However, these simulations do not test how the robot will be to operate manually. To aid with this a secondary simulation was created in a 3D environment, utilising the Webots robotics simulator [104]. This software allows for the creation of 3D environments for the purpose of simulating robotics motion. The kinematic model was thus translated into the software, and a simple environment created the test SASAR. Using this tool, we are able to operate the robot using a handheld games console controller to input the desired velocity and heading. A simulated camera is positioned at the front of the robot, and thus it is possible to navigate through the test environment.

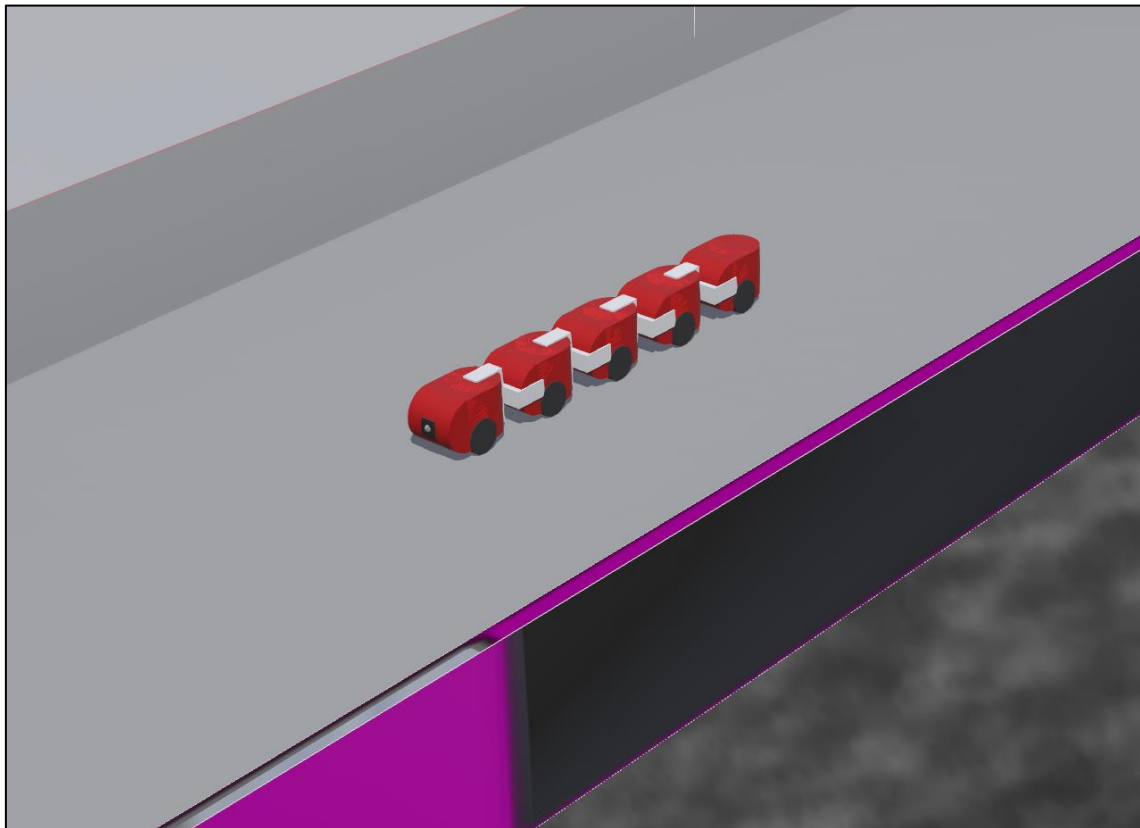


Figure 4.10. Simulated Robot in the Webots Simulation Environment

When controlling the simulation in this way there are a number of observations which were made. Firstly, the control of the robot is very simple and intuitive. Reducing the required inputs to a heading and velocity makes the robot easy to utilise. Secondly, there is a clear and noticeable lateral slip of the linkage wheels. Despite this, the controller works excellently, and the robot can be very easily piloted, even from a first-person perspective.

4.5 Summary

In this chapter we have presented a number of tools for designing our controller for an actively driven, snake-like robot. Firstly, we have created a list of four operational modes for our robot, each with increasing levels of autonomy. These will serve as a guide and a target for future projects utilising this platform. We have also described the motion of a generic serpentine robot in terms of a state and control vector.

Finally, we produced and tested a control method for our new snake-like robot. The control algorithm is based on a kinematic model and simplifies the high number of controllable degrees-of-freedom within a serpentine robot to only two inputs. This algorithm was shown to be suitable for both manual controls, by simulating an operator driving through a 3D virtual environment, and for autonomous control, by including a lateral position controller and simulating a path-following exercise. The modifications made to adapt the kinematic model for use with a Quaternion mechanism have been shown to have little impact on the overall function of the robot, and in some respects improve the performance.

5 SASAR 2: DESIGN AND IMPLEMENTATION

5.1 Introduction

Here we present our final prototype serpentine robot, SASAR 2. The modular design of SASAR 2 is comprised of four modular types: the main Joint Module, the Drive Module, a Head Module and finally a Tail module. We display the robot as a whole, plus exhibit each module independently. The construction of each module is a combination of 3D printed structures and commercially available motors powered by a bespoke printed-circuit board.

We then analyse the function of the robot. Initially, we focus on the performance of the Joint Module in terms of speed, accuracy, and power. We present our testing methodology and compare the results to similar snake-like robots and hyper-redundant manipulators.

5.2 SASAR 2

SASAR 2 is a modular, serpentine robot design. The design features a universal male-female modular interface, with a robust CAN-bus communication system. In its basic configuration, SASAR 2 features a Head module to provide positional feedback to the operator, a number of Joint Modules to actuate the spine of the robot, which are separated by a number of Drive Modules, with actively driven wheels to provide the lateral motion of the robot. Finally, SASAR has a Tail module, simply converting the modular interface into a generic RS-232 socket, for connecting the robot to a central control computer for power and data.

As a CAN bus is used as the communications interface, any compatible CAN enabled device can be used to control the robot. For the central control computer, we use a Nvidia Jetson Nano [105]. The Nano is specifically engineered for use with machine learning tasks, allowing for simpler future implementation of new “AI (Artificial Intelligence)” technologies such as image detection. Additionally, it functions as a traditional Linux-based PC, and thus has compatibility with the Innomaker CAN bus adapter selected [106].

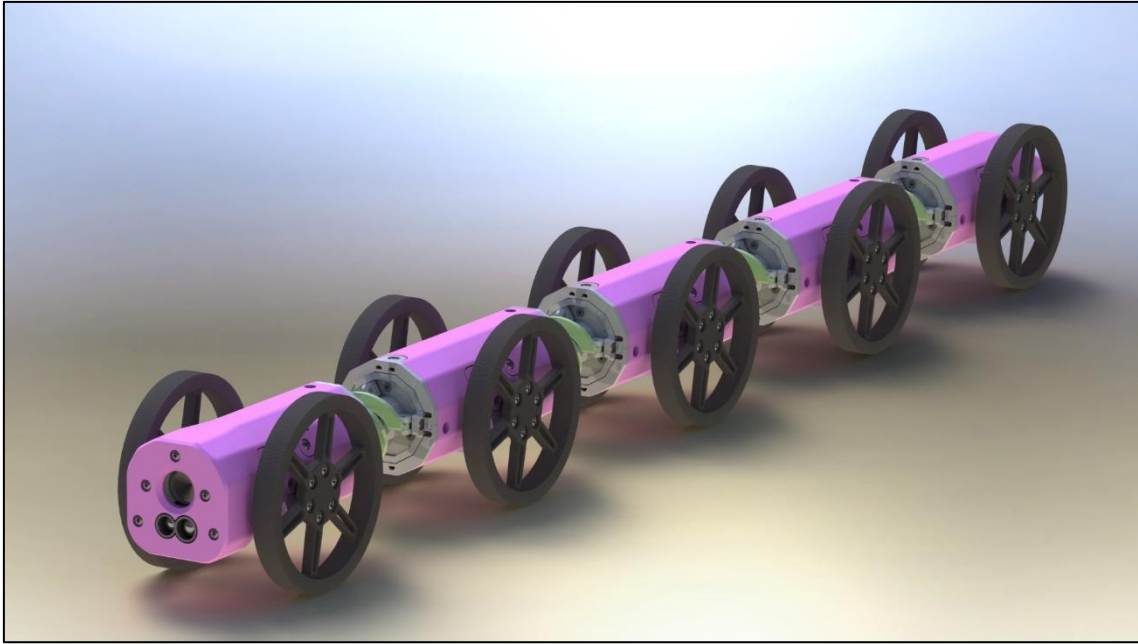


Figure 5.1. Render of SASAR 2

5.3 Joint Module

5.3.1 Design Overview

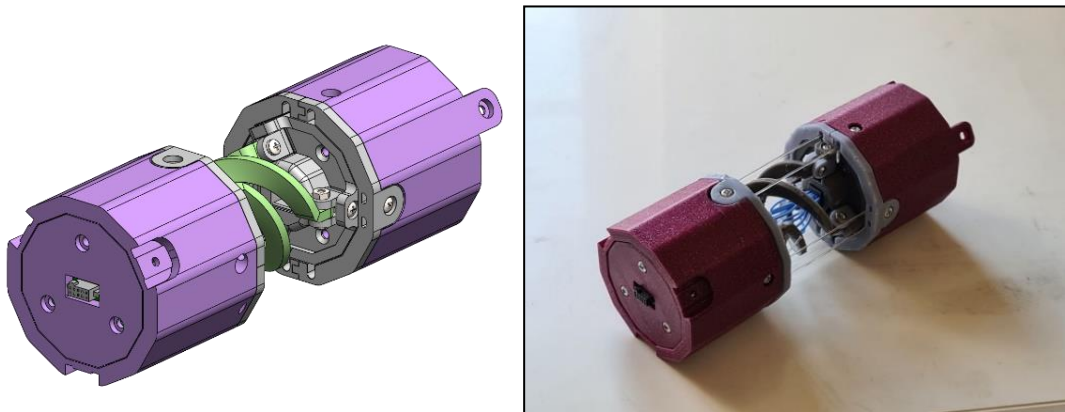


Figure 5.2: Joint Module CAD Model and Prototype

Here we introduce the SASAR 2 Joint Module (Figure 5.2). The design is split into three distinct sections, the Quaternion mechanism in the centre and a housing unit on either side, one for the pitch angle control, and one for the yaw. The two housings contain the main electrical and mechanical components, with the Quaternion mechanism used to give the joint structure, utilising its unique motion properties. To connect to neighbouring modules there is a 10-pin electrical connector and three physical locks in a male-female configuration. The two

housings of each module are connected by a silicone wire loom which runs through the centre of the quaternion mechanism. The nature of the joint means there is always a passageway through the centre, which stays a reasonably constant width and length, although it does curve.

One of the main aims of this design was modularity. As such, an effort was made to ensure most components could be easily replaced or upgraded if necessary. For example, the tensile wires and other drive components do not directly interfere with the Quaternion mechanism. Therefore, if a different mechanism design were required, this could easily be replaced.

Following on from the modularity, all the non-COTS components of the Joint Module are 3D-printed. This both reduces the cost of the module and allows for rapid replacement of defective or damaged components. The only exception to this is the bespoke PCBs (printed circuit boards). Additionally, no permanent bonding methods are used to secure any mechanical components. A combination of captive nuts, brass threaded thermoplastic inserts, and steel dowel pins are used. Therefore, each module can be easily reduced back to its component parts if required.

The new joint module has between 16.3% and 60.5% more output torque than SASAR 1 and double the operational range with both axes overlapping as desired. However, this comes at the cost of additional size, around a 220% increase in volume, a 200% increase in mass, and a huge speed deficit. SASAR 2 is more than six-times slower to respond. These properties are summarised in Table 5-1.

Table 5-1. SASAR 2 Joint Module Properties Comparison

Property	SASAR 2	SASAR 1
Degrees of Freedom	2, [Joint Angle, Bending Plane]	2, [Pitch, Yaw]
Module Size ($\emptyset$ x L)	48mm x 127.4mm	35mm x 75mm
Module Mass	150.0 g	50.4 g
Output Torque	0.53-0.84 Nm	0.45 Nm
Operational Range	± 90 degrees	± 45 degrees
Rotational Speed	59 Degrees per Second (1.03 rad/s)	375 Degrees per Second (6.54 rad/s)

5.3.2 Electrical Architecture and Component Selection

In the simplest of terms, the object of each module is to take an incoming command and move the joint to the desired position. As such, the two most crucial components to select are the motor, and the communication interface. It was chosen to use a CAN interface for the inter-module communication due to its reliability over longer connections. The I2C interface is then used for intra-module connections, mainly due to its ease of use, plus the compatibility with the other components selected. Both the CAN bus and I2C bus require only two data lines to communicate, reducing the complexity of the design.

A combination DC electric motor and gearbox is selected as the main actuator for the robot, which is used in combination with a magnetic position sensor. The Walfront motor was chosen due to its small size, compact 'back-geared' output shaft orientation. This allows for the motor terminals to be on the same surface as the magnetic position sensor, reducing the complexity required for the PCB. Various alternative solutions were trialled, including reconfiguring the Savox SH-0256 motors used in SASAR 1. However, even with the larger size restrictions, it was not possible to construct a suitable solution, thus the Walfront motors are used.

While data of the motor is hard to come by, a torque output of roughly 0.1 Nm was found using a simple lifting test. This test was performed by lifting a set of known masses using a drum of known radius. The mass was increased slowly until the motor stalled. By knowing this mass, a torque value is calculated using the drum radius and the level-arm distance. This test was then performed for a range of drum sizes, and an average taken as the motor torque.

Utilising equation (3-4), a required torque of 0.61 Nm was found. This is to lift three modules given the mass and length properties of SASAR 2. Therefore, using equations (3-11) and (3-12) we determine a maximum tensile load of 39.4 N (Figure 5.3). For the chosen motors this would require a drum of maximum radius 5.1 mm , when a single pulley loop is used. In the final SASAR design we use a drum of radius 4.5 mm , which actuate a quaternion joint to a maximum 0.61 Nm and would therefore require a motor torque of 0.068 Nm , given the physical parameters of SASAR.

This measurement also allows us to specify the tensile cable used, which we use 0.7 mm diameter Nylon wire, rated to 20 kg .

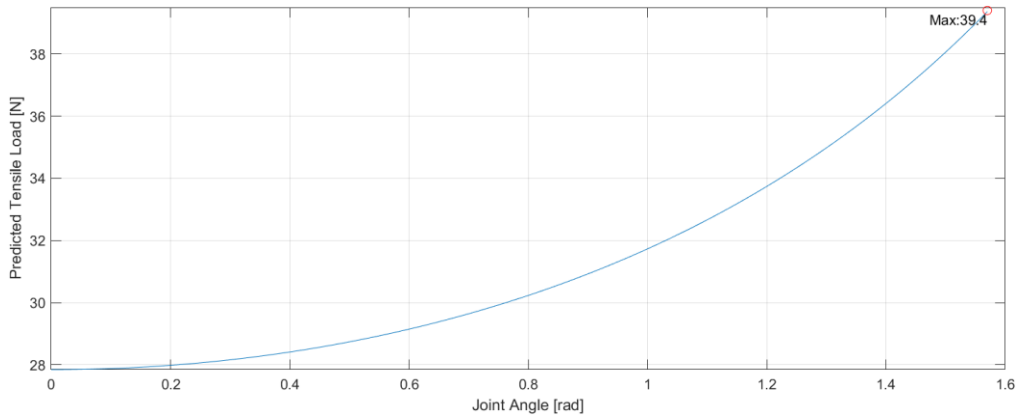


Figure 5.3. Predicted Tensile Load of chosen Geometry

The motors are driven by a DRV8830 motor control unit (MCU), and positionally controlled by an AS5048-B Hall-effect magnetic position sensor (MPS). An MPU-6050 inertial measurement unit (IMU) is included in one of the housings for additional positional feedback. The MCUs, MPSs and IMU are all connected to an ATMEGA32 microprocessor using the I2C interface. The MCP2515 CAN

controller and TJA105T transducer are then connected using the serial peripheral interface (SPI).

To reduce cost, each module is designed to use only a single microprocessor, CAN interface and IMU. To achieve this, we pass the I2C connection through the intra-module wire loom to the secondary housing, connecting the microcontroller from the primary housing to the remaining MCU and MPS in the secondary housing. Additionally, there is the inclusion of four pass-through lanes, which are uninterrupted by the module. These allow for future expandability, and the connection of additional sensors if required. Finally, a universal serial bus (USB) connection is included, to directly connect to the microprocessor. This allows for the firmware of the module to be updated if required. Figure 5.4 shows a basic schematic of the electrical connections within the Joint Module.

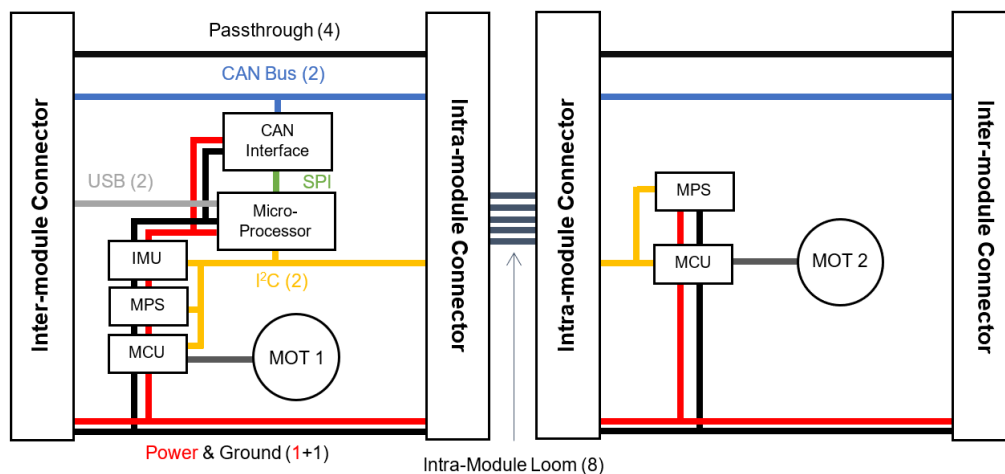


Figure 5.4: Basic Electrical Architecture (Lanes required)

Physically, the components are all mounted to a set of bespoke printed circuit boards (PCBs). Each housing contains three PCB's, a main board containing the component, plus two auxiliary boards for the modular connectors (Figure 5.5). The PCBs are designed in such a way that only three individual designs are required. These are: a main board, an *intra*-module connector board, and a male and female *inter*-module connector. Three solder tabs on the main board are used as hard switches to define if the board is the primary, containing the microcontroller, or the secondary, with only a single motor controller and position sensor. Finally, large contact pads are fitted to the top of the main PCB, for connecting the motor terminals. These pads are then linked to the two captive

nuts which are used for bolting the face plate onto the housings. This allows for fine adjustment of the drum position without powering the whole module, by applying a current to the fixings.

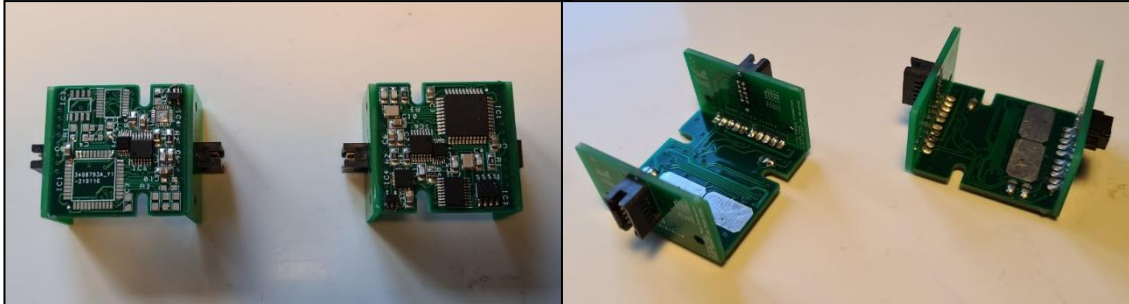


Figure 5.5: SASAR Joint Module PCB's

5.3.3 Mechanical Design

Housings

The housings are the main structure of the module, encompassing and protecting the robotic components. The main mechanical component in each housing is the drum, which actuates the mechanism when driven by the motor. The tensile wire is wound, counter-rotating, around each half of the drum. The wire then exits the housing and is looped around a miniature bearing in the opposite housing collar, before being anchored in the initial housing. Figure 5.6 shows the path of one half of one of these wires.

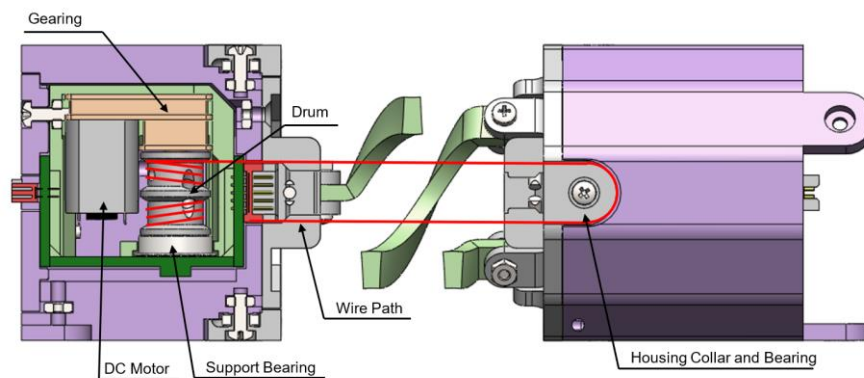


Figure 5.6: Joint Module Partial X-Section showing tensile wire path.

Mechanically, the housing structure follows the modular approach of the rest of the design. The outer housing (in purple) is a tube-like design, which contains the component sled (green), to which all the major components are attached (Figure

5.7). Should a new drivetrain or drum design be required, the sled can be simply removed, and the new components inserted in its place.

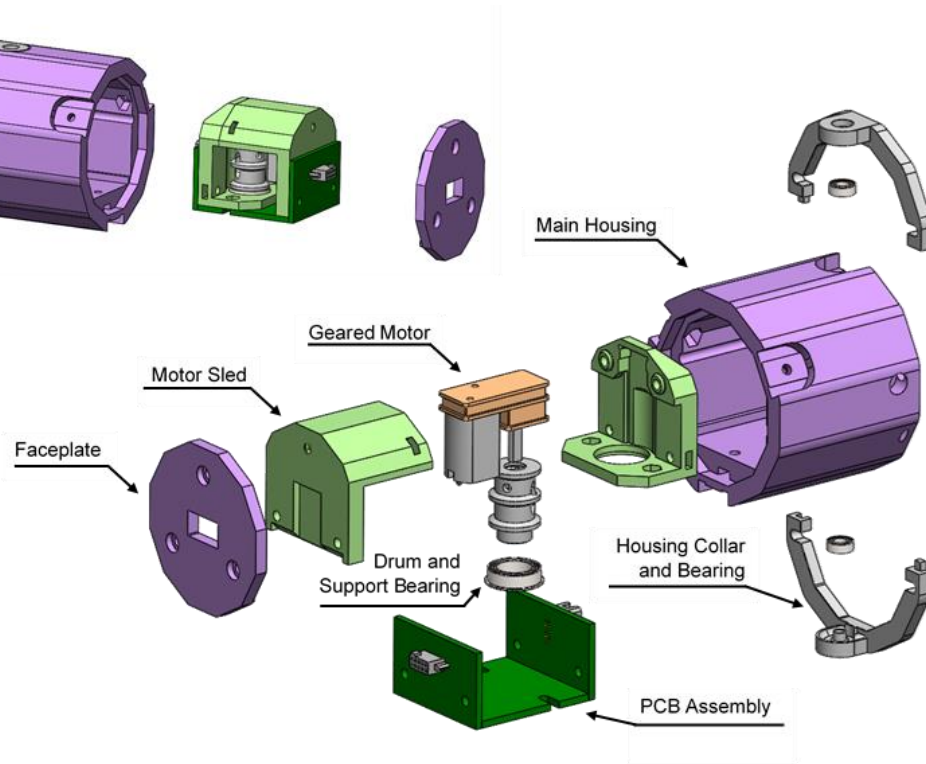


Figure 5.7: Housing Assembly Process

The drum is supported by a bearing, fixed in place into the component sled. This is used both to give the drum additional mechanical support when the joint is loaded, and to centre the Drum over the magnetic position sensor on the PCB. Before final assembly, a small magnet is then inserted into the underside of the drum.

The Main Housing, Faceplate and two Motor Sled components are manufactured using FDM 3D Printing, using PLA, as in SASAR 1. PLA is used due to its ease of printing, and relatively excellent mechanical properties. The main downside of PLA is the low glass transition temperature, which can see it become soft around 60°. Fortunately, SASAR 2 is unlikely to operate in this temperature range.

We utilise the nature of 3D printing to adjust the material properties of each component where necessary. For example, the connection interfaces on the Main Housings are printed with increased perimeter layers compared to the rest of the component. This increases the robustness at the cost of additional material and

printing time. For the Motor Sled, we are able to pause the print at certain intervals, allowing for captive fixings to be inserted. The print is then resumed, and the nut encapsulated within the part, granting additional strength when compared to threading the plastic directly.

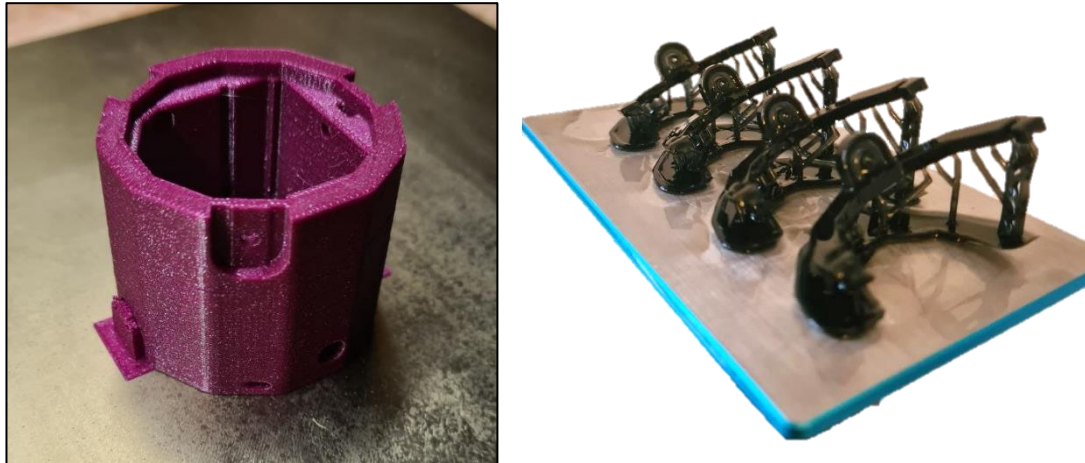


Figure 5.8. 3D Printed Main Housing and SLA Printed Housing Collars Ready for Post-Processing

The final mechanical components are the Housing Collar and Drum. Both of these were initially designed for FDM printing, but the small details were found to be difficult to recreate. Both parts are, therefore, manufactured using resin-based SLA 3D printing. This method is more expensive than FDM, requiring more post-processing to clean the excess resin, but can recreate much finer details.

Quaternion Mechanism

The mechanism design has three major components. They are the three main limbs, six smaller joints, and two end plates. Each corresponds to one of the three key geometric values, for which we select values of $h_c = 30$, $h_0 = 5$ and $w_c = 25$. This combination of values gives an initial joint radius of 20 mm, and the variance is only 0.9 mm over a range of $0 \leq \phi \leq \frac{\pi}{2}$ (Figure 5.9). These three values form the basis of the design.

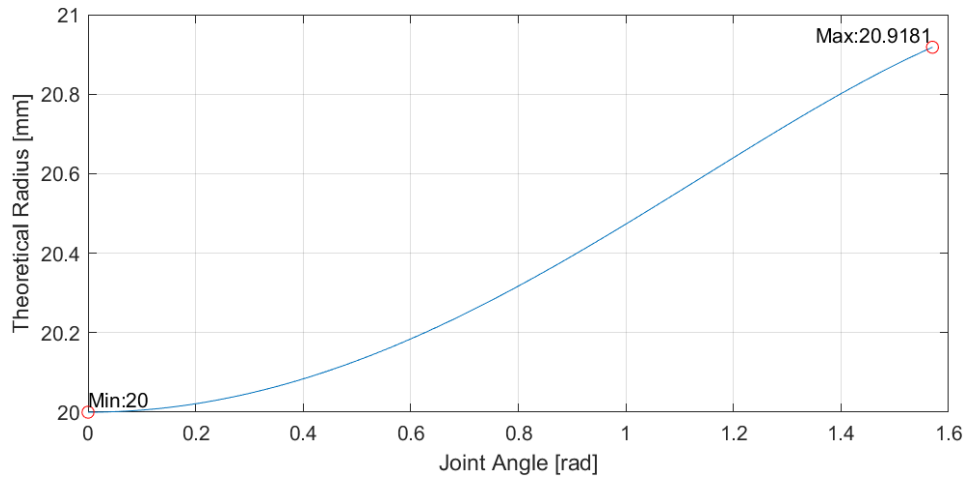


Figure 5.9. Predicted Performance of selected Quaternion Geometry

One of the main issues with the initial designs was the interference between the three main limbs of the joint at extreme joint angles. This is due to the low aspect ratio of the joint in comparison to the LIMS2 quaternion design, with the SASAR joint being much shorter in comparison to its relative width. To counter this, the limbs are designed to follow a spiral pattern with the thickness of the limb reduced. Numerous iterations of design were tested, adjusting various sizes until a suitable mechanism was created (Figure 5.11).

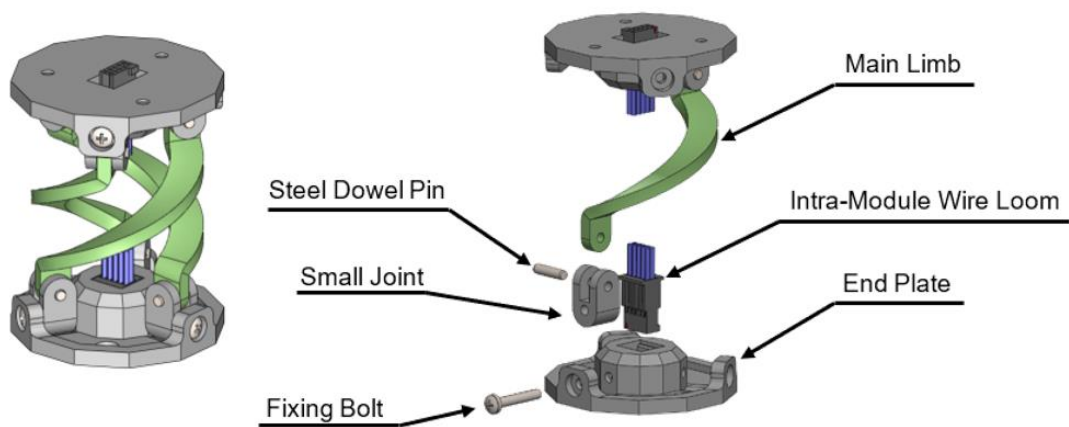


Figure 5.10. SASAR Quaternion Mechanism CAD Model showing Assembly

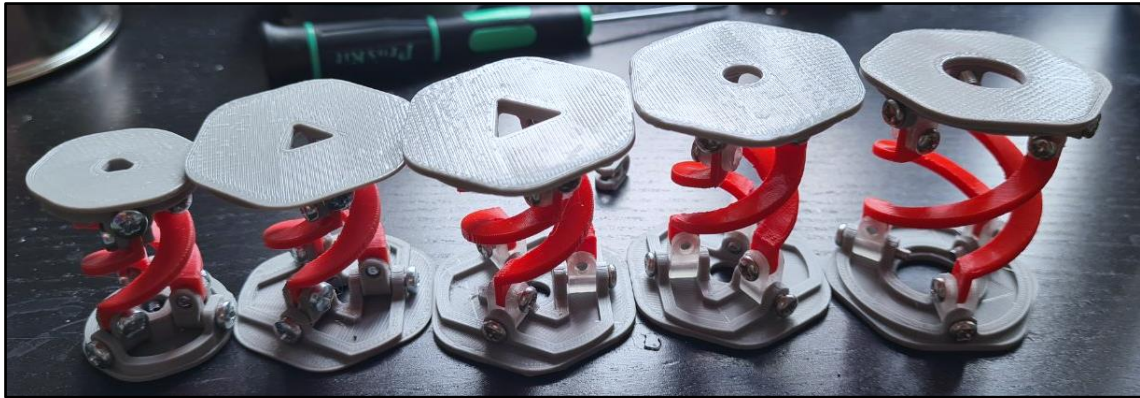


Figure 5.11. Iterations of SASAR Quaternion Joint [Bottom]

This allows the joint to move freely throughout all required angles. This new shape is also practical for manufacture using 3D printing. The main limbs and end plates are both manufactured using FDM printing, and again PLA is the material of choice. The small joint is manufactured using SLA printing and attached to the main arms using a steel dowel pin. While the SLA component is more fragile, this combination is used to reduce the wobble in the mechanism, which had a large amount of play rotating in the x-axis in previous iterations. There was little play in the End Plate to Small Joint interface, thus a standard M2 bolt is used.

5.3.4 Firmware and Control System

The microprocessor is programmed to use the Arduino language. Arduino is used due to its simplicity and compatibility with the chosen components. The firmware itself is a very simple looping structure. The nature of the CAN bus requires each message to be sent with an accompanying identification number. The module firmware is therefore designed to always be checking the CAN bus, waiting for an incoming message with an ID matching the one coded to the individual module. For the joint modules, the messages will contain three pieces of information: the new pose for this joint, which is the joint angle plus a bending plane, and a tertiary number which corresponds to one of several simple commands. Once a new pose has been received, the controller calculates the new positions the two drums must move to. These two values are set as the new setpoint for a pair of PID controllers which run on the system, using the magnetic

positions sensors as the input to the system. The output of the controller is a speed value, which is then sent to each motor controller.

Currently the commands are limited to switching between certain testing functions, such as resetting the position of the joint, zeroing the sensors, or changing PID values for tuning. Again, the aim is to create a system which can be easily extended in the future.

5.3.5 Module Cost and Manufacturing Time

In total the component cost of each Joint Module is an estimated £75.17. The vast majority of this is the commercially purchased components, such as the motors, bearings, and fixings, plus the electrical components required for the PCBs (£70.38). The manufacturing costs of the PCBs and the material costs of the 3D printed components totals only £4.79 per module. Note this cost has been calculated ignoring factors such as delivery fees and import taxes. A breakdown of the module component costs can be found in Appendix D, Table D-1.

The time to manufacture is a more complex estimation to make. Each module can be assembled from its component parts in under an hour. However, the time to 3D print all the components, plus assemble the electrical components on the PCBs is significant. Approximately 19 hours is required to print all the components, and to assemble each set of PCBs takes around 2 hours. This means, when including the required post-processing time and final assembly, each joint module takes around 24 hours to complete. Of course, this time can be greatly reduced, by batch-printing components, or reducing the build quality of the components. Additionally, autonomy is a major benefit of 3D printing. Thus, long prints can be undertaken while other tasks, such as assembling a previous module, are performed.

5.4 Supplementary Modules

5.4.1 Drive Module

The SASAR drive module is the second most complex of the four modules. The basic requirements and architecture of the drive module are much simpler than the Joint Module. It must use the same modular interface, mechanically and

electrically, and be able to receive commands to control the output speed of the wheels. To increase design efficiency, the drive module is designed to reuse as many components from the joint module as possible. In this instance we re-use the design of one of the inter-module connectors, along with numerous electrical components.

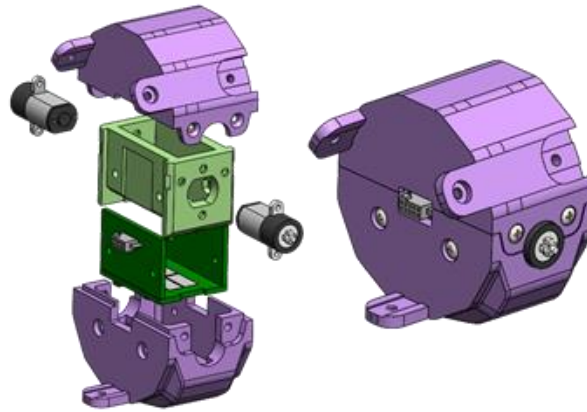


Figure 5.12: CAD model of SASAR 2 Drive Module

The drive module was the final module designed and several improvements were made using the experience gained creating the Joint Module. Mechanically, the main housings of the drive module are now printed in a horizontal orientation. This gives additional strength to the connecting fingers, which tend to break in the joint module. Secondly, while initially designed to have solder pads to connect the motors to the PCBs, the drive module uses small board-to-wire connectors. This both allows the module to be disassembled without the use of a soldering iron, and makes the assembly process easier, as it could be difficult to reach the soldering iron into the correct position when assembling the Joint Module.

Electrically, the design of the Drive module is very similar. The module contains a microcontroller, motor controller and CAN bus components, which are all identical, and connected in a similar manner to the Joint Module. The main electrical difference is the use of two smaller 96 rpm drive motors, which contain a small planetary gearbox. These motors were used as the small size of the module severely limited the available options. Additionally, the four passthrough lanes remain present throughout.

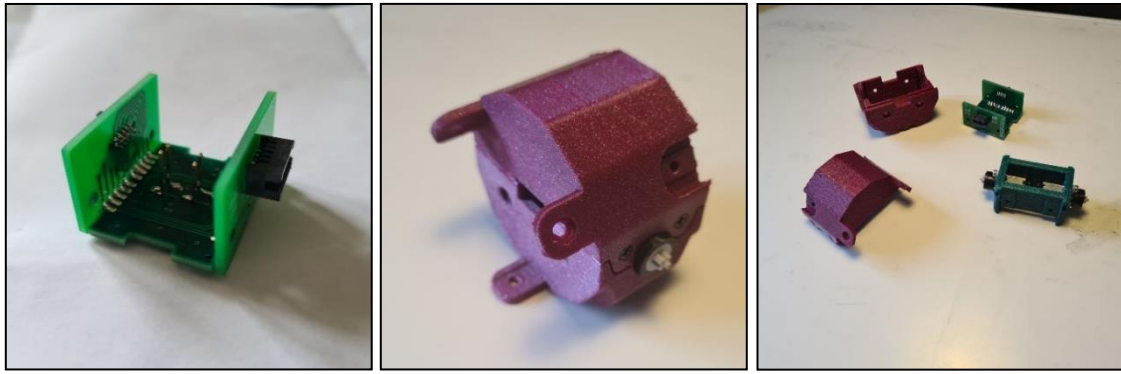


Figure 5.13: Drive Module Prototype

5.4.2 Tail Module

The tail module serves a very simple purpose: to connect SASAR to a central controller via a long tether cable. Internally, the tail module contains the male inter-module connector, which is directly wired to a male DB9 serial connector. DB9 was chosen as the tether cable as it is an existing standard, plus the USB to CAN controller interface used by the central controller uses the DB9 connector. Additionally, it features two locking screws to ensure the connector remains connected. This feature is ideal for the tether cable of a robot designed to delve into unknown structures.

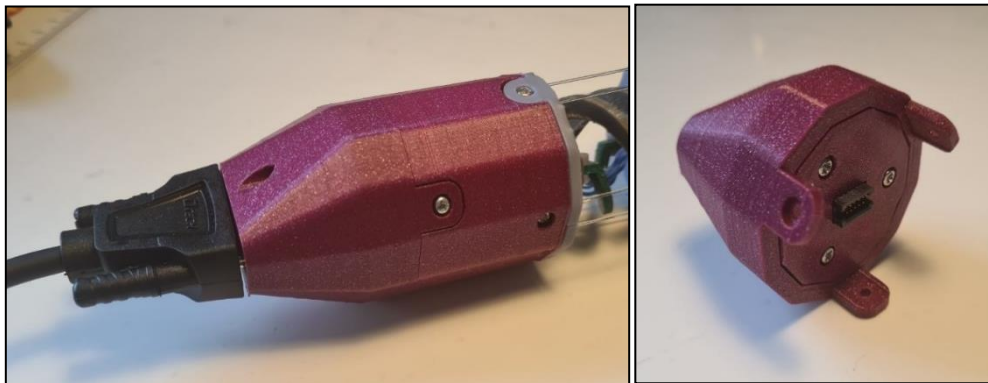


Figure 5.14: SASAR 2 Tail Module

5.4.3 Head Module

The Head Module of SASAR 2 is the final module. For the current testing purposes of SASAR 2 it is somewhat redundant, but mostly serves to display the modularity, and functionality intrinsic to the SASAR design. The current design features a digital camera and miniature LiDAR sensor, which utilise the four

passthrough lanes, plus the main power and ground busses to connect via the tether cable.

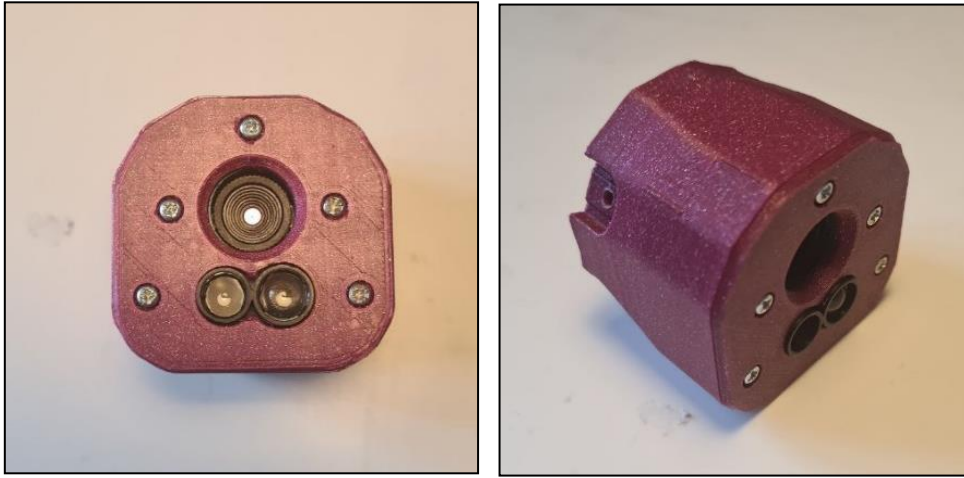


Figure 5.15: SASAR 2 Head Module

5.5 Joint Module Analysis

Here we assess the new SASAR 2 Joint Module in order to compare it to other similar mechanisms. The majority of the properties from Table 5-1 can be found by measuring the physical model. However, the torque and response require further investigation to find an accurate answer. For SASAR 1 we use the properties provided by the motor manufacturer, as the mechanism is directly connected, however this would be difficult to consider accurate for the new Joint Module thanks to the number of interconnecting components between the motor and the actuation. Therefore, two tests are devised for measuring the torque and the responsiveness of the module.

5.5.1 Torque Evaluation

Measuring the torque of the mechanism is not a simple task. As shown in Figure 5.3 the load on the mechanism will be affected by the current joint angle. Additionally, due to the way the system is configured with one module controlling the pitch motion, and the other responsible for the yaw, the current bending plane angle will also affect the maximum torque. When the bending plane is not in one of the cardinal directions, (when $\beta \neq 0, \frac{\pi}{2}, \pi$ or $\frac{3\pi}{2}$) both of the motors should be engaged, thus increasing the torque.

We therefore look to measure both these effects, evaluating the torque output for a range of joint angles, and bending plane angles. We use a static testing setup to measure the output force of the Joint Module and calculate torque by knowing the length from the virtual pivot point to the point of contact. This length is equal to dr , plus the length to the respective contact position, as discussed below.

Testing Method

The testing setup is such: one housing of the Joint Module is clamped in place on the bed of an Instron universal loading frame using a tailored 3D printed clamp. The free end of the module is attached to an ancillary endcap, featuring an embedded steel eyebolt. The testing is performed in two phases. First, we measure the load under compression, and then under tension. This is done for a range of different joint angles and bending planes. To constrain the module to the correct joint angle and plane, a range of different clamps are used, with the desired angle modelled into the part (Figure 5.16). Altering the angle of the fixed half of the mechanism allows us to keep the free half perpendicular to the load cell.

For the compression testing, the module is positioned with the free end in a known position and a minor clearance between the cap and the load cell. This gap is typically less than 1 mm, measured by adjusting the height of the equipment frame until the cell measures a load, before lowering until the load is negligible. We then command the joint to move to a set location, the maximum distance in the bending plane towards the cell, measuring the output of the load cell. The torque is then calculated by taking the maximum load and the lever arm distance from the virtual pivot point at the centre of contact. Each measurement is repeated a total of three times, and an average for the maximum load is taken.

For the tension mode, a loop of the Nylon wire used in the mechanism is tied to the top centre eyebolt, and the opposite end to a steel pin within the load cell. A plumb bob is used to centre the eyelet directly underneath the cell, so as to avoid any loading out of plane. Similarly, the gantry of the loading frame is then lowered until a reading on the load cell is measured. Unlike the compression mode however, we apply a 1 N preload to the system, before commencing

measurements as before. We invert the motion, directing the mechanism to move away from the module to the greatest extent possible.

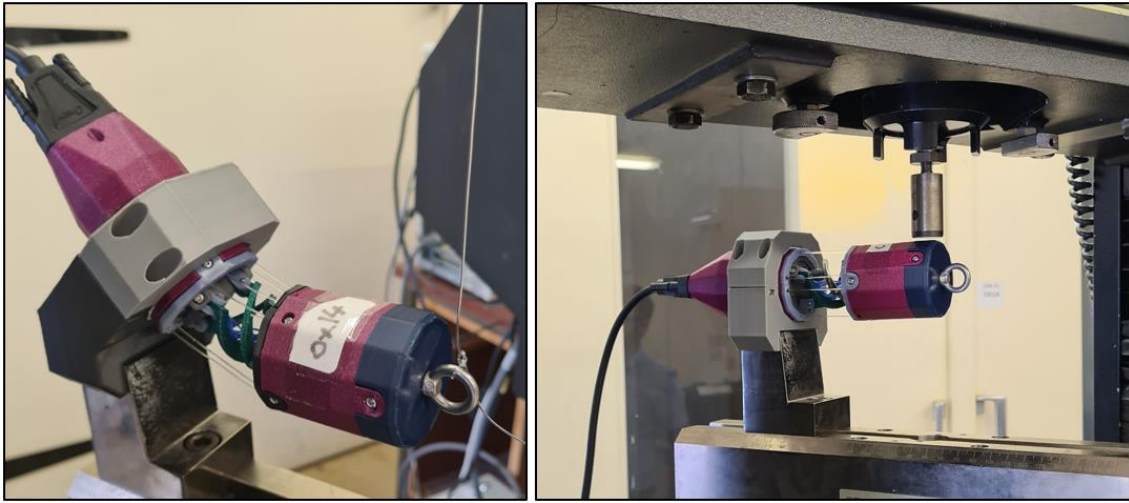


Figure 5.16. Torque testing set up for Tension mode where $\phi = 30, \beta = 30$, and Compression Mode where $\phi = 0, \beta = 0$.

5.5.2 Results and Discussion

The system was tested over seven different configurations, comprising of three joint angles, and three bending planes. The results for the Compression and Tensile testing results are displayed in Table 5-2 and Table 5-3 respectively. The values between Compression and Tension are similar, ranging from a -11% to a 12.6% difference. Table 5-4 displays the average Compression and Tension results for each position.

Table 5-2. Compression Torque Testing Results in Nm

Bending Plane (β) [deg]	Joint Angle (φ) [deg]		
	0	15	45
0	0.659	0.650	0.574
30	0.832	0.818	0.642
45	0.850	-	-

Table 5-3. Tension Torque Testing Results in Nm

Bending Plane (β) [deg]	Joint Angle (φ) [deg]		
	0	15	45
0	0.628	0.568	0.485
30	0.848	0.823	0.713
45	0.798	-	-

Table 5-4. Average Compression/Tension Torque Testing Results, in Nm

Bending Plane (β) [deg]	Joint Angle (φ) [deg]		
	0	15	45
0	0.643	0.609	0.529
30	0.840	0.820	0.678
45	0.824	-	-

We predicted a maximum required output torque of 0.61 Nm and in practice we measure a minimum torque of 0.53 Nm . This is compounded as the maximum required torque of 0.61 Nm is calculated at the maximum joint angle of $\frac{\pi}{2}$, whereas this minimum output is only at $\frac{\pi}{4}$, and as Figure 5.3 shows, the load will continue to increase as the joint angle does. The results do agree with the predicted trend, with the output torque decreasing as the joint angle increases. The bending plane angle also appears to affect the output torque, but not necessarily as expected. Despite the small sample size, it appears the increase in output torque is constant, rising by around 25% when both motors are engaged.

There are two main reasons for the lower measured torque than predicted. Initially we consider the distance measurement of the lever arm used when

calculating the torque. It is assumed the force is applied to the load cell directly in the centre of the contact patch between the top of the housing and the rubber stopper on the end of the load cell. This would only be true in the case where the housing is perfectly perpendicular to the load cell. There may well be a similar issue in the tensile testing, where the centre of the eyebolt is not aligned perfectly to the axis of the load cell. For example, increasing the pivot length by 10 *mm* for the Compression test at $\phi = 0, \beta = 0$ would result in an increase of 13.0% in the calculated torque value. This could account for some error, but it is unlikely to result in the error we see here.

Possibly most critical reason the torque is below expectations is the performance of the Walfront motor. The torque value of 0.1 *Nm* we previously measured is likely to be inaccurate due to the imprecise method used to analyse the motor. Further investigation would require more information on the specifications of the motor, but unfortunately no further information is available for these motors.

5.5.3 Response Evaluation

The purpose of this evaluation is to determine the accuracy and rapidity of response of the Joint Module to an incoming command. Knowing these characteristics will be crucial to the design and fine-tuning of a successful control system. The accuracy is of particular interest, as there is no direct feedback mechanism for the joint angle to the module controller. Instead, the position of the tensile wire drum is controlled. By knowing the drum diameter along with other key geometry, a calculation can be performed to estimate the required drum angle given a desired joint angle. However, there are a number of factors which could cause this calculation to be inaccurate. For example, any loose tension in the wires, or the increase in drum diameter as the wire is wound around it. The purpose of this analysis is to see if these errors do occur, and if possible, create a system to remove them.

Testing Method

The testing method is simple in principle. A secondary sensor is used to measure the “true” joint position, which can then be compared to the value input into the joint. To ensure the test is realistic, the input commands will be sent to the joint

module using the CAN bus, as would be the case in final operation. It would be possible to use the 6-axis gyroscope and accelerometer included in the module design, however at the time of testing this component was not functioning correctly.

Thus, a tertiary system is required. As discussed in section 3.6, the pivot point of a quaternion mechanism is not fixed and increases the virtual length of the joint dependent upon the current joint angle. Therefore, to correctly measure the joint angle, the pivot point must be free to slide vertically with respect to each half of the module. The test rig we use therefore utilises two linear ball-bearing rails to allow the pivot point of the mechanism to move freely in the vertical plane. The two rails are constrained to each half of the module, and also fixed rotationally to each other by a large bearing. Within the bearing sits a magnetic position sensor used to record the position of the module (Figure 5.17).

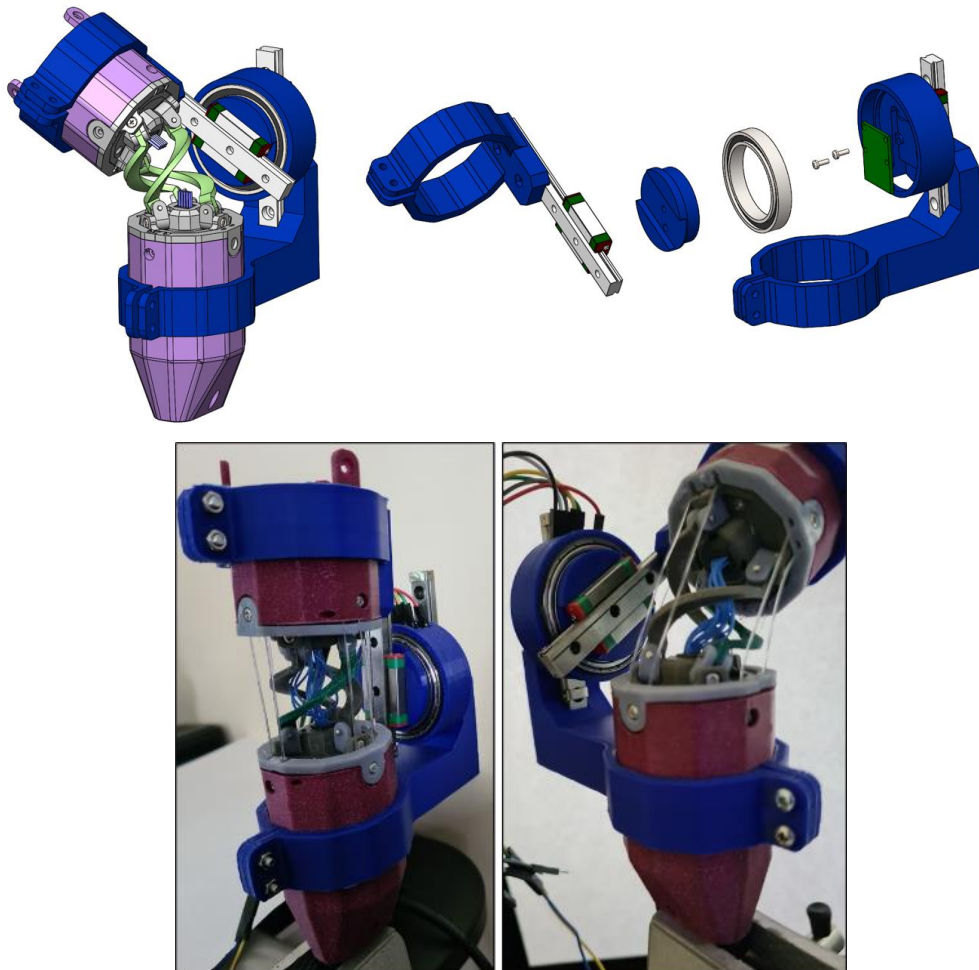


Figure 5.17. Response Testing Rig CAD Model (Top) and Rig in Testing Process (Bottom)

The first input evaluated is a step input response, over a range of different increasing positions. From this we can first calculate a maximum rotation speed, and secondly inspect the steady-state error of the system. Figure 5.18 displays a typical response from the new joint module, in this instance to a 90° input. By measuring the position change over the rise time from 20% to 80% of the steady state value a maximum speed can be calculated. This test is repeated for a total of five attempts, and the average speed taken. From this a maximum rotational speed of 59 degrees per second was found.

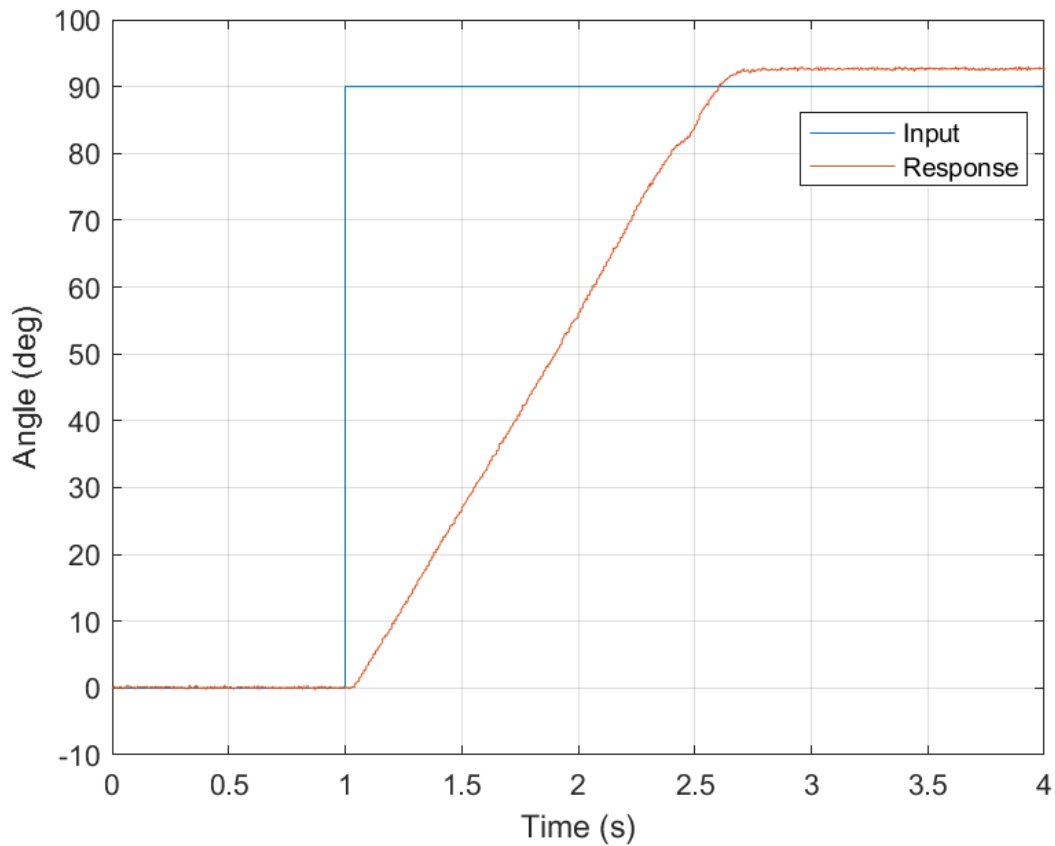


Figure 5.18. Joint Module Response to 90° Step Input

The steady-state error (e_{ss}) is collected by averaging the difference between response and target over the settled period, typically when $t \geq 3$ s. The test is repeated for a target angle (ϕ_t) between 90 and 15 degrees, every 15 degrees. Again, the response is measured a total of five times, and the steady-state error averaged for each position. The results are present in Table 5-5.

Table 5-5. Average Steady-State Errors for a given Input Target

Input Target (deg)	Average Steady State Error (deg)
15.0	-1.49
30.0	-2.74
45.0	2.70
60.0	2.38
75.0	3.05
90.0	2.65

The results are somewhat inconclusive. It appears the device tends to undershoot when moving small distances, but overshoot when moving for longer stretches. To investigate further, we analyse the response to a ramp input, with varying velocity. As Figure 5.19 (e) and (f) show, when the input velocity is too low the response begins to stutter. This is an issue with the selected MCU and motor/gearbox combination, which was first discovered during initial prototyping and testing of the PID controller used in the module. The output from the PID control is a value between -255 and 255, which represents a speed value sent to the motor controller. The output of the motor control is a voltage, between -5v and +5v, which corresponds to the input from the PID controller. However, due to the poor-quality motors, if the input voltage is too low, the motor will not rotate. In an attempt to rectify this, a minimum output velocity was included into the system, however this resulted in the PID controller becoming unstable, due to the backlash in the motor gearbox, thus this was removed. Unfortunately, at the current time there is no way to improve this, meaning the system will not react to small changes in position, or low velocity requests.

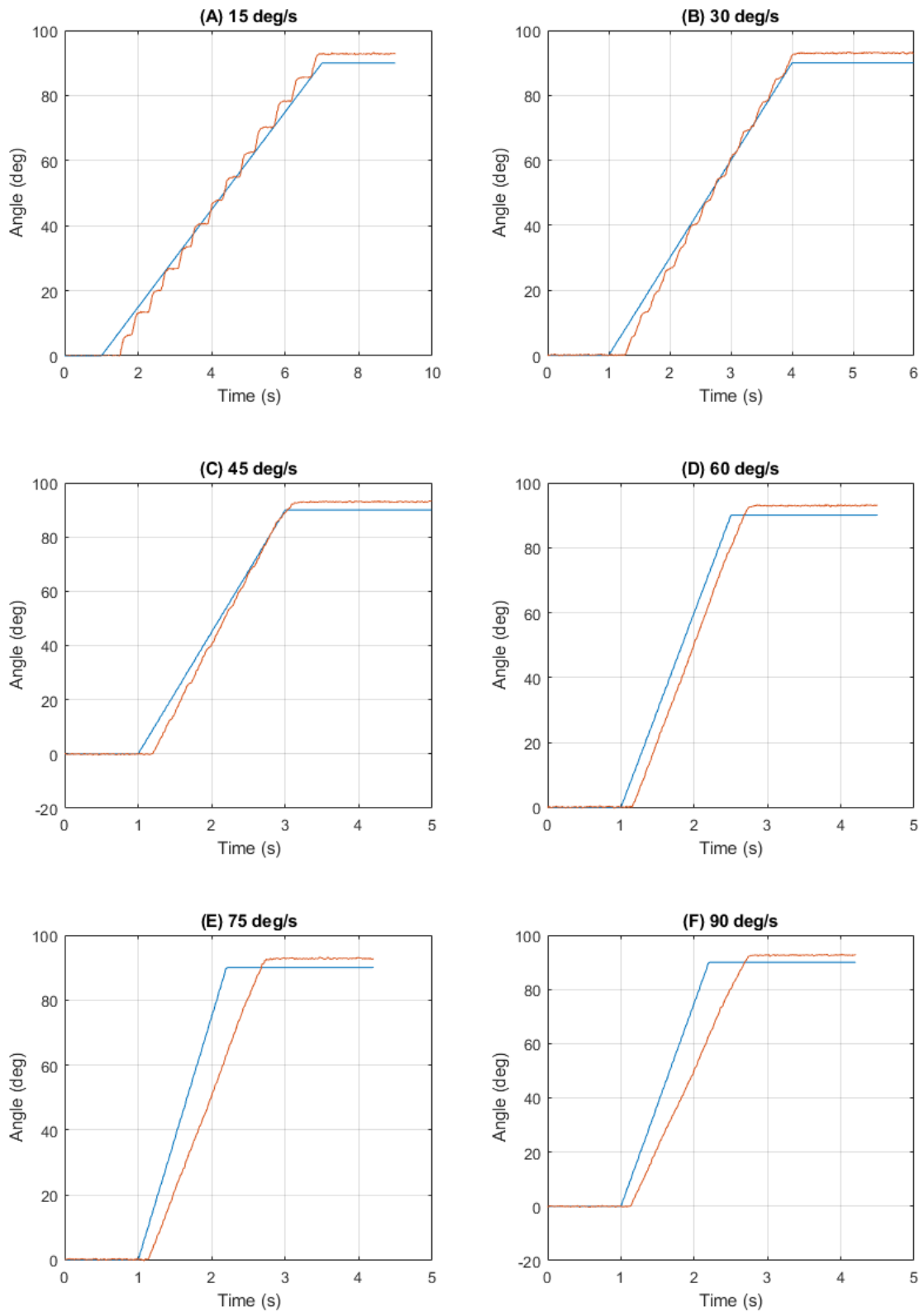


Figure 5.19. Ramp Input Response Graphs for Joint Module (a) 90 deg/s, (b) 75 deg/s, (c) 60 deg/s, (d) 45 deg/s, (e) 30 deg/s, (f) 15 deg/s.

One cause for these errors can be seen when examining the physical models of the SASAR Joint module when in the extreme poses. The joint modules are assembled in the pose $(0, 0)$, as this is the pose at which the motor will be in their 0-position. The final step after assembling the module is to apply tension to the fishing wire used as the driving tensile wire, which is then locked in place using a bolt and washer to clamp the ends of the cable. However, when we examine the state of the wire when in the most extreme pose $\left(\pi, \frac{\pi}{2}\right)$, the tension of the wires in the side which is extending is all but lost. Note, the cables in the opposite, contracting, side remain taught.

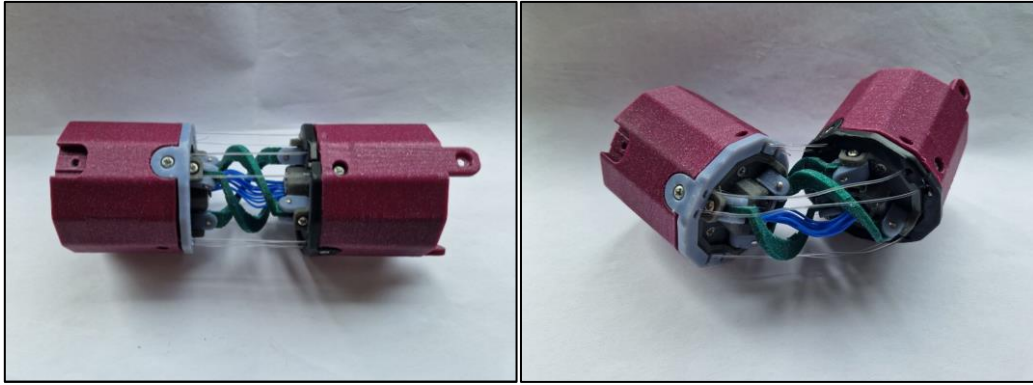


Figure 5.20. SASAR Joint Model in pose $(0, 0)$ [left] and in pose $\left(\pi, \frac{\pi}{2}\right)$ [right], showing how the tension in the cabling is loosened.

A possible explanation can be seen when we investigate Figure 3.24, and equation (3-7) further. In this simple 1-DoF model it is shown that the wire length change is a function of the current joint angle, plus geometric constants such as the number of pulleys and width of the base. In this equation it is assumed that the radius of the mechanism is constant, which, as we have previously shown, it is not. Therefore, one possible solution would be to include an estimation for the wire length lost, or gained, due to the mechanism radius changing. Equation (3-7) would become:

$$\alpha = \frac{nw \sin\left(\frac{\phi}{2}\right) + \Delta r}{r_d} \quad (5-1)$$

And similarly, equation (3-8) would become:

$$\alpha_{pitch} = \frac{nw \sin(\beta) \sin\left(\frac{\phi}{2}\right) + \Delta r}{r_d} \quad \alpha_{yaw} = \frac{nw \cos(\beta) \sin\left(\frac{\phi}{2}\right) + \Delta r}{r_d} \quad (5-2)$$

Where Δr is calculated using equation (3-6). To save on computational resources, this could be pre-calculated and stored in a look-up table within the module firmware.

5.6 Comparison To Similar Mechanisms

Table 5-6 shows a comparison of our two prototype robots against snake-like robot of a similar diameter. In comparative terms, we can see that our designs are typically lighter than their competitors, with similar sizes. However, there is a specific weakness in terms of the maximum torque, and especially for SASAR 2, a low speed. As we can see, the SEA Snake is almost identical to SASAR 2 in length, and only 3 mm larger in diameter, yet has over 8-times the maximum torque output, and over 3-times the speed.

If we analyse our initial module size and torque estimates, summarised in Table 3-1 below, we can see that even the more mature robotics projects would struggle to fit within the 60 mm pipe estimate. Only Woodstock meets the diameter and module length requirements, and these measurements exclude the toroidal skin-drive mechanism, which would take the diameter passed the 41.3 mm limit. Additionally, this is the only robot other than SASAR 1 which has a more restricted operational range.

Table 5-6. SASAR 1 and 2 Performance Comparison verses Similar Snake-Like Robots.

Robot Name	Module Width [mm]	Module Length [mm]	Module Mass [g]	Max Torque [Nm]	Joint Range [$\pm$deg]	Max Speed [deg/s]
SASAR 1	35	75.0	50	0.45	45	375.0
SASAR 2	48	127.4	150	0.84	90	59.0
SEA Snake*	51	128.0	410	7.00	90	198.0
Uncle Sam*	50.8	101.6	320	1.30	90	-
Woodstock**	40.6	101.0	177	9.69	50	151.2
PSR*	40	176.0	400	3.00	90	-
<p><i>*Indicates robots formed of multiple single-DoF modules, thus the length and mass properties are doubled for these cases.</i></p> <p><i>** Woodstock is the name of the HRM used as the basis of the Toroidal skin drive robot. The exact size and mass of the TSD robot is unknown, thus the numbers for Woodstock are shown here.</i></p>						

5.7 Summary

In this chapter we have presented our new serpentine robot, SASAR 2. It features a modular structure and is comprised of four types of modules. The main Joint Module features a novel Quaternion-joint style mechanism, which is unique in the field of snake-like robotics. The supplementary Drive, Head, and Tail modules can be connected to the robot in a wide range of configurations and use a generic electro-mechanical interface, and a CAN communications interface to allow for easy future development.

The majority of the robot is manufactured using 3D printed PLA and UV resin structures, with the inclusion of some COTS fixings and bearings where required. This makes the robot inexpensive to manufacture. With regards to the Joint Module, the majority of the cost is the electric components required to drive the system, such as the microprocessors and magnetic position sensors.

We then analysed the Joint Module utilising two bespoke 3D printed testing rigs, specifically made for the SASAR Joint Module. It was found that the performance was severely lacking, especially the torque output and response time of the system. Regrettably, this was unavoidable due to the lack of options with regards to COTS motors which were able to fit in the small volume. These additionally had the effect of causing unreliable responses at low speeds, causing the stuttering shown in Figure 5.19.

6 RESEARCH ACHIEVEMENTS AND FUTURE RECOMMENDATIONS

6.1 Introduction

In this chapter we review and discuss the findings of this research. We begin by recapping our discoveries and compare the findings to our original aims and objectives. We then discuss in further detail our new snake-like robot design, its strengths and weaknesses, and what improvements should be made for future designs. Finally, we look at what further research should be carried out in order to improve the design and functionality of our snake-like robot.

6.2 Research Findings

The initial goal of this research was to create a functional robot, which met the design specification provided. To achieve this, we introduced three additional core design pillars: Modularity, Semi-Autonomy, and Novelty.

Modularity. This aim was intended to allow for additional functionality to be added to the system if needed, playing into the versatility of snake robots, one of their fundamental benefits over traditional unmanned robots. The three objectives to achieve a modular robot were:

- Introduce a common electro-mechanical modular connection.
- Include an adaptable control system.
- Use a universal standard communications interface.

For the electro-mechanical connection, industrial Samtec 10-pin connectors are used, in combination with the mechanical 3D printed finger joint structures to secure the modules together, shown in Figure 6.1. We also introduced a generic control system based upon a kinematic model. Part of this kinematic model was to treat the linkage lengths as a variable component. This allows us to adjust the length for a particular linkage depending on the current configuration of the robot. For example, if a new module were introduced to carry a payload, the control system could be simply updated to ensure this additional linkage length is included. Finally, we use a CAN bus as our chosen communications interface.

This is a widely used communications standard, although it typically requires additional hardware to interface with microcontrollers. We therefore consider this aim successfully achieved.



Figure 6.1. SASAR 2 Joint Module Showing Connection Interfaced

Semi-Autonomy. This aim was introduced in order to reduce the complexity of controlling a hyper redundant mechanism, such as a snake robot. We also wished to include variable levels to improve the number of tasks the system could undertake, from routine autonomous inspections to manual investigations into unknown environments. To recap, the three objectives targeted were:

- Create multiple, defined, operational modes, each with differing levels of autonomy.
- Creation of a kinematic or dynamic model to enable simulation of the robot.
- Investigate machine learning for the purpose of control and detection.

For the operational modes, we have defined four modes, each with increasing autonomy, and thus increasing features. Table 4-1 summarises the required features for each mode, and each mode is fully defined in Table 4-3 to Table 4-6.

To simulate the performance of the new robot we have derived a kinematic model, based upon an existing model, termed the N-Trailer method. Through both 2D and 3D simulations we have shown that the system can be easily controlled via simple operator inputs, and by a lateral position controller if required.

However, we have not included any machine learning or artificial intelligence application in the system as of yet. The inclusion of a digital camera feed, and the use of computing hardware specific for such tasks (The NVidia Jetson Nano) should help facilitate the introduction of these processes in the future. We would therefore consider this aim partially completed.

Novelty. We targeted a novel design initially as we believed we would be unable to create a functional joint mechanism which would be suitable for the chosen environment. In particular, the size constraints calculated from the desired maximum tube diameters. We also believed that the complex geometric shapes which are made possible through additive manufacturing would help solve this problem. These two objectives were the basis of this aim:

- To create a snake robot which can be scaled if required.
- Use 3D printing for both the new geometry it can create, and to reduce manufacturing costs.

The quaternion mechanism we have produced is unique in snake-like robot design and has so far only been utilised in one other robotic design. Additionally, the mechanism itself is certainly very scalable, and we have managed to produce prototypes of a mechanism with a diameter of less than 15 mm, and length of 25 mm. However, the limiting factor is the motors required to operate the mechanism. For our joint module design, almost exactly two-thirds of the total module length is dedicated to just the motor housing, with the quaternion mechanism responsible for the final third.

With regards to 3D printing, almost all the mechanical structures of the robot are 3D printed, aside from bearings, fixings, and the tensile wire. This was one of the key aspects which allowed for the Quaternion design to be utilised and prototyped with ease. Yet, it too is not without fault. The components manufactured from

resin are particularly brittle, and often would shatter, requiring replacement. However, these were not the largest source of broken components. The three limbs which make up the quaternion mechanism would consistently snap. They are printed in such a way that any bending force upon the limbs will cause the 3D printed layers to separate, similarly to that shown in Figure 6.2. For this aim, we would consider the objectives partially successful. While a novel design has been created, there are still design issues which need to be overcome. Additionally, while the module is mostly 3D printed, there are some components which may need to be manufactured through traditional means or strengthened through other methods.



Figure 6.2. Example Quaternion Mechanism Showing Limb Failure

6.3 SASAR 2 Summary and Future Considerations

6.3.1 SASAR Summary

As discussed above, the goal of this research was to create a novel, semi-autonomous, modular robot, and we have been successful for the most part, with one key exception. We have not yet been able to test the physical robot prototype as a whole unit. Currently, we are unable to produce more than two of the Drive Modules. This is due to a combination of component shortages, leading to an

inability to obtain the required microcontrollers and motors required. Each module type (Head, Joint, Drive, Tail) have been independently tested and verified to be working. For the Head module we have been able to receive images from the digital camera onto the control PC and can additionally use the miniature LiDAR sensor to detect distance of obstacles in front of the robot. The Joint Module was tested as described in Section 5.5, and future testing was performed connecting multiple Joint Modules in series to ensure they could be controlled independently. Finally, for the Drive Module we have undertaken a simple test to ensure that the wheels activate, and their speed can be adjusted. When combined with the 80 mm diameter wheels chosen, the modules can reach a maximum velocity of 96 rpm, equivalent to 0.4 m/s. All of these tests were undertaken using the Tail module to interface the control PC with the various modules.

6.3.2 Future Considerations

As with all designs there are a number of improvements which are discovered along the path to production, which we may have been unable to implement as we would have wished. We separate these into physical considerations, or improvements we could make to the mechanical or electrical design, and control considerations which could be made to the controller or software for the robot.

Physical Considerations

Reduced Size. This is one of the largest improvements which must be made. Our initial target of a 40 mm diameter pipe proved to be a difficult task; thus, the decision was made to remove this requirement, in favour of targeting a more functional device. Now we have a prototype robot, we should consider methods to reduce the size.

Alternate Mechanism. Chapters 2 and 3 of this thesis focus upon researching different mechanisms which could be used for our snake-like robot. While we finally settled upon the Quaternion mechanism, it would be wise to investigate alternatives, and any newer mechanisms or improvements which have surfaced since writing. For example, it may be possible to miniaturise the Pitch-Roll configuration more than initially expected, or there may have been a recent

discovery improving the responsiveness of the TCP muscles used in the Novel Soft Actuator. Otherwise, we could investigate better uses of the Quaternion mechanism. Above we mention that two-thirds of the joint module is dedicated to the motors and controllers to move the mechanism. This is ignoring one of the strongest aspects of this mechanism design, and its implementation in LIMS2-AMBIDEX: the motors can be positioned away from the mechanism. We envisage a system whereby the snake-robot has no on-board motors, only a large system of multiple tensile wires running from the rear of the snake to a control box.

Improved Motors. As we have shown in our physical testing of the Joint Module, the torque and response aspects of the mechanism are poor, especially in comparison to similarly sized comparable mechanisms. Both these issues could be resolved with more powerful, or faster motors. An off-the-shelf solution would be ideal, something similar in size and power to the Savox Servomotors we initially used in SASAR 1, but with the functionality to rotate freely. An early attempt was made to reconfigure these motors, but we were unable to produce a 3D printed structure with enough accuracy to house the new geartrain. Thus, we were reliant on the poorer quality Walfront motors.

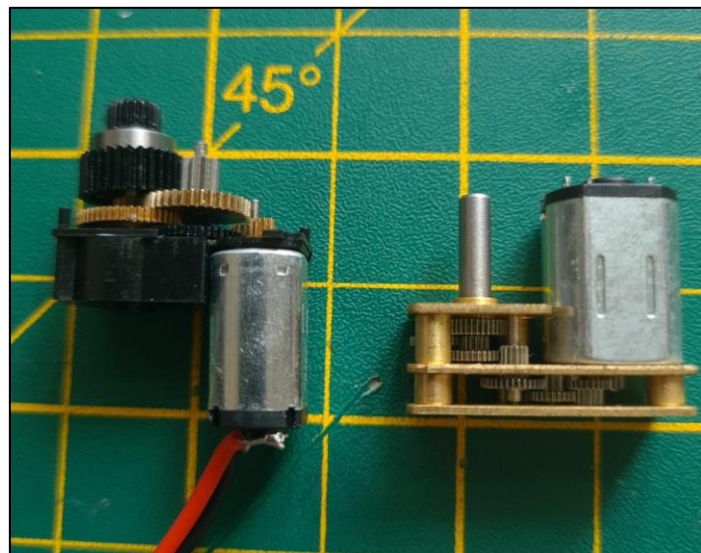


Figure 6.3. A disassembled Savox SH-0256 Servomotor (left) and the Walfront motors used in SASAR (right)

Controller Considerations

A.I. Implementation. As discussed previously, one of the aims of this research was to incorporate some form of machine learning or artificial intelligence into the system, which we were unable to do. This implementation could take many forms, the simplest of which would be to detect obstacles, although using machine learning to train the device how best to overcome the obstacles is also an option.

Obstacle Avoidance. By “Obstacle Avoidance” we mean designing the motion gaits required to overcome any detected obstacles. This work has already been performed by many researchers, for example KAIRO 3 has been taught to climb steps [67]. This would initially involve converting our current kinematic model to a 3D version.

Smoother Motion. One minor control system which could be implemented is to help smooth the motion of the Quaternion modules using interpolation. Currently, when a new pose is requested of the Joint, the desired drum angles are calculated and a pair of PID controllers are used to guide the drum to the desired location. However, this can result in odd motion as one motor may reach its desired location quicker.

6.4 Closing Thoughts

The snake was chosen as the basis of this research for multiple reasons, but the key was the small size. Targeting pipes through which a standard golf ball would be unable to squeeze into is a challenging task, and so it proved. Especially when we attempted to constrain ourselves to using mostly off-the-shelf components. We therefore switched the focus of this research to investigate novel mechanisms for use with a snake-like robot, with a view to scaling the design down to the 40 mm target in a future project.

The result is the unique Quaternion Joint Module. The low-cost 3D printed structure moves in a distinctive manner, mimicking the rolling of two spheres. The joint is able to easily reach a full 90-degree joint angle, in any bending plane from 0 to 360-degrees. The open and consistent length of the central path through the mechanism, which proved a perfect channel for our electrical connections, could

be applied to a variety of situations, carrying a flexible pipe for example, or using a universal linkage as in LIMS2. The use of tensile wires to actuate the joint means configuration for different physical properties can be easily achieved by simply increasing or reducing the total number of pulley loops. This mechanism could easily be repurposed for use in all areas of robotics.

REFERENCES

1. THE REPTILE DATABASE. Available at: <http://www.reptile-database.org/> (Accessed: 8 July 2021)
2. WHO. Venomous snake distribution database. who.int. 2021. Available at: <http://apps.who.int/bloodproducts/snakeantivenoms/database/> (Accessed: 12 January 2021)
3. Hirose S., Umetani Y. Kinematic Control of Active Cord Mechanism with Tactile Sensors. 1975; : 0–4. Available at: DOI:10.9746/sicetr1965.12.543
4. Togawa K., Mori M., Hirose S. Study on three-dimensional active cord mechanism: Development of ACM-R2. IEEE International Conference on Intelligent Robots and Systems. 2000; 3: 2242–2247. Available at: DOI:10.1109/iros.2000.895302
5. Yamada H., Hirose S. Development of Practical 3-Dimensional Active Cord Mechanism ACM-R4. Journal of Robotics and Mechatronics. 2006; 18(3): 305–311. Available at: DOI:10.20965/jrm.2006.p0305
6. Takaoka S., Yamada H., Hirose S. Snake-like active wheel robot ACM-R4.1 with joint torque sensor and limiter. IEEE International Conference on Intelligent Robots and Systems. IEEE; 2011; : 1081–1086. Available at: DOI:10.1109/IROS.2011.6048084
7. Kouno K., Yamada H., Hirose S. Development of active-joint active-wheel high traversability snake-like robot ACM-R4.2. Journal of Robotics and Mechatronics. 2013; 25(3): 559–566. Available at: DOI:10.20965/jrm.2013.p0559
8. Yamada H., Chigisaki S., Mori M., Takita K., Ogami K., Hirose S. Development of amphibious snake-like robot ACM-R5. 36th Int. Symp. Robotics. 2005.
9. Ohashi T., Yamada H., Hirose S. Loop forming snake-like robot ACM-R7 and its serpenoid oval control. IEEE/RSJ 2010 International Conference

- on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings. 2010. pp. 413–418. Available at:
DOI:10.1109/IROS.2010.5651467
10. Komura H., Yamada H., Hirose S. Development of snake-like robot ACM-R8 with large and mono-tread wheel. Advanced Robotics. Robotics Society of Japan; 2 September 2015; 29(17): 1081–1094. Available at:
DOI:10.1080/01691864.2014.971054
 11. Sugita S., Ogami K., Michele G., Hirose S., Takita K. A Study on the Mechanism and Locomotion Strategy for New Snake-Like Robot Active Cord Mechanism – Slime model 1 ACM-S1. Journal of Robotics and Mechatronics. 2008; 20(2): 302–310. Available at:
DOI:10.20965/jrm.2008.p0302
 12. Mori M., Hirose S. Development of active cord mechanism ACM-R3 with agile 3D mobility. IEEE International Conference on Intelligent Robots and Systems. 2001; 3: 1552–1557. Available at:
DOI:10.1109/iros.2001.977200
 13. Transeth AA., Leine RI., Glocker C., Pettersen KY., Liljebäck P. Snake robot obstacle-aided locomotion: Modeling, simulations, and experiments. IEEE Transactions on Robotics. 2008; 24(1): 88–104. Available at:
DOI:10.1109/TRO.2007.914849
 14. Liljebäck P., Pettersen KY., Staudahl Ø. A snake robot with a contact force measurement system for obstacle-aided locomotion. Proceedings - IEEE International Conference on Robotics and Automation. 2010; : 683–690. Available at: DOI:10.1109/ROBOT.2010.5509839
 15. Liljebäck P., Haugstuen IU., Pettersen KY. Path following control of planar snake robots using a cascaded approach. IEEE Transactions on Control Systems Technology. January 2012; 20(1): 111–126. Available at:
DOI:10.1109/TCST.2011.2107516

16. Liljebäck P., Pettersen KY., Stavadahl Ø., Gravdahl JT. Experimental investigation of obstacle-aided locomotion with a snake robot. IEEE Transactions on Robotics. August 2011; 27(4): 792–800. Available at: DOI:10.1109/TRO.2011.2134150
17. Liljebäck P., Pettersen KY., Stavadahl Ø., Gravdahl JT. Snake robot locomotion in environments with obstacles. IEEE/ASME Transactions on Mechatronics. 2012; 17(6): 1158–1169. Available at: DOI:10.1109/TMECH.2011.2159863
18. Liljebäck P., Stavadahl O., Pettersen KY., Gravdahl JT. Mamba - A waterproof snake robot with tactile sensing. IEEE International Conference on Intelligent Robots and Systems. 2014; : 294–301. Available at: DOI:10.1109/IROS.2014.6942575
19. Liljebäck P., Pettersen KY., Stavadahl Ø., Gravdahl JT. Snake Robots - Modelling, Mechatronics, and Control. 2013. Available at: DOI:10.1007/978-1-4471-2996-7
20. Transeth AA., Leine RI., Glocker C., Pettersen KY. 3-D snake robot motion: Nonsmooth modeling, simulations, and experiments. IEEE Transactions on Robotics. April 2008; 24(2): 361–376. Available at: DOI:10.1109/TRO.2008.917003
21. Transeth AA., Leine RI., Glocker C., Pettersen KY., Liljebäck P. Snake Robot Obstacle Aided Locomotion: Modeling, Simulations, and Experiments. Available at: <http://ieeexplore.ieee.org>,
22. Murugendran B., Transeth AA., Fjerdingen SA. Modeling and path-following for a snake robot with active wheels. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009. 2009. pp. 3643–3650. Available at: DOI:10.1109/IROS.2009.5353886
23. Crespi A., Badertscher A., Guignard A., Ijspeert AJ. AmphiBot I: An amphibious snake-like robot. Robotics and Autonomous Systems. 31

March 2005; 50(4): 163–175. Available at:
DOI:10.1016/j.robot.2004.09.015

24. Crespi A., Jan Ijspeert A. AmphiBot II: An Amphibious Snake Robot that Crawls and Swims using a Central Pattern Generator. 2006. Available at: <https://www.researchgate.net/publication/40755126>
25. Shammass Alon Wolf Ben Brown EH., Howie Choset J. New Joint Design for Three-dimensional Hyper Redundant Robots. 2003.
26. Johnson A., Wright C., Tesch M., Lipkin K., Choset H. A Novel Architecture for Modular Snake Robots. Ri.Cmu.Edu. 2011. pp. 1–7. Available at: http://www.ri.cmu.edu/pub_files/2011/8/journal08.pdf
27. Wolf A., Brown HB., Casciola R., Costa A., Schwerin M., Shamas E., et al. A Mobile Hyper Redundant Mechanism for Search and Rescue Tasks. IEEE International Conference on Intelligent Robots and Systems. 2003; 3(October): 2889–2895. Available at: DOI:10.1109/iros.2003.1249309
28. McKenna JC., Anhalt DJ., Bronson FM., Brown H ben., Schwerin M., Shammass E., et al. Toroidal skin drive for snake robot locomotion. Proceedings - IEEE International Conference on Robotics and Automation. 2008; : 1150–1155. Available at: DOI:10.1109/ROBOT.2008.4543359
29. Wright C., Johnson A., Peck A., Mccord Z., Naaktgeboren A., Gianfortoni P., et al. Design of a Modular Snake Robot.
30. Rollinson D., Ford S., Brown B., Choset H. DESIGN AND MODELING OF A SERIES ELASTIC ELEMENT FOR SNAKE ROBOTS. 2013. Available at: DOI:10.0/Linux-x86_64
31. Rollinson D., Bilgen Y., Brown B., Enner F., Ford S., Layton C., et al. Design and architecture of a series elastic snake robot. IEEE International Conference on Intelligent Robots and Systems. Institute of Electrical and Electronics Engineers Inc.; 2014. pp. 4630–4636. Available at: DOI:10.1109/IROS.2014.6943219

32. Snake Robot Search & Rescue - TEEX Disaster City - YouTube.
Available at: <https://www.youtube.com/watch?v=A9OdUPY3OyE>
(Accessed: 27 April 2021)
33. Wright C., Buchan A., Brown B., Geist J., Schwerin M., Rollinson D., et al.
Design and architecture of the unified modular snake robot. Proceedings -
IEEE International Conference on Robotics and Automation. 2012; :
4347–4354. Available at: DOI:10.1109/ICRA.2012.6225255
34. Schreiber DA., Richter F., Bilan A., Gavrilov P v., Lam HM., Price CH., et
al. ARCSnake: An Archimedes' Screw-Propelled, Reconfigurable
Serpentine Robot for Complex Environments. 2020; Available at:
DOI:10.0/Linux-x86_64
35. Borenstein J., Granosik G., Hansen M. The OmniTread OT-4 serpentine
robot: design and field performance. In: Gerhart GR, Shoemaker CM,
Gage DW (eds.) 2005. p. 324. Available at: DOI:10.1117/12.601486
36. Gray J. The mechanism of locomotion in Snakes. Journal of Morphology.
1946; 23(2): 101–120. Available at: DOI:10.1002/jmor.1050220107
37. Crotalus Cerastes - Sidewinder Rattlesnake - USA Snakes. Available at:
<https://usasnakes.com/crotalus-cerastes-sidewinder-rattlesnake/>
(Accessed: 12 July 2021)
38. Granosik G., Borenstein J. Integrated joint actuator for serpentine robots.
IEEE/ASME Transactions on Mechatronics. October 2005; 10(5): 473–
481. Available at: DOI:10.1109/TMECH.2005.856222
39. Paljug E., Ohm T., Hayati S. The JPL Serpentine Robot: a 12-DOF
system for inspection. Proceedings of 1995 IEEE International
Conference on Robotics and Automation. IEEE; 1995. pp. 3143–3148.
Available at: DOI:10.1109/ROBOT.1995.525732
40. 2. Industrial Robot Functionality and Coordinate Systems – Inlearc.
Available at: <https://www.tthk.ee/inlearcs/industrial-robot-functionality-and-coordinate-systems/> (Accessed: 2 August 2021)

41. FLSUN Auto leveling Kossel Metal Delta 3D Printer Kits | 3D Printers Bay. Available at: <https://www.3dprintersbay.com/flsun-autolevelling-kossel-diy-kit> (Accessed: 2 August 2021)
42. Endo G., Togawa K., Hirose S. Study on self-contained and terrain adaptive active cord mechanism. IEEE International Conference on Intelligent Robots and Systems. 1999; 3(1): 1399–1405. Available at: DOI:10.7210/jrsj.18.419
43. Crespi A., Badertscher A., Guignard A., Ijspeert AJ. AmphiBot I: An amphibious snake-like robot. Robotics and Autonomous Systems. 2005; 50(4): 163–175. Available at: DOI:10.1016/j.robot.2004.09.015
44. Crespi A., Ijspeert AJA., F EP. AmphiBot II : An Amphibious Snake Robot that Crawls and Swims using a Central Pattern Generator. ... Conference on Climbing and Walking Robots (.... 2006; (September): 19–27.
45. Zhu W., Guo X., Fang Y. Design of a modular snake robot and control with Internet of Things. Proceedings - 2017 Chinese Automation Congress, CAC 2017. 2017; 2017-Janua: 850–854. Available at: DOI:10.1109/CAC.2017.8242884
46. Yamada H., Hirose S. Development of Practical 3-Dimensional Active Cord Mechanism ACM-R4. Journal of Robotics and Mechatronics. 2006; 18(3): 305–311. Available at: DOI:10.20965/jrm.2006.p0305
47. Mori M., Hirose S. Development of active cord mechanism ACM-R3 with agile 3D mobility. IEEE International Conference on Intelligent Robots and Systems. 2001; 3: 1552–1557. Available at: DOI:10.1109/iros.2001.977200
48. Takaoka S., Yamada H., Hirose S. Snake-like active wheel robot ACM-R4.1 with joint torque sensor and limiter. IEEE International Conference on Intelligent Robots and Systems. IEEE; 2011; : 1081–1086. Available at: DOI:10.1109/IROS.2011.6048084

49. Kouno K., Yamada H., Hirose S. Development of active-joint active-wheel high traversability snake-like robot ACM-R4.2. *Journal of Robotics and Mechatronics*. 2013; 25(3): 559–566. Available at: DOI:10.20965/jrm.2013.p0559
50. Wright C., Buchan A., Brown B., Geist J., Schwerin M., Rollinson D., et al. Design and architecture of the unified modular snake robot. *Proceedings - IEEE International Conference on Robotics and Automation*. 2012; : 4347–4354. Available at: DOI:10.1109/ICRA.2012.6225255
51. Rollinson D., Bilgen Y., Brown B., Enner F., Ford S., Layton C., et al. Design and architecture of a series elastic snake robot. *IEEE International Conference on Intelligent Robots and Systems*. IEEE; 2014; (Iros): 4630–4636. Available at: DOI:10.1109/IROS.2014.6943219
52. Liljebäck P., Staudahl O., Pettersen KY., Gravdahl JT. Mamba - A waterproof snake robot with tactile sensing. *IEEE International Conference on Intelligent Robots and Systems*. 2014; : 294–301. Available at: DOI:10.1109/IROS.2014.6942575
53. Ni F., Li Y., Zhou Y., Zhao L., Liu H. Design of a Hierarchical Control System for Tetherless Snake Robot. *Proceedings of 2019 IEEE International Conference on Mechatronics and Automation, ICMA 2019*. IEEE; 2019; : 1254–1259. Available at: DOI:10.1109/ICMA.2019.8816398
54. Sanfilippo F., Helgerud E., Stadheim PA., Aronsen SL. Serpens, A Low-Cost Snake Robot with Series Elastic Torque-Controlled Actuators and A Screw-Less Assembly Mechanism. *2019 5th International Conference on Control, Automation and Robotics, ICCAR 2019*. IEEE; 2019; : 133–139. Available at: DOI:10.1109/ICCAR.2019.8813482
55. Tanaka M., Kon K., Nakajima M., Matsumoto N., Fukumura S., Fukui K., et al. Development and field test of the articulated mobile robot T2 Snake-4 for plant disaster prevention. *Advanced Robotics*. Taylor & Francis; 2020; 34(2): 70–88. Available at: DOI:10.1080/01691864.2019.1680316

56. Fukushima H., Satomura S., Kawai T., Tanaka M., Kamegawa T. Modeling and Control of a Snake-Like Robot Using the Screw-Drive Mechanism. *IEEE Transactions on Robotics*. IEEE; 2012; 28(3): 541–554. Available at: DOI:10.1109/TRO.2012.2183050
57. Fjerdingen SA., Liljebäck P., Transeth AA. A snake-like robot for internal inspection of complex pipe structures (PIKo). 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009. IEEE; 2009; : 5665–5671. Available at: DOI:10.1109/IROS.2009.5354751
58. Liljebäck P., Pettersen KY., Stavadahl Ø. A snake robot with a contact force measurement system for obstacle-aided locomotion. *Proceedings - IEEE International Conference on Robotics and Automation*. 2010; : 683–690. Available at: DOI:10.1109/ROBOT.2010.5509839
59. Wang H., Liu T., Xu W., Wang H. A snake-like robot composed of 2-DOFs modularized spherical-shape joints for space application. 2016 14th International Conference on Control, Automation, Robotics and Vision, ICARCV 2016. IEEE; 2017; 2016(November): 13–15. Available at: DOI:10.1109/ICARCV.2016.7838674
60. Schreiber DA., Richtery F., Bilanz A., Gavrilovz P V., Lamz HM., Pricez CH., et al. ARC snake: An archimedes' screw-propelled, reconfigurable robot snake for complex environments. *arXiv*. 2019; : 7029–7034.
61. Brown H Ben., Schwerin M., Shammass E., Choset H. Design and Control of a Second-Generation Hyper-Redundant Mechanism. 2007; : 2603–2608.
62. McKenna JC., Anhalt DJ., Bronson FM., Brown H Ben., Schwerin M., Shammass E., et al. Toroidal skin drive for snake robot locomotion. *Proceedings - IEEE International Conference on Robotics and Automation*. 2008; : 1150–1155. Available at: DOI:10.1109/ROBOT.2008.4543359

63. Shamas E., Wolf A., Choset H. Three degrees-of-freedom joint for spatial hyper-redundant robots. *Mechanism and Machine Theory*. 2006; 41(2): 170–190. Available at: DOI:10.1016/j.mechmachtheory.2005.04.008
64. Rome E., Hertzberg J., Kirchner F., Licht U., Christaller T. Towards autonomous sewer robots: The MAKRO project. *Urban Water*. 1999; 1(1): 57–70. Available at: DOI:10.1016/s1462-0758(99)00012-6
65. Kolesnik M., Birlinghoven S., Sankt D., Streich H., Birlinghoven S., Sankt D-. Visual Orientation and Motion Control of MAKRO – Adaptation to the Sewer Environment. 1999;
66. Scholl KU., Kepplin V., Berns K., Dillmann R. Controlling a multijoint robot for autonomous sewer inspection. *Proceedings - IEEE International Conference on Robotics and Automation*. 2000. pp. 1701–1706. Available at: DOI:10.1109/robot.2000.844841
67. Pfotzer L., Staehler M., Hermann A., Roennau A., Dillmann R. KAIRO 3: Moving over stairs & unknown obstacles with reconfigurable snake-like robots. 2015 European Conference on Mobile Robots, ECMR 2015 - Proceedings. Institute of Electrical and Electronics Engineers Inc.; 2015. Available at: DOI:10.1109/ECMR.2015.7324209
68. Pfotzer L., Ruehl S., Heppner G., Roennau A., Dillmann R. KAIRO 3: A modular reconfigurable robot for search and rescue field missions. 2014 IEEE International Conference on Robotics and Biomimetics, IEEE ROBIO 2014. Institute of Electrical and Electronics Engineers Inc.; 2014. pp. 205–210. Available at: DOI:10.1109/ROBIO.2014.7090331
69. KAIRO - MODULAR SNAKE-LIKE INSPECTION ROBOT. <https://www.fzi.de/>. Available at: <https://www.fzi.de/forschung/projekt-details/kairo/> (Accessed: 20 July 2021)
70. Wolf A., Brown HB., Casciola R., Costa A., Schwerin M., Shamas E., et al. A Mobile Hyper Redundant Mechanism for Search and Rescue Tasks.

- IEEE International Conference on Intelligent Robots and Systems. 2003; 3(October): 2889–2895. Available at: DOI:10.1109/iros.2003.1249309
71. Togawa K., Mori M., Hirose S. Study on three-dimensional active cord mechanism: Development of ACM-R2. IEEE International Conference on Intelligent Robots and Systems. 2000; 3: 2242–2247. Available at: DOI:10.1109/iros.2000.895302
 72. Ohashi T., Yamada H., Hirose S. Loop forming snake-like robot ACM-R7 and its serpenoid oval control. IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010 - Conference Proceedings. IEEE; 2010; : 413–418. Available at: DOI:10.1109/IROS.2010.5651467
 73. Komura H., Yamada H., Hirose S. Development of snake-like robot ACM-R8 with large and mono-tread wheel. Advanced Robotics. Taylor & Francis; 2015; 29(17): 1081–1094. Available at: DOI:10.1080/01691864.2014.971054
 74. Sugita S., Ogami K., Michele G., Hirose S., Takita K. A Study on the Mechanism and Locomotion Strategy for New Snake-Like Robot Active Cord Mechanism – Slime model 1 ACM-S1. Journal of Robotics and Mechatronics. 2008; 20(2): 302–310. Available at: DOI:10.20965/jrm.2008.p0302
 75. Yamada H., Chigisaki S., Mori M., Takita K., Ogami K., Hirose S. Development of amphibious snake-like robot ACM-R5. 36th Int. Symp. Robotics. 2005.
 76. Borenstein J., Granosik G., Hansen M. The OmniTread serpentine robot: design and field performance. Unmanned Ground Vehicle Technology VII. SPIE; 2005. p. 324. Available at: DOI:10.1117/12.601486
 77. Granosik G., Hansen MG., Borenstein J. The OmniTread serpentine robot for industrial inspection and surveillance. Industrial Robot. 2005; 32(2): 139–148. Available at: DOI:10.1108/01439910510582264

78. Sanfilippo F., Helgerud E., Stadheim PA., Aronsen SL. Serpens: A highly compliant low-cost ROS-based snake robot with series elastic actuators, stereoscopic vision and a screw-less assembly mechanism. *Applied Sciences (Switzerland)*. 2019; 9(3). Available at: DOI:10.3390/app9030396
79. Kolesnik M., Birlinghoven S., Sankt D., Streich H., Birlinghoven S., Sankt D-. Visual Orientation and Motion Control of MAKRO – Adaptation to the Sewer Environment. 1999;
80. Granosik G., Hansen MG., Borenstein J. The OmniTread serpentine robot for industrial inspection and surveillance. *Industrial Robot*. 2005; 32(2): 139–148. Available at: DOI:10.1108/01439910510582264
81. Borenstein J., Granosik G., Hansen M. The OmniTread OT-4 serpentine robot: design and field performance. In: Gerhart GR, Shoemaker CM, Gage DW (eds.) 2005. p. 324. Available at: DOI:10.1117/12.601486
82. Wu L., Chauhan I., Tadesse Y. A Novel Soft Actuator for the Musculoskeletal System. *Advanced Materials Technologies*. Wiley-Blackwell; 1 May 2018; 3(5). Available at: DOI:10.1002/admt.201700359
83. Kim YJ., Kim JI., Jang W. Quaternion Joint: Dexterous 3-DOF Joint Representing Quaternion Motion for High-Speed Safe Interaction. *IEEE International Conference on Intelligent Robots and Systems*. 2018; : 935–942. Available at: DOI:10.1109/IROS.2018.8594301
84. Verleg MN. Wrist Prosthesis: New Two Degrees-Of-Freedom Hydraulic Wrist Mechanism for Hand Prostheses. 2015.
85. Brown H ben., Schwerin M., Shammas E., Choset H. Design and Control of a Second-Generation Hyper-Redundant Mechanism. 2007; : 2603–2608.
86. Wright C., Johnson A., Peck A., McCord Z., Naaktgeboren A., Gianfortoni P., et al. Design of a modular snake robot. *IEEE International Conference*

- on Intelligent Robots and Systems. 2007; : 2609–2614. Available at: DOI:10.1109/IROS.2007.4399617
87. Savox - Servo - SH-0256 - Digital - DC Motor » Brushed Motor » Servos » SAVOX-SERVO.COM. Available at: <https://www.savox-servo.com/Servos-c-1338/Brushed-Motor-c-1340/Savox-Servo-SH-0256-Digital-DC-Motor/> (Accessed: 5 August 2021)
 88. Tesch M., Lipkin K., Brown I., Hatton R., Peck A., Rembisz J., et al. Parameterized and scripted gaits for modular snake robots. *Advanced Robotics*. 1 June 2009; 23(9): 1131–1158. Available at: DOI:10.1163/156855309X452566
 89. Schmit-Nielsen K. *Scaling. Why is animal size so important?* Cambridge University Press; 1984. Available at: DOI:10.1002/ajpa.1330690114
 90. Autonomous definition and meaning | Collins English Dictionary. Available at: <https://www.collinsdictionary.com/dictionary/english/autonomous> (Accessed: 19 November 2021)
 91. SAE International. *Automated Driving - Levels of driving automation*. SAE International Standards. 2016; SAE J3016.
 92. Sheridan TB. *Telerobotics, Automation, and Human Supervisory Control*. The MIT Press; 1992. Available at: <https://mitpress.mit.edu/books/telerobotics-automation-and-human-supervisory-control>
 93. Parasuraman R., Sheridan TB., Wickens CD. *A Model for Types and Levels of Human Interaction with Automation*. SYSTEMS AND HUMANS. 2000.
 94. Boyd JR. *The Essence of Winning and Losing*. 2010. Available at: www.dnipogo.org (Accessed: 19 November 2021)
 95. Proud RW., Hart JJ., Mrozinski RB. *Methods for Determining the Level of Autonomy to Design into a Human Spaceflight Vehicle: A Function*

Specific Approach. Performance Metrics for Intelligent Systems. Available at: <https://ntrs.nasa.gov/citations/20100017272> (Accessed: 22 November 2021)

96. Proud RW., Hart JJ., Mrozinski RB. Methods for Determining the Level of Autonomy to Design into a Human Spaceflight Vehicle: A Function Specific Approach. PerMIS '03. 2003;
97. Tanaka M., Kon K., Nakajima M., Matsumoto N., Fukumura S., Fukui K., et al. Development and field test of the articulated mobile robot T2 Snake-4 for plant disaster prevention. *Advanced Robotics*. Robotics Society of Japan; 17 January 2020; 34(2): 70–88. Available at: DOI:10.1080/01691864.2019.1680316
98. Yamada H., Takaoka S., Hirose S. A snake-like robot for real-world inspection applications (the design and control of a practical active cord mechanism). *Advanced Robotics*. 2013. pp. 47–60. Available at: DOI:10.1080/01691864.2013.752318
99. Altafini C. Some properties of the general n-trailer. *International Journal of Control*. 10 March 2001; 74(4): 409–424. Available at: DOI:10.1080/00207170010010579
100. Michałek MM. A highly scalable path-following controller for N-trailers with off-axle hitching. *Control Engineering Practice*. Elsevier Ltd; 2014; 29: 61–73. Available at: DOI:10.1016/j.conengprac.2014.04.001
101. Hoffmann GM., Tomlin CJ., Montemerlo M., Thrun S. Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing †. 2007.
102. Snider JM. Automatic Steering Methods for Autonomous Automobile Path Tracking. 2009.
103. Dominguez-Quijada S., Ali A., Garcia G., Martinet P., Dominguez S. Comparison of lateral controllers for autonomous vehicle : experimental results. 2016. Available at: DOI:10.0/Linux-x86_64

104. Webots: robot simulator. Available at:
<https://cyberbotics.com/#cyberbotics> (Accessed: 7 January 2022)
105. Jetson Nano Developer Kit for AI and Robotics | NVIDIA. Available at:
<https://www.nvidia.com/en-gb/autonomous-machines/embedded-systems/jetson-nano/> (Accessed: 23 December 2021)
106. USB2CAN-Module,CE,FCC,Windows/Linux/Mac Os/ – Raspberry PI|Industrial Camera|USB TO CAN FD|. Available at: <https://www.inno-maker.com/product/usb-can/> (Accessed: 23 December 2021)

APPENDICES

Appendix A Customer Provided Information

Table A-1. Customer Design Specification

The robot <i>must</i> be able to...	<ul style="list-style-type: none">• Bridge a gap of at least 80mm.• Traverse a tube of minimum diameter 80mm.• Turn through a curve of minimum radius of 80mm.• Ascend a ramp of a gradient of 45 degrees. (1:1 ratio)• Delve a minimum of 10 metres into a structure.
The robot <i>should</i> be able to...	<ul style="list-style-type: none">• Traverse a tube of minimum diameter 60mm.• Climb a step of height 75mm.• Reach a speed of 5 metres per minute in a straight line.• Carry a payload of 50g.• Last 1 hour of constant usage.
The robot <i>could</i> be able to...	<ul style="list-style-type: none">• Traverse a tube of minimum diameter 40mm.• Operate without out fault in light watery conditions.• Operate in a temperature range of 0 °C to 50 °C.• Be able to cross loose rocky and oiled surfaces.

Note: For the Customer design specifications, the requirements are listed from most crucial to least crucial. The “*must*” requirements being the most, and the “*could*” requirements being the least.

Appendix B Prototype Mechanisms

B.1 Mechanism Range Simulations

The positions of the end effectors in Cartesian coordinates (x y z) are calculated in terms of the relative module length (L), and the joint angles (ϕ). This is done by compounding rotation and translation matrices from the origin to the end effector. For example, the Pitch-Yaw mechanism is calculated as a translation in Z by $0.25L$, compounded with a rotation about the x axis by ϕ_{pitch} , another Z translation of $0.5L$, a rotation about y of ϕ_{yaw} , and a final $0.25L$ z translation. This creates a matrix which can be used to calculate the x , y , and z positions of the end effector, as shown below:

B.1.1 Pitch-Yaw Position Equations

$$\begin{aligned}x &= \left[\frac{L \sin(\phi_{yaw})}{4} \right] \\y &= - \left[\frac{L \sin(\phi_{pitch})}{2} \right] - \left[\frac{L \cos(\phi_{yaw}) \sin(\phi_{pitch})}{4} \right] \\z &= \frac{L}{4} + \left[\frac{L \cos(\phi_{pitch})}{2} \right] + \left[\frac{L \cos(\phi_{pitch}) \cos(\phi_{yaw})}{4} \right]\end{aligned}$$

B.1.2 Pitch Roll Position Equations

$$\begin{aligned}x &= \left[\frac{L \sin(\beta) \sin(\phi)}{4} \right] \\y &= - \left[\frac{L \cos(\beta) \sin(\phi)}{4} \right] \\z &= \frac{3L}{4} + \left[\frac{L \cos(\phi)}{4} \right]\end{aligned}$$

B.1.3 Quaternion Position Equations

$$x = \left[\frac{L \sin(\beta) \sin(\phi)}{4} \right] + \left[\frac{L \sin(\beta) \sin(\phi)}{4 \cos\left(\frac{\phi}{2}\right)} \right]$$

$$y = \left[\frac{L \cos(\beta) \sin(\phi)}{4} \right] - \left[\frac{L \cos(\beta) \sin(\phi)}{4 \cos\left(\frac{\phi}{2}\right)} \right]$$

$$z = \frac{L}{4} + \left[\frac{L}{4 \cos\left(\frac{\phi}{2}\right)} \right] + \left[\frac{L \cos(\phi)}{4} \right] + \left[\frac{L \cos(\phi)}{4 \cos\left(\frac{\phi}{2}\right)} \right]$$

B.1.4 Spherical Position Equations

$$x = \left[\frac{L \sin(\beta) \sin(\phi)}{2} \right]$$

$$y = - \left[\frac{L \cos(\beta) \sin(\phi)}{2} \right]$$

$$z = \frac{L}{2} + \left[\frac{L}{4 \cos\left(\frac{\phi}{2}\right)} \right] + \left[\frac{L \cos(\phi)}{2} \right]$$

Appendix C Estimation of Quaternion Joint Performance

C.1 Derivation by Kim et al.

Kim et al, [83] derive the following expression to determine the theoretical radius of the hemisphere created by the quaternion mechanism.

$$r(\phi) = \frac{h_o + \sqrt{h_o^2 + \left(1 + \left(\frac{w_c}{h_c}\right)^2 \tan^2 \phi\right) \left(\left(\frac{w_c}{2}\right)^2 - h_o^2\right)}}{\cos \phi \left(1 + \left(\frac{w_c}{h_c}\right)^2 \tan^2 \phi\right)}$$

The constant values of $w_c = 30$, $l_c = 85.75$ and $h_o = 4.84$ are also provided. We therefore assume h_c is found using the Pythagorean theorem, $l_c^2 = h_c^2 + w_c^2$.

However, when inputting these values and solving across the range $0 \leq \phi \leq \frac{\pi}{2}$ we get significant errors, and an Δr error over 100%. Therefore, we assume that there is a mistake somewhere within the paper. Hence, we derive the equation ourselves, using the same method.

C.2 New Equation Derivation

We derive the equation using the same methodology. First, we define an ellipse, with the centre p_o at the point $(0, -h_c)$, and a point on that ellipse p_c , with coordinates (x_c, y_c) . This point is a vector $r(\phi)$ from the centre of the ellipse, which is an angle ϕ from the y-axis.

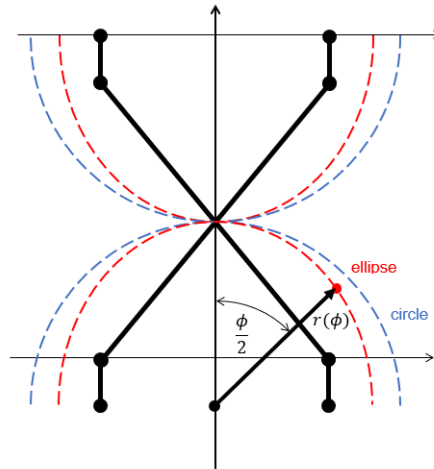


Figure C.1. Quaternion Mechanism Ellipses Approximating Rolling Motion.

We begin with the ellipse equation:

$$\frac{4x_c^2}{l_c^2} + \frac{4y_c^2}{h_c^2} = 1$$

We can re-arrange this, to solve for y_c .

$$\left(\frac{y_c}{h_c}\right)^2 = \frac{1}{4} - \left(\frac{x_c}{l_c}\right)^2$$

The equation of the line will be:

$$y_c = \frac{1}{\tan \phi} x_c - h_o$$

We can then insert the equation for y_c .

$$\left(\frac{1}{\tan \phi} x_c - h_o\right)^2 = h_o^2 \left(\frac{1}{4} - \left(\frac{x_c}{l_c}\right)^2\right)$$

We then arrange this in terms of x_c .

$$\left(\frac{1}{\tan^2 \phi} + \left(\frac{h_c}{l_c}\right)^2\right) x_c^2 - \left(\frac{2h_o}{\tan \phi}\right) x_c + \left(h_o^2 - \frac{h_c^2}{4}\right) = 0$$

We use the quadratic formula to solve this equation for x_c .

$$x_c = \frac{\frac{2h_o}{\tan \phi} + \sqrt{\frac{4h_o^2}{\tan^2 \phi} - 4\left(\frac{1}{\tan^2 \phi} + \left(\frac{h_c}{l_c}\right)^2\right)\left(h_o^2 - \frac{h_c^2}{4}\right)}}{2\left(\frac{1}{\tan^2 \phi} + \left(\frac{h_c}{l_c}\right)^2\right)}$$

We can simplify this significantly.

$$x_c = \frac{\sin \phi \left(h_o + \sqrt{h_o^2 + \left(1 + \tan^2 \phi \left(\frac{h_c}{l_c}\right)^2\right) \left(\left(\frac{h_c}{2}\right)^2 - h_o^2\right)} \right)}{\cos \phi \left(1 + \tan^2 \phi \left(\frac{h_c}{l_c}\right)^2 \right)}$$

We can also define the length of the line r as:

$$r(\phi) = \frac{x_c}{\sin \phi}$$

Therefore, we have the final equation:

$$r(\phi) = \frac{h_o + \sqrt{h_o^2 + \left(1 + \tan^2 \phi \left(\frac{h_c}{l_c}\right)^2\right) \left(\left(\frac{h_c}{2}\right)^2 - h_o^2\right)}}{\cos \phi \left(1 + \tan^2 \phi \left(\frac{h_c}{l_c}\right)^2\right)}$$

Appendix D SASAR 2 Costings

Table D-1. SASAR 2 Joint Module Costs

Component	Unit Quantity	Cost per Unit [£]	Total Cost [£]
COTS Components			
MR63 Bearing	4	1.88	7.52
MF128ZZ Bearing	2	2.88	5.76
Hobby Geared Motor	2	3.00	6.00
Samtec 10 Pin Male Connector	3	1.50	4.50
Samtec 10 Pin Female Connector	1	1.89	1.89
Samtec 10 Pin Male-Male Wire	1	4.70	4.70
Microcontroller - ATMEGA 32UR4RC-AUR	1	3.03	3.03
CAN Controller - MCP2515	1	1.47	1.47
CAN Transceiver - TJA105T	1	0.99	0.99
IMU - MPU-6050	1	4.91	4.91
Magnetic Position Sensor - AS5048-B	2	6.29	12.58
Motor Driver - DRV8830	2	1.53	3.06
100nF Capacitor	8	0.25	2.00
18pF Capacitor	4	0.03	0.12
1nF Capacitor	2	1.13	2.26
2.2nF Capacitor	2	0.08	0.16
2.2uF Capacitor	2	0.23	0.46

22pF Capacitor	4	0.41	1.64
10k Ohm Resistor	3	0.04	0.12
1k Ohm Resistor	1	0.07	0.07
0.2 Ohm Current Sense Resistor	2	0.62	1.24
8Mhz Crystal	1	0.48	0.48
Polyamide Tensile Thread (per metre)	0.5	0.054	0.03
M2 Nut	37	0.05	1.94
M2 x 5 Countersunk Bolt	2	0.04	0.09
M2 x 6 Bolt	17	0.04	0.71
M2 x 10 Countersunk Bolt	4	0.05	0.19
M2 x 10 Bolt	8	0.03	0.22
M2 x 12 Bolt	6	0.05	0.29
2 x 8 mm Steel Dowel Pin	6	0.08	0.47
Bespoke Manufacture Components			
Intra-Module PCB Board	2	0.07	0.14
Inter-Module Pitch PCB Board	1	0.07	0.07
Inter-Module Yaw PCB Board	1	0.07	0.07
Main Joint PCB	2	1.02	2.04
3D Printed Components			
Drum	2	£0.03	£0.06
Drum Sled	2	£0.08	£0.16
Housing Collars	4	£0.05	£0.20
Housing Faceplates	2	£0.07	£0.14
Main Housing	2	£0.58	£1.16
Mechanism End Plate	2	£0.16	£0.32
Mechanism Joints	6	£0.01	£0.06
Mechanism Linkage Arm	3	£0.05	£0.15
TOTALS			
COTS Components			£70.38
Bespoke PCBs			£2.32
3D Printed Components			£2.47
TOTAL			£75.17

Note: When costing the 3D printed components, only the cost of the material is calculated. This value is calculated in the 3D printer slicer software, by inputting the value of 1 kg of filament, or 1 L of resin. For SASAR 2 the selected PLA filament is £15/kg, and the UV resin is £25/L.