FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Robotic System for Arbutus Berry Harvesting

Tomás Rodrigo Ribas Silva

Master in Electrical and Computer Engineering

Supervisor: Prof. Filipe Neves Santos

July 31, 2026

© Tomás Silva, 2026

Resumo

A crescente procura global por uma produção alimentar sustentável, associada à escassez de mão de obra, às pressões climáticas e à limitação de recursos, tem impulsionado avanços significativos na robótica agrícola. Com base em trabalho anterior sobre sistemas robóticos de colheita automatizada de medronhos assente em perceção visual, esta dissertação foca-se na concretização desse sistema em hardware físico, com particular ênfase no projeto e avaliação do manipulador robótico.

Enquanto o trabalho anterior estabeleceu uma arquitetura baseada em perceção, integrando sensores LiDAR, câmaras RGB-D e técnicas de *deep-learning* para deteção, localização e avaliação do grau de maturação dos frutos, a presente investigação procura fazer evoluir o sistema no sentido da sua aplicação em contexto real. A contribuição central consiste no desenvolvimento de um manipulador e na formulação do respetivo modelo cinemático, que permite o posicionamento do efetuador em coordenadas-alvo no espaço de trabalho. Trata-se de uma etapa fundamental para a futura colheita seletiva de medronhos, cujo medronheiro apresenta uma morfologia complexa, com copas densas, ramos frágeis e uma distribuição não uniforme dos frutos.

A validação do sistema foi realizada em ambiente laboratorial, demonstrando a capacidade do manipulador de atingir as coordenadas-alvo no seu espaço de trabalho. Estes resultados constituem uma base para sistemas robóticos de colheita fisicamente implementáveis, dirigidos a culturas não convencionais como o medronho.

Abstract

The growing global demand for sustainable food production, coupled with labour shortages, climate pressures, and limited resources, has driven significant progress in agricultural robotics. Building upon previous work on vision-based robotic systems for the automated harvesting of arbutus berries, this dissertation focuses on realizing such a system on physical hardware, with particular emphasis on the design and evaluation of the robotic manipulator.

While previous work established a perception-based framework integrating LiDAR sensors, RGB-D cameras, and *deep-learning* techniques for fruit detection, localization, and ripeness assessment, the present research seeks to advance the system toward real-world application. Its central contribution lies in the development of a manipulator and the formulation of its kinematic model, which enables the positioning of the end-effector at target coordinates within the workspace. This is a fundamental step toward the future selective harvesting of arbutus berries, whose tree presents a complex morphology of dense canopies, fragile branches, and a non-uniform fruit distribution.

The system was validated in a laboratory environment, demonstrating the manipulator's ability to reach target coordinates within its workspace. These results provide a foundation for physically deployable harvesting robots aimed at unconventional crops such as the arbutus berry.

UN Sustainable Development Goals

The United Nations Sustainable Development Goals (SDGs) provide a global framework to achieve a better and more sustainable future for all. It includes 17 goals[1] to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice.

This dissertation contributes to the advancement of robotic systems for agricultural harvesting, with particular emphasis on the development and evaluation of a robotic manipulator. By enabling the precise positioning required for the future selective harvesting of arbutus berries, the work supports the broader transition toward more efficient, less wasteful, and more sustainable agricultural practices. Its contributions relate primarily to the following Sustainable Development Goals:

SDG 2 End hunger, achieve food security and improved nutrition and promote sustainable agriculture

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

SDG 12 Ensure sustainable consumption and production patterns

SDG	Target	Contribution	Performance Indicators and Metrics
2	2.4	Automated harvesting addresses agricultural labour shortages and supports the sustainable production of arbutus berries, an underexploited native crop.	Feasibility of automating a labour-intensive task; applicability of the approach to non-conventional crops.
9	9.5	The development and laboratory validation of a robotic manipulator and its kinematic model advance the technological capabilities available for agricultural automation.	Positioning accuracy within the workspace; demonstration of kinematic control on physical hardware beyond simulation.
12	12.2	Precise, controllable manipulation lays the groundwork for selective harvesting that targets only ripe fruit, reducing damage to plants and minimizing crop waste.	Reduction in unintended plant contact during positioning; potential decrease in fruit loss enabled by selective targeting.

[1] https://sdgs.un.org/goals

Acknowledgements

I would like to thank both my supervisors, Prof. Filipe Neves dos Santos and Prof. Sandro Augusto Magalhães, for their availability and for all the help and guidance they provided throughout this work.

I am also grateful to all the fellow researchers at *TRIBE Lab* for their help and for the welcoming environment they created.

Finally, I would like to thank my parents for their support.

Tomás Silva

Contents

1 Introduction **1**
- 1.1 Context and Motivation . . . 1
- 1.2 Arbutus Berry Harvesting and Its Challenges . . . 1
- 1.3 Problem Statement . . . 2
- 1.4 Objectives and Contributions . . . 3
- 1.5 Dissertation Structure . . . 3

2 Background **5**
- 2.1 Agricultural Robotics . . . 5
- 2.2 Arbutus Berry Harvesting: Physical and Mechanical Constraints . . . 6
- 2.3 Manipulators for Harvesting: From Rigid to Continuum . . . 6
- 2.4 Kinematics of Continuum Manipulators . . . 7
- 2.5 Actuation of Continuum Manipulators . . . 8
- 2.6 Communication for Distributed Actuation . . . 9
- 2.7 End-Effectors for Agricultural Harvesting . . . 10

3 State of the Art **11**
- 3.1 Robotic Harvesting Systems . . . 11
  - 3.1.1 Infrastructure-Guided Harvesting Robots . . . 11
  - 3.1.2 Mobile Single-Arm Outdoor Robots . . . 12
  - 3.1.3 Specialized Robots for Densely Clustered Fruits . . . 13
  - 3.1.4 Multi-Arm Collaborative Systems . . . 14
  - 3.1.5 Discussion . . . 15
- 3.2 Manipulation in Cluttered Canopies . . . 15
- 3.3 Continuum and Soft Manipulators . . . 16
- 3.4 Harvesting End-Effectors . . . 17
- 3.5 Vision-Based Perception . . . 19
- 3.6 Research Gaps and Positioning of This Work . . . 20
- 3.7 Summary . . . 20

4 Proposed Solution **21**
- 4.1 Design Requirements and Constraints . . . 21
- 4.2 Design Rationale . . . 22
- 4.3 Design Iterations . . . 23
  - 4.3.1 First Iteration: Elastic-Tube Backbone . . . 24
  - 4.3.2 Second Iteration: Articulated-Joint Backbone . . . 25
- 4.4 Final Design . . . 27
  - 4.4.1 Backbone and Sections . . . 28

4.4.2 Spacer Disks and Tendon Offset . . . 29
4.4.3 Tendon Layout . . . 29
4.4.4 Actuation Base . . . 30
4.5 Summary . . . 31

5 Kinematic Modelling **32**
5.1 Constant-Curvature Model . . . 32
5.1.1 Arc Parameters . . . 33
5.1.2 The Two-Mapping Decomposition . . . 33
5.1.3 Modelling Assumptions . . . 34
5.2 Forward Kinematics . . . 35
5.2.1 Geometry of a Section in the Bending Plane . . . 35
5.2.2 Tip Position in Three Dimensions . . . 36
5.2.3 Tip Orientation . . . 37
5.2.4 Homogeneous Transformation of a Section . . . 38
5.2.5 The Straight-Section Limit . . . 38
5.2.6 Composition of the Two Sections . . . 39
5.3 Cable-to-Arc Mapping . . . 39
5.3.1 Length Change of a Single Tendon . . . 39
5.3.2 Antagonistic Pairs and Bending in a Plane . . . 40
5.3.3 Recovering the Arc Parameters . . . 41
5.3.4 From Motor Rotation to Tendon Displacement . . . 43
5.3.5 Limitations of the Mapping . . . 43
5.4 Inter-Section Coupling . . . 44
5.4.1 Arc Parameters as Orthogonal Curvatures . . . 44
5.4.2 Coupling as a Rotation of the Curvature Components . . . 44
5.4.3 Equivalent View in Cable Length . . . 45
5.4.4 Significance . . . 45
5.5 Inverse Kinematics . . . 46
5.5.1 Single-Section Analytic Inverse . . . 46
5.5.2 Two-Section Numerical Inverse . . . 46
5.6 Summary . . . 48

6 Hardware Implementation **50**
6.1 System Architecture Overview . . . 50
6.2 Actuation: Motors and Drivers . . . 51
6.2.1 Actuation Chain . . . 51
6.2.2 Stepper Motors . . . 52
6.2.3 Drivers . . . 52
6.2.4 Step Resolution . . . 53
6.2.5 Controller and Actuation Boards . . . 53
6.3 Inter-Board Communication: CAN . . . 53
6.3.1 Selection of Communication Protocol . . . 54
6.3.2 Implementation . . . 55
6.4 Physical Prototype . . . 55
6.5 Summary . . . 56

7 Software Implementation **57**
7.1 Software Architecture . . . 57
7.2 Operating Modes . . . 58
7.2.1 Single-Section Mode . . . 58
7.2.2 Two-Section Mode . . . 58
7.2.3 Direct and Safety Commands . . . 58
7.3 Stepper Control . . . 59
7.4 Command Interface . . . 59
7.5 Control Flow . . . 60
7.6 Summary . . . 61

8 Experimental Validation and Results **62**
8.1 Experimental Setup . . . 62
8.2 Methodology . . . 63
8.3 Results . . . 64
8.4 Discussion . . . 67
8.5 Summary . . . 68

9 Conclusions and Future Work **69**
9.1 Conclusions . . . 69
9.2 Contributions . . . 70
9.3 Future Work . . . 70

References **72**

List of Figures

1.1 Arbutus unedo fruits at different ripeness stages 2

2.1 Arbutus berry 6
2.2 Intrinsic actuation on a continuum manipulator 8
2.3 Extrinsic actuation on a continuum manipulator 9

3.1 The SWEEPER pepper harvesting robot 12
3.2 Mobile single-arm harvesting robot 13
3.3 Cherry tomato harvesting robot 13
3.4 Dual-arm harvesting robot 14
3.5 Harvesting in densely clustered fruit 16
3.6 Example of continuum manipulator 17
3.7 End-effector for apple harvesting 18
3.8 Vacuum suction end-effector 18
3.9 High-tolerance soft-rigid gripper 18
3.10 Vision-based fruit detection 19

4.1 First design iteration: spacer disk 24
4.2 First design iteration: elastic-tube backbone 24
4.3 First design iteration: elastic-tube backbone 25
4.4 Candidate backbone joint 25
4.5 Final design in CAD 26
4.6 Motors disposition in CAD 27
4.7 Final manipulator design 28
4.8 Backbone and Sections 29
4.9 Tendon angular layout 30
4.10 View of the motors disposition 30

5.1 Constant-curvature arc parameters 33
5.2 Spaces and mappings 34
5.3 Arc in the bending plane 35
5.4 Derivation of the position components 36
5.5 Orientation of the bending plane 37
5.6 Composition of the two sections 39
5.7 Tendon length change under bending 40
5.8 Bending of each pair as a vector 42
5.9 Vector sum of the two pairs 42
5.10 Pulley of the motors 43

6.1 System architecture 51
6.2 Actuation chain 52
6.3 Controller and actuation boards 53
6.4 Control and actuation electronics 56

7.1 Software control flow 61

8.1 Positioning test cases 66

List of Tables

4.1 Design requirements and constraints for the manipulator. 22

6.1 Specifications of the SF2424-10B41 stepper motor. 52
6.2 Comparison of candidate inter-board communication protocols. 54
6.3 CAN message set. 55

7.1 USB-serial command set accepted by the master. 59

8.1 Commanded targets and measured positioning error for the eight cases. 64

List of Acronyms

CAD	Computer-Aided Design
CAN	Controller Area Network
DLS	damped least squares
DoF	degrees of freedom
I^2C	Inter-Integrated Circuit
LiDAR	Light Detection and Ranging
RGB	Red, Green, Blue
RGB-D	Red, Green, Blue plus Depth
SMA	Shape-Memory Alloy
SPI	Serial Peripheral Interface
TRIBE	Laboratório de Robótica e IoT para Agricultura e Floresta de Precisão Inteligente
UART	Universal Asynchronous Receiver-Transmitter
USB	Universal Serial Bus

List of Symbols

Geometry and arc parameters

κ	Curvature of a section, $\kappa = 1/r$; a straight section has $\kappa = 0$
r	Radius of curvature of a section, $r = 1/\kappa$
ϕ	Bending-plane angle, orienting the plane of the arc about the base z-axis
θ	Bending angle of a section, $\theta = \kappa\ell$
ℓ	Arc length of a section, measured along the backbone
L	Total backbone length, used in single-section mode (whole backbone as one arc)
$\kappa_x,\ \kappa_y$	Orthogonal curvature components, $\kappa_x = \kappa\cos\phi$, $\kappa_y = \kappa\sin\phi$
$\kappa^*_{i,x},\ \kappa^*_{i,y}$	Curvature components of section i corrected for inter-section coupling

Position, orientation, and pose

x, y, z	Cartesian coordinates of the tip in the base frame
$p_\parallel$	Tip position component along the tangent (z-axis), $p_\parallel = r\sin\theta$
$p_\perp$	Tip position component in the bending plane, $p_\perp = r(1-\cos\theta)$
$\mathbf{p}$	Tip position vector, $\mathbf{p}(\kappa,\phi,\ell)$
$\mathbf{R}$	Rotation matrix of the tip frame relative to the base frame
$\mathbf{R}_z(\phi), \mathbf{R}_y(\theta)$	Elementary rotations about the z- and y-axes
$\mathbf{T}$	Homogeneous transformation of a section, $\mathbf{T}(\kappa,\phi,\ell)$
$\mathbf{T}^0_d$	Pose of the distal tip expressed in the base frame
$\mathbf{I}$	Identity matrix
$\mathbf{0}$	Zero vector

Tendons and actuation

d	Radial offset of the tendons from the central axis ($d = 28\,\text{mm}$)
ρ	Pulley radius ($\rho = 8\,\text{mm}$)
α	Motor rotation angle
τ	Motor holding torque
F	Tendon force, $F = \tau/\rho$
$\Delta\ell$	Length change of a tendon, $\Delta\ell = \pm d\theta = \rho\alpha$
$\ell_{\text{in}}, \ell_{\text{out}}$	Length of the inner / outer tendon of a pair under bending
θ_A, θ_B	Bending angles produced by antagonistic pairs A and B
$\Delta\ell_A, \Delta\ell_B$	Length changes of the tendons of pairs A and B
β_{d}	Angular position of a distal tendon on the disc
α_{d}	Rotation of a distal motor

$\Delta\ell_{\text{bend}}$	Tendon displacement producing distal bending (coupling-corrected)
$\Delta\ell_{\text{c}}$	Tendon length consumed by the proximal bend (coupling term)

Inverse kinematics

$\mathbf{q}$	Configuration vector, $\mathbf{q} = [u_{x,p}, u_{y,p}, u_{x,d}, u_{y,d}]^\top$
$\mathbf{u}$	Cartesian bending components of a section, $\mathbf{u} = \theta\,[\cos\phi, \sin\phi]^\top$
u_x, u_y	Components of $\mathbf{u}$
$\mathbf{s}(\mathbf{q})$	Tip position given by the forward kinematics
$\mathbf{t}$	Target tip position
$\mathbf{e}$	Position error, $\mathbf{e} = \mathbf{t} - \mathbf{s}(\mathbf{q})$
$\mathbf{J}$	Jacobian of the forward map, $\mathbf{J} = \partial\mathbf{s}/\partial\mathbf{q}$
$\Delta\mathbf{q}$	Configuration update per iteration
λ	Damping constant of the damped least-squares method
σ_i	Singular values of the Jacobian

Experimental evaluation

$\mathbf{p}_c$	Commanded tip position, (x_c, y_c, z_c)
$\mathbf{p}_m$	Measured (reached) tip position, (x_m, y_m, z_m)
Δx, Δy, Δz	Per-axis positioning error, $\mathbf{p}_m - \mathbf{p}_c$
$\|\mathbf{e}\|$	Euclidean positioning error, $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$

Operators and accents

$(\cdot)^\top$	Transpose of a vector or matrix
$(\cdot)^{-1}$	Inverse of a matrix
$(\cdot)^+$	Moore–Penrose pseudoinverse
$(\cdot)^*$	Quantity corrected for inter-section coupling
$\|\cdot\|$	Euclidean norm
$\operatorname{atan2}(\cdot,\cdot)$	Two-argument arctangent

Notation

$\mathbf{M}$	Bold uppercase letters denote matrices
$\mathbf{v}$	Bold lowercase letters denote vectors
a	Italic letters denote scalars
$(\cdot)_p$, $(\cdot)_d$	Subscripts p and d refer to the proximal and distal sections
$(\cdot)_A$, $(\cdot)_B$	Subscripts A and B refer to the two antagonistic pairs of a section
$(\cdot)_c$, $(\cdot)_m$	Subscripts c and m refer to commanded and measured quantities

Chapter 1

Introduction

1.1 Context and Motivation

Agricultural production faces increasing pressure to improve productivity and sustainability while operating under constrained resources and a reduced availability of manual labour. The growing adoption of automated technologies reflects a larger shift toward precision agriculture, where sensing, monitoring, and intelligent decision-making are used to improve efficiency and reduce waste. In this context, robotic systems have emerged as an important technology to support agricultural operations and reduce dependence on labour-intensive practices [1], [2], [3].

Despite significant progress, harvesting remains one of the most challenging tasks to automate. Unlike structured industrial environments, agricultural fields are inherently unstructured, with variable lighting conditions, foliage-induced occlusions, and highly irregular plant geometries. These characteristics complicate perception, navigation and manipulation, and surveys consistently identify harvesting as a task for which robust operation under real-world conditions remains an open challenge [4], [5]. Beyond perception and planning, the physical interaction between the robot and the plant represents a critical bottleneck in practical deployment where harvesting mechanisms must safely handle delicate fruits while tolerating positioning errors and operating within constrained and cluttered canopies [6], [7].

1.2 Arbutus Berry Harvesting and Its Challenges

This work focuses on the automated harvesting of the arbutus berry (*Arbutus unedo*). The arbutus is a shrub of notable ecological and economic value. Its fruit is used as raw material for products such as arbutus spirit (*medronho*), honey and jams, and it is also valued in confectionery for its antioxidant properties [8], [9]. Ecologically, the plant's high resilience to fire, rapid regeneration capacity, and tolerance of poor soils make it valuable for establishing forests that are more resilient to wildfire [8], [10].

From a robotics standpoint, however, the arbutus tree is a demanding target. Its canopy is dense and rounded, its branches are thin and flexible, and its fruit is distributed irregularly within

the foliage, increasing occlusions and the likelihood of unintended contact during manipulation. A further distinctive feature is the simultaneous presence of flowers and fruits at different ripening stages on the same branches, which requires selective harvesting that collects ripe berries while preserving the flowers that contribute to future yield (Figure 1.1). These morphological characteristics prevent harvesting robots developed for more structured crops from being directly transferred to the arbutus, motivating the development of a dedicated solution.

Figure 1.1: *Arbutus unedo* [11].

1.3 Problem Statement

Modern robotic harvesting relies heavily on vision-based perception for fruit detection, localization and ripeness assessment, and advances in Red, Green, Blue (RGB), Red, Green, Blue plus Depth (RGB-D); learning-based methods have shown that selective harvesting is feasible at the perception level even under occlusion and variable lighting [4], [5]. At this level of maturity, the principal obstacle is no longer recognizing the fruit but reliably acting on that information. A system that detects fruit accurately is of limited use unless a manipulator can position itself at the corresponding location in the workspace, safely and repeatably.

This transition from perception to physical action is the open problem left by prior research on arbutus berry harvesting. The work of Pereira [11] established a perception-driven framework integrating Light Detection and Ranging (LiDAR) and RGB-D sensing with deep-learning models for fruit detection, localization and ripeness classification. A robotic manipulator and harvesting strategies *in simulation* were evaluated. While that work demonstrated the feasibility of the perception pipeline and released annotated datasets for the arbutus berry, its manipulation component was validated exclusively in simulation and was not realized in physical hardware.

Building on the same perception context, this dissertation addresses the physical manipulation problem through a different line of approach. Rather than a rigid serial manipulator, it investigates a continuum manipulator, whose distributed compliance and shape adaptability make it a candidate for operating within the dense and irregular canopy of the arbutus tree. The work focuses

on designing such a manipulator, building a physical prototype, formulating and implementing its kinematic model, and verifying that it can be commanded to reach target coordinates in its workspace under laboratory conditions.

1.4 Objectives and Contributions

The main objective of this dissertation is to advance the vision-based arbutus berry harvesting system from simulation toward physical realization, focusing on the robotic manipulator and its kinematic control. This general objective is divided into the following specific objectives:

- Design a robotic manipulator suited to the physical and morphological constraints of arbutus berry harvesting, exploring multiple design iterations;
- Build a physical prototype of the selected manipulator design;
- Formulate and implement the manipulator's kinematic model, covering both forward and inverse kinematics;
- Experimentally validate the accuracy of the manipulator's positioning under laboratory conditions.

Accordingly, this work makes four main contributions. The first is the design of an arbutus-harvesting manipulator developed through successive iterations. The second is the construction of a physical prototype that extends a previously simulation-only system to real hardware. The third is the implementation of its kinematic model for commanded positioning of the end-effector. The fourth is an experimental evaluation of the prototype's positioning accuracy. Together, these validate the manipulator's positioning capability and provide a foundation for future field deployment. The selective harvesting of arbutus berries itself is regarded as the longer-term goal that this work is intended to enable, and is discussed as a direction for future work.

1.5 Dissertation Structure

In addition to this introduction, this dissertation is organized as follows.

- **Chapter 2 — Background** introduces the core concepts needed to follow the work, agricultural robotics, the mechanical constraints of arbutus harvesting, continuum manipulators and their kinematics, as well as the supporting communication technologies.
- **Chapter 3 — State of the Art** reviews robotic harvesting platforms, manipulation in cluttered canopies, continuum and soft manipulators, end-effectors, and the inherited perception stack, and positions the contribution of this work.
- **Chapter 4 — Proposed Solution** defines the design requirements, the rationale for a tendon-driven continuum manipulator, the design iterations, and the final design.

- **Chapter 5 — Kinematic Modelling** develops the constant-curvature model, the forward and inverse kinematics, the cable-to-arc mapping, and the inter-section coupling.
- **Chapter 6 — Hardware Implementation** describes the actuation, the distributed control electronics, and the assembled physical prototype.
- **Chapter 7 — Software Implementation** presents the control architecture, stepper control, command interface, and control flow.
- **Chapter 8 — Experimental Validation and Results** reports the experimental setup, methodology, and positioning-accuracy results.
- **Chapter 9 — Conclusions and Future Work** summarizes the achievements against the objectives and outlines future work, including the end-effector, perception integration, and field harvesting.

Chapter 2

Background

This chapter introduces the concepts required to understand the work developed in this dissertation. It is not intended as an exhaustive treatment of each topic, but rather as the minimal background needed to follow the design and evaluation of the proposed manipulator. It covers agricultural robotics and the specific mechanical constraints of arbutus harvesting, the manipulator and kinematics fundamentals on which the work builds, and the supporting technologies used in the physical implementation.

2.1 Agricultural Robotics

Agricultural robotics is an interdisciplinary field that combines mechanical design, sensing, control, and artificial intelligence to enable autonomous or semi-autonomous operation in agricultural environments. In contrast to industrial robotics, which typically operates in structured and predictable settings, agricultural robots must function under unstructured, dynamic, and highly variable conditions, where plant geometry, terrain, and environmental factors change continuously [1], [12].

Robotic systems in agriculture support tasks such as monitoring, spraying, pruning, and harvesting. Although these tasks differ in their requirements, they share common challenges related to limited sensing visibility, environmental uncertainty, and physical interaction with deformable and fragile biological structures. For this reason, agricultural robots are generally designed as integrated systems that combine mobility, perception, manipulation, and task-specific actuation [3], [13]. A typical harvesting robot can be described in terms of four main subsystems: a locomotion or positioning platform, a perception module for crop and environment understanding, a manipulation subsystem for reaching the target, and an end-effector responsible for the physical interaction with the plant. The effectiveness of the overall system depends not only on each component individually, but on their integration and coordinated operation [12]. Beyond perception and planning, harvesting requires direct physical interaction with delicate targets under occlusion and positioning uncertainty, which makes the manipulation and end-effector subsystems decisive for success in real conditions.

2.2 Arbutus Berry Harvesting: Physical and Mechanical Constraints

The arbutus tree (*Arbutus unedo*) exhibits physical and structural characteristics that directly influence the mechanical interaction between a harvesting system and the plant. Its fruits are approximately spherical, with relatively small diameters and textured surfaces, and are attached to thin, flexible peduncles connected to fragile branches. These branches are highly compliant and deflect noticeably when external forces are applied during manipulation [11].

A distinctive feature of the arbutus tree is the simultaneous presence of flowers and fruits at different ripening stages along the same branches. Mechanically, this produces a heterogeneous distribution of elements with varying stiffness, size, and attachment strength within a confined region, while the fruits themselves are often partially embedded in foliage and surrounded by leaves and smaller branches [8], [9] (Figure 2.1). The compliance of both branches and fruit attachments introduces variability in the plant's response. When a force is applied to a target fruit, neighbouring structures may deform or move, altering the relative position of the fruit and the elements around it.

From a manipulation standpoint, these characteristics impose constraints on the admissible contact forces, the directions of interaction, and the required positioning accuracy. Rigid or force-intensive interactions can propagate motion throughout the branch structure and compromise harvesting, particularly within the dense and irregular regions of the canopy. Effective harvesting therefore requires a manipulation approach that accommodates the deformable nature of the plant and the limited structural support offered by its branches.

Figure 2.1: *Arbutus berry* in different ripeness stages [11].

2.3 Manipulators for Harvesting: From Rigid to Continuum

The manipulation subsystem is responsible for bringing the end-effector to the fruit, and its design strongly influences a harvesting robot's ability to operate in cluttered canopies. Manipula-

tors used in agriculture broadly fall along a spectrum, from rigid-link mechanisms to continuum and soft structures, each with distinct compromises.

Rigid serial manipulators, formed by links connected through discrete revolute or prismatic joints, are the conventional choice. Their kinematics and control are well understood, and they provide high precision and payload capacity. However, because their motion is concentrated at the joints, reaching targets deep inside a dense canopy is difficult since the rigid links are prone to colliding with branches and foliage, and access to occluded fruit is limited [12], [14].

Continuum and soft manipulators occupy the opposite end of the spectrum. Instead of discrete joints, they consist of a continuous, flexible backbone along which compliance and motion are distributed. Inspired by biological structures such as elephant trunks and tentacles, they bend smoothly along their length and conform to their surroundings [15], [16]. This distributed compliance reduces the risk of damage when navigating through dense vegetation and allows safer interaction with flexible plant structures, which makes continuum mechanisms attractive for operation in constrained and cluttered environments [17], [18]. In the specific context of selective harvesting, however, continuum mechanisms have been explored considerably less than rigid ones, and their potential for reaching fruit in cluttered canopies remains an open research direction [19].

A common way to actuate a continuum manipulator is through tendons (cables) routed along its length and pulled from the base. By coordinating the tension in different cables, the structure can be made to bend in different directions, without placing actuators along the backbone itself. Multi-section designs extend this principle by stacking independently actuated segments, increasing the range of shapes the manipulator can adopt at the cost of more complex actuation and coupling between segments.

2.4 Kinematics of Continuum Manipulators

Kinematics describes the relationship between a manipulator's actuation and the pose of its end-effector, without considering the system dynamics involved. Two problems are usually distinguished. *Forward kinematics* computes the end-effector pose from known actuation values, and generally has a unique solution. *Inverse kinematics* solves the opposite, and more practically relevant, problem: determining the actuation required to place the end-effector at a desired pose. The inverse problem is typically harder, as it may admit multiple solutions, no solution, or singular configurations.

For rigid manipulators, the kinematic model is commonly built using the Denavit–Hartenberg convention, which assigns coordinate frames to successive rigid links. This formulation, however, does not apply to continuum structures, whose continuous bending cannot be described by a finite set of rigid links and discrete joints. Instead, continuum manipulators are most often modelled under the *constant-curvature* assumption, in which each section is approximated as an arc of constant curvature. Under this assumption, the configuration of a section can be captured by a small number of arc parameters, which can in turn be related to the actuation (for example, cable lengths) on one side, and to the end-effector pose on the other [20].

When an analytic inverse-kinematics solution is unavailable or impractical, numerical methods are used instead. A widely adopted approach is the damped least squares (DLS) method, which iteratively updates the configuration using a regularized inverse of the manipulator's Jacobian. The damping term improves numerical stability near singularities, at the cost of a small steady positioning error, making the method robust for reaching targets within the manipulator's workspace [21].

2.5 Actuation of Continuum Manipulators

A defining design decision for any continuum manipulator is how its continuous backbone is actuated. Actuation methods are commonly divided into *intrinsic* schemes, in which the actuators are distributed along the backbone itself, and *extrinsic* schemes, in which the actuators are located at the base and transmit motion through the structure [22]. The choice of actuation strongly influences the manipulator's achievable workspace, payload, controllability, and suitability for a given environment.

Intrinsic actuation includes pneumatic and hydraulic chambers, Shape-Memory Alloy (SMA), electroactive polymers, and other materials. Pneumatic and fluidic actuators can produce large deformations and inherent compliance, and are widely used in soft robots, but they require compressors, pressure regulators, and relatively complex control, and tend to suffer from limited bandwidth, vibration, and reduced positioning accuracy [22]. SMA and other smart-material actuators allow highly compact designs, but face challenges in response time, hysteresis, and thermal management (Figure 2.2). These methods are best suited to applications where compliance and miniaturization dominate over precise positioning and payload.

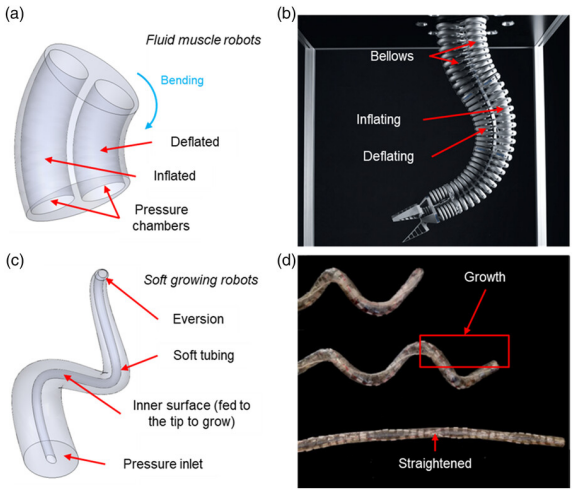


Figure 2.2: Intrinsic actuation [22]

Extrinsic actuation is dominated by two families: concentric-tube robots and tendon-driven

robots [22]. Concentric-tube robots are formed by nested, pre-curved elastic tubes whose combined shape is controlled by rotating and translating each tube. They excel at small-diameter applications but offer limited payload and workspace for larger tasks. Tendon-driven continuum robots, by contrast, route sets of cables along an elastic backbone and actuate bending by pulling them from the base (Figure 2.3). This approach is among the simplest and most effective to implement. An elastic backbone provides structural stability and restoring force, while base-mounted motors keep the actuated structure lightweight and allow large length-to-diameter ratios [23]. Compared with fluidic actuation, tendon drives offer faster response and more direct control, which has made them the predominant choice for continuum robots operating in confined and cluttered environments [22].

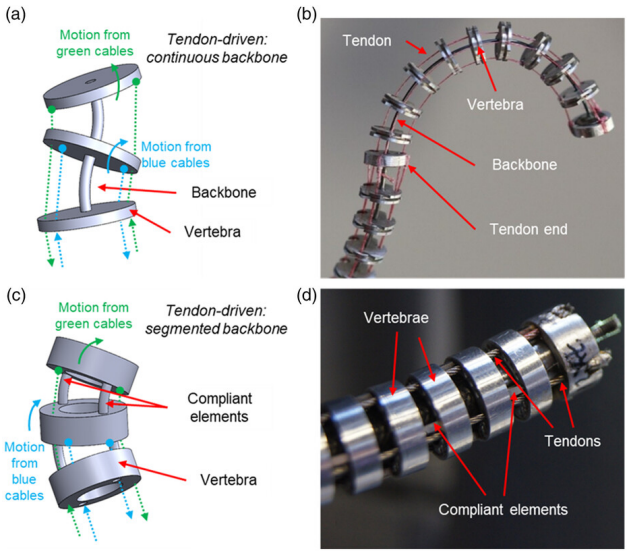


Figure 2.3: Extrinsic actuation [22]

Within tendon-driven designs, the arrangement of cables determines how each section can bend. A common and efficient configuration uses antagonistic tendon pairs, in which a single motor drives two opposing cables, paying out one as it pulls in the other. This halves the number of actuators relative to driving each cable independently, while still allowing controlled bending of a section in a given plane. The workspace and repeatability achievable with one motor per antagonistic pair have been characterized experimentally, confirming that this scheme provides a favourable balance between mechanical simplicity and positioning performance [23]. In multi-section manipulators, separate groups of tendons actuate each section, enabling richer three-dimensional shapes at the cost of additional actuators and inter-section coupling.

2.6 Communication for Distributed Actuation

When a manipulator's actuators are distributed across several microcontroller boards, those boards must exchange commands and status information reliably and in real time. Point-to-point

wiring between every pair of devices scales poorly and is sensitive to electrical noise, which motivates the use of a shared communication bus.

The Controller Area Network (CAN) is a serial communication bus originally developed for automotive systems and now widely used in robotics and industrial automation [24]. It is a multi-master, message-based protocol in which frames are identified by content rather than by node address, and bus access is arbitrated by message priority. Its differential signalling provides strong immunity to electrical noise, and its built-in error detection and fault confinement make it well suited to the distributed, noisy environment of a motor control system. These properties make CAN a natural choice for coordinating actuation across multiple controller boards, as is the case in the manipulator developed in this work.

2.7 End-Effectors for Agricultural Harvesting

The end-effector is the component that interacts directly with the crop, and its design determines whether a detected and reached fruit can actually be harvested. Agricultural end-effectors must operate on delicate, deformable, and biologically variable targets, often under occlusion, which makes their design strongly crop-dependent [12].

Several categories have been explored. Mechanical grippers are simple and provide high holding force, but depend on accurate fruit pose estimation and can bruise delicate produce [25]. Cutting and shearing tools achieve clean detachment for crops with well-defined peduncles, but require precise alignment. Suction-based end-effectors stabilize the fruit through negative pressure, improving tolerance to positioning errors and reducing contact forces [7]. To improve robustness, compliant and soft designs incorporate flexible materials or underactuated mechanisms that tolerate misalignment and reduce the risk of damage [6], [26]. No single design is universally applicable, and the choice of end-effector is closely tied to the morphology of the target crop. In this work, the end-effector itself is not the focus since it is regarded as part of the future harvesting capability that the proposed manipulator is intended to enable.

Chapter 3

State of the Art

This chapter reviews the current landscape of robotic systems applied to agricultural harvesting. It begins with a detailed overview of existing harvesting platforms, then narrows to the problem of manipulation within cluttered canopies, the continuum manipulators and kinematic models on which this work builds, and finally the perception and end-effector context inherited from prior research. The chapter closes by identifying the limitations of existing approaches and positioning the contribution of this dissertation.

3.1 Robotic Harvesting Systems

Selective harvesting robots have been developed for a wide range of specialty crops, including apple, kiwifruit, strawberry, sweet pepper, tomato, and others [27], [28], [29]. In contrast to bulk mechanical harvesting, these systems aim to pick fruit individually, using controlled mechanisms that reduce bruising and damage. Regardless of the target crop, a selective harvesting robot can generally be decomposed into three cooperating subsystems: a perception system for fruit detection and localization, a manipulation system for approaching and retracting from the fruit, and an end-effector responsible for detachment [12], [19].

Robotic harvesting platforms can be broadly categorized by their mobility and manipulation strategies, which are strongly influenced by crop structure and operating environment. The following subsections review examples across this spectrum, from infrastructure-supported systems in controlled environments to fully mobile and multi-arm robots, highlighting their respective strengths and limitations.

3.1.1 Infrastructure-Guided Harvesting Robots

In controlled environments such as greenhouses, harvesting robots often exploit existing infrastructure to constrain motion and reduce navigation complexity, allowing greater emphasis on

perception and manipulation performance. Rail or track guided platforms provide stable and repeatable positioning while simplifying localization and motion planning. A representative example is the SWEEPER sweet pepper harvesting robot of Arad et al. [29], developed for commercial greenhouses, which combined an industrial manipulator with a custom end-effector (Figure 3.1).

Figure 3.1: The SWEEPER pepper harvesting robot [29].

Modern high-density orchard structures, such as V-trellis and vertical fruiting-wall systems, pursue a similar philosophy in the field. By training trees into structured, planar layouts, they reduce occlusion and improve accessibility, greatly facilitating robotic harvesting [27]. However, while such structured environments simplify navigation and perception, harvesting performance remains strongly influenced by plant morphology, occlusion, and the interaction between the end-effector and the crop. Consequently, even in structured environments, manipulation strategy and end-effector design remain critical factors for successful robotic harvesting [29].

3.1.2 Mobile Single-Arm Outdoor Robots

Outdoor harvesting robots must operate in unstructured environments characterized by uneven terrain, variable lighting, and significant plant-to-plant variability. Many systems therefore adopt mobile platforms equipped with a single robotic arm, balancing mechanical simplicity with sufficient reach and dexterity for harvesting tasks. Single-arm configurations are commonly mounted on wheeled or tracked bases that navigate between crop rows, relying on vision-based perception for fruit detection and localization, followed by sequential manipulation actions [30] (Figure 3.2).

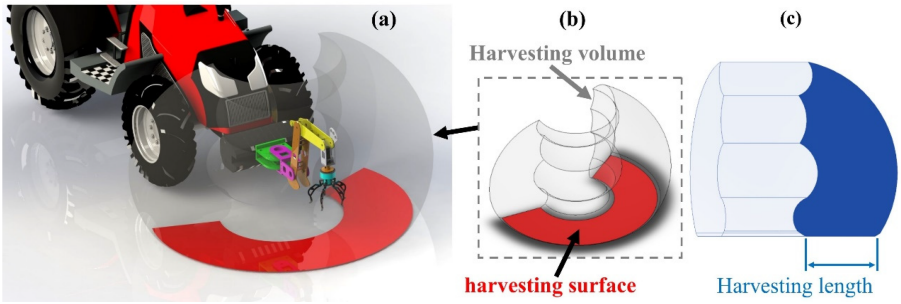


Figure 3.2: Mobile single-arm harvesting robot operating in an outdoor environment [30].

Such systems have been demonstrated across diverse crops. The autonomous pumpkin harvesting system of Roshanianfard et al. [30] integrated a robotic arm with a dedicated end-effector for heavy, ground-level fruit, while sweet pepper systems have combined single manipulators with cutting and suction mechanisms for greenhouse rows [31]. Field evaluations consistently demonstrate the feasibility of mobile single-arm harvesting, but also reveal a strong dependence on crop structure, end-effector capability, and positioning accuracy. While this approach reduces hardware complexity relative to multi-arm systems, it frequently struggles with occlusions, dense canopies, and constrained access to fruit clusters, which limits both success rate and harvesting speed [30], [31].

3.1.3 Specialized Robots for Densely Clustered Fruits

Some crops grow in dense clusters where fruits are tightly packed and mutually occluding, and several harvesting systems have been purpose-built for these conditions. As complete platforms, they typically pair a dedicated end-effector with a perception pipeline tuned for cluttered scenes, and are evaluated by their harvesting success and cycle time under realistic clustering. The cherry-tomato harvesting robot of Rong et al. [28] is representative of this approach (Figure 3.3). It combines a cam-driven end-effector that performs cutting and grasping from a single actuator with a coarse-to-fine visual pipeline for locating pedicel cutting points.

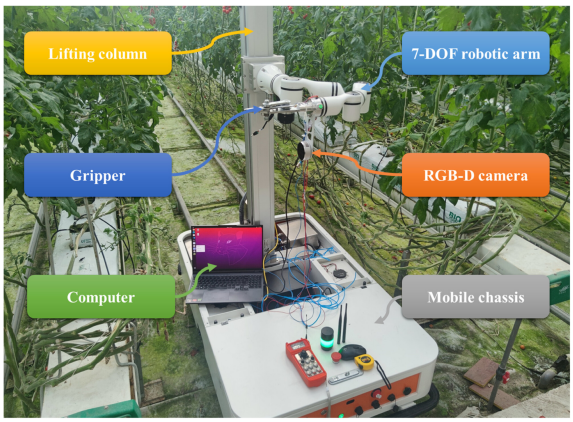


Figure 3.3: Cherry tomato harvesting robot [28].

Such systems demonstrate that hardware specialization can extend harvesting to crops that general-purpose platforms handle poorly, but their reported success rates remain markedly lower than those achieved on well-separated fruit. This reinforces that, in clustered scenarios, the limiting factor is typically not detection but the reliability of reaching and detaching the target fruit without disturbing its neighbours.

3.1.4 Multi-Arm Collaborative Systems

Multi-arm robotic systems have been explored to improve harvesting efficiency and manipulation robustness by coordinating multiple manipulators for tasks such as grasping, cutting, and obstacle handling. The dual-arm harvesting robot of Yoshida et al. [32] coordinates two arms to manage occluded fruit and accelerate the harvesting cycle, while the dual-arm mango harvesting system of Yin et al. [14] demonstrated improved task success and operational flexibility through integrated design and field evaluation (Figure 3.4). Apple harvesting has also benefited from multi-arm coordination. The dual-arm system of Zhang et al. [27] achieved a high field success rate in well-structured orchards, with the majority of fruit picked on the first attempt, by combining an occlusion-aware detector with coordinated arm motion.

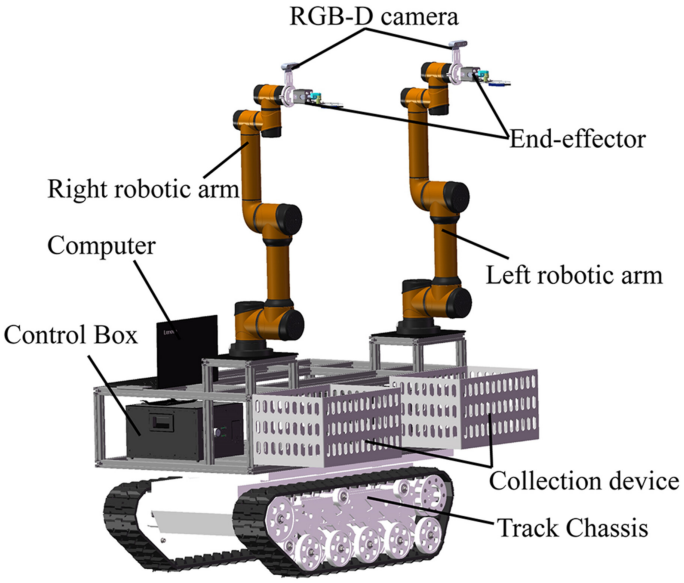


Figure 3.4: Dual-arm robotic harvesting system [14].

More recent work has extended multi-manipulator harvesting beyond fixed bases. The hierarchical tri-manual planning approach of Liu et al. [33] coordinates multiple arms mounted on a quadrupedal platform for vision-assisted fruit picking, combining mobility with redundant manipulation. However, multi-arm systems introduce increased mechanical complexity, higher control and synchronization requirements, and greater system cost. As a result, their deployment is often limited to structured environments or research platforms, motivating the investigation of simpler yet robust alternatives [14], [32].

3.1.5 Discussion

Across these categories, a consistent trade-off emerges between structural support and operational flexibility. Infrastructure-guided systems achieve reliable, repeatable positioning and simplified perception, but are restricted to environments and crops compatible with fixed structures. Mobile single-arm platforms offer far greater versatility at lower hardware cost, yet must contend with unstructured terrain, variable lighting, and heterogeneous plant morphologies, which degrade both success rate and speed. Specialized cluster-handling and multi-arm systems improve access and efficiency in cluttered canopies, but at the expense of mechanical complexity, control requirements, and cost. Regardless of platform architecture, the decisive limitation is increasingly the reliability of physical interaction with the crop within complex, occluded canopies, rather than the detection of the fruit itself.

3.2 Manipulation in Cluttered Canopies

The systems reviewed above differ in hardware, but they share a common underlying difficulty: once a target fruit has been detected and localized, the manipulator must reach it without colliding with the surrounding plant. In cluttered canopies this is a demanding problem in its own right, independent of the platform. Fruits are frequently occluded by leaves and branches, access paths are narrow, and the plant structures involved are fragile, so the reachability of a target depends not only on the manipulator's workspace but on its ability to follow a viable path to a fruit embedded in foliage [34].

Two broad strategies have emerged for handling the obstacles encountered along that path. The first treats surrounding structures as obstacles to be *avoided*, planning a collision-free trajectory around them using motion-planning techniques adapted for orchard and greenhouse environments [34]. Such approaches work well when a clear path to the fruit exists, but degrade sharply when the target is so densely surrounded that no collision-free trajectory remains. Analysing sweet-pepper harvesting in a dense obstacle environment, Bac et al. [34] observed that the manipulator occasionally collided with unseen plant parts, illustrating how reachability degrades as clutter increases.

The second strategy treats obstacles as elements to be *interacted with* rather than strictly avoided. Active obstacle-separation methods deliberately push or part neighbouring fruit and foliage to open an access path to the target. Xiong et al. [35] demonstrated this principle by pushing occluding fruit aside (Figure 3.5), and Mghames et al. [36] formulated the pushing of occluding pieces as an interactive motion-planning problem. These methods extend the set of reachable fruit beyond what pure avoidance allows, but they introduce deliberate contact with the plant, raising the risk of damage and demanding careful, compliant interaction.

Both strategies expose the same fundamental limitation. Rigid-link manipulators, whose segments cannot conform to their surroundings, are intrinsically constrained inside dense, fragile

canopies. Whether a path is planned around obstacles or forced through them, the rigid links remain prone to unintended collisions, and positioning must be accurate enough to avoid damaging the plant. This limitation motivates the exploration of manipulators that are inherently compliant and able to conform to the canopy.

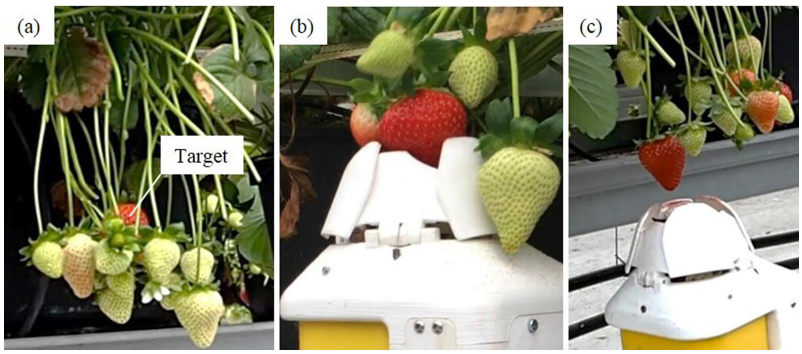


Figure 3.5: Robotic harvesting system for operation in densely clustered fruit [35].

3.3 Continuum and Soft Manipulators

As introduced in Section 2.3, continuum and soft manipulators differ fundamentally from conventional rigid-link arms: instead of concentrating motion at discrete joints, they distribute bending and compliance along a continuous, flexible backbone and conform to the shape of their surroundings [15], [16]. This intrinsic compliance makes them well suited to operation in constrained and cluttered environments, where contact with the surroundings is difficult to avoid. Rather than risking damage on collision, a continuum structure can deform around obstacles, reducing interaction forces and improving tolerance to positioning uncertainty [17], [18].

These properties are directly relevant to agricultural harvesting, where access to fruit is obstructed by dense, fragile vegetation. A small but growing body of work has begun to apply continuum and tendon-driven mechanisms to harvesting. The two-section tendon-driven continuum manipulator of Pachouri and Pathak [37] was designed explicitly for agricultural applications such as vegetable harvesting, using two serially stacked planar sections to position its tip in three dimensions while requiring relatively few actuators (Figure 3.6). Huertas Niño et al. [38] developed a hybrid tendon-driven continuum robot with a spring-and-disc backbone and a constant-curvature control law, demonstrating a payload suitable for fruit and vegetable harvesting while explicitly mitigating torsional deformation. Earlier, the elephant-trunk-inspired manipulator of Tiefeng et al. [15] showed the feasibility of continuum mechanisms for fruit harvesting.

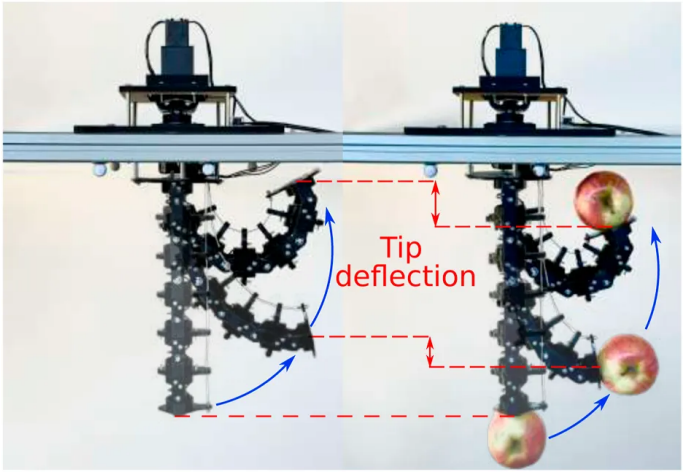


Figure 3.6: Continuum manipulator [37]

Despite these examples, continuum manipulators remain considerably less explored than rigid mechanisms in the harvesting domain. The recent review of selective harvesting by Rajendran et al. [19] notes that, while continuum mechanisms offer compliance and adaptability well suited to positioning an end effector in cluttered picking scenarios, they have received comparatively little attention relative to rigid arms. This gap, between the demonstrated advantages of continuum manipulators in other constrained domains and their limited adoption in agriculture, is one of the motivations for the present work.

3.4 Harvesting End-Effectors

The end-effector is the component that interacts directly with the crop and is ultimately responsible for detaching the fruit. Because it operates on delicate, deformable, and biologically variable targets, often under occlusion, its design is strongly crop-dependent, and a range of strategies have been developed, each with characteristic trade-offs [12], [19].

Mechanical clamping grippers are among the most common solutions, offering simplicity and high holding force, but they depend on accurate fruit pose estimation and can bruise delicate produce or damage stems when misaligned [25]. Cutting and shearing tools achieve clean detachment for crops with well-defined peduncles, but require precise alignment and can transmit cutting forces to surrounding branches in dense canopies [39] (Figure 3.7).

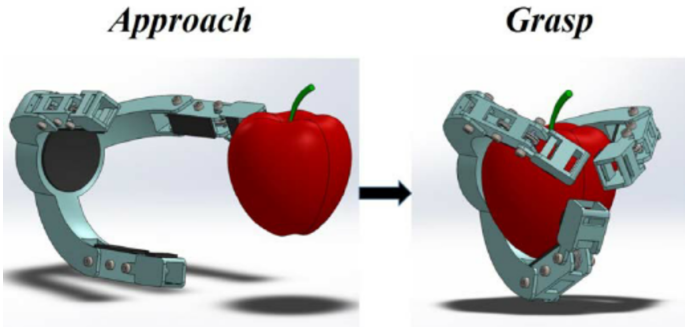


Figure 3.7: End-effector for apple harvesting [39]

Suction-based end-effectors stabilize the fruit through negative pressure, improving tolerance to positioning errors and reducing contact forces (Figure 3.8). Their performance, however, is sensitive to surface texture, air leakage, and contamination, and they often require combination with additional detachment mechanisms [7], [40].

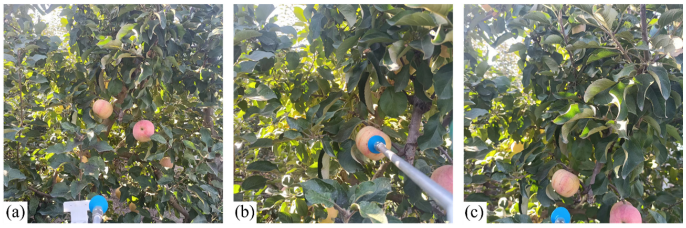


Figure 3.8: Vacuum suction end-effector [7]

To improve robustness across variable conditions, compliant and soft end-effectors incorporate flexible materials or underactuated mechanisms that tolerate misalignment and reduce the risk of damage [6], [26] (Figure 3.9). Hybrid designs combining suction, gripping, and cutting have also been proposed to broaden applicability, at the cost of increased mechanical and control complexity.

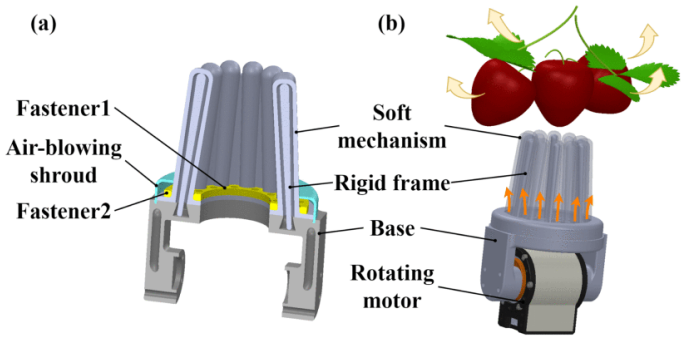


Figure 3.9: High-tolerance soft-rigid gripper [6]

No single end-effector design is universally applicable, the choice is closely tied to crop morphology and harvesting strategy. In the present work, the end-effector itself is not the focus and

is regarded as part of the future harvesting capability that the proposed manipulator is intended to enable.

3.5 Vision-Based Perception

Vision-based perception provides the spatial and semantic information required to guide harvesting, enabling fruit detection, localization, and ripeness assessment in unstructured agricultural environments. RGB cameras are the most widely adopted sensors, due to their low cost and high resolution, but they lack direct depth information and are sensitive to illumination changes. RGB-D and stereo systems add three-dimensional information that supports grasp planning and collision avoidance, while LiDAR provides robust range measurements under varying lighting, though with lower spatial resolution. These modalities are frequently combined to compensate for their individual limitations [4].

For detection and ripeness estimation, modern systems increasingly rely on deep learning. Convolutional neural networks and related architectures learn discriminative features directly from data, achieving robust fruit detection in cluttered scenes and supporting ripeness classification from colour and texture cues [5]. Despite this maturity, perception outputs remain approximate where occlusions cause missed or partial detections, depth-estimation errors propagate to manipulation, and centroid or bounding-box outputs may be insufficient for precise end-effector alignment in cluttered canopies (Figure 3.10). As a result, accurate perception alone does not guarantee successful harvesting unless the manipulation system can tolerate localization uncertainty.

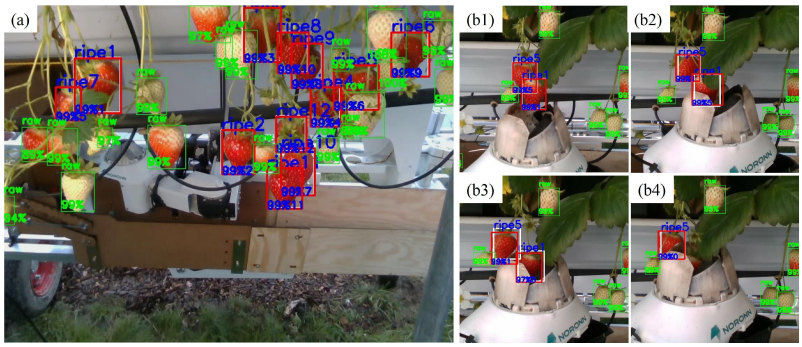


Figure 3.10: Vision-based detection and tracking of fruit in clusters [5].

The perception framework underlying the present work was established in prior research on arbutus berry harvesting by Pereira [11], which integrated LiDAR and RGB-D sensing with deep-learning models for fruit detection, localization, and ripeness classification. As perception is inherited rather than contributed by this dissertation, it is not treated further here. The focus instead falls on the physical realization and kinematic control of the manipulator that must act on those perception outputs.

3.6 Research Gaps and Positioning of This Work

The literature reviewed in this chapter reveals a consistent pattern. Robotic harvesting has been demonstrated across many crops and platform architectures, from infrastructure-guided greenhouse systems to mobile single-arm, multi-arm, and cluster-handling robots. Vision-based perception, meanwhile, has reached a relatively mature stage, achieving reliable detection, localization, and ripeness estimation. Taken together, these observations confirm the pattern noted in Section 3.1.5: the binding constraint in practical harvesting is physical interaction with the crop in dense, fragile canopies, not the detection of the fruit.

The manipulation literature points to a specific reason for this. As established in Section 3.2, rigid-link manipulators cannot conform to cluttered, fragile canopies and remain prone to collisions and plant damage, whereas the distributed compliance of continuum and soft manipulators lets them conform to obstacles and tolerate positioning uncertainty. However, as the recent selective-harvesting review notes, continuum mechanisms remain markedly underexplored in agriculture relative to rigid arms [19], and the few existing agricultural continuum manipulators have largely been validated in simulation or on isolated prototypes rather than integrated into a full perception-driven harvesting context.

This dissertation is positioned precisely in that gap. Building on a perception-driven arbutus berry harvesting framework that was validated in simulation [11], it investigates a two-section tendon-driven continuum manipulator as the physical-interaction layer of the system. The work addresses the design of such a manipulator, its physical realization, the formulation and implementation of its constant-curvature kinematics including inter-section coupling, and the experimental validation of its positioning accuracy under laboratory conditions.

3.7 Summary

This chapter reviewed the state of the art relevant to the work developed in this dissertation. It surveyed the landscape of robotic harvesting systems and the trade-offs between platform architectures, and identified physical interaction in cluttered canopies as the principal remaining bottleneck. It then examined manipulation in cluttered canopies, showing the limitations of rigid arms, and reviewed continuum and soft manipulators as a compliant alternative, together with their actuation, kinematics and control, and sensing. Finally, it covered harvesting end-effectors and the inherited perception stack, and synthesized the research gaps that motivate this work. The following chapters build on this foundation, presenting the design, implementation, and experimental evaluation of the proposed continuum manipulator.

Chapter 4

Proposed Solution

This chapter presents the proposed manipulator, from the requirements that shaped it to the final design that was built. It begins by deriving the design requirements and constraints from the characteristics of the arbutus tree, the intended harvesting strategy, and the available means of fabrication. It then sets out the rationale for adopting a tendon-driven continuum manipulator, before describing the successive design iterations that were explored and the final two-section design selected for implementation.

4.1 Design Requirements and Constraints

The design of the manipulator is driven by the physical and morphological characteristics of the arbutus tree established in Chapter 2, together with the intended harvesting strategy and the practical constraints of in-house fabrication. Most requirements are qualitative, reflecting the variability of the target environment, while a few impose concrete bounds on the structure. They are summarized in Table 4.1 and discussed below.

Table 4.1: Design requirements and constraints for the manipulator.

Requirement	**Description and rationale**
Compliance and safe interaction	Must conform to fragile branches and peduncles rather than displacing or damaging them. This becomes a primary motivation for a continuum (rather than rigid) structure.
Reach within the canopy	Must enter a dense, occluded canopy and position its tip at fruit distributed at varying depths.
Backbone length	Approximately 600 mm, to reach across the depth of a typical arbutus canopy.
Positioning accuracy	Tip must reach a commanded target to within approximately the size of an arbutus berry, so a future end-effector can engage the fruit despite residual error.
Hollow backbone for aspiration	Backbone and spacer discs must be hollow and of sufficient diameter to route a central aspiration tube for suction-based fruit handling.
In-house fabrication	Must be manufacturable with available resources, primarily 3D-printed components and off-the-shelf parts.

The most influential of these is compliance: the fragility of the arbutus branches makes a rigid manipulator unsuitable and motivates the continuum structure developed in the following sections. The remaining requirements act as constraints on that structure. Reach and backbone length fix its overall dimensions, the hollow-backbone requirement constrains its internal layout, the positioning-accuracy target sets the standard against which the kinematics are later evaluated, and in-house fabrication limits the available materials and manufacturing processes. Together they shape the design decisions presented next.

4.2 Design Rationale

The manipulator is a tendon-driven continuum structure built from two antagonistically actuated sections. This section explains that configuration through the four design decisions behind it, taken in turn, and relates each to the requirements of Section 4.1 and to the literature of Chapter 3: a continuum rather than rigid structure, tendon-driven rather than intrinsic actuation, two sections rather than one, and antagonistic tendon pairs rather than independent cables.

The primary driver is the requirement for compliant, safe interaction with the plant. The arbutus canopy is composed of thin, flexible branches and fragile peduncles, and Section 3.2 showed that a rigid arm cannot conform to such a canopy and tends either to collide with unseen structures or to force its links through them. A continuum manipulator instead distributes compliance along

its backbone and can deform around the same obstacles, reducing interaction forces and tolerating the positioning uncertainty inherent to a cluttered scene. Two-section tendon-driven manipulators have been built for vegetable harvesting [37], and spring-backbone continuum robots for fruit and vegetable picking [38]. Their use in comparable settings supports the suitability of a continuum structure for the foliage of the arbutus tree, and this compliance is the decisive reason for adopting it over a conventional rigid arm.

Among the actuation schemes available for continuum manipulators, a tendon-driven approach was selected. As discussed in Section 2.5, tendon drives keep the actuators at the base. This leaves the moving structure lightweight, allows a large length-to-diameter ratio, and gives faster and more direct control than fluidic alternatives. The decisive factor for this work, however, is that tendon actuation leaves the central axis of the manipulator free. The requirement for a hollow backbone able to route an aspiration tube (Section 4.1) rules out intrinsic schemes such as pneumatic chambers, which would occupy the core of the structure. It equally rules out the alternative extrinsic family, concentric-tube actuation, which is confined to small-diameter applications that leave no room for an aspiration channel. A tendon-driven backbone, with cables routed along the periphery and pulled from the base, is therefore the actuation family that best reconciles the compliance and central-aspiration requirements.

The manipulator is built from two independently actuated sections rather than one. A single continuum section bends as one circular arc and can place its tip only over a surface. Two serially stacked sections position the tip within a volume rather than over a surface, and the resulting redundancy lets the same target be reached through different backbone configurations. This added freedom is what allows the manipulator to thread into a cluttered canopy and reach fruit distributed at varying depths, and it comes at the cost of more actuators and of inter-section coupling, both of which are addressed in the chapters that follow.

Finally, within each section the tendons are arranged as antagonistic pairs, a single motor driving two opposing cables so that one is paid out as the other is pulled in. This halves the number of actuators needed to bend a section in a given plane, relative to driving each cable independently. As Section 2.5 reported, this one-motor-per-pair scheme has been characterised experimentally and offers a favourable balance between mechanical simplicity and positioning performance [23]. It keeps the actuation compact while preserving the bending range the canopy demands, and it fixes the cable layout described in Section 4.4.

4.3 Design Iterations

The architecture fixed in Section 4.2, a tendon-driven structure with a hollow backbone, still left open how that backbone should be physically realised. Two distinct realisations were explored: an elastic-tube backbone, which was built and tested, and an articulated-joint backbone, which addressed the limitations of the first and became the final design. This section describes each in turn, the reasoning behind it, and the findings that motivated the transition.

4.3.1 First Iteration: Elastic-Tube Backbone

The first concept reused a readily available component as the backbone, a flexible vacuum-cleaner tube. Its corrugated wall was expected to bend freely, while the hollow core provided the central channel required for aspiration (Section 4.1). Around this tube a set of spacer disks was designed, each profiled to seat onto the corrugations and to hold the tendon-routing holes at a fixed radius from the centre. The tip went through several revisions and was shaped into a funnelled, mouth-like intake intended to guide individual berries toward the central channel, drawing on the cluster-handling end-effector reviewed in Section 3.2.

Figure 4.1: Spacer disk in CAD.

Figure 4.2: First design iteration in CAD.

The discs were 3D-printed and assembled onto the tube to form a first physical prototype (Figure 4.3), and tendons of polished flax twine were routed through the discs so that the actuation force could be assessed manually. Each cable was pulled through a handheld luggage scale to obtain a rough indication of the tension needed to bend the structure. The measurement was indicative rather than rigorous, but its outcome was unambiguous. The tubing, being new, was considerably stiffer than anticipated and bending it demanded a large amount of tension, and even

under that tension the achievable curvature was small. The prototype therefore could not bend tightly enough to be useful within a canopy. The stiffness of the tube itself, rather than any detail of the discs or tendons, was the limiting factor, and it could not be resolved without abandoning the tube as the backbone.

Figure 4.3: First design iteration, physical build.

4.3.2 Second Iteration: Articulated-Joint Backbone

The lesson from the first prototype was not that the backbone should be free of bending resistance because a degree of stiffness is in fact desirable, providing the restoring force that holds a section's shape, as in the spring-backbone designs discussed in Section 3.3. The problem was the curvature the tube could attain, since it resisted bending into a tight enough radius, so even when forced it would not form the sharp curves a section must reach to be useful inside a canopy. Attention therefore shifted from a continuous elastic tube to an articulated backbone, in which bending is concentrated at discrete joints between rigid segments. Concentrating the deformation at joints allows each one to rotate through a large angle, so the backbone as a whole can reach much tighter curvatures than the tube permitted. Two joint types were designed and compared: a Cardan (universal) joint, allowing bending about two perpendicular axes, and a spherical (ball-and-socket) joint.

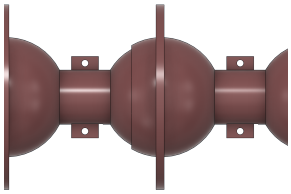


Figure 4.4: Spherical joint.

The spherical joint was selected for three reasons, each tied to a requirement of Section 4.1 or to the kinematic model adopted for the manipulator. The decisive one is that the body of the joint is itself hollow and tubular, so the same component that provides articulation also forms the central aspiration channel, satisfying the hollow-backbone requirement with no additional part. The Cardan joint, as its cross-and-yoke construction occupies the centre of the joint and leaves no clear bore for an aspiration tube. Second, a spherical joint bends isotropically since it pivots in the same way in every direction about its centre, so the section can bend in an arbitrary plane purely as a function of the tendon tensions. A Cardan joint instead has two structurally privileged hinge axes, and bending in an intermediate plane must be synthesised by coordinating both axes, with a range and resistance that vary with direction. The directional uniformity of the spherical joint matches the constant-curvature model of Chapter 5, in which a section bends as a single arc of arbitrary orientation, whereas the Cardan joint's built-in anisotropy does not. Third, fabrication favours the spherical joint since it reduces to two printed parts that nest together, whereas the Cardan joint requires a central cross and two yokes held to fine tolerances, making the simpler joint better suited to the in-house, 3D-printed fabrication on which this work relies.

A backbone composed of a series of these spherical joints bends through the accumulation of many small rotations, approximating the constant-curvature behaviour assumed by the kinematic model of Chapter 5, while reaching the tight curvatures the tube could not. This spherical-joint backbone was carried forward and built as the manipulator described in detail in Section 4.4.

Figure 4.5: Final design in CAD.

Settling the backbone left a separate problem at the base: how to mount the four motors that drive the eight tendons. The constraint is geometric. Each motor must pull its antagonistic cable pair in line with where those cables enter the backbone, yet four motors and eight cables have to share the limited footprint beneath the arm without the motor bodies colliding or one motor's cables fouling another's. Routing the cables straight down from the backbone called for the motor shafts to lie close to the central axis, which left no room to place four of them side by side.

One option considered was a system of rollers that would turn each cable through 90°, decoupling the motor positions from the cable entry points and allowing the motors to be arranged freely. This was rejected since the added rollers, mounts, and routing raised the mechanical complexity

disproportionately for a prototype, and each redirection introduces a further point of friction and wear. The simpler solution adopted instead was to tilt each motor slightly and offset it from the centre, so that all four fit beneath the base while keeping each cable run close enough to straight to avoid binding. The arrangement is not perfectly symmetric, but it accommodates the full set of actuators within the available space with no additional parts, which suited the prototyping stage.

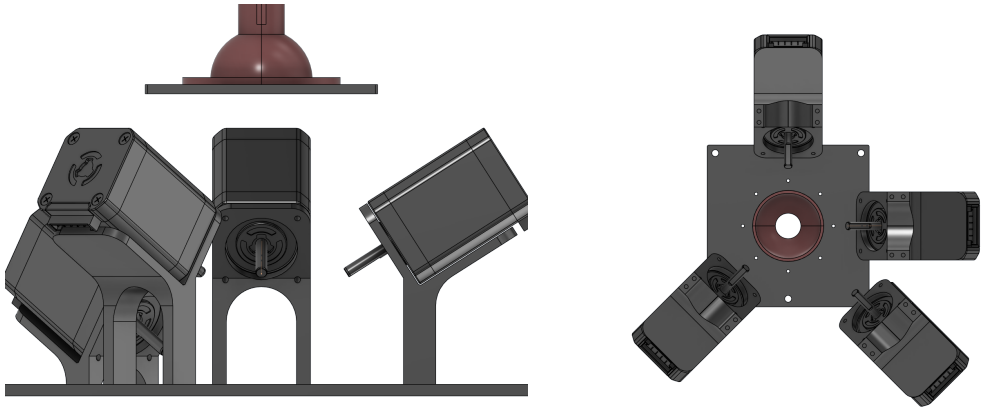


Figure 4.6: Motors disposition in CAD.

4.4 Final Design

The design iterations led to the manipulator shown in Figure 4.7: a two-section, tendon-driven continuum arm built on the articulated spherical-joint backbone selected in Section 4.3.2. This section describes the mechanical design as built. The actuation electronics and the control architecture are the subject of Chapters 6 and 7, and the kinematic model relating tendon lengths to tip pose is developed in Chapter 5.

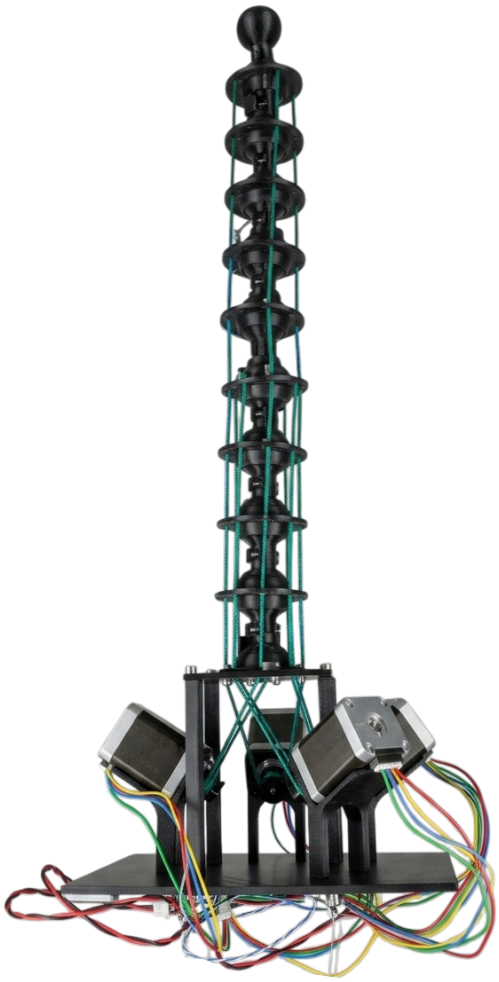


Figure 4.7: The final two-section tendon-driven manipulator.

4.4.1 Backbone and Sections

The backbone is a chain of hollow spherical joints, each acting as one vertebra of an articulated spine whose central bore forms the continuous aspiration channel required by Section 4.1. The chain is divided into two independently actuated sections, a proximal section of approximately 228 mm and a distal section of approximately 346 mm, giving a total backbone length of about 574 mm. Each section bends as an approximately constant-curvature arc, and the two in series allow the tip to be positioned within a volume rather than over a single surface, as argued in Section 4.2.

Figure 4.8: Close up of the backbone.

4.4.2 Spacer Disks and Tendon Offset

Tendon-routing discs are mounted at intervals along the backbone, ten in total across the two sections. Each disk holds the tendons at a fixed radius of 28 mm from the central axis. This offset sets the moment arm through which a tendon acts on the backbone. A larger offset increases the bending produced per unit of tendon displacement, while remaining within the space available around the central channel. The discs do not bear load themselves, they constrain the tendon paths so that pulling a cable produces a predictable bend rather than allowing the cables to cut across the structure.

4.4.3 Tendon Layout

Each section is driven by four tendons of low-stretch polyester rope, arranged as two antagonistic pairs (Section 4.2), giving eight tendons in total. The low elongation of the rope is important because a tendon that stretched appreciably under load would weaken the relationship between motor rotation and tip displacement on which the kinematic model depends.

A complication arises because the distal section's tendons must pass through the proximal section to reach their anchor points, sharing the same discs as the proximal tendons. To keep the two sets from interfering, they are routed at different angular positions: the proximal tendons run through the diagonal positions of the discs, and the distal tendons through the cardinal positions (Figure 4.9). This separation lets all eight tendons pass through the proximal section without contact. Routing the distal tendons through the proximal section does, however, couple the two since bending the proximal section changes the length of distal tendon consumed within it. This inter-section coupling is accounted for in the kinematic model of Chapter 5.

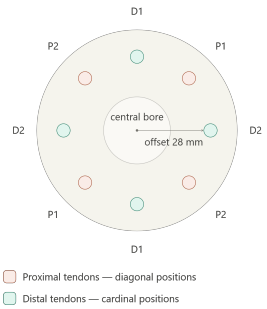


Figure 4.9: Cross-section of the disks.

4.4.4 Actuation Base

At the base, the eight tendons are driven by four stepper motors, one per antagonistic pair, each tendon wound onto a pulley of 16 mm diameter that converts motor rotation into tendon displacement. As described in Section 4.3.2, the four motors are tilted and offset from the central axis so that all four fit beneath the base while keeping each tendon run close to straight. The motors, drivers, and control boards that make up this actuation stage are detailed in Chapter 6.

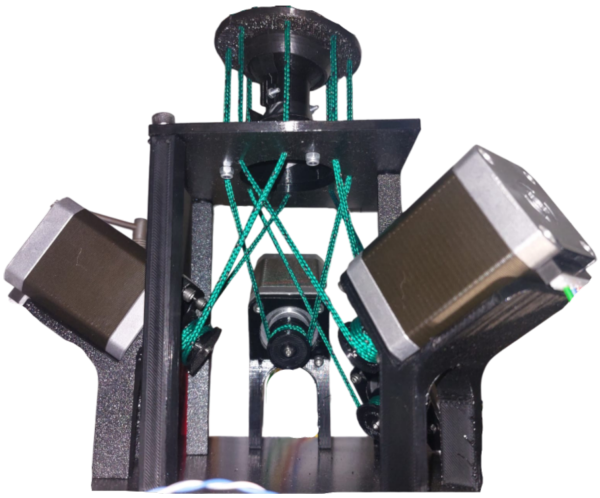


Figure 4.10: View of the motors disposition.

It should be noted that the manipulator realised here is the positioning structure only. The funnelled aspiration tip explored in the first iteration (Section 4.3.1) is not part of the built system, and no end-effector or perception components are integrated. The backbone is designed to accommodate a central aspiration tube, but suction-based fruit handling and the integration with perception are left as future work.

4.5 Summary

This chapter developed the proposed manipulator from its requirements through to the prototype that was built. The requirements were derived from the morphology of the arbutus tree, the intended suction-based harvesting strategy, and the limits of in-house fabrication, with compliant interaction emerging as the dominant driver and a hollow backbone, sufficient reach, and adequate positioning accuracy as the principal constraints. From these, the design rationale followed as a sequence of four decisions:

- a continuum rather than rigid structure, for compliance in a fragile canopy;
- tendon-driven rather than intrinsic actuation, to keep the central axis free for aspiration;
- two sections rather than one, for more dexterity;
- antagonistic tendon pairs, to halve the number of actuators.

Two physical realisations of the backbone were then explored. The first, built around a flexible vacuum-cleaner tube, proved unable to attain a tight enough curvature to be useful and was set aside. The second, an articulated backbone of hollow spherical joints, resolved this while satisfying the hollow-backbone requirement through the joints' own bores, and was carried forward. The resulting final design is a two-section manipulator with a spherical-joint backbone of approximately 574 mm, fitted with ten spacer discs that route eight low-stretch polyester tendons at a 28 mm offset, the tendons grouped diagonally for the proximal section and cardinally for the distal one to avoid interference, and driven by four base-mounted stepper motors.

Chapter 5

Kinematic Modelling

This chapter develops the kinematic model that relates how its tendons are driven to where its tip ends up. Because a continuum backbone bends continuously rather than at discrete joints, the rigid-link conventions used for conventional manipulators do not apply, and the model is instead built on the constant curvature assumption [20], under which each section is treated as an arc of constant curvature. This reduces the description of a section to a small set of arc parameters and lets the kinematics be expressed as a composition of two mappings: a robot-specific mapping from the tendon lengths to the arc parameters, which depends on the cable geometry of this particular manipulator, and robot-independent mapping from the arc parameters to the tip pose, which follows from the geometry of a circular arc.

The chapter is organised around these two mappings. Section 5.1 introduces the constant-curvature framework and the arc parameters used throughout. The forward kinematics are then assembled from the two mappings in turn: Section 5.2 derives the robot-independent arc-to-pose mapping and composes it across the two sections, and Section 5.3 derives the robot-specific cable-to-arc mapping from the eight-tendon, antagonistic-pair layout of the built manipulator, which Section 5.4 extends to account for the coupling introduced by routing the distal tendons through the proximal section. Finally, Section 5.5 addresses the inverse problem, of finding the actuation required to reach a desired tip pose, using an analytic solution for a single section and a DLS numerical method for the full two-section manipulator.

5.1 Constant-Curvature Model

A continuum section bends along its entire length rather than at discrete joints, so the rigid-link conventions used for conventional manipulators do not describe it. The standard alternative, and the one adopted here, is the *constant-curvature* model formalised by Webster and Jones [20], in which each section of the backbone is approximated as a circular arc of constant curvature lying in a single plane. The approximation holds well for a slender, uniformly actuated section under negligible external load, and reduces the description of a section from an infinite-dimensional shape to three scalar parameters.

5.1.1 Arc Parameters

Under the constant-curvature assumption, the shape of a single section is described by three *arc parameters* (Figure 5.1):

- the curvature κ, the reciprocal of the radius r of the arc ($\kappa = 1/r$), which measures how sharply the section bends. A straight section has $\kappa = 0$;
- the bending-plane angle ϕ, the angle about the section's base z-axis that orients the plane in which the arc lies;
- the arc length ℓ, the length of the backbone measured along the arc, taken as constant for an inextensible backbone.

It is often convenient to work instead with the *bending angle*

$$\theta = \kappa\ell, \tag{5.1}$$

the total angle subtended by the arc, which combines curvature and length into the single quantity that the section actually rotates through. The pair (κ, ℓ) and the pair (θ, ℓ) are equivalent descriptions, related by Equation (5.1). The expressions that follow are written in whichever form is clearer, with θ used for the geometric transform and κ retained where the dependence on physical curvature must be explicit.

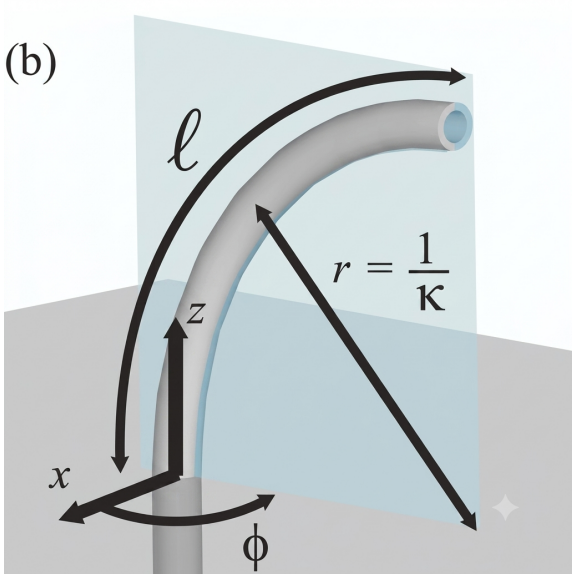


Figure 5.1: A single section model [20].

5.1.2 The Two-Mapping Decomposition

The central insight of the Webster–Jones framework is that the kinematics of a constant-curvature section factor into two independent mappings, joined by the arc parameters [20]. The

first is *robot-specific*: it relates the actuator variables of the particular mechanism, here the lengths of the tendons, to the arc parameters (κ, ϕ, ℓ). This mapping depends on how the section is actuated, on the number of tendons, their offset from the centre, and the geometry of the routing. The second is *robot-independent*: it relates the arc parameters to the homogeneous transformation describing the tip pose, and follows purely from the geometry of a circular arc, so it is identical for every constant-curvature robot regardless of how it is built. Schematically:

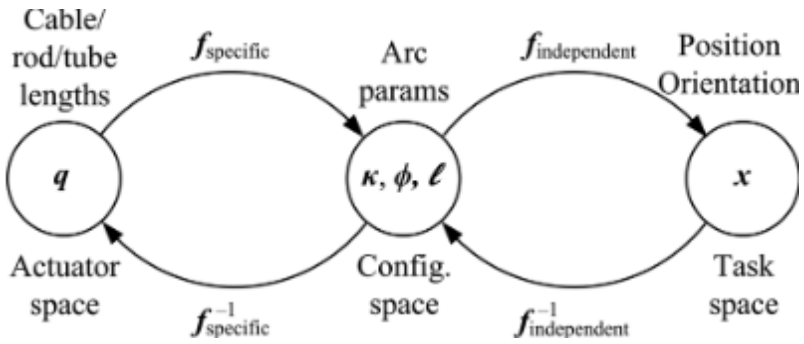


Figure 5.2: Spaces and mappings between them [20].

This decomposition organises the rest of the chapter, and together the two mappings constitute the forward kinematics. The robot-independent mapping, common to all constant-curvature robots, is developed in Section 5.2 and composed across the two sections. The robot-specific mapping, particular to the eight-tendon layout of the manipulator built in this work, is derived in Section 5.3 and extended in Section 5.4 to account for the coupling introduced by routing the distal tendons through the proximal section. The inverse problem, of recovering the actuation required for a desired pose, is treated in Section 5.5.

5.1.3 Modelling Assumptions

The constant-curvature model rests on assumptions that should be stated, since they bound the accuracy that the kinematics can be expected to achieve.

- Each section is taken to bend as a single planar arc of uniform curvature;
- the backbone is treated as inextensible, so ℓ is constant and bending arises only from the differential lengths of the tendons;
- the effects of gravity, external contact, friction in the tendon paths, and torsion about the backbone axis are neglected.

The articulated spherical-joint backbone of this manipulator (Section 4.4) approximates a continuous arc through a discrete chain of joints, which the model treats as the equivalent smooth arc. The validity of these assumptions for the physical manipulator is examined experimentally in Chapter 8.

5.2 Forward Kinematics

The forward kinematics relate the manipulator's actuation to the pose of its tip. Following the two-mapping decomposition introduced in Section 5.1, this relation factors into two mappings joined by the arc parameters: the robot-specific mapping (tendon lengths $\rightarrow (\kappa, \phi, \ell)$), treated in Section 5.3, and the robot-independent mapping ($(\kappa, \phi, \ell) \rightarrow$ pose), developed here. The latter depends only on the geometry of a circular arc and is therefore identical for every constant-curvature manipulator [20]. A geometric derivation of its components is given below, validated at the end against the model of [20].

5.2.1 Geometry of a Section in the Bending Plane

The property that makes the model tractable is that a constant-curvature section is *planar*, the arc traced by the backbone lies in a single plane containing the base z-axis. Rather than solving the problem directly in three dimensions, it is solved first within the bending plane as a two-dimensional problem, and the plane is then oriented in space through the angle ϕ.

Consider the section from its base, with the z-axis tangent to the backbone (Figure 5.3). The arc is part of a circle of radius

$$r = \frac{1}{\kappa}, \tag{5.2}$$

where κ is the curvature of the section. For this circle to be tangent to the z-axis at the base, its centre of curvature lies perpendicular to the tangent, at a distance r from the base. Travelling an arc length ℓ along the backbone, the angle subtended at the centre is, by the definition of arc length,

$$\theta = \frac{\ell}{r} = \kappa\ell, \tag{5.3}$$

referred to as the *bending angle* of the section.

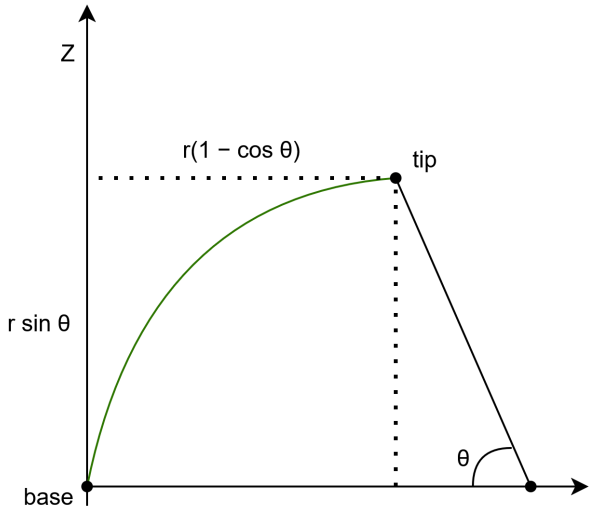


Figure 5.3: A section modelled as a constant-curvature arc in its bending plane.

The tip position within this plane follows from the geometry of the triangle formed by the base, the centre of curvature, and the tip (Figure 5.4). Place the origin at the centre of curvature and measure the angle from the radius pointing to the base. In this frame, the base lies on the horizontal side, a distance r from the centre, at zero height. The tip is the same radius r rotated by θ, so it lies $r\cos\theta$ from the centre horizontally and is raised $r\sin\theta$ vertically.

The components of the tip position *relative to the base* are obtained by subtracting the base coordinates from those of the tip. Along the tangent direction (the z-axis), the base is level with the centre and the tip is $r\sin\theta$ above it, giving the component

$$p_{\parallel} = r\sin\theta. \tag{5.4}$$

In the perpendicular direction (lateral, within the plane), the base is a distance r from the centre and the tip only $r\cos\theta$, giving

$$p_{\perp} = r - r\cos\theta = r(1-\cos\theta). \tag{5.5}$$

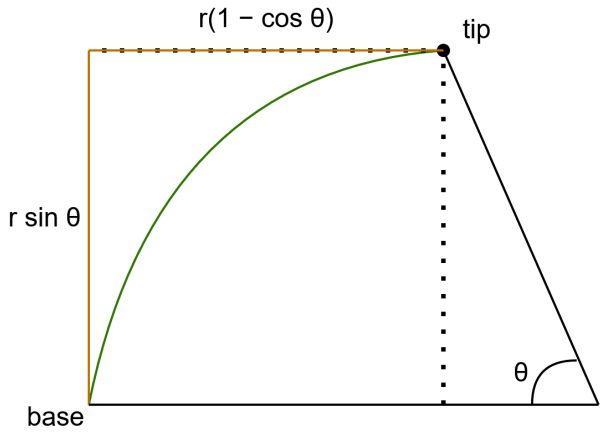


Figure 5.4: Components of the tip position in the bending plane, obtained as the legs of the triangle formed by the base, the centre of curvature, and the tip.

5.2.2 Tip Position in Three Dimensions

Components (5.4) and (5.5) are in the bending plane. It remains to orient that plane in space, which is the role of the angle ϕ, measured about the base z-axis from the x-axis (Figure 5.5). The component along the tangent coincides with the z-axis and does not depend on ϕ. The lateral component is a horizontal vector of magnitude $p_{\perp}$ pointing in the ϕ direction, projecting onto the x- and y-axes as

$$x = p_{\perp}\cos\phi, \qquad y = p_{\perp}\sin\phi. \tag{5.6}$$

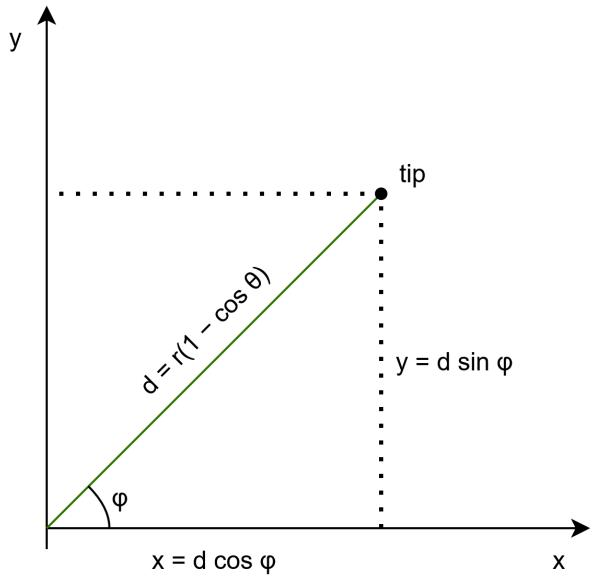


Figure 5.5: Top view along the z-axis.

Substituting $r = 1/\kappa$ and $\theta = \kappa\ell$ into Equations (5.4)–(5.6), the tip position in the base frame is

$$\mathbf{p}(\kappa,\phi,\ell) = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \dfrac{\cos\phi\,(1-\cos\kappa\ell)}{\kappa} \\ \dfrac{\sin\phi\,(1-\cos\kappa\ell)}{\kappa} \\ \dfrac{\sin\kappa\ell}{\kappa} \end{bmatrix}. \tag{5.7}$$

Each term has an identifiable origin:

- the factor $(1-\cos\kappa\ell)/\kappa$ is the lateral component of the bending plane;
- $\cos\phi$ and $\sin\phi$ distribute it between x and y;
- $\sin(\kappa\ell)/\kappa$ is the component along the tangent, with no dependence on ϕ

5.2.3 Tip Orientation

The position locates the tip, but the complete pose also requires its orientation, that is, the rotation matrix $\mathbf{R}$ describing the tip frame relative to the base frame. This orientation is essential for composing sections, since the orientation of one section's tip defines the inclination of the next section's base.

The orientation is obtained as the composition of two elementary rotations. First, a rotation of ϕ about z aligns the x-axis with the bending plane. Then the bending of the section corresponds to a rotation of $\theta = \kappa\ell$ about the resulting y-axis :

$$\mathbf{R}(\kappa,\phi,\ell) = \mathbf{R}_z(\phi)\,\mathbf{R}_y(\theta). \tag{5.8}$$

This convention leaves the tip x-axis aligned with the bending plane, matching the one adopted by [20]. The elementary rotations are

$$\mathbf{R}_z(\phi)) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{5.9}$$

whose columns are the images of the axes after each rotation. Carrying out the product of Equation (5.8) entry by entry gives

$$\mathbf{R}(\kappa,\phi,\ell) = \begin{bmatrix} \cos\phi\cos\kappa\ell & -\sin\phi & \cos\phi\sin\kappa\ell \\ \sin\phi\cos\kappa\ell & \cos\phi & \sin\phi\sin\kappa\ell \\ -\sin\kappa\ell & 0 & \cos\kappa\ell \end{bmatrix}. \tag{5.10}$$

5.2.4 Homogeneous Transformation of a Section

Combining the orientation (5.10) and the position (5.7) into a single homogeneous transformation, the pose of a section's tip relative to its base is

$$\mathbf{T}(\kappa,\phi,\ell) = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}^\top & 1 \end{bmatrix} = \begin{bmatrix} \cos\phi\cos\kappa\ell & -\sin\phi & \cos\phi\sin\kappa\ell & \dfrac{\cos\phi\,(1-\cos\kappa\ell)}{\kappa} \\ \sin\phi\cos\kappa\ell & \cos\phi & \sin\phi\sin\kappa\ell & \dfrac{\sin\phi\,(1-\cos\kappa\ell)}{\kappa} \\ -\sin\kappa\ell & 0 & \cos\kappa\ell & \dfrac{\sin\kappa\ell}{\kappa} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.11}$$

Equation (5.11) is the robot-independent mapping for a single section and coincides, term by term, with the constant-curvature transformation presented by Webster and Jones [20] (with $s \equiv \ell$), validating the derivation.

5.2.5 The Straight-Section Limit

Equations (5.7) and (5.11) contain a factor $1/\kappa$ and are undefined at $\kappa = 0$ (the straight section), where ϕ also loses meaning. Expanding as $\kappa \to 0$,

$$\frac{\sin\kappa\ell}{\kappa} \to \ell, \qquad \frac{1-\cos\kappa\ell}{\kappa} \to 0, \tag{5.12}$$

so the position tends to $\mathbf{p} \to [0,\,0,\,\ell]^\top$ and the rotation to the identity, recovering a straight section of length ℓ aligned with z, as expected. In implementation, this singularity is handled by replacing the expressions with their limits for values of κ below a small threshold, avoiding the numerical $0/0$ with negligible error.

5.2.6 Composition of the Two Sections

The manipulator has two serially stacked sections (Section 4.4), each described by its own arc parameters: $(\kappa_p, \phi_p, \ell_p)$ for the proximal section and $(\kappa_d, \phi_d, \ell_d)$ for the distal one. Since the base of the distal section coincides with the tip of the proximal section, the pose of the manipulator's tip is obtained by composing the two section transformations (Figure 5.6):

$$\mathbf{T}_d^0 = \mathbf{T}(\kappa_p, \phi_p, \ell_p)\,\mathbf{T}(\kappa_d, \phi_d, \ell_d), \tag{5.13}$$

where $\mathbf{T}_d^0$ expresses the distal tip in the base frame of the proximal section. The composition chains the inclinations automatically: the rotation block of $\mathbf{T}(\kappa_p, \phi_p, \ell_p)$ makes the base of the distal section start already oriented along the end of the proximal section, so that the matrix product repositions and reorients the entire distal section.

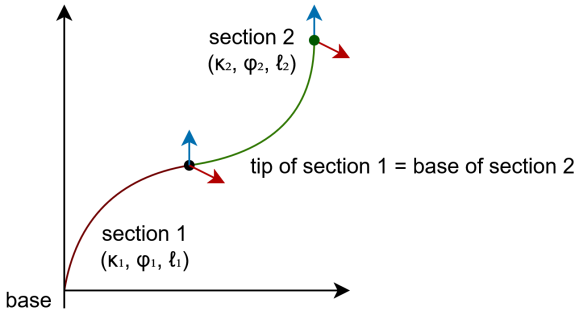


Figure 5.6: Chaining of the two sections.

Equation (5.13) constitutes the complete robot-independent mapping, from the arc parameters of both sections to the tip pose. The arc parameters it consumes are supplied by the robot-specific mapping derived in Section 5.3, including the inter-section coupling treated in Section 5.4, which makes (κ_d, ϕ_d) depend also on the proximal configuration.

5.3 Cable-to-Arc Mapping

This section derives the robot-specific mapping introduced in Section 5.1.2: the relation between the tendon lengths and the arc parameters (κ, ϕ, ℓ) of a section. Unlike the robot-independent mapping of Section 5.2, this one depends on the particular cable geometry of the manipulator built in this work, the offset of the tendons from the backbone, their angular arrangement on the discs, and the actuator geometry. The derivation assumes an inextensible backbone and tendons and rigid discs, so that all length changes are geometric.

5.3.1 Length Change of a Single Tendon

Considering a single tendon running parallel to the backbone at a fixed radial offset d from the central axis, held there by the routing discs. When the section is straight, the tendon has the same

length as the backbone, ℓ. When the section bends, the tendon lies on a circle concentric with the backbone arc but at a different radius, so its length changes (Figure 5.7).

Let the section bend with curvature κ, radius $r = 1/\kappa$, and bending angle $\theta = \kappa\ell$, so the backbone subtends θ at the centre of curvature and has length $\ell = r\theta$. A tendon on the inside of the bend lies at radius $r - d$ and, subtending the same angle θ, has length

$$\ell_{\text{in}} = (r - d)\,\theta, \tag{5.14}$$

while a tendon on the outside lies at radius $r + d$ and has length $\ell_{\text{out}} = (r + d)\,\theta$. The change relative to the straight length $\ell = r\theta$ is therefore

$$\Delta\ell = \ell_{\text{in}} - \ell = (r - d)\theta - r\theta = -d\,\theta, \tag{5.15}$$

for the inner tendon, and $+d\,\theta$ for the outer one. The radius r cancels: the length change of a tendon depends only on its offset d and on the bending angle θ, not on how tightly the section is curved. This linear relation, $\Delta\ell = \pm d\,\theta$, is the basis of the entire mapping.

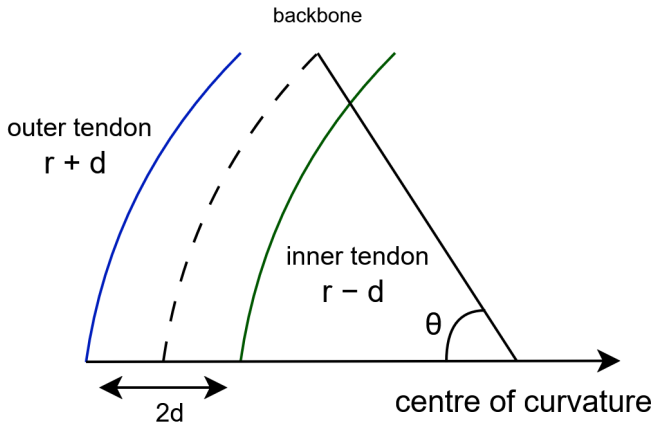


Figure 5.7: Two opposing tendons of an antagonistic pair under bending.

5.3.2 Antagonistic Pairs and Bending in a Plane

A single tendon can only pull, bending the section one way in one plane. To control bending in both directions of a plane, the tendons are arranged as antagonistic pairs (Section 4.2): two tendons on opposite sides of the axis, at $+d$ and $-d$ within that plane, driven by one motor so that one is paid out as the other is pulled in. By Equation (5.15), bending by θ in that plane shortens one tendon by $d\theta$ and lengthens the other by $d\theta$. The pair thus measures the bending angle of the section in its own plane.

A single pair, however, only spans one bending plane. To let the section bend in an arbitrary direction, each section carries two antagonistic pairs in two perpendicular planes, driven by two motors (Section 4.4). Denote the two pairs A and B, and let θ_A and θ_B be the bending angles each

pair would produce in its own plane, recovered from the measured tendon displacements through

$$\theta_A = \frac{\Delta\ell_A}{d}, \qquad \theta_B = \frac{\Delta\ell_B}{d}, \tag{5.16}$$

where $\Delta\ell_A$ and $\Delta\ell_B$ are the magnitudes of the length change of one tendon in each pair.

5.3.3 Recovering the Arc Parameters

The previous subsection related each antagonistic pair to a bending angle in its own plane. It remains to combine the two pairs of a section into its arc parameters (κ, ϕ, ℓ). Rather than asserting the combination, it is derived here from the way the two bending contributions superpose.

The bending produced by a single pair carries two pieces of information, how much the section bends (the angle θ_A or θ_B) and in which plane (the direction of that pair). This is the structure of a vector, and each pair's bending is represented as a vector in the plane of the disc (Figure 5.8): pair A contributes a vector of magnitude θ_A along its plane, taken as the x-direction, and pair B a vector of magnitude θ_B along the perpendicular y-direction.

For small deformations, the curvature contributions of independent actuations superpose. Bending the section in pair A's plane and in pair B's plane simultaneously produces a total bending equal to the vector sum of the two individual contributions. Since the two vectors are orthogonal, their sum is the diagonal of the rectangle they span (Figure 5.9). The magnitude of this diagonal is the total bending angle of the section, by the Pythagorean theorem,

$$\theta = \sqrt{\theta_A^2 + \theta_B^2}, \tag{5.17}$$

and its direction is the bending-plane angle,

$$\phi = \operatorname{atan2}(\theta_B, \theta_A). \tag{5.18}$$

The curvature follows from $\theta = \kappa\ell$ with the inextensible arc length ℓ,

$$\kappa = \frac{\theta}{\ell} = \frac{\sqrt{\theta_A^2 + \theta_B^2}}{\ell}. \tag{5.19}$$

This is the exact inverse of the position decomposition of Section 5.2.2: there, a known (κ, ϕ) was split into the orthogonal components $x = p_\perp \cos\phi$ and $y = p_\perp \sin\phi$; here, the two measured orthogonal components θ_A and θ_B are recombined into magnitude and direction. The cartesian-to-polar form of Equations (5.17)–(5.18) reflects this.

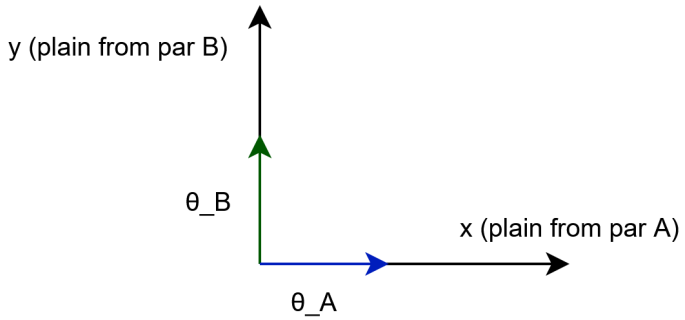


Figure 5.8: Top view along the backbone axis. The bending of each antagonistic pair is a vector in the plane of the disc.

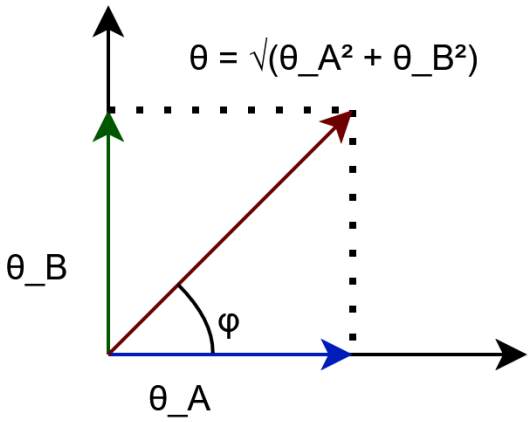


Figure 5.9: The two orthogonal bending vectors sum to the section's total bending. The diagonal has magnitude $\theta = \sqrt{\theta_A^2 + \theta_B^2}$, giving κ, and direction ϕ.

The combination is verified in the limiting cases:

- actuating only pair A ($\theta_B = 0$) gives $\theta = \theta_A$ and $\phi = 0$ bending in pair A's;
- actuating only pair B gives $\phi = 90°$;
- Equal actuation ($\theta_A = \theta_B$) gives $\phi = 45°$, bending along the diagonal.

The vector sum of curvatures is exact in the limit of small deformations, where the contributions superpose linearly. For large bending angles, the two contributions do not commute exactly, so the combination is an approximation that improves as the curvature decreases, consistent with the small-deformation basis of the constant-curvature model itself (Section 5.1.3).

5.3.4 From Motor Rotation to Tendon Displacement

The arc parameters above are expressed in terms of tendon displacements, but the quantity actually commanded is the motor rotation. Each tendon is wound on a pulley of radius $\rho = 8\,\text{mm}$ (a diameter of 16 mm). A motor rotation α winds a length of cable equal to the arc swept by the pulley,

$$\Delta\ell = \rho\,\alpha. \tag{5.20}$$

Substituting Equation (5.20) into the pair relation $\theta = \Delta\ell/d$ of Equation (5.16) gives the bending angle of a pair directly in terms of its motor rotation,

$$\theta = \frac{\rho}{d}\,\alpha, \tag{5.21}$$

so each pair bends in proportion to its motor rotation, with the scale factor $\rho/d = 8\,\text{mm}/28\,\text{mm} \approx 0.286$. The conversion of motor rotation into discrete actuator steps, and the resulting positioning resolution, are properties of the actuation hardware and are treated in Chapter 6. Equation (5.21), applied to the two pairs of a section and combined through Equations (5.17)–(5.19), closes the robot-specific mapping from the commanded motor rotations to the arc parameters (κ, ϕ, ℓ).

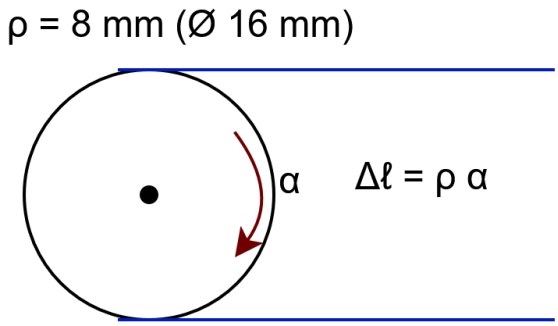


Figure 5.10: Pulley (circle) at the center with radius ρ marked, rotation arrow α, and blue cable coming out of the pulley to the manipulator.

5.3.5 Limitations of the Mapping

Two simplifications underlie the mapping. The first is geometric coupling between the sections: the mapping treats each section in isolation, assuming that a motor's displacement translates entirely into bending of its own section. This holds for the proximal section, but not for the distal one, whose tendons are routed through the proximal section (Section 4.4); bending the proximal section also consumes distal tendon length, so part of the displacement commanded for the distal section is absorbed by the proximal bend rather than producing distal curvature. The correction for this inter-section coupling is the subject of Section 5.4.

The second relates to the motor mounting. In the built manipulator, the four motors are tilted and offset from the central axis to fit beneath the base (Section 4.3.2), so each cable leaves its pulley at a slight angle to its routing path. While the geometric relationship of the wound cable

length remains $\rho\alpha$ regardless of this tilt, the angled routing introduces additional friction where the cable changes direction, which is not modelled here. Any residual effect is left to be assessed in the experimental validation of Chapter 8, where it could be addressed by per-motor calibration rather than by extending the geometric model.

5.4 Inter-Section Coupling

The cable-to-arc mapping of Section 5.3 treats each section in isolation. For a multi-section manipulator whose tendons run parallel to the backbone at an offset from the central axis, this is incomplete: the actuation of one section is coupled to that of the sections before it. As Hannan and Walker [41] note for the elephant-trunk manipulator, whose construction is of the same family as the manipulator built here, "the design of the cable drive system has the cable pairs of each plane running parallel to each other off the center axis of the manipulator by some distance," which "has the effect of coupling the cable actuation of a section with all of the previous sections." When a proximal section bends, the tendons of the distal section, because they are not on the central axis, change length, so the distal curvature is altered even if the distal motors do not move.

In the manipulator built here, this applies directly: the distal tendons are routed through the proximal section at the same offset d as all other tendons (Section 4.4), so bending the proximal section changes the length of distal tendon consumed within it. This section formulates the compensation, following the multi-section treatment of Hannan and Walker [41].

5.4.1 Arc Parameters as Orthogonal Curvatures

The compensation is expressed most naturally in terms of two orthogonal curvatures rather than the curvature–angle pair (κ, ϕ). Following Hannan and Walker [41], a section's bending is resolved into curvature components κ_x and κ_y about two perpendicular axes, related to (κ, ϕ) by

$$\kappa_x = \kappa\cos\phi, \qquad \kappa_y = \kappa\sin\phi, \tag{5.22}$$

and recovered by the inverse transformation [41]

$$\kappa = \sqrt{\kappa_x^2 + \kappa_y^2}, \qquad \phi = \arctan\left(\frac{\kappa_y}{\kappa_x}\right). \tag{5.23}$$

Equation (5.23) is the same cartesian-to-polar combination derived for the two antagonistic pairs in Section 5.3.3: each pair controls one orthogonal curvature component, so (κ_x, κ_y) are exactly the per-pair bending contributions, and (κ, ϕ) their resultant.

5.4.2 Coupling as a Rotation of the Curvature Components

The key observation of Hannan and Walker [41] is that, viewed down the backbone, bending a proximal section rotates the frames of all subsequent sections about the backbone axis by that section's bending-plane angle. The curvature components of a section must therefore be expressed

relative to the accumulated rotation of all preceding sections. For section i, the corrected components $(\kappa^*_{i,x}, \kappa^*_{i,y})$ are obtained from the section's own components by a rotation through the sum of the preceding bending-plane angles,

$$\begin{bmatrix} \kappa^*_{i,x} \\ \kappa^*_{i,y} \end{bmatrix} = \begin{bmatrix} \cos(\phi_{i-1} + \phi_{i-2} + \cdots) & -\sin(\phi_{i-1} + \phi_{i-2} + \cdots) \\ \sin(\phi_{i-1} + \phi_{i-2} + \cdots) & \cos(\phi_{i-1} + \phi_{i-2} + \cdots) \end{bmatrix} \begin{bmatrix} \kappa_{i,x} \\ \kappa_{i,y} \end{bmatrix}. \tag{5.24}$$

For the two-section manipulator built here, this reduces to its simplest form. The proximal section has no preceding section and needs no correction. The distal section is corrected by the bending-plane angle of the proximal section alone,

$$\begin{bmatrix} \kappa^*_{d,x} \\ \kappa^*_{d,y} \end{bmatrix} = \begin{bmatrix} \cos\phi_p & -\sin\phi_p \\ \sin\phi_p & \cos\phi_p \end{bmatrix} \begin{bmatrix} \kappa_{d,x} \\ \kappa_{d,y} \end{bmatrix}. \tag{5.25}$$

5.4.3 Equivalent View in Cable Length

The same coupling can be read directly in terms of the tendon lengths, which is the form closest to the actuation of this manipulator. A distal tendon at angular position β_d, passing through a proximal section that bends by $\theta_p = \kappa_p \ell_p$ in the plane ϕ_p, changes length by

$$\Delta\ell_\mathrm{c} = -\,d\,\theta_p \cos(\beta_\mathrm{d} - \phi_p), \tag{5.26}$$

the projection of the proximal bend onto that tendon's position. The displacement applied by a distal motor, $\rho\,\alpha_\mathrm{d}$, therefore divides between the part that bends the distal section and the part consumed by the proximal bend,

$$\rho\,\alpha_\mathrm{d} = \Delta\ell_\mathrm{bend} + \Delta\ell_\mathrm{c}, \tag{5.27}$$

so the displacement available to bend the distal section is recovered by subtracting the coupling term,

$$\Delta\ell_\mathrm{bend} = \rho\,\alpha_\mathrm{d} + d\,\theta_p \cos(\beta_\mathrm{d} - \phi_p). \tag{5.28}$$

Equations (5.25) and (5.28) are two views of the same correction. The former rotates the distal curvature components by the proximal angle, the latter adjusts each distal tendon length by the cable consumed in the proximal bend. Both depend only on the proximal configuration (κ_p, ϕ_p) and are computed deterministically.

5.4.4 Significance

This coupling is not a negligible effect. Hannan and Walker [41] identify it explicitly as a principal source of positioning inaccuracy in their manipulator, attributing "a great deal of the robot's inaccuracy" to the curvature determination "and the coupling between the cables of the sections." Since the present work evaluates positioning accuracy (Chapter 8), the coupling is compensated rather than neglected. The compensation is deterministic and inexpensive, requiring only the proximal configuration, which is already known from the proximal cable-to-arc mapping.

Residual coupling error, arising from friction between the distal tendons and the proximal discs along the routed path, is not captured by the geometric model and is assessed experimentally in Chapter 8.

5.5 Inverse Kinematics

The forward kinematics of the preceding sections map a configuration to the tip pose. Control requires the inverse: given a desired tip position, find the configuration that reaches it, and hence the motor commands. This section presents two cases: a closed-form inverse for the single-section mode, and a numerical inverse for the full two-section manipulator.

5.5.1 Single-Section Analytic Inverse

In single-section mode, the whole backbone is treated as one constant-curvature arc of length L. For a target tip position (x, y, z) in the base frame, the arc that reaches it is found in closed form. The bending plane is the plane containing the target and the base axis,

$$\phi = \operatorname{atan2}(y, x), \tag{5.29}$$

and, from the constant-curvature geometry of Section 5.2, the bending angle follows from the in-plane coordinates,

$$\theta = 2\,\operatorname{atan2}\left(\sqrt{x^2+y^2},\, z\right), \qquad \kappa = \frac{\theta}{L}. \tag{5.30}$$

This single-section case has a unique solution and needs no iteration. With two degrees of freedom (DoF) it reaches a surface rather than a volume, and the tip orientation is not independently controllable.

5.5.2 Two-Section Numerical Inverse

For the full two-section manipulator the inverse cannot be written in closed form. Two sections provide four bending DoF $(\kappa_p, \phi_p, \kappa_d, \phi_d)$ against three position constraints, so the problem is redundant, and the mapping from configuration to tip position is non-linear. As Buss [21] notes for inverse kinematics in general, in such cases "there may not be a unique (best) solution… even in well-behaved situations, there may be no closed form equation for the solution," and an iterative method is used instead.

5.5.2.1 Problem Formulation

Following the Jacobian-based formulation of Buss [21], let $\mathbf{q}$ be the configuration vector and $\mathbf{s}(\mathbf{q})$ the tip position given by the forward kinematics. For a target $\mathbf{t}$, the position error is

$$\mathbf{e} = \mathbf{t} - \mathbf{s}(\mathbf{q}), \tag{5.31}$$

and the Jacobian $\mathbf{J} = \partial \mathbf{s}/\partial \mathbf{q}$ linearises the forward map, $\Delta \mathbf{s} \approx \mathbf{J}\,\Delta \mathbf{q}$. The iteration seeks an update $\Delta \mathbf{q}$ that drives $\mathbf{e}$ to zero, updating $\mathbf{q} \leftarrow \mathbf{q} + \Delta \mathbf{q}$ until the tip error falls below a tolerance.

5.5.2.2 Configuration Parametrisation

The configuration is not parametrised directly by (κ, ϕ) of each section, because ϕ is undefined when a section is straight ($\kappa = 0$) and wraps discontinuously, a singularity noted for this representation by Jones and Walker [42]. Instead, each section is represented by the Cartesian bending components

$$\mathbf{u} = \theta\,[\cos\phi,\ \sin\phi]^{\top}, \qquad \theta = \|\mathbf{u}\|, \quad \phi = \operatorname{atan2}(u_y, u_x), \quad \kappa = \theta/\ell, \tag{5.32}$$

so that a straight section is the well-behaved point $\mathbf{u} = \mathbf{0}$. The configuration vector for the two sections is $\mathbf{q} = [\,u_{x,p}, u_{y,p}, u_{x,d}, u_{y,d}\,]^{\top}$, giving the four bending DoF in a form free of the $\kappa = 0$ singularity.

5.5.2.3 Damped Least Squares

The update is computed by DLS (the Levenberg–Marquardt method), following Buss [21]. Rather than solving $\mathbf{e} = \mathbf{J}\,\Delta \mathbf{q}$ directly, which is unstable near singularities where $\mathbf{J}$ loses rank, the damped method minimizes

$$\|\mathbf{J}\,\Delta \mathbf{q} - \mathbf{e}\|^2 + \lambda^2 \|\Delta \mathbf{q}\|^2, \tag{5.33}$$

where λ is a non-zero damping constant [21]. The solution is

$$\Delta \mathbf{q} = \mathbf{J}^{\top} \left(\mathbf{J}\mathbf{J}^{\top} + \lambda^2 \mathbf{I} \right)^{-1} \mathbf{e}, \tag{5.34}$$

which is Buss's Equation (11) [21]. This form is preferred over the equivalent $(\mathbf{J}^{\top}\mathbf{J} + \lambda^2 \mathbf{I})^{-1}\mathbf{J}^{\top}\mathbf{e}$ because the matrix inverted is only 3×3 (the dimension of the position target) rather than 4×4 [21]. The damping λ smooths the behaviour near singularities, as Buss [21] shows through the singular value decomposition, each singular value σ_i is replaced by $\sigma_i/(\sigma_i^2 + \lambda^2)$, which tends to zero rather than diverging as $\sigma_i \to 0$, so the pseudoinverse instability at singularities is removed. The damping must be "large enough so that the solutions for $\Delta \mathbf{q}$ are well-behaved near singularities, but if it is chosen too large, then the convergence rate is too slow" [21].

The Jacobian is evaluated numerically by finite differences, perturbing each component of $\mathbf{q}$ and recomputing the tip position through the forward kinematics. To prevent unstable large steps, the bending magnitude of each section is clamped at each iteration, following Buss's recommendation that "it is almost always a good idea to clamp the maximum angle change in a single update" [21]. The iteration terminates when the tip error falls below 0.5 mm or a maximum iteration count is reached. If the target is unreachable, the damped method still returns the nearest configuration rather than diverging, a robustness property of DLS that the pseudoinverse lacks [21].

5.5.2.4 Position-Level Inverse and the Actuators

This formulation solves for tip *position*. The tip orientation is not constrained, but follows from the configuration found, the redundant DoF being resolved by the minimum-norm nature of the DLS update. This contrasts with the velocity-level resolved-rate control of the multi-section formulations of Hannan and Walker [41] and Jones and Walker [42], which integrate $\dot{\mathbf{q}} = \mathbf{J}^{+}\dot{\mathbf{x}}$ for real-time teleoperation with continuous position feedback. The manipulator built here is actuated by open-loop stepper motors commanded to absolute positions (Chapter 6). A position-level inverse, solving directly for the target configuration, matches the position-commanded nature of the actuators and avoids the integration drift that velocity-level control would accumulate without continuous feedback.

Once the configuration is found, the arc parameters of both sections are converted to motor commands by the inverse robot-specific mapping (Section 5.3), with the inter-section coupling of Section 5.4 applied in that conversion. The numerical inverse itself operates purely in arc-parameter space through the robot-independent forward model; the robot-specific coupling enters only at the final arc-to-motor step, keeping the two mappings cleanly separated.

5.6 Summary

This chapter developed the kinematic model of the manipulator as the composition of a robot-specific cable-to-arc mapping and a robot-independent arc-to-pose mapping, joined by the arc parameters (κ, ϕ, ℓ) of each section. The forward kinematics assemble these mappings from the commanded motor rotations to the tip pose. The inverse kinematics recover the actuation required for a desired tip position. The principal relations are collected below.

Forward kinematics (actuation $\rightarrow$ pose). Each motor rotation produces a per-pair bending angle through the pulley radius ρ and the tendon offset d, the two pairs of a section combine into its arc parameters, and the two section transforms compose into the tip pose:

$$\theta_{A,B} = \frac{\rho}{d}\,\alpha_{A,B}, \qquad \text{(motor rotation to pair bending, Eq. (5.21))} \tag{5.35}$$

$$\kappa = \frac{\sqrt{\theta_A^2 + \theta_B^2}}{\ell}, \qquad \phi = \operatorname{atan2}(\theta_B, \theta_A), \qquad \text{(pairs to arc parameters, Eqs. (5.17)–(5.19))} \tag{5.36}$$

$$\mathbf{T}(\kappa, \phi, \ell) = \begin{bmatrix} \mathbf{R}(\kappa, \phi, \ell) & \mathbf{p}(\kappa, \phi, \ell) \\ \mathbf{0}^{\top} & 1 \end{bmatrix}, \qquad \text{(arc to section pose, Eq. (5.11))} \tag{5.37}$$

$$\mathbf{T}_d^0 = \mathbf{T}(\kappa_p, \phi_p, \ell_p)\,\mathbf{T}(\kappa_d, \phi_d, \ell_d). \qquad \text{(two-section composition, Eq. (5.13))} \tag{5.38}$$

The distal arc parameters are corrected for inter-section coupling, the distal curvature components being rotated by the proximal bending-plane angle,

$$\begin{bmatrix} \kappa^{*}_{d,x} \\ \kappa^{*}_{d,y} \end{bmatrix} = \begin{bmatrix} \cos\phi_p & -\sin\phi_p \\ \sin\phi_p & \cos\phi_p \end{bmatrix} \begin{bmatrix} \kappa_{d,x} \\ \kappa_{d,y} \end{bmatrix}, \qquad \text{(coupling, Eq. (5.25))} \tag{5.39}$$

equivalently, in cable length, $\Delta\ell_{\text{bend}} = \rho\,\alpha_{\text{d}} + d\,\theta_p \cos(\beta_{\text{d}} - \phi_p)$ (Eq. (5.28)).

Inverse kinematics (pose → actuation). For a single-section arc, the inverse is closed form:

$$\phi = \operatorname{atan2}(y,x), \qquad \theta = 2\,\operatorname{atan2}\left(\sqrt{x^2+y^2}, z\right), \qquad \kappa = \frac{\theta}{L}. \qquad \text{(Eqs. (5.29)–(5.30))} \tag{5.40}$$

For the two-section manipulator, the tip position is solved numerically by DLS, iterating on the configuration $\mathbf{q} = [u_{x,p}, u_{y,p}, u_{x,d}, u_{y,d}]^{\top}$ with $\mathbf{u} = \theta[\cos\phi, \sin\phi]^{\top}$:

$$\mathbf{e} = \mathbf{t} - \mathbf{s}(\mathbf{q}), \qquad \Delta\mathbf{q} = \mathbf{J}^{\top}\left(\mathbf{J}\mathbf{J}^{\top} + \lambda^2\mathbf{I}\right)^{-1}\mathbf{e}, \qquad \mathbf{q} \leftarrow \mathbf{q} + \Delta\mathbf{q}. \qquad \text{(Eq. (5.34))} \tag{5.41}$$

The recovered arc parameters are mapped back to motor rotations by inverting the cable-to-arc mapping, with the coupling of Equation (5.39) applied at the arc-to-motor step.

Chapter 6

Hardware Implementation

This chapter describes the hardware that executes the rotations calculated from the kinematics. The actuators that drive the tendons, the electronics that command them, and the communication that coordinates the two halves of the manipulator. The design problem is shaped by the fact that the manipulator's four motors are split across two actuation boards, one per section, which must act as a single coordinated system. The chapter follows this thread:

- Section 6.1 gives the overall control architecture;
- Section 6.2 details the motors, drivers, and controller boards;
- Sections 6.3.1 and 6.3.2 motivate and describe the communication bus that links the two boards;
- Section 6.4 presents the assembled prototype.

6.1 System Architecture Overview

The manipulator is actuated by four stepper motors, two per section, driving the eight tendons in antagonistic pairs (Chapter 4). These four motors are controlled by two RP2040-based boards, each responsible for the two motors of one section. The two boards operate in a *master–follower* arrangement linked by a CAN bus (Figure 6.1).

The *master* is the board driving the distal section. It carries the entire kinematic model of Chapter 5: it receives the operator's command, solves the inverse kinematics, applies the cable-to-arc mapping and the inter-section coupling, and produces the absolute step targets for all four motors. It then drives its own two motors directly (the distal B motors) and transmits the remaining two step targets (the proximal A motors) over the CAN bus to the follower.

The *follower* is the board driving the proximal section. It carries no kinematics. It listens on the bus for the step targets sent by the master and drives its two motors to those targets, reporting the reached positions back to the master. This division keeps all computation on one board and

reduces the follower to a commanded actuator, so that the two physically separated sections behave as one manipulator under a single kinematic solver.

The control flow from command to motion is therefore:

1. the operator issues a target (a tip position, or direct arc parameters) to the master;
2. the master solves the kinematics and computes the four motors' absolute step targets;
3. the master drives its two distal motors and sends the two proximal targets over CAN;
4. the follower drives its two proximal motors to the received targets and returns their reached positions.

The system runs *open-loop*: the stepper motors carry no encoders, and a motor's position is tracked solely as a count of the steps issued to it. The only feedback on the bus is the follower's report of its commanded step count, which confirms execution but does not measure the actual shape or tip position of the manipulator. No sensor closes the loop on the physical pose. This is consistent with the position-commanded nature of stepper actuation exploited in the inverse kinematics (Section 5.5.2): the controller commands a configuration computed to be correct and the steppers move to it by exact step counts, without measuring the result. The accuracy achievable under this open-loop scheme and the error sources it leaves uncompensated are assessed in Chapter 8.

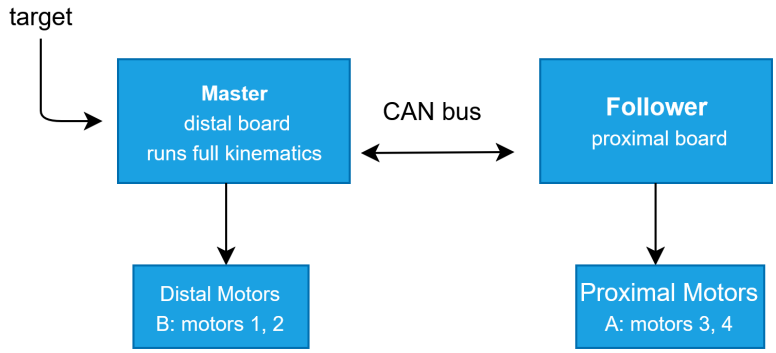


Figure 6.1: Master–follower control architecture.

6.2 Actuation: Motors and Drivers

6.2.1 Actuation Chain

The actuation of each motor follows a fixed signal chain. The RP2040 controller emits a `STEP` pulse train and a `DIR` level on dedicated GPIO lines. The DRV8825 driver translates each step pulse into a commutation of the current in the two motor windings, advancing the rotor by one (micro)step in the commanded direction. The SF2424 stepper motor converts this into mechanical rotation of the pulley, which reels the tendon (Figure 6.2). Because a stepper advances by a fixed angle per step, the controller tracks position simply by counting the steps it has issued, without any position sensor on the motor (Section 6.1).

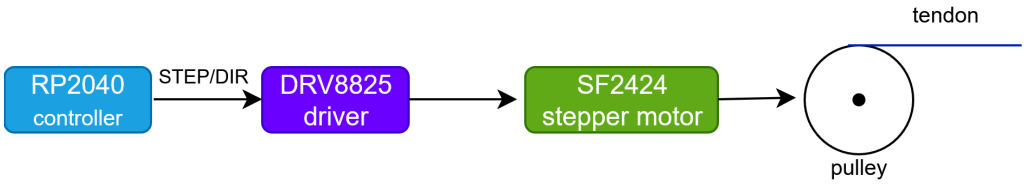


Figure 6.2: The actuation chain for one motor.

6.2.2 Stepper Motors

The motors are SF2424-10B41 bipolar stepper motors. The relevant specifications are summarised in Table 6.1. The basic step angle of 1.8° gives 200 full steps per revolution, the count from which all step-to-angle conversion in the firmware is derived. The holding torque of 0.8 N m sets the available tendon tension: through the 8 mm pulley radius (Section 5.3.4), it corresponds to a tendon force of up to

$$F = \frac{\tau}{\rho} = \frac{0.8\,\mathrm{N\,m}}{0.008\,\mathrm{m}} = 100\,\mathrm{N}, \tag{6.1}$$

a theoretical ceiling at the holding torque; the dynamic torque while turning is lower, but the margin over the force needed to bend the spring-loaded backbone is generous. The rated current of 1 A per phase sets the current limit configured on the drivers.

Table 6.1: Specifications of the SF2424-10B41 stepper motor.

Parameter	Value
Basic step angle	1.8° (200 steps/rev)
Holding torque	0.8 N m
Rated current	1 A per phase
Wiring	Bipolar
Flange size	42 mm
Motor length	59.5 mm
Mass	0.51 kg

6.2.3 Drivers

Each motor is driven by a DRV8825 stepper driver. The driver receives the `STEP` and `DIR` signals from the RP2040 and regulates the winding currents by chopping, holding the current to a limit set by a sense resistor and a reference voltage. On the actuation board the sense resistors are 200 mΩ, sized for the motor's 1 A per phase rating. The step mode is selected by three mode pins (`M0`, `M1`, `M2`) wired to the controller, which the firmware sets at start-up.

6.2.4 Step Resolution

Although the DRV8825 supports microstepping, in which the winding currents are driven in graded levels so that the rotor settles at intermediate positions between full steps, the manipulator is driven in *full-step* mode. All four motors are configured for full stepping (mode pins `M0`, `M1`, `M2` low), giving

$$200 \text{ steps per revolution}, \tag{6.2}$$

or an angular resolution at the motor of $360^\circ/200 = 1.8^\circ$ per step. Through the actuation scale factor $\rho/d \approx 0.286$ of Equation (5.21), this corresponds to a section bending resolution of approximately

$$\frac{\rho}{d} \times 1.8^\circ \approx 0.51^\circ \tag{6.3}$$

per step. The positioning accuracy actually achieved by the manipulator, which is also affected by backlash, tendon compliance, and the unmodelled effects noted in Chapter 5, is evaluated in Chapter 8. The motor position is tracked as an exact count of the steps issued, the step pulses being emitted and counted directly so that the count is not lost under communication load (Section 6.3.2).

6.2.5 Controller and Actuation Boards

The electronics of each node are built from two stacked boards: a TL.HEART control board carrying the RP2040 microcontroller and the CAN transceivers, and a TL.HEART#STEPPER actuation board carrying the two DRV8825 drivers and the connectors for that node's two motors (Figure 6.3). The two boards stack through their inter-board connectors, so each node is a control board and an actuation board driving two motors. Two such nodes, one per section, make up the four-motor system. The TL.HEART is a general-purpose board developed at the Laboratório de Robótica e IoT para Agricultura e Floresta de Precisão Inteligente (TRIBE) and carries additional interfaces not used in this work.

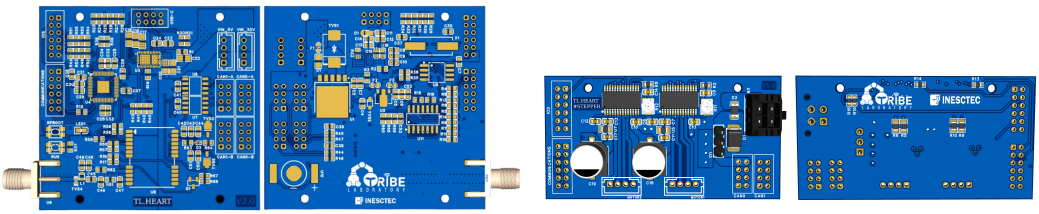


Figure 6.3: The two boards forming one node: the TL.HEART control board and the TL.HEART#STEPPER actuation board.

6.3 Inter-Board Communication: CAN

The four motors are split across two nodes, two motors per node (Section 6.2). Because the kinematics are solved on a single board (Section 6.1), that board must command the two motors

of the other node, so the two nodes must exchange data reliably and in real time. This section motivates the choice of a CAN bus for this link and describes its implementation.

6.3.1 Selection of Communication Protocol

The communication need is modest in volume but strict in timing: a few small messages (motor targets and status) must pass between the two nodes promptly and without loss, in an environment where four stepper drivers switch large currents close to the signal wires. Several standard embedded buses could carry this traffic. Table 6.2 compares the four most common candidates across the properties that matter here.

Table 6.2: Comparison of candidate inter-board communication protocols.

	UART	I^{2}C	SPI	CAN
Topology	point-to-point	multi-device	multi-device	multi-master
Nodes	2	many (addressed)	few (per CS)	many (arbitrated)
Signal wires	2	2	3+ (+CS/node)	2 (differential)
Signalling	single-ended	single-ended	single-ended	differential
Typical range	short	on-board	on-board	long
Complexity	low	low	medium	high

UART is the easiest to set aside: it is a point-to-point link, so it doesnt scale beyond two nodes. The remaining three are all workable, and it is worth being clear that, for the scale of this system, two nodes a short distance apart exchanging a few bytes, a simpler master–slave bus such as I^{2}C or Serial Peripheral Interface (SPI) would have been sufficient and lighter to implement. The volume and distance of the traffic do not, on their own, demand CAN.

CAN [24] was nonetheless chosen, for robustness and headroom rather than strict necessity. The decisive property is its differential signalling, which is markedly more immune to electrical noise than the single-ended I^{2}C and SPI: the signal wires run alongside four DRV8825 drivers switching coil currents of the order of an ampere, exactly the conditions under which single-ended buses are prone to corrupted transfers. CAN further provides built-in error detection and automatic retransmission of corrupted frames, and its message-based, identifier-addressed protocol maps naturally onto the small fixed set of command and status messages used here (Section 6.3.2). It also leaves room to add nodes without rewiring should the manipulator grow. Its cost is a more complex protocol than I^{2}C or SPI, but here that complexity is largely absorbed by the `can2040` software implementation (Section 6.3.2), which made CAN practical on the RP2040 without dedicated hardware. The choice is therefore best read as trading a small amount of implementation effort for noise immunity and scalability that a motor-driven system benefits from, even though a simpler bus could have met the bare requirement.

6.3.2 Implementation

The bus is realised in hardware by a MAX33041 CAN transceiver on each control board, with the standard split termination on the bus pair. On the controller side, rather than a dedicated CAN peripheral, the protocol is implemented in software by the `can2040` library, which uses one of the RP2040's PIO blocks to handle the CAN bit-level timing. The bus runs at 500 kbit s^{-1}.

The protocol uses a small fixed set of message identifiers rather than a dynamic addressing scheme, which is sufficient for two nodes with predetermined roles. Three messages are defined (Table 6.3): a `MOVE` message carrying the two proximal motor targets, a `HOME` message to zero the position counters, and a `STATUS` message by which the follower reports its reached positions back to the master.

Table 6.3: CAN message set.

Message	ID	Direction	Payload
`MOVE`	`0x200`	master → follower	two `int32` absolute step targets
`STATUS`	`0x201`	follower → master	two `int32` reached step counts
`HOME`	`0x202`	master → follower	none (zeroes the counters)

A deliberate feature of the protocol is that motor commands are sent as *absolute* step targets, not incremental moves. Each `MOVE` carries the target position the follower's motors should reach, expressed as an absolute step count from the home pose. This makes the link self-correcting against a lost or corrupted message: if one `MOVE` is missed, the next one still carries the complete intended position, so the follower converges to the correct configuration on the following command rather than accumulating the error a missed *incremental* step would cause. Combined with CAN's own error detection, this keeps the two nodes' motors consistent without any higher-level acknowledgement protocol.

6.4 Physical Prototype

Figure 6.4 shows the assembled control and actuation electronics described in this chapter: the two stacked board pairs that form the master and follower nodes, and the wiring that links them to power, to the motors, and to each other. The mechanical manipulator these electronics drive, the two-section tendon-driven backbone and the stepper motors at its base, is presented in Chapter 4.

The wiring in the figure carries the three signals the system needs. The red and black leads bring the power supply to the two nodes; the multicoloured four-wire cables run from each node to its stepper motors, carrying the two motor phases; and the white-and-blue twisted pair links the two nodes and carries the CAN bus.

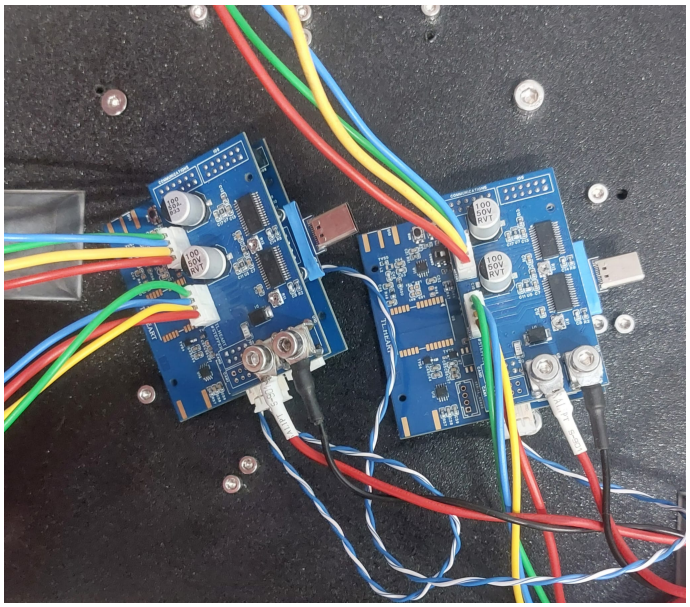


Figure 6.4: The two control and actuation nodes, each a stacked control and actuation board pair, with the power leads, the stepper-motor cables, and the white-and-blue CAN pair linking the two nodes.

6.5 Summary

This chapter described the hardware that executes the kinematic model of Chapter 5. The manipulator is actuated by four SF2424 stepper motors, two per section, each driving an antagonistic tendon pair through an 8 mm pulley. The motors are driven by DRV8825 drivers configured for full stepping, giving a design actuation resolution of approximately 0.51° of section bending per step, and are commanded by two RP2040-based control nodes in a master–follower arrangement: the master solves the full kinematics and drives the two distal motors, while the proximal motors are commanded over a $500\,\mathrm{kbit\,s^{-1}}$ CAN bus carrying absolute step targets, a scheme that is self-correcting against lost messages. The system runs open-loop, tracking position as an exact count of issued steps with no position feedback from the manipulator. The accuracy achievable under this implementation is the subject of Chapter 8.

Chapter 7

Software Implementation

The hardware of Chapter 6 provides the actuators and the link between the two control nodes. The software described here drives them according to the kinematic model of Chapter 5. Its task is to turn a commanded tip position into coordinated motion of the four motors, distributed across the two RP2040 boards. This chapter describes how the firmware is organised, how the kinematics are implemented as software modules, how the two boards cooperate, and how motor motion is produced. The kinematic equations themselves are not repeated here, they are developed in Chapter 5, and this chapter is concerned with their realisation in embedded code.

7.1 Software Architecture

The kinematic model of Chapter 5 is realised in firmware as two separate modules whose division mirrors the robot-independent and robot-specific decomposition introduced there. Keeping the two halves in separate units, rather than merging them into a single kinematic routine, isolates the universal geometry from everything that depends on this particular manipulator, so that the parts most likely to change with the mechanical design are confined to one module.

The first module implements the robot-independent mapping. It takes the arc parameters of a section, the curvature, the bending-plane angle, and the fixed length, and returns the homogeneous transform relating the section's base frame to its tip frame, together with the composition of two such transforms into the full tip pose. This mapping is the same for any constant-curvature continuum robot and makes no reference to cables, motors, or the dimensions of the build.

The second module implements the robot-specific mapping for this eight-cable, four-motor, two-section design. It is where the mechanical constants live, the cable offset, the two section lengths, the pulley radius, the cardinal and diagonal cable layout, and the antagonistic motor pairing, and it converts between the sections' arc parameters and the four motor angles in both directions. The inverse direction also accounts for the inter-section coupling of Chapter 5, and a final conversion expresses the motor angles as the integer step counts the motors are commanded with. A single run-time flag held in this module selects whether the manipulator is treated as one section or two, matching the operating modes of Section 7.2.

The two modules exchange data through a small set of plain structures, the arc parameters of a section and the four motor angles, which form the vocabulary in which the rest of the firmware expresses a configuration. Both modules run on the master node, which carries the entire kinematic computation. The follower node carries neither and only actuates (Section 6.1).

7.2 Operating Modes

The firmware supports two operating modes, selectable at run time, which differ in how many sections of the manipulator are actuated and in the inverse-kinematics path taken to reach a target. The mode is held in a single state variable and switched by a command, so the user can move between them without reflashing.

7.2.1 Single-Section Mode

In single-section mode, the entire backbone is treated as one constant-curvature arc driven by the two distal motors, while the proximal motors are held at their home position. For a target tip position, the firmware uses the closed-form single-section inverse of Section 5.5.1: the arc parameters are computed directly, with no iteration. The robot-specific module is placed in a matching state in which the proximal section is forced straight, so that the inter-section coupling term vanishes and the distal solve reduces to a plain single-section mapping.

7.2.2 Two-Section Mode

In two-section mode, both sections are actuated, and the full four-motor model applies. For a target tip position, the firmware invokes the numerical DLS solver of Section 5.5.2, which iterates on the configuration of both sections until the tip reaches the target or the iteration limit is met. The solver operates through the robot-independent model. The inter-section coupling is then applied when the resulting arc parameters are converted to motor commands. The two proximal motor targets are dispatched over CAN while the two distal motors are driven locally, so the four motors move toward the same solved configuration (Section 6.1). This mode reaches a volume with one redundant DoF, and is the mode used for the positioning evaluation of Chapter 8.

7.2.3 Direct and Safety Commands

Alongside the position-target commands, the firmware accepts direct arc commands, in which the arc parameters of the sections are specified explicitly rather than solved for, bypassing the inverse kinematics. These are used to exercise individual sections during testing and to verify the forward model against the physical manipulator. Before any solved or direct configuration is sent to the motors, the firmware checks it against the mechanical bending limit and refuses to move if a section's bend exceeds that limit, or if a position target is found to lie outside the reachable workspace (reported by the solver as a large residual). This keeps an unreachable or unsafe command from driving the motors past their intended range.

7.3 Stepper Control

Once a configuration has been reduced to four absolute step targets, the firmware must drive each motor to its target and keep an accurate record of where the motor is. Both tasks are handled by a stepper class that wraps the DRV8825 driver and presents a single operation: move the motor to an absolute step count. Because the system is open-loop (Section 6.1), this recorded count is the only estimate of motor position the controller has, so it must remain exact.

The control was therefore written so that the count is produced by the same loop that produces the motion, with no interrupt in the counting path. To move a motor, the firmware computes the signed difference between the target and the recorded position and emits exactly that many pulses, toggling the `STEP` line directly and incrementing the recorded position as each pulse is issued. The count is thus generated in lock-step with the pulses and cannot be desynchronised by `can2040`, restoring the exact step tracking on which the open-loop scheme and the actuation resolution of Equation (6.3) depend. The pulse rate is held at a fixed value and the driver is put to sleep between moves. Targets are absolute counts measured from the home pose, so a homing command simply zeroes the recorded position, consistent with the absolute-target convention of the CAN protocol (Section 6.3.2).

This emit-and-count loop is blocking: a move occupies the master until its last pulse is issued. One consequence is that the two distal motors are driven one after the other rather than simultaneously, and that command parsing and bus servicing are suspended for the duration of a move. For the short, low-speed moves of the bench positioning evaluated in Chapter 8 this is acceptable, but it precludes the coordinated, simultaneous motion that a dynamic trajectory would require, and is noted as a limitation in Chapter 9.

7.4 Command Interface

The manipulator is operated from a host computer over a USB serial console attached to the master. The interface is line-oriented: characters are read as they arrive, without blocking the control loop, and accumulated until a line is complete, at which point the line is interpreted by its leading token. The command set is small and is summarised in Table 7.1.

Table 7.1: USB-serial command set accepted by the master.

Command	Arguments	Action
`x y z`	tip position (mm)	move the tip to the target position
`a`	ϕ, θ (deg)	single-section direct arc
`b`	$\phi_p, \theta_p, \phi_d, \theta_d$ (deg)	two-section direct arcs
`s1` / `s2`	—	select single- or two-section mode
`z`	—	straighten all sections
`h`	—	set the current pose as home

The position command is the primary one: a target tip position, given in millimetres in the base frame, is passed to the single- or two-section solver according to the mode currently selected (Section 7.2). The two direct-arc commands bypass the inverse kinematics and specify the arc parameters of the sections explicitly. They are the means by which individual sections are exercised and the forward model is checked against the physical manipulator (Section 7.2.3). The mode commands switch between single- and two-section operation at run time, the straighten command returns every section to its home, zero-curvature shape, and the home command declares the current pose to be the zero of the step counters, zeroing the master's own counters and broadcasting the same instruction to the follower over CAN. Before any command that results in motion, the bending-angle limit and workspace check of Section 7.2.3 are applied, so that an out-of-range target or an excessive bend is rejected rather than driven.

7.5 Control Flow

The preceding sections describe the firmware's parts in isolation; this section follows a single command through them. The hardware-level division of labour, the master solving all the kinematics and the follower only actuating, is described in Section 6.1 and is not repeated here; what follows is the software path on the master that turns one tip-position command into the four step targets that division requires (Figure 7.1).

A position command in two-section mode enters the numerical position solver of Section 7.2.2, which returns the proximal and distal arc parameters that place the tip at the target. These arc parameters are passed to the robot-specific inverse, which solves the proximal section first and then the distal section with the inter-section coupling added back, yielding the four motor angles; the coupling thus enters at the arc-to-cable stage, not in the position solve itself. The motor angles are converted to absolute step counts at the resolution of Section 6.2.4. The two proximal targets are then transmitted to the follower over CAN while the two distal targets are driven locally, and the follower reports its reached counts back over the bus (Section 6.3.2). In single-section mode the path is shorter: the closed-form whole-arc inverse of Section 7.2.1 produces the two distal targets directly, and the proximal motors are commanded to their home position.

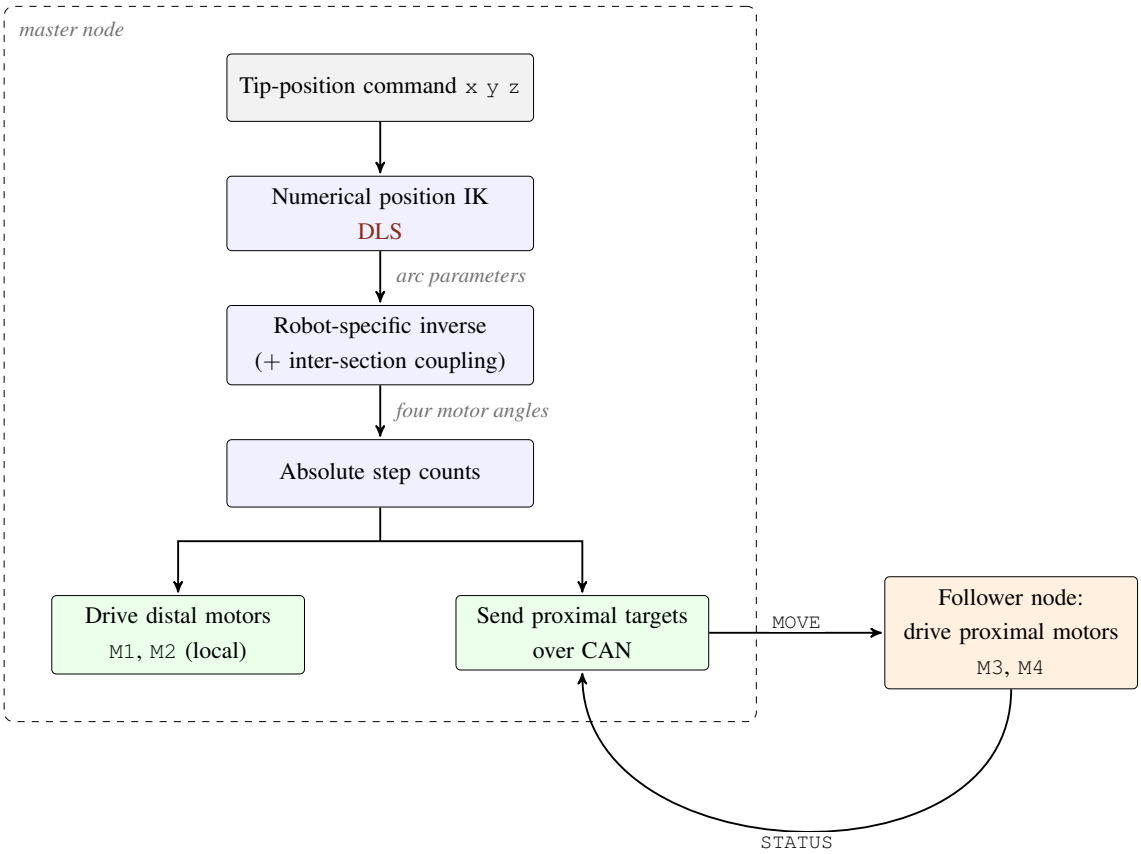


Figure 7.1: Control flow on the master, from a tip-position command to coordinated motor motion.

7.6 Summary

This chapter described how the kinematic model of Chapter 5 is realised in the firmware that drives the manipulator. The kinematics are implemented as two modules following the robot-independent and robot-specific split, the first a hardware-free evaluation of the constant-curvature geometry and the second the cable, coupling, and step-count mapping particular to this build. The firmware offers two operating modes, a fast closed-form single-section solve and a numerical two-section solve, selectable at run time and reached through a small USB-serial command set that also exposes direct-arc and homing commands. Motor motion is produced by an exact-step-count driver that emits and counts each pulse in a single blocking loop, a design adopted so that the position count survives the interrupt load of the CAN stack, at the cost of driving the local motors sequentially. A command is carried through the firmware by solving the inverse kinematics, applying the inter-section coupling in the cable mapping, converting to absolute step targets, and dispatching those targets to the local motors and, over CAN, to the follower. The positioning accuracy this implementation achieves is evaluated in Chapter 8.

Chapter 8

Experimental Validation and Results

The preceding chapters described a manipulator that was designed, built, and given a kinematic model and the firmware to drive it. This chapter reports how well the assembled system performs in placing its tip at a commanded position. It is a positioning evaluation only. The manipulator does not yet carry an end-effector or any perception, and harvesting was not attempted. What is assessed here is the accuracy with which the open-loop, two-section system reaches a set of commanded tip positions, and the chapter is deliberately explicit about the limits of the measurement on which that assessment rests.

8.1 Experimental Setup

The manipulator was commanded to eight target tip positions, and for each the position actually reached by the tip was measured and compared with the command. All targets were expressed in the base frame used by the firmware, with its origin at the first spacer disc (disc 0), the same frame in which the kinematic model of Chapter 5 is defined. The system was operated in two-section mode (Section 7.2.2), the mode intended for general positioning, with the numerical solver placing both sections to reach each target.

The eight targets were chosen to sample the reachable workspace at differing depths and directions rather than to lie on any regular grid, so that the evaluation exercises the manipulator across its range rather than at a single favourable pose. Three considerations guided the selection. First, the principal directions were covered: targets 1–4 command a full outward bend along $+x$, $-x$, $+y$, and $-y$ respectively, with both sections bending equally in a single plane, which probes whether the manipulator behaves symmetrically about its base axis. Second, intermediate directions were added, targets 5 and 7 reaching diagonally between the principal axes, so that the directions sampled are not confined to the planes in which the model is most easily visualised. Third, and most importantly, the targets were chosen to elicit qualitatively different backbone *shapes*: in targets 1–5 and 7 the two sections bend in the same plane and form a planar C-shape, whereas target 6 commands the proximal and distal sections into opposing planes and forms a non-planar S-shape,

which exercises the two-section capability and the inter-section coupling directly. Depth was varied alongside direction, targets 1–5 and 7 sitting at the lower height $z = 379\,\text{mm}$, where the target demands a large outward bend, and targets 6 and 8 sitting higher at $z = 528\,\text{mm}$, where the tip is reached with less curvature. Together the eight cases place the manipulator in different planes and different shapes, from a near-vertical low-curvature pose to fully bent reaches and a double-curved S, rather than at a single representative configuration.

For each target, the tip position reached by the manipulator was measured by hand with a tape measure against a physical reference, rather than against an external tracking system. To reduce the influence of individual reading errors, each reported position is the median of fifteen measurements of the same reached pose. This manual reference is the central limitation of the setup and is discussed in Section 8.4. Without a motion-capture or other instrumented ground truth, the figures reported here are indicative of the system's positioning behaviour rather than a statistically characterised accuracy. The equipment used was the assembled prototype of Figure 4.7, driven from a host computer over the USB-serial command interface of Section 7.4.

8.2 Methodology

The quantity evaluated is the positioning error, the difference between the commanded tip position and the position the tip actually reached. For a target commanded at $\mathbf{p}_c = (x_c, y_c, z_c)$ and measured at $\mathbf{p}_m = (x_m, y_m, z_m)$, the per-axis error is

$$(\Delta x, \Delta y, \Delta z) = (x_m - x_c,\ y_m - y_c,\ z_m - z_c), \tag{8.1}$$

and the overall error is its Euclidean norm,

$$\|\mathbf{e}\| = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}. \tag{8.2}$$

The test protocol was the same for each of the eight targets. The manipulator was first straightened and homed (Section 7.4), so that all step counts began from a known zero. The target was then commanded in two-section mode. The firmware solved the inverse kinematics, applied the inter-section coupling, and drove the four motors to the resulting configuration (Section 7.5). Once motion had completed, the reached tip position was measured (as the median of fifteen hand measurements) and the per-axis error recorded. The manipulator was then re-homed before the next target.

This protocol has two limitations that shape what its results can show:

- **Accuracy limitation.** Selected targets were commanded several times, and the reached position was consistent between repetitions where the manipulator returns to nearly the same point each time. The error reported here is therefore systematic rather than random, a repeatable offset between the commanded and the reached position rather than scatter. What is not characterised is the full statistical distribution of that error. The repetitions were

informal and limited in number, so a formal repeatability figure (a standard deviation over many trials at each target) is not reported, even though the qualitative consistency was clear.

- **Manual reference, not an instrumented ground truth.** The reached tip was measured by hand against a physical reference, rather than against an external tracking system such as a motion-capture rig. The measurement therefore has an uncertainty of its own, and that uncertainty is not separated from the error of the manipulator. Taking the median of fifteen readings reduces the random component of this reading error, but its systematic part remains, and the reported figure still combines how far the tip landed from the target with how accurately that landing point could be read off. With an instrumented ground truth the two could be told apart, and the manipulator's own error isolated.

A more rigorous characterisation would measure each target against an external tracking system over a large number of repetitions, generating a mean error, a standard deviation, and a formal repeatability figure. Such a characterisation and its absence is revisited in Section 8.4.

8.3 Results

Table 8.1 reports the commanded position and the measured per-axis and Euclidean error for each of the eight targets. Figure 8.1 shows the manipulator at each commanded target.

Table 8.1: Commanded targets and measured positioning error for the eight cases.

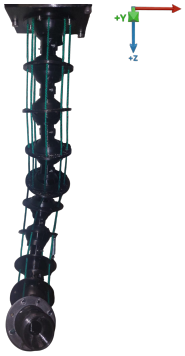
	Commanded [mm]			Error [mm]			$\|\mathbf{e}\|$ [mm]
Target	x_c	y_c	z_c	Δx	Δy	Δz	
1	370	0	379	-115	0	-99	151.7
2	-370	0	379	125	0	-86	151.7
3	0	370	379	0	-107	-115	157.1
4	0	-370	379	0	93	-132	161.5
5	262	262	379	-89	-54	-19	105.8
6	-136	-136	528	26	22	-22	40.5
7	-220	220	379	70	-50	51	100.0
8	-80	0	528	30	0	-52	60.0
						Mean $\|\mathbf{e}\|$	116.1
						Minimum $\|\mathbf{e}\|$	40.5
						Maximum $\|\mathbf{e}\|$	161.5

The errors recorded across the eight targets are summarised in Table 8.1. The Euclidean error ranges from 40.5 mm at target 6 to 161.5 mm at target 4, with a mean of 116.1 mm. The five targets commanded furthest from the base axis (targets 1–5, at $z_c = 379$ mm with a horizontal

reach near 370 mm) carry the largest errors, the tip falling short of the commanded radius by 93 mm to 125 mm. The two targets with the smallest commanded reach, targets 6 and 8, give the smallest errors (40.5 mm and 60.0 mm). In every case the reached tip is closer to the base axis than commanded and, except at target 7, lower in z. The vertical axis carries the largest mean per-axis error (72 mm), ahead of x (57 mm) and y (41 mm). The error thus grows with the commanded outward bend: the further the manipulator is asked to reach from its base axis, the larger the shortfall.

(a) Target 1

(b) Target 2

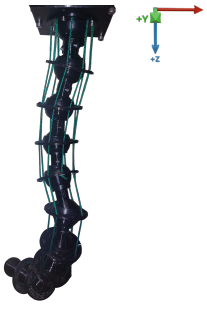


(c) Target 3

(d) Target 4

(e) Target 5

(f) Target 6



(g) Target 7

(h) Target 8

Figure 8.1: The manipulator at each of the eight commanded targets.

8.4 Discussion

The manipulator reached the eight commanded targets with a mean positioning error of 116.1 mm and a maximum of 161.5 mm. These errors are large, and this section is explicit about that.

The requirement of Section 4.1 was that the tip reach a commanded target to within approximately the size of an arbutus berry, on the order of 20 mm in diameter, so that a future end-effector could engage the fruit despite residual error. Measured against this requirement, the manipulator does not yet meet it: with a mean error of 116.1 mm, more than five times the berry scale, bare positioning would not, on its own, place a gripper reliably onto a fruit. As a point of comparison, the spring-backbone continuum robot for fruit and vegetable picking of Huertas Niño et al. reports tip positioning errors of the order of 6 mm to 24 mm [38]; the errors obtained here, a mean of 116.1 mm, are several times larger, which is consistent with this being an early, open-loop prototype rather than a refined system.

One observation, however, qualifies the picture in the system's favour. When the same target was commanded repeatedly, the reached position was consistent between attempts: the error is *systematic*, a repeatable offset between the commanded and the reached pose, rather than random scatter. This matches the high repeatability reported for the single-motor-per-antagonistic-pair actuation scheme adopted here by Marzi et al. [23], the mechanism returns reliably to the same shape for the same command, so what is deficient is accuracy, not repeatability. A systematic error is, in principle, correctable, by calibration or by feedback, which is what makes the sources below worth identifying precisely.

Several effects combine to produce the error, and they are largely deterministic, which is why it repeats. The most fundamental is that the system is open-loop (Section 6.1): the controller computes a configuration it believes to be correct and drives the motors to it by step count, with no measurement of the actual shape, so no error is sensed or corrected and any modelling or mechanical imperfection passes straight through to the tip. The kinematic model is itself a simplification, it is purely geometric and assumes a perfect constant-curvature arc produced rigidly by the tendon displacements, with an inextensible backbone and no account of forces, gravity, or structural stiffness. The physical manipulator departs from each of these assumptions. Within the transmission, the cable can slip on the pulley, so a commanded motor rotation is not fully delivered as cable travel and the section bends less than intended, and friction in the tendon paths and in the spherical joints both withholds commanded motion and introduces hysteresis on reversals. Finally, the backbone has no restoring or stiffening element: at more demanding poses it sags under its own weight and the spacer discs can rotate relative to one another, so the realised shape drifts away from the modelled arc by an amount that grows with the reach and load of the pose. The inter-section coupling (modelled only approximately in Chapter 5), the coarse full-step actuation resolution of about 0.51° per step (Equation (6.3)), and the residual compliance of the tendons add further, smaller contributions on top of these.

Crucially, these sources are not intrinsic limits of the approach but properties of this early prototype. The results of this chapter should therefore be read as an indicative characterisation,

within the measurement limitations set out in Section 8.2, rather than a final accuracy specification. They show a manipulator that positions its tip repeatably but with a large, systematic error whose causes are identified above. Closing that gap, through the structural and feedback improvements just described, is the natural continuation of this work.

8.5 Summary

This chapter evaluated the positioning of the assembled manipulator by commanding it to eight target tip positions in two-section mode and measuring the error at each. The per-axis and Euclidean errors are reported in Table 8.1, and the manipulator is shown at each target in Figure 8.1. The tip reached the targets with a mean error of 116.1 mm and a maximum of 161.5 mm. The errors are large but systematic: the tip reaches a repeatable position for each command, consistently offset from the target rather than scattered around it. This behaviour is consistent with the deterministic sources identified above, chiefly the open-loop control and the simplifications of the geometric model, none of which is sensed or corrected. The evaluation is, moreover, a positioning check measured against a manual reference rather than an instrumented ground truth, so it indicates how the system behaves rather than fixing a final accuracy or repeatability figure. A statistically characterised evaluation, and the structural and feedback improvements that the error sources point to, are left to future work (Chapter 9).

Chapter 9

Conclusions and Future Work

This final chapter assesses the work against the objectives set out in Section 1.4, restates what it delivered, and outlines the directions that would carry it toward its longer-term goal.

9.1 Conclusions

The general aim of this dissertation was to advance a vision-based arbutus-harvesting system from simulation toward physical realisation, concentrating on the robotic manipulator and its kinematic control. Measured against the four specific objectives, the work delivered the first three and carried out the fourth with an informative result.

A two-section, tendon-driven continuum manipulator was designed for the constraints of reaching fruit within a canopy, arrived at through successive design iterations (Chapter 4), and built as a physical prototype: the spacer-disk backbone, the eight-tendon antagonistic actuation driven by four stepper motors, and the two-node RP2040 control system communicating over CAN (Chapter 6). Its kinematics were then formulated and implemented in both directions (Chapter 5), following the robot-independent and robot-specific decomposition, with a closed-form single-section solution and a numerical two-section solver, and realised in firmware that turns a commanded tip position into coordinated motor motion (Chapter 7). To the extent of these three objectives, the result is a complete, controllable manipulator that can be commanded to a tip position and that drives itself there.

The fourth objective, the experimental validation of positioning accuracy under laboratory conditions, was carried out in Chapter 8, and its outcome is the most qualified of the four. The manipulator reaches commanded positions repeatably, returning to nearly the same pose each time a target is issued, but with a large error between the reached and the commanded position. Importantly, this error is systematic, and its likely sources were discussed (Section 8.4) as being the open-loop control, the simplifications of the purely geometric model, and the structural and transmission imperfections of an early prototype. The validation therefore did not certify an accuracy figure fit for harvesting. What it established is that the system positions its tip in a repeatable manner, and that the gap to a useful accuracy is likely due to correctable mechanical and modelling

limitations rather than to any fundamental inconsistency in the control approach. That is a good foundation to build on, and it sets a clear agenda for the work that should follow.

9.2 Contributions

In line with the objectives, this work makes four contributions. It presents the design of a tendon-driven continuum manipulator for arbutus harvesting, developed through successive iterations toward a buildable form. It delivers a physical prototype that extends a previously simulation-only system to real hardware, including its actuation and its distributed control electronics. It provides an implementation of the manipulator's forward and inverse kinematics for commanded positioning of the tip, structured so that the universal geometry and the design-specific cable mapping are kept separate. And it provides an experimental evaluation of the prototype's positioning, which, beyond a measurement of error, characterises the behaviour of the system and points to the probable dominant sources of its inaccuracy. Together these establish the manipulator's positioning capability and its limits, and provide the basis on which a field-capable harvesting system could be developed.

Additionally, the work developed in this dissertation resulted in the submission of a scientific paper to a conference, which is currently under review.

9.3 Future Work

The potential error sources discussed in Section 8.4 point to two concrete improvements for a future revision of the manipulator. The first is mechanical: a spring or compliant sheath running along the backbone would give the manipulator a restoring force to hold its commanded shape instead of sagging at demanding poses, and would resist the relative rotation of the spacer discs, adding rigidity and torsional stability. This comes at a cost, the motors must then work against the spring as well as the load, so it raises the actuation force required, but it would directly attack the structural component of the error. The second is sensory: a camera or other visual feedback closing the loop around the tip would let the controller measure the actual tip position and correct toward the target, absorbing exactly the systematic offset that the present open-loop scheme cannot.

Independently of those improvements, the positioning itself should be characterised more rigorously than was possible here. Commanding each target many times and measuring every attempt against an instrumented ground truth, such as a motion-capture system, would yield a mean error, a dispersion, and a formal repeatability figure, separating the manipulator's own error from the uncertainty of a manual measurement and turning the indicative results of Chapter 8 into a quantitative specification.

Beyond the manipulator, the larger system remains to be assembled. The prototype carries no end-effector. A gripper or detachment mechanism suited to the arbutus berry would have to be designed and integrated at the tip, and the kinematics extended to account for its geometry. The perception side, the vision that would locate ripe berries and supply the target positions this work

assumes are given, has likewise still to be integrated, so that targets come from the scene rather than from an operator. With an accurate manipulator, an end-effector, and perception in place, the system could then be taken from the laboratory to field testing on real plants.

The destination of all of this is the selective harvesting that names the project: picking ripe arbutus berries while preserving the flowers that set the following season's fruit. That selectivity is a demanding goal, requiring the accuracy, the end-effector, and the perception above to act together in an unstructured canopy, and it is the natural continuation toward which the manipulator presented in this dissertation is a first physical step.

References

[1] A. Singh, T. Biswal, U. Maharana, A. Al Khatib, and P. Mishra, "Big data in agriculture: Acquisition, processing and implications," in *Smart Farming, Smarter Solutions: Revolutionizing Agriculture with Artificial Intelligence*, CRC Press, 2025, pp. 23–46. DOI: 10.1201/9781003536932-2.

[2] H. Shankarappa and V. Kumar, "Automating agriculture: Robotics and ai for a greener future," in *Sustainable Agriculture Applications Using Large Language Models*, Bentham Science Publishers, 2025, pp. 80–95. DOI: 10.2174/9798898810184125010008. [Online]. Available: https://doi.org/10.2174/9798898810184125010008.

[3] M. Naik et al., "Robotics on the farm: Automating tasks for increased productivity," in *Smart Farming, Smarter Solutions: Revolutionizing Agriculture with Artificial Intelligence*, CRC Press, 2025, pp. 99–117. DOI: 10.1201/9781003536932-5. [Online]. Available: https://doi.org/10.1201/9781003536932-5.

[4] Z. Chen et al., "A comprehensive review of obstacle avoidance for autonomous agricultural machinery in multi-operational environment," *Artificial Intelligence in Agriculture*, vol. 16, no. 1, pp. 139–163, 2026. DOI: 10.1016/j.aiia.2025.10.001. [Online]. Available: https://doi.org/10.1016/j.aiia.2025.10.001.

[5] Y. Xiong, Y. Ge, and P. From, "An improved obstacle separation method using deep learning for object detection and tracking in a hybrid visual control loop for fruit picking in clusters," *Computers and Electronics in Agriculture*, vol. 191, p. 106 508, 2021. DOI: 10.1016/j.compag.2021.106508. [Online]. Available: https://doi.org/10.1016/j.compag.2021.106508.

[6] L. Yang, H. Li, S. Lin, and Y. Xiong, "High-tolerance soft-rigid gripper for low-damage robotic strawberry harvesting," *IEEE Robotics and Automation Letters*, vol. 11, no. 1, pp. 121–128, 2026. DOI: 10.1109/LRA.2025.3630528. [Online]. Available: https://doi.org/10.1109/LRA.2025.3630528.

[7] W. Hua et al., "Vacuum suction end-effector development for robotic harvesters of fresh market apples," *Biosystems Engineering*, vol. 249, pp. 28–40, 2025. DOI: 10.1016/j.biosystemseng.2024.11.011. [Online]. Available: https://doi.org/10.1016/j.biosystemseng.2024.11.011.

[8] C. Fonseca, *Medronheiro: Uma espécie multifuncional*, https://florestas.pt/conhecer/medronheiro-uma-especie-multifuncional/, Accessed: 2026-05-23, Sep. 2022.

[9] C. F. Gomes, G. Botelho, and R. Guilherme, *Medronheiro — Manual de Boas Práticas para a Cultura*. Cooperativa Portuguesa de Medronho CRL, 2017, ISBN: 9789729920561.

[10] The Navigator Company, *Medronheiro*, https://biodiversidade.com.pt/biogaleria/medronheiro/, Accessed: 2026-05-23.

[11] J. C. de Sousa Pereira, "Vision-based robotic manipulator for arbutus berry harvesting," Master's Dissertation, Faculty of Engineering, University of Porto, Porto, Portugal, Jul. 2025. [Online]. Available: https://hdl.handle.net/10216/168751.

[12] A. Bechar and C. Vigneault, "Agricultural robots for field operations: Concepts and components," *Biosystems Engineering*, vol. 149, pp. 94–111, 2016. DOI: 10.1016/j.biosystemseng.2016.06.014. [Online]. Available: https://doi.org/10.1016/j.biosystemseng.2016.06.014.

[13] A. Kutyrev et al., "Robotic platform for horticulture: Assessment methodology and increasing the level of autonomy," *Sensors*, vol. 22, no. 22, p. 8901, 2022. DOI: 10.3390/s22228901. [Online]. Available: https://doi.org/10.3390/s22228901.

[14] C. Yin, J. Huang, Y. Xia, H. Zheng, W. Fu, and B. Zhang, "Design, development, integration and field evaluation of a dual robotic arm mango harvesting robot," *Journal of Field Robotics*, vol. 42, no. 7, pp. 3705–3725, 2025. DOI: 10.1002/rob.22580. [Online]. Available: https://doi.org/10.1002/rob.22580.

[15] S. Tiefeng, Z. Libin, D. Mingyu, B. Guanjun, and Y. Qinghua, "Fruit harvesting continuum manipulator inspired by elephant trunk," *International Journal of Agricultural and Biological Engineering*, vol. 8, no. 1, pp. 57–63, 2015. DOI: 10.3965/j.ijabe.20150801.008. [Online]. Available: https://doi.org/10.3965/j.ijabe.20150801.008.

[16] Y. Yang and W. Zhang, "Et arm: Highly compliant elephant-trunk continuum manipulator," in *Intelligent Robotics and Applications (ICIRA 2014), Lecture Notes in Computer Science*, vol. 8917, Springer, 2014, pp. 288–299. DOI: 10.1007/978-3-319-13966-1_29. [Online]. Available: https://doi.org/10.1007/978-3-319-13966-1_29.

[17] R. Peng, Y. Wang, M. Lu, and P. Lu, "A dexterous and compliant aerial continuum manipulator for cluttered and constrained environments," *Nature Communications*, vol. 16, no. 1, p. 889, 2025. DOI: 10.1038/s41467-024-55157-2. [Online]. Available: https://doi.org/10.1038/s41467-024-55157-2.

[18] S. Zhang et al., "A versatile continuum gripping robot with a concealable gripper," *Cyborg and Bionic Systems*, vol. 4, p. 0003, 2023. DOI: 10.34133/cbsystems.0003. [Online]. Available: https://doi.org/10.34133/cbsystems.0003.

[19] V. Rajendran et al., "Towards autonomous selective harvesting: A review of robot perception, robot design, motion planning and control," *Journal of Field Robotics*, vol. 41, no. 7, pp. 2247–2279, 2024. DOI: 10.1002/rob.22230. [Online]. Available: https://doi.org/10.1002/rob.22230.

[20] R. J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010. DOI: 10.1177/0278364910368147. [Online]. Available: https://doi.org/10.1177/0278364910368147.

[21] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," University of California, San Diego, Technical Report, 2004. [Online]. Available: https://mathweb.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf.

[22] M. Russo et al., "Continuum robots: An overview," *Advanced Intelligent Systems*, vol. 5, no. 5, p. 2 200 367, 2023. DOI: 10.1002/aisy.202200367. [Online]. Available: https://doi.org/10.1002/aisy.202200367.

[23] C. Marzi, C. Kunz, and F. Mathis-Ullrich, "Continuum robot actuation by a single motor per antagonistic tendon pair: Workspace and repeatability analysis," *at - Automatisierungstechnik*, vol. 71, no. 7, pp. 553–562, 2023. DOI: 10.1515/auto-2023-0066. [Online]. Available: https://doi.org/10.1515/auto-2023-0066.

[24] Robert Bosch GmbH, *CAN specification version 2.0*, Robert Bosch GmbH, Stuttgart, Germany, 1991.

[25] M. Fu et al., "Design and experiment of multi-fruit gripping and cutting linkage kiwifruit picking end-effector," *INMATEH - Agricultural Engineering*, vol. 72, no. 1, pp. 710–719, 2024. DOI: 10.35633/inmateh-72-63. [Online]. Available: https://doi.org/10.35633/inmateh-72-63.

[26] J. Amador and A. Carrasco, "Soft gripper end effector prototype for fruit handling," in *ACM International Conference Proceeding Series*, 2023, pp. 94–99. DOI: 10.1145/3655532.3655547. [Online]. Available: https://doi.org/10.1145/3655532.3655547.

[27] K. Zhang, K. Lammers, P. Chu, Z. Li, and R. Lu, "An automated apple harvesting robot—from system design to field evaluation," *Journal of Field Robotics*, vol. 41, no. 7, pp. 2384–2400, 2024. DOI: 10.1002/rob.22268. [Online]. Available: https://doi.org/10.1002/rob.22268.

[28] J. Rong, P. Wang, Q. Yang, and F. Huang, "A selective harvesting robot for cherry tomatoes: Design, development, field evaluation analysis," *Journal of Field Robotics*, vol. 41, no. 7, pp. 2564–2582, 2024. DOI: 10.1002/rob.22377. [Online]. Available: https://doi.org/10.1002/rob.22377.

[29] B. Arad et al., "Development of a sweet pepper harvesting robot," *Journal of Field Robotics*, vol. 37, no. 6, pp. 1027–1039, 2020. DOI: 10.1002/rob.21937. [Online]. Available: https://doi.org/10.1002/rob.21937.

[30] A. Roshanianfard, N. Noguchi, S. Ardabili, C. Mako, and A. Mosavi, "Autonomous robotic system for pumpkin harvesting," *Agronomy*, vol. 12, no. 7, p. 1594, 2022. DOI: 10.3390/agronomy12071594. [Online]. Available: https://doi.org/10.3390/agronomy12071594.

[31] Q. Pan, D. Wang, J. Lian, Y. Dong, and C. Qiu, "Development of an automatic sweet pepper harvesting robot and experimental evaluation," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 811–15 817. DOI: 10.1109/ICRA57147.2024.10610866. [Online]. Available: https://doi.org/10.1109/ICRA57147.2024.10610866.

[32] T. Yoshida, Y. Onishi, T. Kawahara, and T. Fukao, "Automated harvesting by a dual-arm fruit harvesting robot," *ROBOMECH Journal*, vol. 9, no. 1, p. 19, 2022. DOI: 10.1186/s40648-022-00233-9. [Online]. Available: https://doi.org/10.1186/s40648-022-00233-9.

[33] Z. Liu, J. Zhou, and K. Karydis, "Hierarchical tri-manual planning for vision-assisted fruit harvesting with quadrupedal robots," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 7153–7159. DOI: 10.1109/ICRA55743.2025.11127561. [Online]. Available: https://doi.org/10.1109/ICRA55743.2025.11127561.

[34] C. W. Bac, T. Roorda, R. Reshef, S. Berman, J. Hemming, and E. J. van Henten, "Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment," *Biosystems Engineering*, vol. 146, pp. 85–97, 2016. DOI: 10.1016/j.biosystemseng.2015.07.004. [Online]. Available: https://doi.org/10.1016/j.biosystemseng.2015.07.004.

[35] Y. Xiong, Y. Ge, and P. From, "An obstacle separation method for robotic picking of fruits in clusters," *Computers and Electronics in Agriculture*, vol. 175, p. 105 397, 2020. DOI: 10.1016/j.compag.2020.105397. [Online]. Available: https://doi.org/10.1016/j.compag.2020.105397.

[36] S. Mghames, M. Hanheide, and A. Ghalamzan, "Interactive movement primitives: Planning to push occluding pieces for fruit picking," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2616–2623. DOI: 10.1109/IROS45743.2020.9341728. [Online]. Available: https://doi.org/10.1109/IROS45743.2020.9341728.

[37] V. Pachouri and P. M. Pathak, "Design and modeling of a planar-to-spatial tendon-driven continuum manipulator subjected to uncertain forces," *Robotics and Autonomous Systems*, vol. 170, p. 104 551, 2023. DOI: https://doi.org/10.1016/j.robot.2023.104551. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889023001902.

[38] M. P. Huertas Niño, M. Boutayeb, and D. Martinez, "A hybrid tendon-driven continuum robot that avoids torsion under external load," *Frontiers in Robotics and AI*, vol. 12, p. 1 576 209, 2025. DOI: 10.3389/frobt.2025.1576209. [Online]. Available: https://doi.org/10.3389/frobt.2025.1576209.

[39] J. Davidson and C. Mo, *Conceptual design of an end-effector for an apple harvesting robot*, https://www.researchgate.net/publication/288840721, Available on ResearchGate, 2014. [Online]. Available: https://www.researchgate.net/publication/288840721.

[40] K. Oguma, S. Higuchi, F. Ikeda, Y. Murakami, and S. Toyama, "Development of automatic tomato harvesting system using universal vacuum gripper and rgb-d camera," in *15th International Conference on Motion and Vibration Control (MoViC 2020)*, 2020. DOI: 10.1299/jsmeintmovic.2020.15.10050. [Online]. Available: https://doi.org/10.1299/jsmeintmovic.2020.15.10050.

[41] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of Robotic Systems*, vol. 20, no. 2, pp. 45–63, 2003. DOI: 10.1002/rob.10070.

[42] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43–55, 2006. DOI: 10.1109/TRO.2005.861458. [Online]. Available: https://doi.org/10.1109/TRO.2005.861458.