

MATHEMATICAL MODELING, DESIGN VALIDATION, AND
SIMULATION OF MORPHING MECHANISM FOR WINGLET
APPLICATIONS

by

Upasana Choudhuri

Bachelor of Engineering, Ryerson University, 2016

A thesis

presented to Ryerson University

in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the program of
Aerospace Engineering

Toronto, Ontario, Canada 2019

©Upasana Choudhuri 2019

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

MATHEMATICAL MODELING, DESIGN VALIDATION AND SIMULATION OF
MORPHING
MECHANISM FOR WINGLET APPLICATIONS

Upasana Choudhuri

Master of Applied Science, Aerospace Engineering, Ryerson University, Toronto (2019)

ABSTRACT

Presented within this thesis is the preliminary design phases for the development of a morphing winglet mechanism. The mathematical models and analyses conducted within this thesis provide the means for bringing the design concept stage to the testing and validation phases. The kinematic modeling of a proposed design is developed. The inverse kinematics of the system are used to determine the required inputs to meet the range of motion. The velocity models for the system are established for both the forward and inverse cases. The inverse velocity models are used to establish a synchronous behaviour between the two serial linkages. Thus, allowing system operation as a redundantly actuated parallel mechanism. The results of implementation are evaluated for the initial and optimized designs. A proposed velocity profile is developed to facilitate control and desired operation of the system. This is then validated by the testing of the system response and error.

ACKNOWLEDGEMENTS

I would like to express my gratitude to the staff and professors in the Aerospace Department at Ryerson University for the opportunity and knowledge to progress as an aerospace engineer. I sincerely thank my supervisor Dr. Fengfeng Jeff Xi, for giving me the opportunity to be part of this project, and providing the knowledge, guidance and support to complete my research.

I would like to express my deep gratitude to my colleague Gabriel Campos, for his help and advice during my thesis. As well as for his great work and cooperation on this project, with the design of the controller and collaboration in prototype testing.

Finally, I would like to sincerely thank my family and close friends for the encouragement, tolerance, and support during my studies.

TABLE OF CONTENTS

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A THESIS	ii
ABSTRACT.....	iii
ACKNOWLEDGEMENTS.....	iv
List of Tables	ix
List of Figures.....	x
Nomenclature.....	xiv
CHAPTER 1 Introduction.....	1
1.1 Motivation	1
1.1.1 Effect of Winglets on Aerodynamics.....	5
1.1.2 Types of Winglets Currently Used	5
1.2 Project Overview	7
1.3 Thesis Outline.....	9
CHAPTER 2 Literature Review	10
2.1 Morphing Winglets.....	10
2.2 Morphing Winglet Designs	10
2.2.1 Bourdin Variable Cant Angle Winglet	13
2.2.2 Falcao Morphing Wingtip.....	16
2.2.3 Kim Bi-stable Winglet.....	17
2.3 Mechanism Analysis.....	18

2.3.1	Mobility Analysis	18
2.3.2	Position and Orientation	19
2.3.3	Rotation and Translation	19
2.3.4	Kinematic Analysis	21
2.4	Kinematic Analysis of Spatial Linkages in Literature.....	22
2.4.1	Analysis of Spatial 6-bar Mechanism	22
2.4.2	Kinematic solution of a Stephenson-III mechanism.....	24
2.4.3	Kinematic Solution for Spatial Parallel Linkage	26
2.4.4	Kinematic Analysis of a 6R Overconstrained Spatial Mechanism.....	26
2.4.5	Analysis and Simulation of Kinematic Error in a Spatial Linkage.....	27
2.5	Redundantly Actuated Systems	28
2.5.1	Analysis of Redundantly Actuated Systems	29
CHAPTER 3 Kinematic Modeling		30
3.1	Coordinate Set-up	30
3.2	Geometry and Model Definition	31
3.3	Mobility Analysis	32
3.4	Detailed Model Definition.....	36
3.5	Forward Kinematics Analysis	41
3.6	Workspace Definition.....	45
3.7	Inverse Kinematics.....	47

3.8	Stroke Length Determination for Target Workspace	50
CHAPTER 4 Velocity Analysis		52
4.1	Fwd linkage Forward Velocity.....	52
4.2	Aft linkage Forward Velocity.....	53
4.3	Fwd linkage Inverse Velocity.....	54
4.4	Aft Linkage Inverse Velocity	56
4.5	Velocity Synchronization	57
CHAPTER 5 Implementation.....		59
5.1	Simulation.....	59
5.2	Control system set-up.....	59
5.3	Design Prototyping.....	61
5.4	Implementation Issues.....	61
5.4.1	Error Evaluation	64
5.5	Optimized Design	67
5.5.1	Velocity Tuning for Control.....	72
5.5.2	Model Set-up	76
5.5.3	Error Analysis.....	79
CHAPTER 6 Conclusion		86
6.1	Contributions	86
6.2	Future Work.....	87

Appendix A.....	88
Forward Kinematics.....	88
Inverse Kinematics.....	95
Inverse Kinematics Aft Linkage.....	98
Velocity Fwd Linkage	101
Velocity Aft Linkage	105
Velocity Tuning	108
Goal Attainment	108
Velocity Function used for Tuning.....	109
Appendix B.....	116
Appendix C.....	117
Appendix D	118
Non-computational analytic solution for FK.....	118
Non-computational Analytic Solution Error	119
References.....	120

List of Tables

Table 2-1: Forward Kinematics Results for Stephenson III 6-bar [14]	25
Table 2-2: Inverse Kinematics of Stephenson III 6-bar [14].....	25
Table 5-1: Goal Attainment Initializing Parameters	73
Table 5-2: Delay Results for various initial inputs	74

List of Figures

Figure 1-1: Sketch of airfoil in relation to wing geometry [1]	1
Figure 1-2: Airfoil geometry and nomenclator [1]	1
Figure 1-3: Airfoil with aerodynamic forces indicated [1].....	2
Figure 1-4: Relation between aerodynamic forces and normal and axial components	2
Figure 1-5: Airflow vortices in finite wings[1]	3
Figure 1-6: Effect of tip vortices on airflow [2]	4
Figure 1-7: Airbus A310 wingtip fence [5].....	5
Figure 1-8: 737-200Adv. Mini-winglets [6].....	6
Figure 1-9: Boeing advanced technology winglet [6]	7
Figure 1-10: AT winglet lift profile comparison [6].....	7
Figure 1-11: Standard flight profile with optimal winglet cant angles [7]	8
Figure 2-1: NACA 653218[3]	11
Figure 2-2: Prototype wing and winglet[3]	11
Figure 2-3: Lift coefficient with varying cant angles [3]	12
Figure 2-4: Drag coefficient with varying cant angles [3].....	12
Figure 2-5: L/D ratio for varying cant angles[3]	12
Figure 2-6: Bourdin morphing winglet [8].....	13
Figure 2-7: Bourdin wing prototype with winglets at 0 and 90 degrees respectively [8]	14
Figure 2-8: Effect of winglet deflection on aerodynamic c.g. in Bourdin design [8]	15
Figure 2-9: Bourdin wing, winglet cant angle and roll rate [8]	15
Figure 2-10: Falcao morphing wingtip prototype [9].....	16
Figure 2-11: Falcao wingtip morphing mechanism [9].....	16

Figure 2-12: Two- roll bending device [10]	17
Figure 2-13: Bistable winglet. a) 0 degree cant, b) 60 degree cant [10]	17
Figure 2-14: Figure representing transformation of single body[11].....	20
Figure 2-15: Multibody system with global and body fixed frames [11].....	20
Figure 2-16: Spatial 6-bar mechanism: a) planar slider crank part A, b) planar slider crank part B	23
Figure 2-17: Mechanism Coordinate Frames[14]	24
Figure 2-18: Coordinate frames for 6R mechanism analysed by Kong [16]	26
Figure 3-1: Global reference frame setup.....	31
Figure 3-2: Model Geometry.....	32
Figure 3-3: fwd Linkage joint motions	33
Figure 3-4: Link 3,4, and 5 forming bell crank link	34
Figure 3-5: Bell crank links aft linkage	35
Figure 3-6: Fwd linkage vector setup.....	37
Figure 3-7: Fwd linkage vector loop definition - (a) Loop 1 (planar loop), (b) Loop 2 (spatial loop)	38
Figure 3-8: Aft linkage vector setup.....	39
Figure 3-9: Aft linkage vector loop definition – (a) Loop 1 (planar loop), (b) Loop 2 (spatial loop)	40
Figure 3-10: Preliminary Workspace (a) XZ-plane, (b) XY-plane.....	46
Figure 3-11: Relation between Winglet interface angle and linear actuator stroke length	51
Figure 4-1: Fwd linkage actuator speed vs. Input angle	58
Figure 4-2: Aft Linkage Actuator speed vs. Input angle	58
Figure 5-1: Final Controller design.....	60
Figure 5-2: Prototype Image.....	61
Figure 5-3: General controller design algorithm	62
Figure 5-4: Controller input.....	63

Figure 5-5: Actuator Lengths vs. Time	63
Figure 5-6: Reference plane for calibration.....	65
Figure 5-7: Points used to generate rotation axis	65
Figure 5-8: Rotation arc definition	65
Figure 5-9: System error for decreasing winglet cant angles	66
Figure 5-10: System error for increase cant angles.....	66
Figure 5-11: Optimized design simulation at 80° configuration	68
Figure 5-12: Optimized design simulation at 10° configuration	68
Figure 5-13: IK results for optimized design	70
Figure 5-14: Stroke length vs. Target angle for inverse velocity results.....	70
Figure 5-15: Fwd linkage actuator speed vs. current winglet angle	71
Figure 5-16: Aft linkage actuator speed vs. current winglet angle	71
Figure 5-17: Stroke length vs. winglet angle output for $\omega = 5^\circ/s$	73
Figure 5-18: Fwd linkage Actuator speed vs Time with delay of 2.8 seconds	74
Figure 5-19: Aft linkage Actuator speed vs Time with delay of 2.8 seconds	75
Figure 5-20: Stroke length vs. Input angle with delay	75
Figure 5-21: Prototype test setup Top view.....	76
Figure 5-22: Prototype test setup with Winglet at 10°	77
Figure 5-23: Prototype test setup with Winglet at 80°	77
Figure 5-24: Polynomial fit used for estimation of IK relationship for input to controller	78
Figure 5-25: Quadratic fit used for Inverse Velocity relation estimation for input to the controller..	78
Figure 5-26: Final controller design	79
Figure 5-27: Actuator speed vs. winglet angle without ramp up	80
Figure 5-28: Stroke length vs. Winglet angle without ramp up.....	80

Figure 5-29: System input for case with ramp up to target angular rate	81
Figure 5-30: System response to ramped case	81
Figure 5-31: Resultant angular velocity of winglet for ramped case.....	82
Figure 5-32: Stroke length vs. Winglet angle ramped case.....	82
Figure 5-33: System input with Winglet load.....	84
Figure 5-34: System response with Winglet load	84
Figure 5-35: Output Winglet angular velocity with Winglet load	85
Figure 5-36: Stroke length vs. Winglet angle for loaded case.....	85

Nomenclature

Symbol	Definition
x,y,z	Axis coordinate
P	Spatial point
r	Spatial vector
s_i	Linear velocity vector
R	Rotational matrix
Θ, ϕ	Rotation angles
u	Unit vector
α	Angle of attack
η	Normalized spanwise location of wing
a	Speed of sound in air
AR	Aspect Ratio
AoA	Angle of Attack
b	Wingspan
Cant angle	Angle about axis perpendicular to wing span axis (See Appendix B)
CD	Coefficient of drag
CL	Coefficient of lift
CN	Normal force coefficient
C_p	Pressure coefficient
CT	Specific fuel consumption
CD_i	Induced drag
CD_o	Zero lift drag

CL_{max}	Maximum lift coefficient
e	Span efficiency factor
Mach	Mach number
Re	Reynolds number
S	Wing platform area
Toe angle	Angle about wing span axis (See Appendix B)
UAV	Unmanned Aerial Vehicle

CHAPTER 1 Introduction

1.1 Motivation

The airfoil of a wing is defined as indicated in Figure 1-1, the shape found from the intersection of the wing and the plane perpendicular. Figure 1-2 outlines the airfoil geometry and nomenclature.

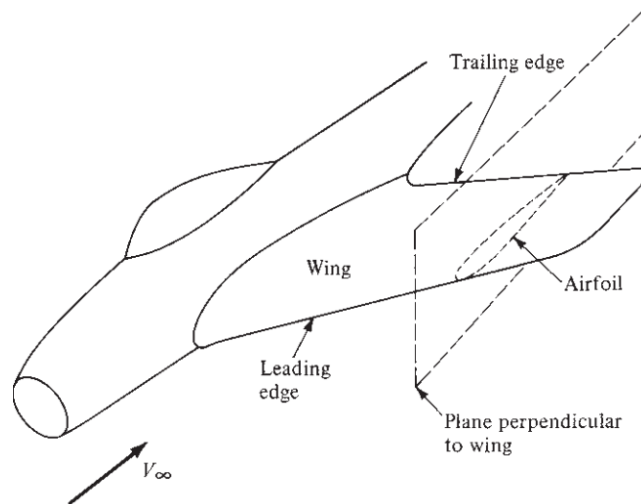


Figure 1-1: Sketch of airfoil in relation to wing geometry [1]

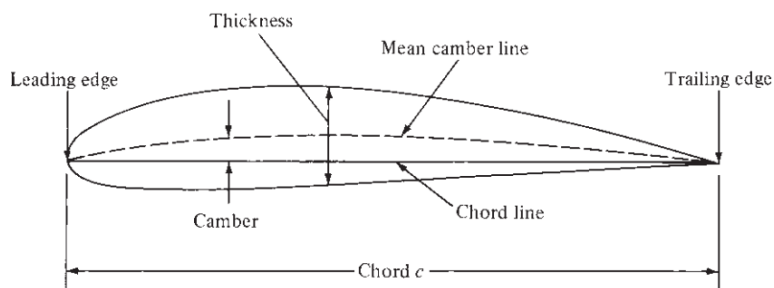


Figure 1-2: Airfoil geometry and nomenclature [1]

The aerodynamic forces are taken to act at the quarter chord point, shown in Figure 1-3. The normal and axial forces are the components of the Lift and Drag forces that act normal/axial to the airfoil. The angle between the relative wind and chord line is the angle of attack [1]. The relation between the aerodynamic forces and normal and axial forces is indicated in the equations below [1].

$$L = N \cos \alpha - A \sin \alpha \quad (1-1)$$

$$D = N \sin \alpha + A \cos \alpha \quad (1-2)$$

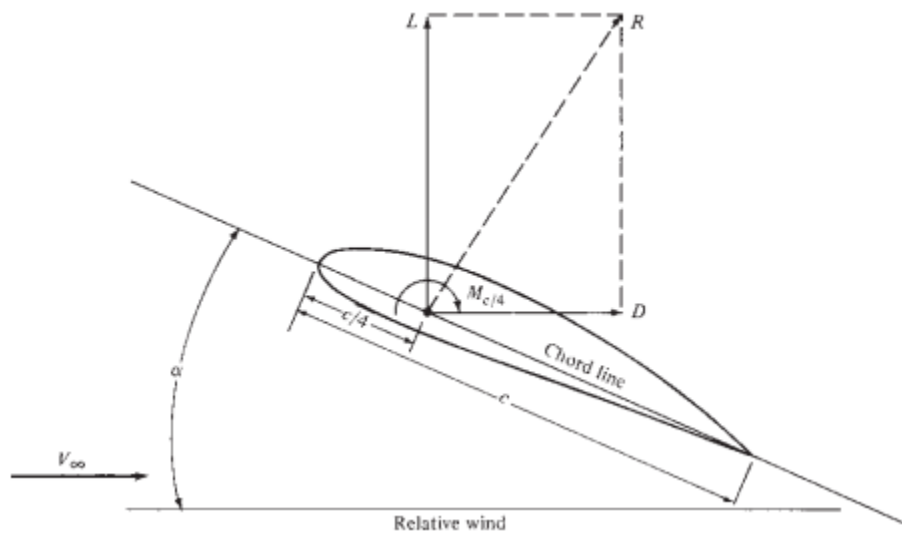


Figure 1-3: Airfoil with aerodynamic forces indicated [1]

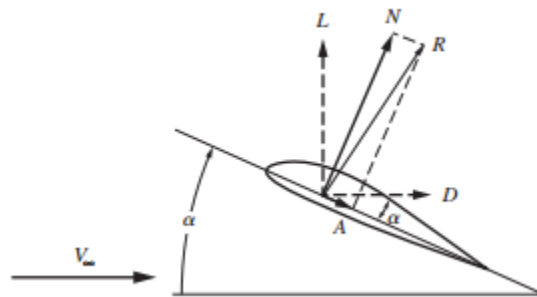


Figure 1-4: Relation between aerodynamic forces and normal and axial components

The major factors influencing Lift, Drag, and Moment acting on an airfoil are: [1]

- Angle of attack
- Flight velocity
- Wing area
- Airfoil geometry
- Viscosity coefficient
- Compressibility of the airflow

An important aspect to consider is that the winglet resembles a finite wing segment, thus the forces will be three-dimensional and differ from the standard infinite solutions for the lift/drag etc. coefficients. Figure 1-5 shows the primary difference between finite and infinite wings, the generation of wing-tip vortices due to shear created at the trailing edge causing rolling of the airflow [1].

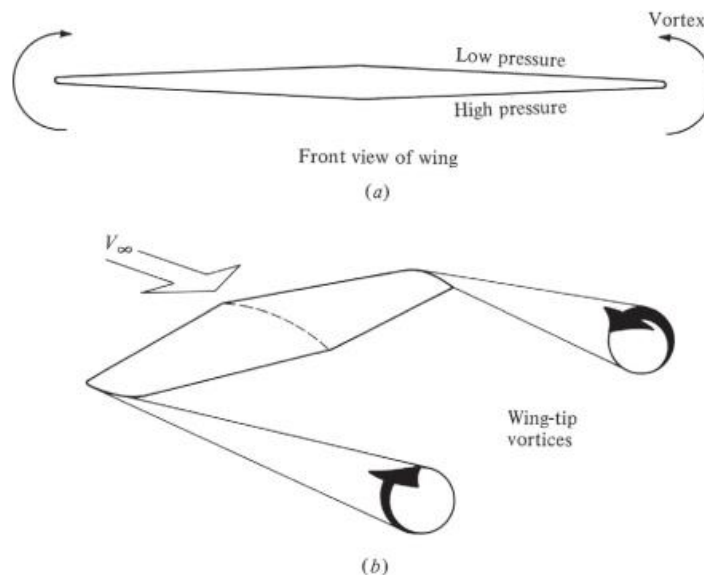


Figure 1-5: Airflow vortices in finite wings[1]

The vortices contribute to downwash, a downward velocity component. The consequences of downwash are:[1]

- The effective angle of attack is reduced
- The drag force is increased, referenced as induced drag

This origin downwash on the flow section near the wingtip is shown in Figure 1-6. The induced drag, and effective angle of attack can be calculated as indicated in Equation (1-3), and Equation (1-4) below [1].

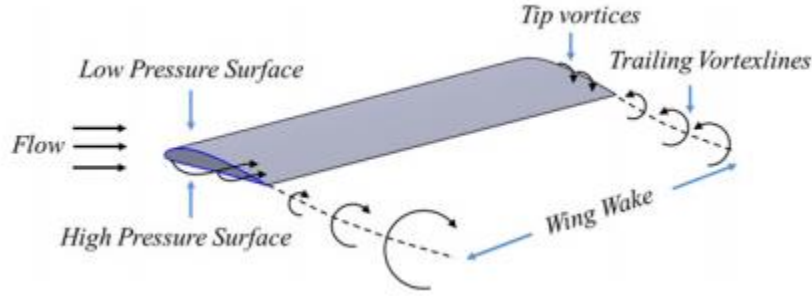


Figure 1-6: Effect of tip vortices on airflow [2]

$$D_i = L \sin \alpha_i \quad (1-3)$$

$$\alpha_i = \frac{57.3 C_L}{\pi e_1 AR} \quad (1-4)$$

Here the span effectiveness factor, e_1 , is 1 for an elliptical lift profile, and AR is the aspect ratio. As the effective angle of attack is reduced, so is the lift curve for the airfoil. The new slope of the lift curve for a finite airfoil can be defined in terms of the infinite lift slope as shown in Equation (1-5). Note that both Equation (1-4) and (1-5) provide results in terms of degrees, for radians the 57.3 component would be removed from the calculations.

$$a = \frac{a_0}{1 + \left(\frac{57.3 a_0}{\pi e_1 AR} \right)} \quad (1-5)$$

The effects of these vortices are the main reason for the use of winglets in modern commercial aircraft.

1.1.1 Effect of Winglets on Aerodynamics

The wingtips, reduce the pressure gradient that results in the shear causing the wingtip vortices, a small addition to the wing extending vertically. This was further tested in 1974 by NASA on the Boeing KC-135A for cruise flight conditions. It was found that a near vertical winglet provided optimal drag reduction and efficiency under cruise flight conditions[3].

1.1.2 Types of Winglets Currently Used

Winglets currently installed on commercial aircraft all fit the near vertical description and provide optimal wing efficiency improvement at cruise flight conditions. These include the initial winglets used in commercial flight. These designs are pictured in the figures below. The airbus A310 incorporated the ‘wingtip fence’ which included both upward and downward components as is shown in Figure 1-7 [4].



Figure 1-7: Airbus A310 wingtip fence [5]

Boeing also incorporated winglets into their aircraft. The figure below shows mini-winglets attached to the 737-200Adv. These resulted in a range increase of 3%, with 3% fuel savings[6].



Figure 1-8: 737-200Adv. Mini-winglets [6]

Currently Boeing is installing the ‘Advanced Technology’ winglets on the 737 MAX family. These winglets combine rake tips and dual feather winglets, resulting in the winglet design shown in Figure 1-9. These winglets are claimed to provide 1.5% greater fuel efficiency when compared to current winglet designs. The claim is that these winglets balance the effective increase in wingspan between the upper and lower wing, generating greater lift and higher drag reduction. The diagram in Figure 1-10 shows this theorized lift distribution compared with the ‘no winglet’ and ‘traditional winglet’ cases [6].



Figure 1-9: Boeing advanced technology winglet [6]

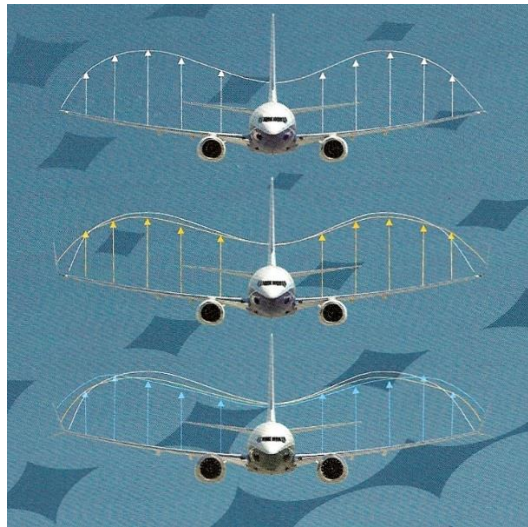


Figure 1-10: AT winglet lift profile comparison [6]

1.2 Project Overview

A morphing winglet design has been proposed by the corresponding research team at Ryerson. This winglet has been designed to meet the requirements outlined by Bombardier Aerospace. The main kinematic and velocity requirements of the system being, a 10° to 80° angular range of the *cant* angle and a $1^\circ/\text{s}$ angular velocity. The design kinematics, and velocity capabilities must be established and analysed in detail to ascertain whether these requirements can be met. Furthermore, the system kinematic and dynamic relations are required for the design and implementation of a controller and

test prototype. The proposed design consists of two serial linkages operating synchronously, forming a redundantly actuated parallel mechanism. The second linkage is present to assist with the load bearing requirements of the system, particularly in-flight. The proposed design must meet strict operating conditions while fitting highly constrained sizing requirements. These operational requirements have been generated according to the optimal winglet configurations during a standard flight profile. This is shown in Figure 1-11 [7]. As a result, the system model must be clearly defined so as to prevent clashing of the two linkages, while maintaining synchronous operation. Due to the spatial nature of the mechanism the analysis of its kinematics and dynamics is complicated. A major complexity in the system design lies in the mechanism output motion occurring in an alternate plane than the input motion. The input motion occurs in the XY plane as defined for this system in later sections, while the output motion takes place in the XZ plane. This thesis outlines the numerical formulation of the system, the optimization and adjustment of the system motion requirements for control, and the testing of these formulations on a scaled prototype.

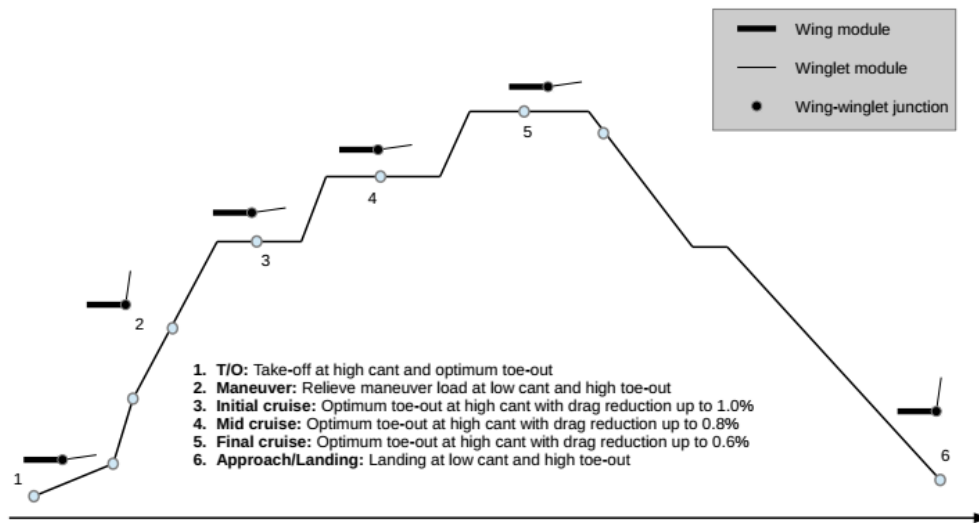


Figure 1-11: Standard flight profile with optimal winglet cant angles [7]

1.3 Thesis Outline

The literature review of current morphing winglets, spatial linkages, and redundantly actuated systems is provided in Chapter 2.

Chapter 3 outlines the kinematic modeling of the system. The system motion and capability is analysed and a workspace plot is developed. The inverse kinematics of the mechanism are derived and used to establish system stroke requirements for target motion.

Chapter 4 covers the velocity modeling of the system. The forward and inverse velocity relations are developed, and the synchronization requirements of the system are determined.

Chapter 5 discusses the implementation of the mathematical models develop in this thesis. The prototyping and testing of the system is presented in this chapter as a comparison between the theoretical system model and the test prototype. This chapter also discusses the simulation of an optimized design, and the comparison of the theoretical system response with the experimental. Further, an optimized velocity profile is developed in this chapter for facilitation of more accurate control.

Chapter 6 provides the concluding statements and discussion of future work.

CHAPTER 2 Literature Review

2.1 Morphing Winglets

The concept of a morphing winglet has arisen in aircraft design, due to the benefits of variable cant angle on performance. A variety of morphing winglet designs have been proposed, some of which are discussed below. These design concepts use a variety of methods to achieve morphing winglets, such as a bi-stable set-up, or servo actuated hinges. These designs all include a single actuation mechanism with concepts for relatively small-scale winglets. Furthermore, while these concepts present a preliminary kinematic analysis, velocity and dynamic analysis, as well as velocity control, are yet to be determined. Section 2.2 below consists of an overview of the effects of various cant angle configurations on aircraft performance, and a summary of the current morphing winglet design concepts.

2.2 Morphing Winglet Designs

Beechock and Wang [3] studied the effects of variable cant angle on aircraft performance. The airfoil studied was the NACA 653218 series, shown in Figure 2-1. Four winglet configurations were studied, cant angle at 0, 30, 45, and 60 degrees. The prototype wing and winglets used for analysis are shown in Figure 2-2. Testing indicated that winglets at cant angle of 45 degrees resulted in the highest lift, as summarized in Figure 2-3. Tests also indicated an increase in drag coefficient with an increase in cant angle, shown in Figure 2-4. The highest L/D ratio was found to be at 45 degrees cant, this relation is summarized in Figure 2-5. Furthermore, Beechock and Wang indicate that large vortices were formed at high cant angles (45 or 60 degrees) and high AoAs. At low AoA (as would occur during cruise) the high cant angles produced minimal drag and vortices [3].

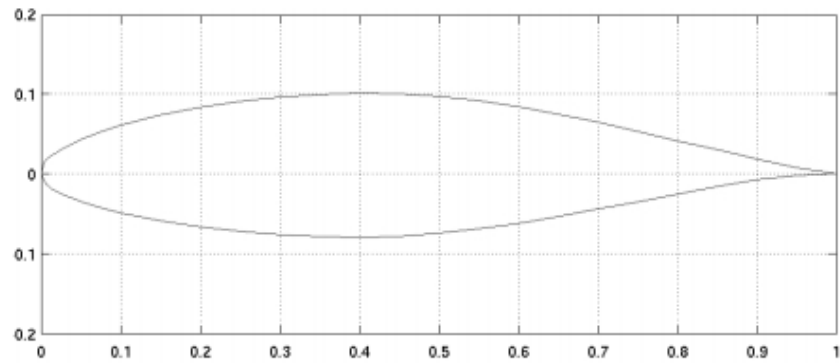


Figure 2-1: NACA 653218[3]



Figure 2-2: Prototype wing and winglet[3]

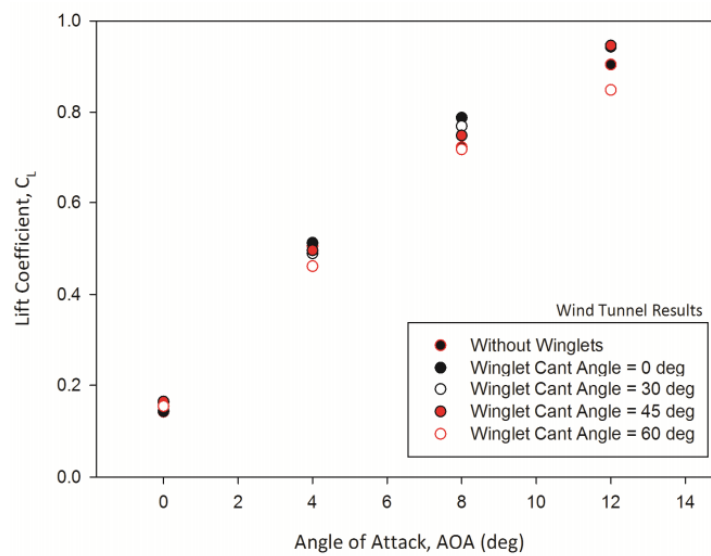


Figure 2-3: Lift coefficient with varying cant angles [3]

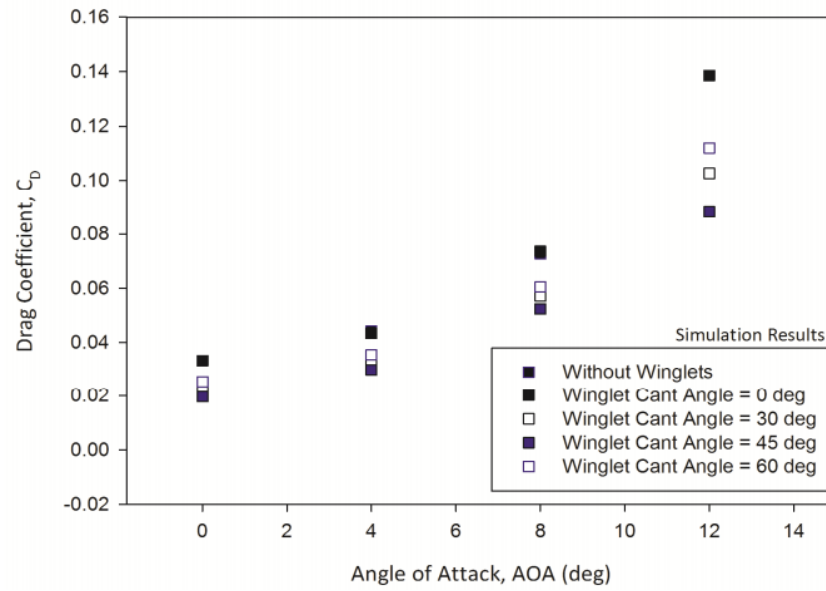


Figure 2-4: Drag coefficient with varying cant angles [3]

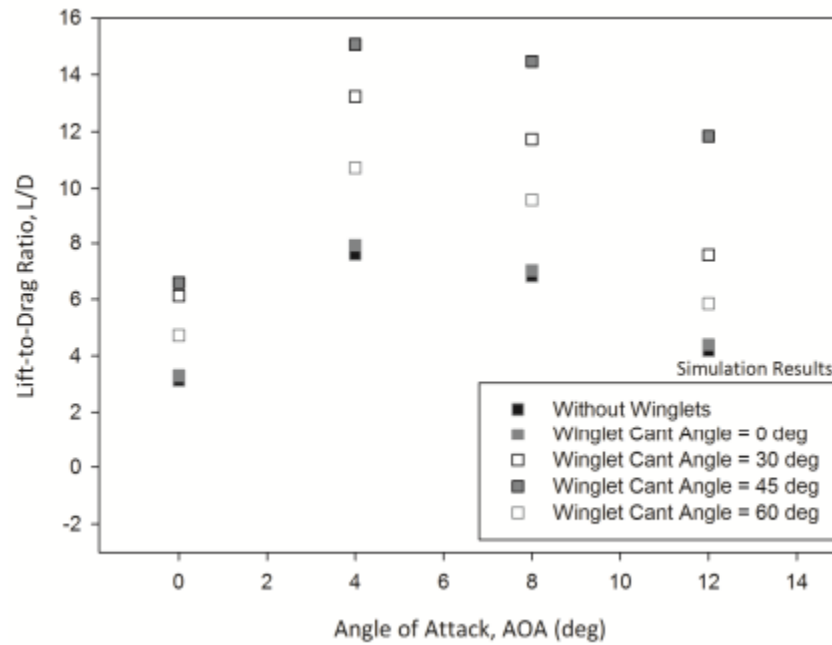


Figure 2-5: L/D ratio for varying cant angles[3]

2.2.1 Bourdin Variable Cant Angle Winglet

Bourdin et al. [8] have proposed a design for morphing winglets with a cant angle range of -90 to 90 degrees. The wing design proposed consisted of a planar wing with a 30° leading edge sweep, aspect ratio (AR) of 4.6 and taper ratio of 0.56. The wingtips were modified with the addition of servo articulated hinges mounted within the wing profile. These servo driven hinges provide the actuation of the winglets from -90 degrees to 90 degrees [8].

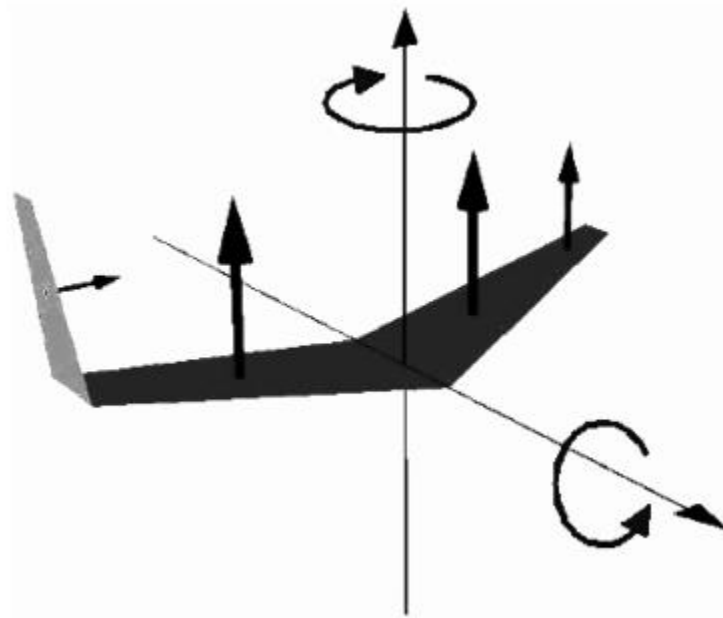


Figure 2-6: Bourdin morphing winglet [8]

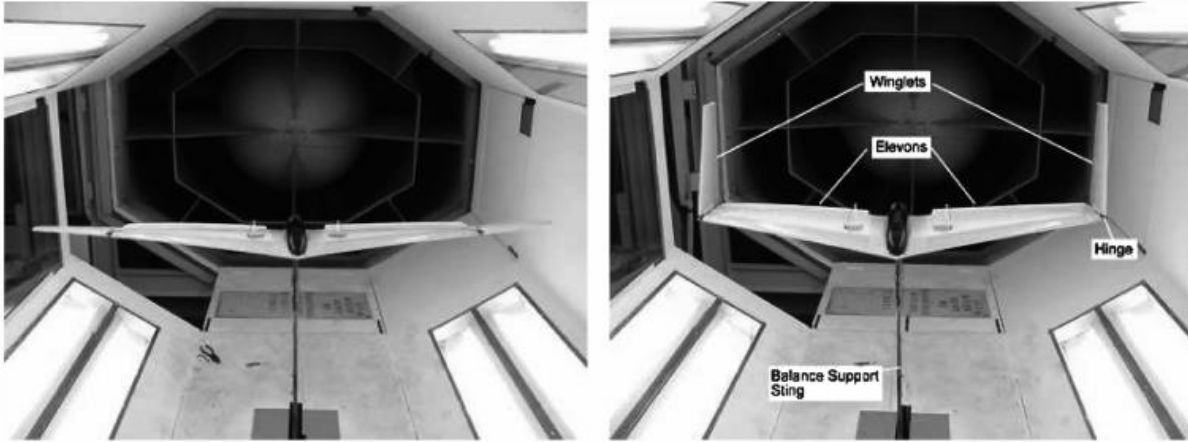


Figure 2-7: Bourdin wing prototype with winglets at 0 and 90 degrees respectively [8]

The Bourdin design was tested to determine the effect of winglet deflection on aerodynamic center of gravity (c.g.), the results are summarized in Figure 2-8. As can be seen larger winglet deflections result in the aerodynamic c.g. moving further ahead in the chordwise direction. This displacement of the aerodynamic c.g. provided potential for control of the aircraft longitudinally. Additionally, the effects of deflecting only one winglet on the lateral control and balance were analysed, it was found that the winglet cant angle is directly proportionate to the roll rate, increasing the roll rate as the cant angle was increased. This is shown in Figure 2-9.

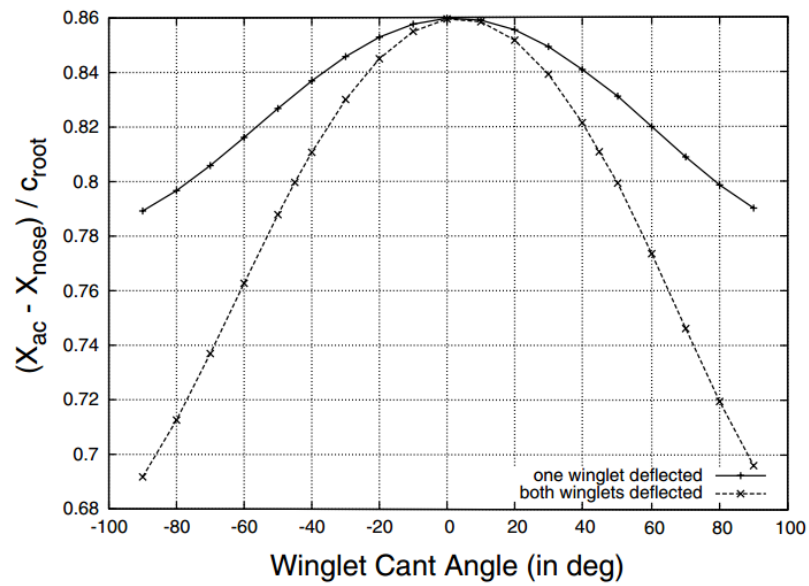


Figure 2-8: Effect of winglet deflection on aerodynamic c.g. in Bourdin design [8]

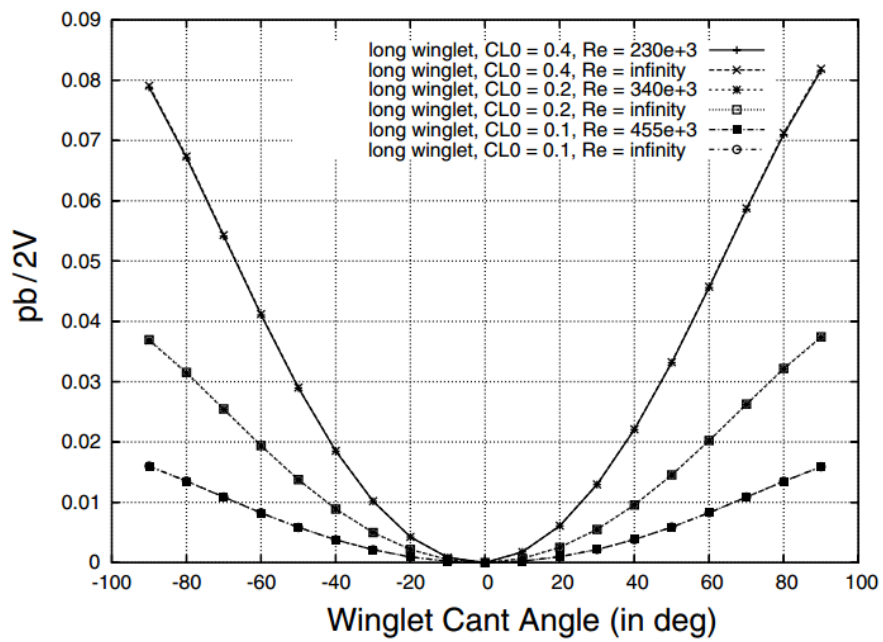


Figure 2-9: Bourdin wing, winglet cant angle and roll rate [8]

2.2.2 Falcao Morphing Wingtip

Falcao et al. [9] designed a morphing wingtip with servo-actuated control of the cant and toe angles, pictured in Figure 2-10. This design was analysed for use with a multi-mission UAV. Tests indicated that the morphing wingtip improved aircraft performance by 25%, a 4% drag reduction, and takeoff ground roll reduction of 20%. [9]



Figure 2-10: Falcao morphing wingtip prototype [9]

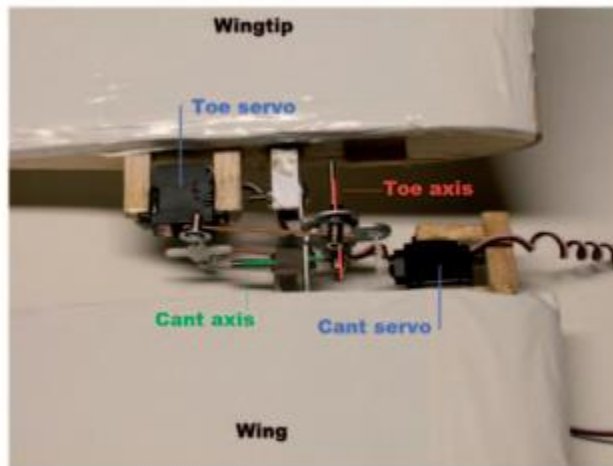


Figure 2-11: Falcao wingtip morphing mechanism [9]

2.2.3 Kim Bi-stable Winglet

Kim et al. [10] proposed a design for a bistable structure for a morphing winglet in a UAV. The design of a two-roll bending device was proposed to achieve the residual stress in the metal sheet required to accomplish the bistability of winglet (Figure 2-12). Using a bistable metallic structure, polymer cover, and two-roll bending device, Kim et al. produced a prototype bistable morphing winglet able to provide a cant angle of 0 degrees and 60 degrees (Figure 2-13) [10].

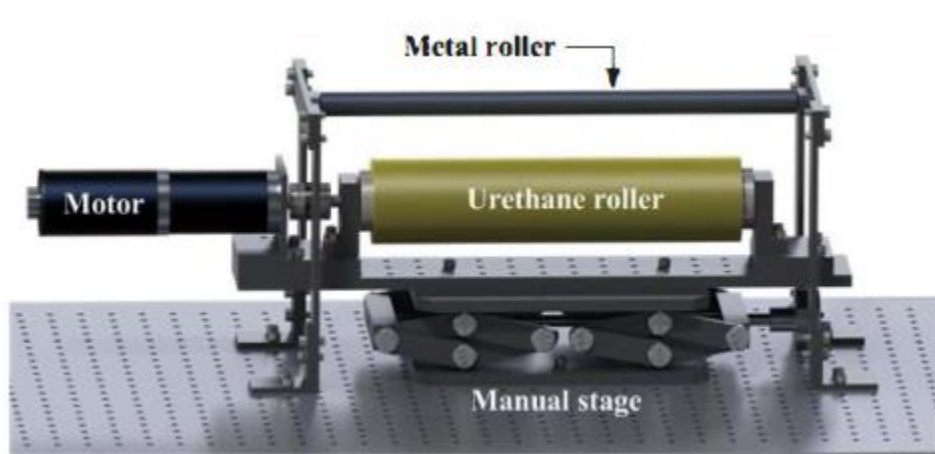


Figure 2-12: Two-roll bending device [10]

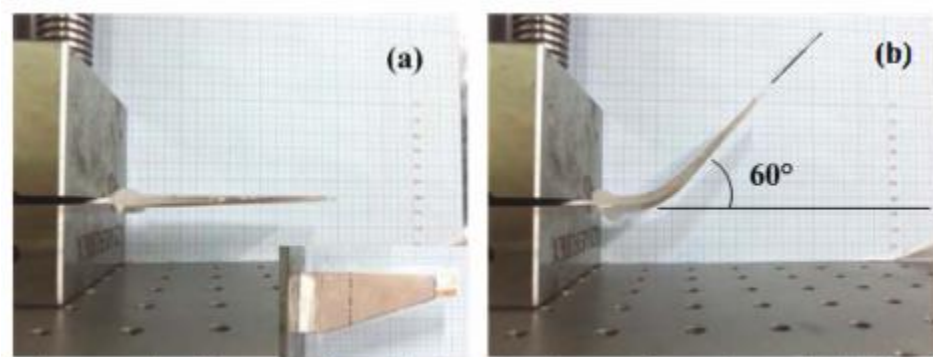


Figure 2-13: Bistable winglet. a) 0 degree cant, b) 60 degree cant [10]

2.3 Mechanism Analysis

General mechanism analysis is separated into the kinematics and dynamics. The system kinematics can be represented by the forward and inverse kinematics. Forward kinematics involve the determination of the relationship between the input, and output position and orientation.

2.3.1 Mobility Analysis

The degrees of freedom for a mechanism are defined by the mobility. Mobility analysis determines the number of independent variables required to define the linkage positions on a particular reference frame[11]. The mobility of a system can be defined by the Chebychev-Grubler-Kutzbach criterion, this is presented in the equation below [11]

$$M = \lambda(N_l - N_j - 1) + \sum_{i=1}^{N_j} f_i \quad (2-1)$$

In the above equation, λ is the order of the system. This represents the default degrees of freedom (DOF) of the system, 3 for planar mechanisms, and 6 for spatial linkages. N_l and N_j are the number of links and joints respectively. Finally, the joint DOF is represented by f_i .

Using the Chebychev mobility equation, the movability and redundancy of the system can be identified. As would be expected, if the mobility, M , of the mechanism is greater than zero the linkage is movable and can be termed a mechanism. Likewise, if M is zero or less the system is a structure and not movable[11]. The redundancy of the mechanism is found by comparing the mobility, M , with the order of the system, λ . The system is redundant if M is greater than λ , non-redundant if the two are equal, and constrained if M is less than λ [11].

It is important to note that, while the mobility criterion is necessary not sufficient. Further analysis in the design of a mechanism is required, such as the actuator locations, and linkage relative motion.

These considerations are required for the initial design stages, accounting for required tasks, and feasible configurations [11].

2.3.2 Position and Orientation

The motion of a body takes place in Euclidean space and can be described by the Cartesian frame. This frame is formed by an axis triad, using the right-hand convention. A point in space is expressed in Cartesian coordinates using its x, y, and z position. A body in space has both a position and orientation, known as the pose, and the motion of a body is represented by its translation and rotation. [11] The coordinate selection for a closed chain mechanism includes defining the Cartesian coordinates, the pose for each link, using the relative coordinates, the relative pose between links, starting at the input. Due to this, it is beneficial to define the global frame with origin at the input joint or fixed joint on the mechanism[11].

2.3.3 Rotation and Translation

The rotation of a single body about an axis can be defined using the Rotation Matrix for the corresponding axis. These rotation matrices can be used to define rotations about the x, y, or z axis in the global frame. In this manner the motion of a single body can be described by Equation (2-2) below.

$$p = Rb' + h \quad (2-2)$$

Here p represents the transformed coordinates of the body, h is the translation vector, R is the rotation matrix, and b' is the original vector.[11]

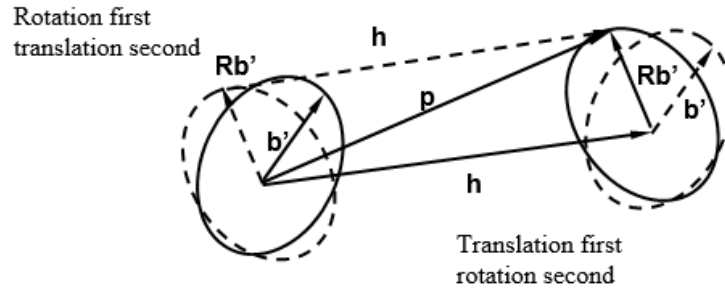


Figure 2-14: Figure representing transformation of single body[11]

$$p_i = p_{i-1} + R_{oi-1}b'_{i-1} \quad (2-3)$$

This method provides the tools to transform the body fixed coordinates, in a multibody system, to the global/fixed reference frame. The figure below shows a multibody system with global and body fixed coordinates. The vector method for expressing the positions of each joint in this system in the global frame involves Equation (2-3) shown above to find p_i for the i^{th} joint. This is done as shown in Equation (2-4) below.

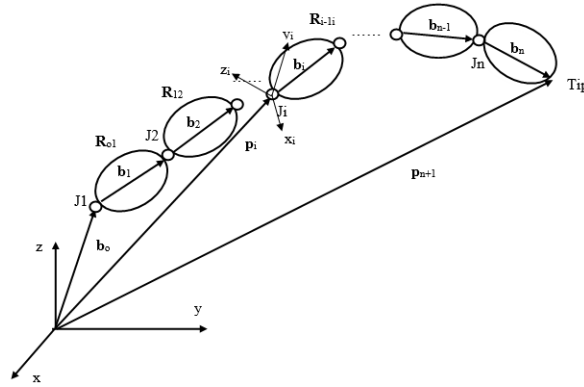


Figure 2-15: Multibody system with global and body fixed frames [11]

$$p_i = b_o + R_{o1}(b'_1 \cdots + R_{i-1i}b'_i) \quad (2-4)$$

Here R_{i-1i} is the rotation matrix from i to $i-1$.

2.3.4 Kinematic Analysis

Kinematic analysis consists of the forward/direct kinematics and the inverse. The direct kinematics involves the relationship between the joint angles and the end effector pose. The inverse kinematics likewise relates the required end effector pose to the corresponding joint angles. The sections below outline the general procedure that is used in the kinematic analysis of mechanisms.

2.3.4.1 Direct Kinematics

As stated above, the forward kinematics provide the end effector pose for given joint variable values. This is often used to develop a workspace for a kinematic chain. There are two types of kinematic chains, open and closed. Open chains possess one fixed end, while the end effector link is left free. Closed chain systems form a closed loop with the links, having two fixed ends. Closed chains have an input link and an output link rather than an end effector. Section 2.3.1 outlines the mobility analysis for a system, this defines the mechanical structure of the chain providing the general posture of the linkage [12]. The mobility is used to determine the number articulated joints, giving the number of joint variables. These joint variables are then used as the inputs for the direct kinematics. There are two main methods for solving the direct kinematics of a system: graphical analysis, and systematic procedural analysis [12]. The benefits of the graphical approach can be summarized as a simplification of the problem, though success of this approach is based on the analysis methodology. The procedural analysis involves evaluating the Denavit-Hartenberg (DH) parameters for the chain and using these parameters to establish the transformation matrix. This method is the optimal choice for long, complex, open kinematic chains. While DH analysis can be modified for analysis of closed loop chains, the geometric approach is often desirable, as it can lead to simplification of potentially complex equations.

2.3.4.2 Inverse Kinematics and Velocity Analysis

As stated earlier, the inverse kinematics provide the required joint variables for given end effector poses. This is a fundamental component to the design and control of a mechanism, as the joint motions necessary to achieve specified motions and tasks are provided via this step. The inverse kinematics are often more complex than the forward kinematics. This is due to nonlinear equations in the solutions for joint variables, the possibility for multiple – even infinite – solutions, or potentially no solutions. The mechanism workspace must be analysed beforehand to determine if the solutions are possible. Often it is the case, particularly with complex systems, that the solution is dependant upon geometric intuition of the problem solver [12].

The differential kinematics, or velocity analysis, provides the relationship between the joint velocities and the end effector velocity. This includes both the angular and linear velocities.

2.4 Kinematic Analysis of Spatial Linkages in Literature

2.4.1 Analysis of Spatial 6-bar Mechanism

Xie et al. [13] have conducted kinematic analysis on the spatial 6-bar linkage mechanism used in metallurgical equipment door operators. The paper discusses a method to simplify the analysis of a 6-bar spatial linkage by separating the mechanism into 2 planar slider-crank mechanisms for simplification of analysis. Using this method, the kinematic equations for a slider-crank mechanism were applied to determine the kinematics of this overall system. The following equations are the equations of motion for each slider-crank as determine by Xie et al. The two parts of the mechanism are indicated in the figure bellow.

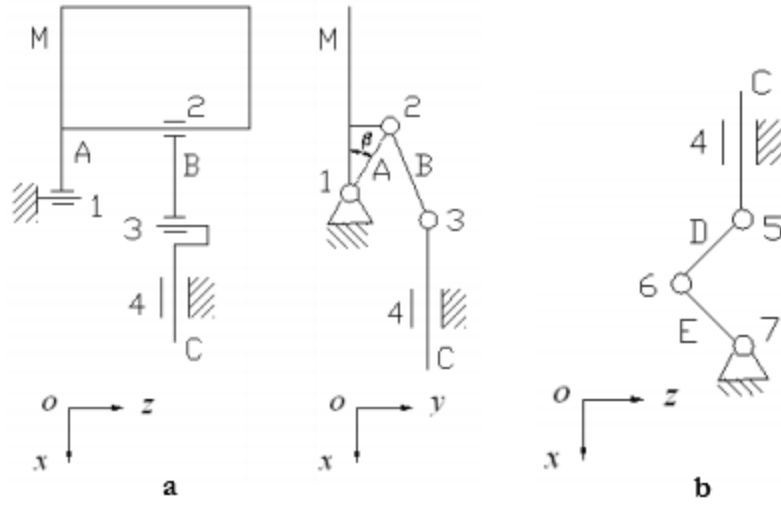


Figure 2-16: Spatial 6-bar mechanism: a) planar slider crank part A, b) planar slider crank part B

$$\sin\phi_2 = \frac{h - l_1 \sin\phi_1}{l_2} \quad (2-5)$$

$$s = l_1 \cos\phi_1 + \sqrt{l_2^2 - (h - l_1 \sin\phi_1)^2} \quad (2-6)$$

In the above equations ϕ_1 is the position angle of link A or E, ϕ_2 is the position angle of link B or D, l_1 and l_2 are the lengths of links 1 and 2 respectively, s is the displacement of the slider, and h is the offset of the slider[13].

The velocity and acceleration equations are determined via differentiation of the above equations (2-5) and (2-6). The actual solution of the system is conducted by relating the two parts, A and B, via the link C. The following equation represents this relationship:

$$s_a + s_b + l_c = H \quad (2-7)$$

here s_a and s_b are the displacements of joint 5 in part A and B respectively, l_c is the length of C, and H is the distance between joints 1 and 7 [13]. In this equation s_b is known, and thus the above

equation is used to simplify equation [11] into a linear one variable equations. The reference then utilizes MATLAB to complete the solution process and simulation [13].

2.4.2 Kinematic solution of a Stephenson-III mechanism

Lui and Yang [14] present a spherical Stephenson 6-bar mechanism for kinematic analysis. The analysis of this linkage is conducted by decomposing the mechanism into a 4-bar, and a 2-bar spherical mechanism. The figure below represents the coordinate frames of the mechanism. Using reference frame analysis the following solution is presented by Lui and Yang. [14]

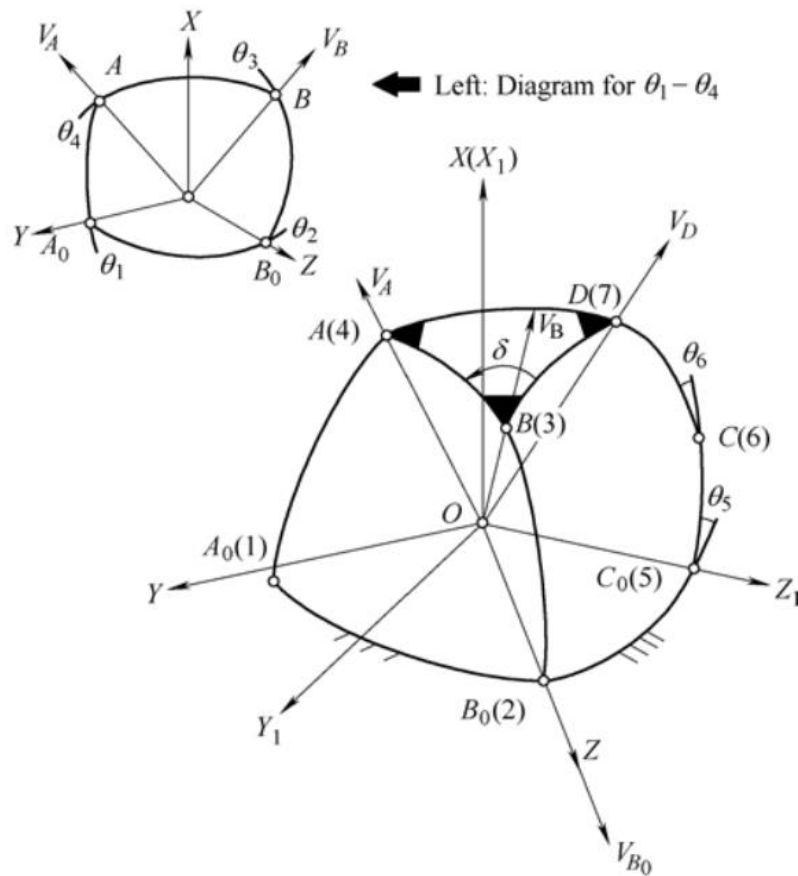


Figure 2-17: Mechanism Coordinate Frames[14]

Table 2-1: Forward Kinematics Results for Stephenson III 6-bar [14]

Joint angle θ_4 (°)	Output angle θ_5 (°)
-32.22	-61.93
	33.06
	83.84
	175.22
116.7	-143.51
	-61.81
	65.47
	85.53

Similarly, Lui and Yang determined the inverse kinematics of the system for an input angle given as $\theta_5 = -143.51$. The table below displays these results [14]. These forward and inverse kinematic solutions are found in the paper by constructing constraint equations based on the reference frame analysis of the mechanisms from previous work, and further simplifying the relations to eliminate extraneous roots, thus providing a more simplified solution to the problem.

Table 2-2: Inverse Kinematics of Stephenson III 6-bar [14]

θ_1 (°)	-139.88	60.00	111.17	125.12
----------------	---------	-------	--------	--------

2.4.3 Kinematic Solution for Spatial Parallel Linkage

Gao and Wu [15] outline the solution of the forward kinematics for a spatial parallel linkage based on the corresponding constraint equations. The first stage of this solution is presented as the breakdown of the linkage into lower pairs of linkages and establishing the constraint equations for these pairs. Table C-1 in the Appendix provides the relevant constraint equations for the lower pairs. The reference provides the case study solution of the forward kinematics of a Stewart platform using this analysis[15].

2.4.4 Kinematic Analysis of a 6R Overconstrained Spatial Mechanism

This solution proposed by Kong involves the use of the transformation matrix analysis to determine the forward kinematics of the 6R overconstrained mechanism. The figure below is the matrix that is analysed in this reference [16]. The corresponding general form of the transformation matrix is given in Equation (2-8) and the closed loop kinematic relation is given by Equation (2-9).

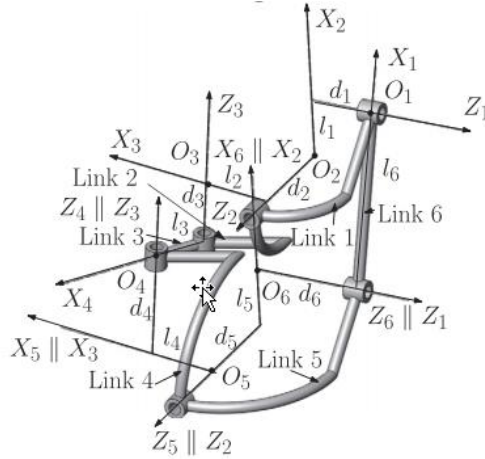


Figure 2-18: Coordinate frames for 6R mechanism analysed by Kong [16]

$$T_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & C\theta_i l_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & S\theta_i l_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2-8)$$

$$T_1 T_2 T_3 T_4 T_5 T_6 = I \quad (2-9)$$

2.4.5 Analysis and Simulation of Kinematic Error in a Spatial Linkage

Xu and Zhang [17] present the analysis and simulation of the kinematic error in a spatial linkage using the stochastic model of output error without input error. This formula is given in Equation (2-10).

$$\Delta\phi = \sum_{i=1}^n \left(\frac{\delta\phi}{\delta q_i} \right) \Delta q_i + \sum_{i=1}^m \left(\frac{\delta\phi}{\delta c_i} \right) \Delta c_i \quad (2-10)$$

Here Δq_i represent the aberrations in the linkage parameters, and Δc_i the position anomalies of the pin in relation to the pin sleeve[17].

The paper further applies rejection method to sample random variables with any density function for spatial linkages. This is a Monte Carlo simulation method, in the reference it is used to model the kinematic error for spatial linkages. The choice of a random variable, $x \in (1, u)$, with a density function $f(x)$ is given by Equation (2-11),

$$x = (u - 1)\eta_1 + 1 \quad (2-11)$$

If

$$\eta_2 \leq \frac{f(x)}{f_0} \quad (2-12)$$

where η_1 and η_2 are random numbers between 0 and 1, and f_0 is the maximum value of $f(x)$. [17].

The authors present the sampling formulas of random vectors of common joints in spatial linkages, then simulate the stochastic model of the output error. Once the models are developed, the

clearance characteristic element and clearance space parameters are introduced. Thus, giving the full effect of joint clearance on kinematic error. These simulation methods can provide higher accuracy for cad models of linkages [17].

2.5 Redundantly Actuated Systems

A system containing two serial manipulators with a common payload is equivalent to a redundantly actuated parallel manipulator [18]. Actuator redundancy involves inclusion of a greater number of actuators than the minimum needed to achieve the manipulator DOFs [19]. Justification for redundant actuation consists of 3 main benefits: elimination of singularities, higher system stiffness, and distribution of loads among actuated chains. Actuation redundancy can be achieved in two ways: actuation of passive joints, and inclusion of additional kinematics chains without increasing the system DOF [20]. Additional actuator chains mean greater complexity in the design and analysis of the system, however, have greater mechanical feasibility, and are more effective in reducing singularities – thus increasing the workspace of the manipulator [21]. The distribution of loads among the actuators reduces the individual actuator loads, thus increasing the dynamic capability of the system. Greater stiffness increases the accuracy of the system, as well as eliminating joint backlash [22]. The solution to the inverse force dynamics of a redundantly actuated system provides infinite possible results. This allows for optimization of the joint torques and forces [18]. As a result, redundant actuation facilitates a more homogenous force output. Allowing for a lighter and faster mechanical structure. [23] Furthermore, redundantly actuated systems have a higher safety in the event of breakdown, as the system can still be controlled despite breakdown of one actuation chain [23].

2.5.1 Analysis of Redundantly Actuated Systems

Solving the forward kinematics of redundantly actuated systems consists of two main methods: analytical methods, and numerical methods. Analytical methods include the use of additional sensors to evaluate the forward kinematics relations, as well as using the Newton-Raphson method to solve a more generalized case. A popular solution method is the use of artificial neural networks, this has shown some promising results for the FK problem. However, the disadvantages to this method include: inability to be adopted for on-line application (necessary for control of parallel manipulators), accuracy heavily depends on the training of system (a large data set is needed to achieve accurate output), slow convergence speed, local minima and poor generalization [19]. Due to the complexity of the system dynamics the inverse force solution for redundantly actuated systems cannot be found using standard inverse dynamics methods. Most generally the pseudo-inverse is found as the solution to the inverse force problem. This method is equivalent to minimizing the 2-norm of the torques and forces. The pseudo-inverse solution however, does not yield internal forces in the manipulators and thus does not provide the actual force capabilities of the system. To determine the maximum force capabilities of redundantly actuated systems optimization is necessary [18].

CHAPTER 3 Kinematic Modeling

The first stage in design verification of a mechanism is the kinematic modeling. This involves the coordinate system set-up, definition of the model geometry, mobility analysis, establishment of the reference frames, and finally the forward kinematics analysis. These steps are outlined in the following sections in this chapter.

3.1 Coordinate Set-up

The coordinate system set-up involves the establishment of the global reference frame. The coordinate system was selected by setting the direction of the x-axis as positive towards the wingtip. Once the x-axis was selected the positive direction of the z-axis was selected to be upwards out of the winglet surface. The y-axis definition was determined as the common normal between the x and z axes. This coordinate system selection is outlined in the figure below. The x-axis is indicated in blue, the z-axis in red and the y-axis in black. As can be seen the origin was selected as the fixed anchor joint of the fwd mechanism input actuator, this was selected to simplify vector and reference frame setup for the later stages in analysis. This is an important stage in the forward kinematics analysis, as it provides the basis for the geometric specifications and constraint definition of the system components. Following the determination of the global reference frame, the model geometry and system mobility can be established, as carried out in the following sections.

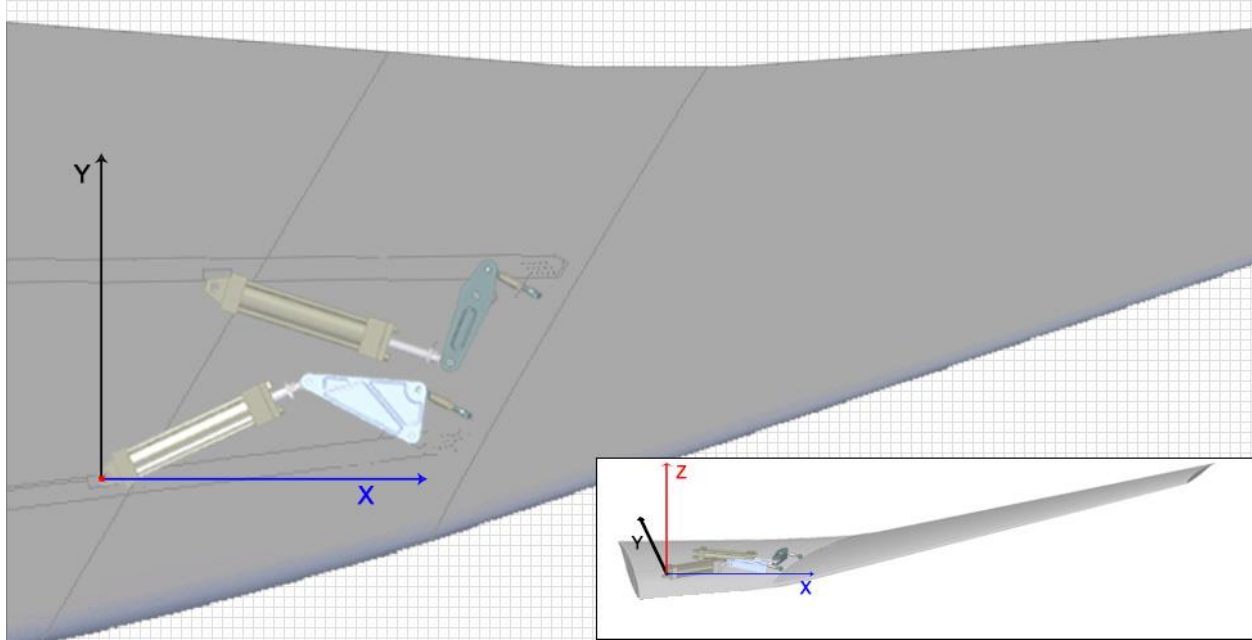


Figure 3-1: Global reference frame setup

3.2 Geometry and Model Definition

The mechanism is shown in detail in Figure 3-2 below. The system is composed of two closed loop linkages acting as a parallel mechanism. There are 3 fixed joints on each chain: A, D, G, and H, K, N. Joints B and I, are prismatic actuators, and the inputs for the two chains respectively. Each chain is essentially a bell-crank mechanism. Lines 1-2 and 8-9 represent two linear actuators. Two triangles formed by lines 3,4,5 and lines 10,11,12 are called bell. Lines 6 and 13 are called toggle links. Lines 7 and 14 present the same winglet at two different locations.

In terms of the connection, joints A, D, G, H, K, and N are mounted on the base called spar. The joints F, and M are the connection between the mechanism and the winglet, these joints are mounted on a winglet interface mount. This connection transfers the mechanism motion into pure winglet rotation

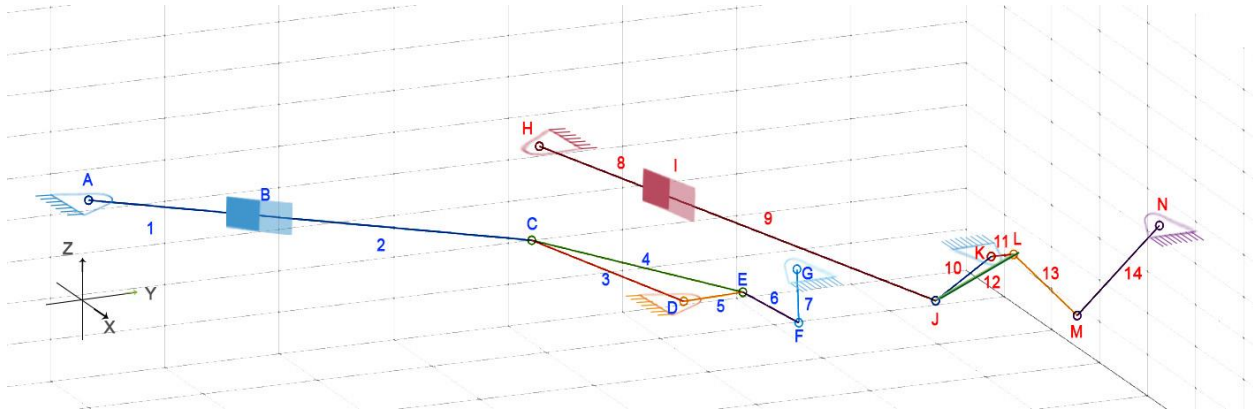


Figure 3-2: Model Geometry

3.3 Mobility Analysis

The mobility analysis for the system will be conducted separately for each linkage chain. The figure below shows a more detailed diagram of the fwd linkage with each of the joint types labelled. Joints A, D, G are revolute joints, B is a prismatic joint, and C E and F are spherical joints. The system mobility is calculated as:

$$M = 6(6 - 7 - 1) + (1 + 1 + 1 + 1 + 3 + 3 + 3) \quad (3-1)$$

$$M = 1 \quad (3-2)$$

Taking the first chain as an example, Joint A is connected to the actuator and fixed to the base, rotating about the Z-axis. Joint B is a prismatic actuator, with translation in the XY plane. Joint D is connected to the triangle plate and fixed to the base rotating about the Z-axis. Joint G is fixed the base to pivot the winglet about the X-axis. One of our design criteria is to minimize the toggle link, link 6, lateral movement. This requires the motion of joint E to remain as close to a straight line as possible. This would effectively change joint C into a revolute joint, hence the motion of the bell (triangle plate) would be a pivot about joint D in the XY plane.

As mentioned before, lines 3,4 and 5 are treated as one link forming the bell and as such the internal angles, c , e , and d (shown in Figure 3-4) are design parameters. The output of the end effector is a Y axis rotation, with the axis fixed at G. This can be seen more clearly in Figure 3-3. Due to this motion constraint, despite the spatial positioning of joints E and F, the motion of these points is entirely planar. Joint E moves in the XY plane, and joint F moves in the XZ plane. This provides a constraint on the system that simplifies the kinematic analysis and is used in the following chapters.

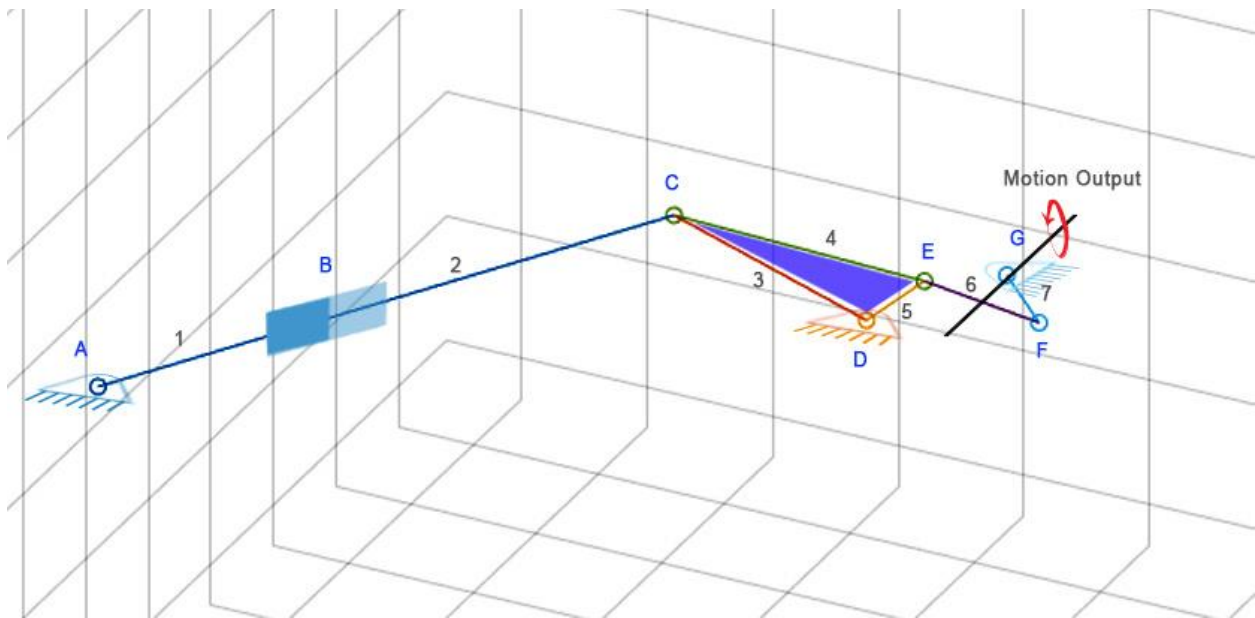


Figure 3-3: fwd Linkage joint motions

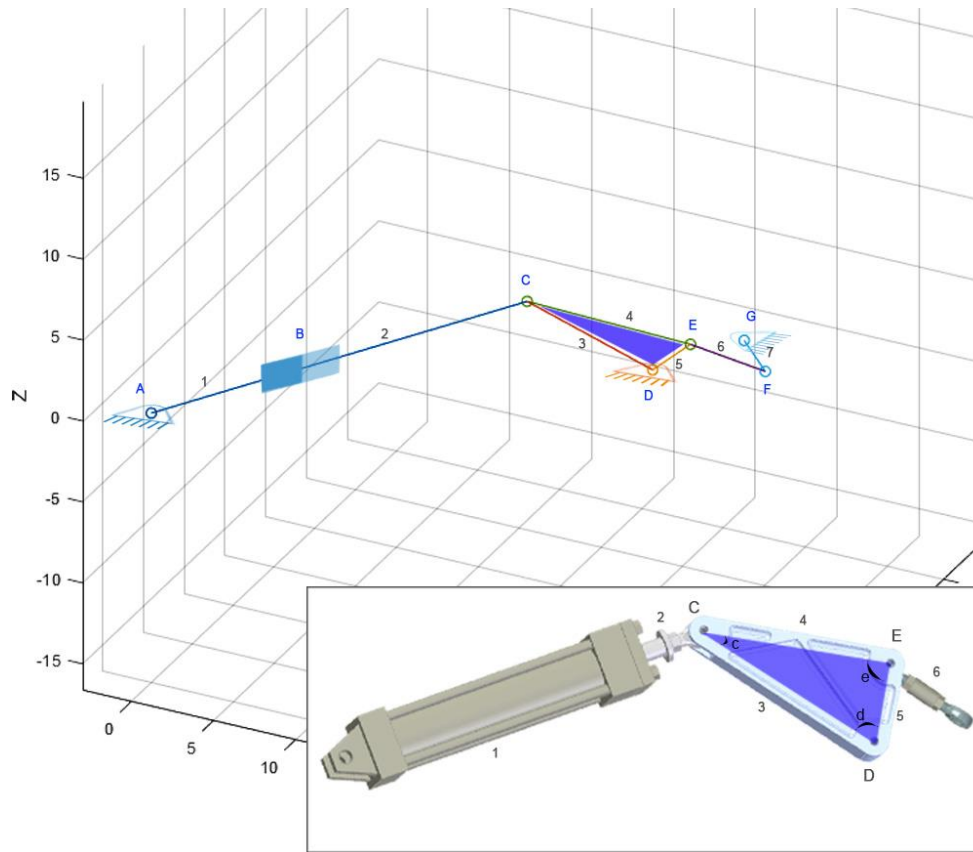


Figure 3-4: Link 3,4, and 5 forming bell crank link

The second chain is similarly structured, beginning with a fixed hinge with DOF of 1. Joint I is a prismatic actuator, with ad DOF of 1. Joints J, K, and N are also rotary actuators with DOF of 1. Joints L and M are spherical joints similar to joints E and F in the fwd linkage, the same motion constraints apply to these joints as well. Lines 10, 11, and 12 can be considered a single link as the angle between these two links is fixed, making up the bell crank for the aft linkage (shown in Figure 3-5).

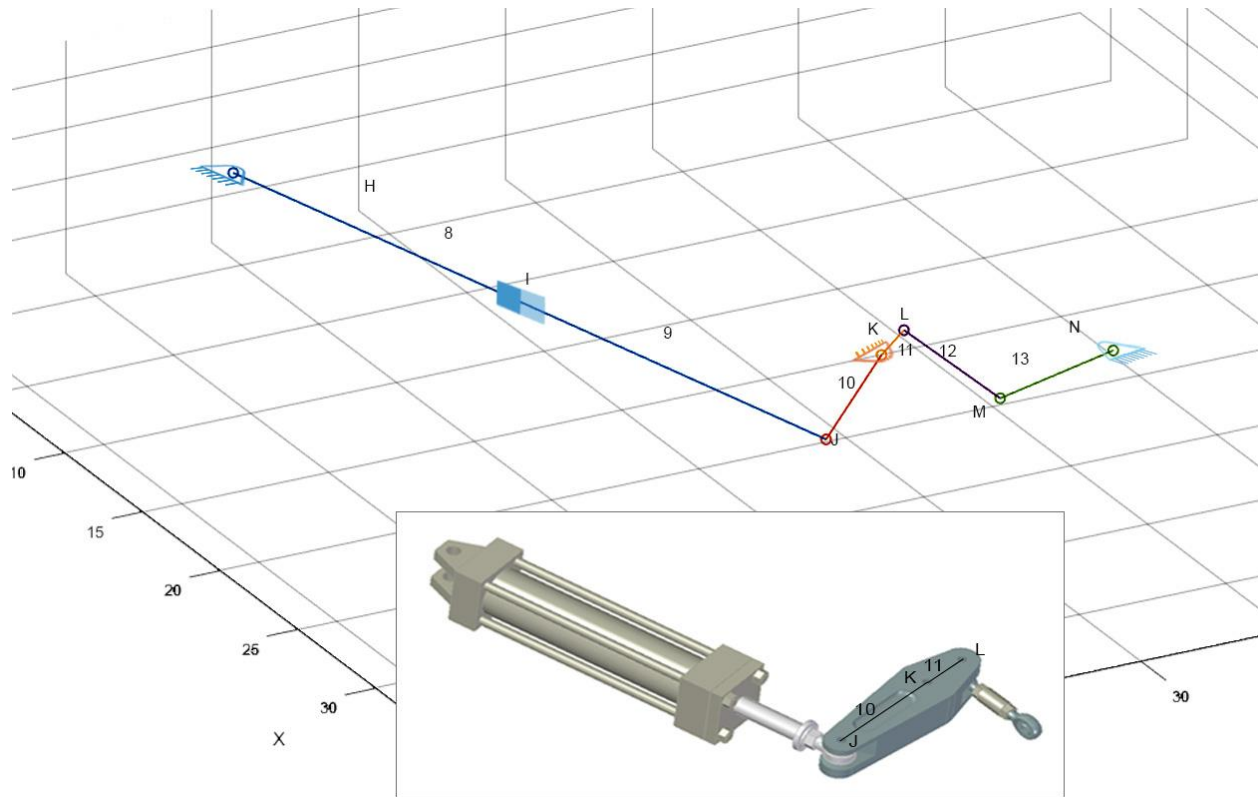


Figure 3-5: Bell crank links aft linkage

The fwd (first) and aft (second) mechanisms operate in an inverse relation to one another. The fwd linkage extends to increase the system output (winglet) angle, while the aft linkage is in compression during this process. This manner of motion accounts for the load sharing capability of the system and prevents possible collisions between the mechanisms. The two serial chains are connected to the same platform (winglet/winglet interface) forming a parallel mechanism. Furthermore, the DOF of the system is 1, a rotation about the Y-axis, accomplished using both the actuators. This indicates that the design can be considered a redundantly actuated parallel mechanism.

3.4 Detailed Model Definition

Throughout the following analyses the notation \mathbf{p}_i is used to designate vectors from the origin to the joint locations, \mathbf{r}_i is used to designate the vectors between joints. As a further note, considering the output of the system is a frame that runs along the Y axis between points p_6 and p_{12} , as such the Y axis coordinates of these points is used as the Y axis coordinate of p_5 and p_{11} accordingly. This simplifies the analysis by allowing the vectors \mathbf{r}_5 and \mathbf{r}_{11} to be in the XZ plane. There is no affect on the accuracy of the analysis as the motion is a pivot about the Y axis attached to this frame. As discussed previously the mechanism model is composed of the fwd and aft linkage chains. Each chain, for example, the fwd linkage is made up of 2 closed loops. The 2 chains are shown in Figure 3-7. The second loop is the spatial loop, due to the vectors \mathbf{r}_5 and \mathbf{r}_6 . The angle θ_3 between vectors \mathbf{r}_3 and \mathbf{r}_4 is fixed and a design parameter. Angle θ_1 is an internal angle used in combination with ϕ_1 , to total θ_2 , for the definition of p_2 . Point p_4 can be defined in two ways, the simplest method is to use internal angle θ_3 and r_4 from fixed point p_3 . The output vector of this loop can be considered \mathbf{r}_4 . In the case of loop 2 the fixed point p_6 is used as the starting point. The angle θ_6 is directly related to the winglet angle and will be taken as the output for the fwd mechanism. The interface vector \mathbf{r}_4 is present in both loops and acts as the input for this loop.

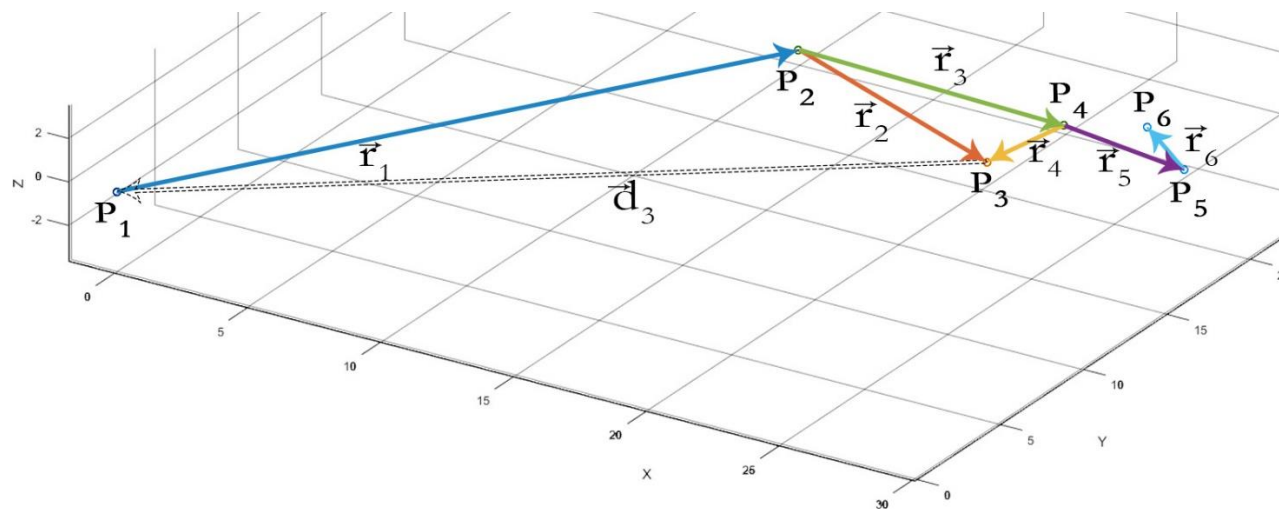
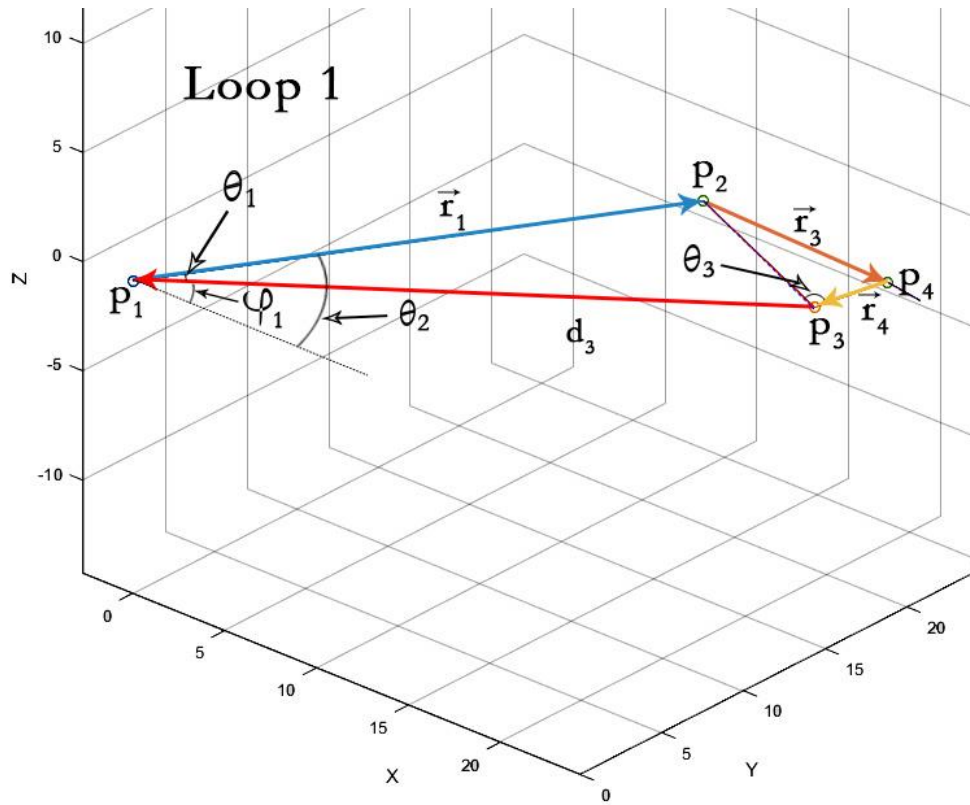
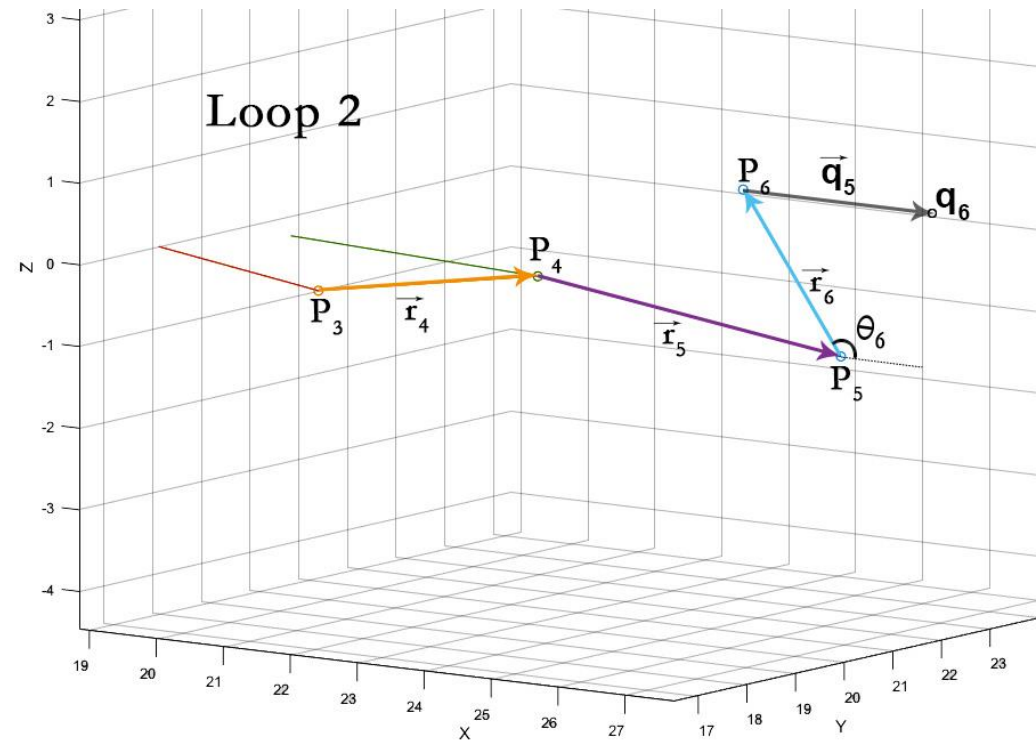


Figure 3-6: Fwd linkage vector setup



(a)



(b)

Figure 3-7: Fwd linkage vector loop definition - (a) Loop 1 (planar loop), (b) Loop 2 (spatial loop)

The aft linkage vector setup is similar to the fwd mechanism. The system is composed of two closed loop chains, one that can be considered planar, and a second spatial chain. The angle θ_7 is used to determine p_8 . The point p_9 is fixed, and r_8 is considered the output link. The link r_9 is considered an extension of the output link r_8 . The angle θ_8 is a design parameter and is used to determine the pose of link r_9 . The output of the second loop is θ_{11} , this is the same as the output of the fwd linkage as the two systems are connected to the same output platform.

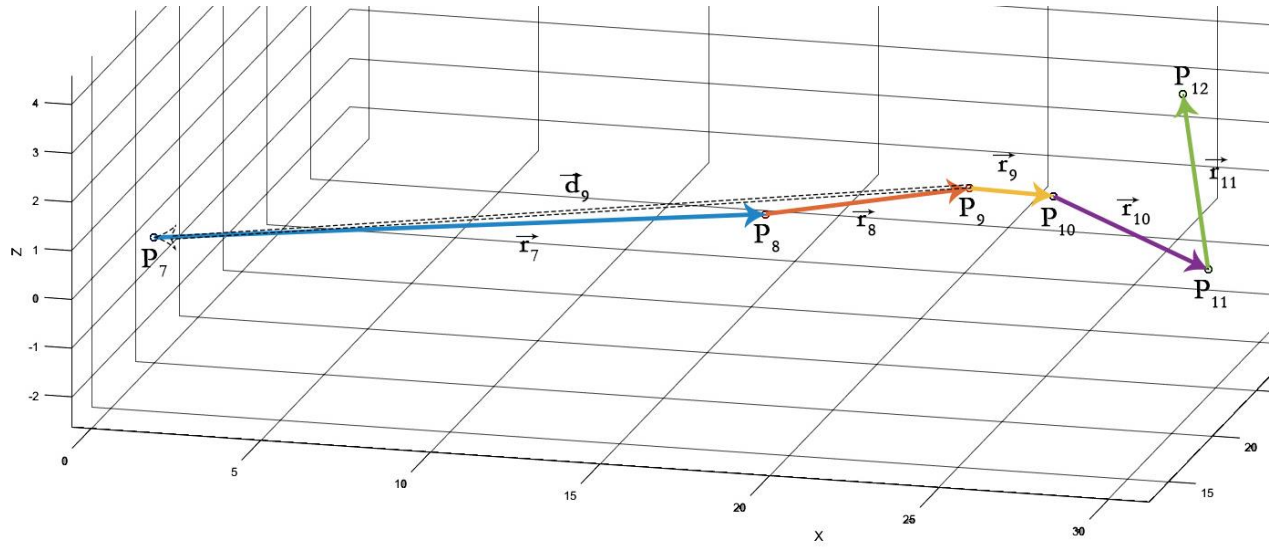
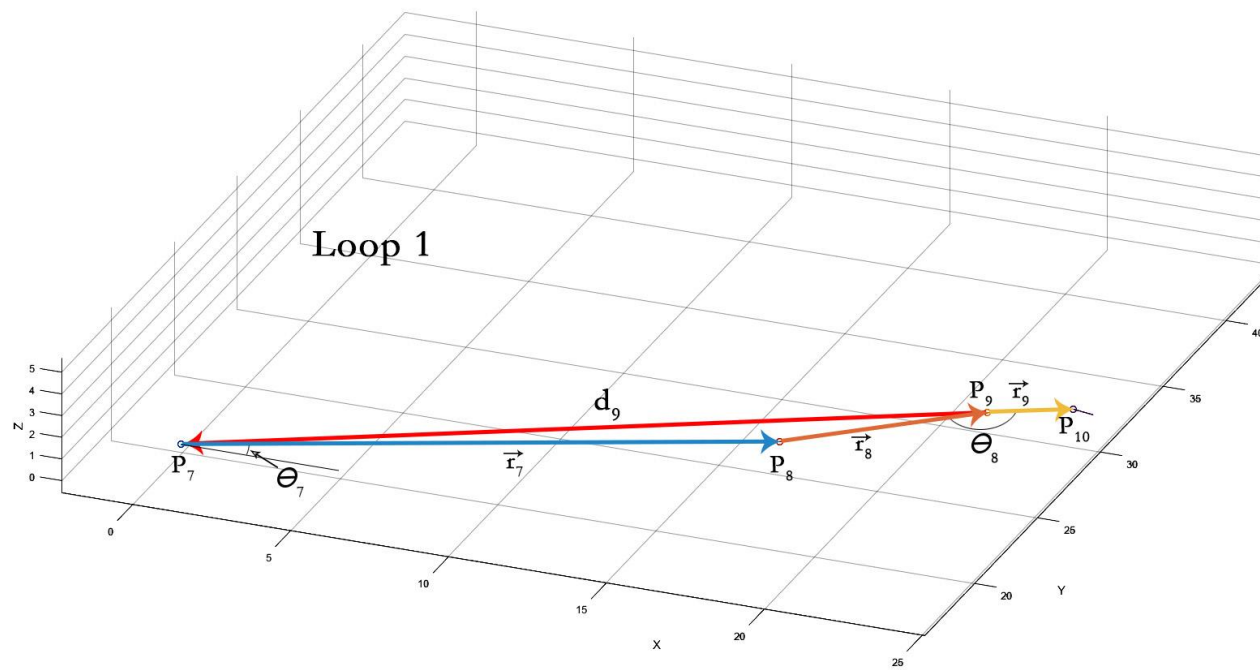


Figure 3-8: Aft linkage vector setup

(a)



(b)

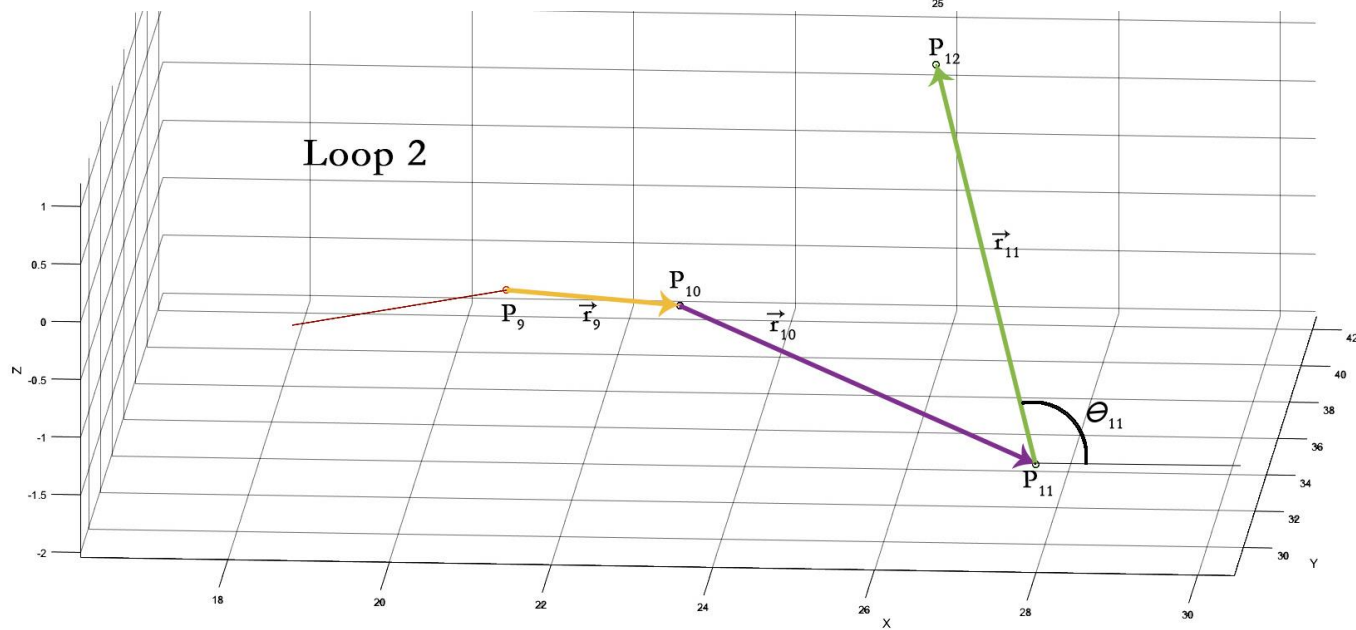


Figure 3-9: Aft linkage vector loop definition – (a) Loop 1 (planar loop), (b) Loop 2 (spatial loop)

3.5 Forward Kinematics Analysis

The forward kinematics was modeled with the linear actuator length as the input joint variable. In order to define point p_2 from here, the joint angle θ_2 is required. This angle can be represented as the summation of the internal angle θ_1 , and ϕ_1 the design angle between the vector from p_1 to p_3 , this is shown in Figure 3-9(a). Applying the cosine law to the triangle formed by p_1 , p_2 , and p_3 (shown in Figure 3-7)– with the input r_1 and fixed link length d_3 – yields θ_1 .

$$\theta_2 = \theta_1 + \phi_1 \quad (3-3)$$

$$\theta_1 = \arccos \left[\frac{r_1^2 + d_3^2 - r_2^2}{2r_1 d_3} \right] \quad (3-4)$$

Using the x and y coordinates of point p_3 , ϕ_1 can be defined.

$$\phi_1 = \arctan \left(\frac{p_{3y}}{p_{3x}} \right) \quad (3-5)$$

Given θ_2 point p_2 is defined using a z-axis rotation matrix:

$$\mathbf{p}_2 = \mathbf{r}_1 \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-6)$$

With point p_2 defined, using the design angle θ_3 , point p_4 can be defined using the equation below, which involves the Z-axis rotation and scaling of r_2 to establish the pose of r_3 .

$$\mathbf{p}_4 = \mathbf{p}_3 + R_4 \mathbf{u}_2 r_3 \quad (3-7)$$

R_4 is a z-axis rotation matrix, with $-\theta_3$ as the rotation angle. The angle θ_3 is negative by convention, as the rotation is clockwise.

$$R_4 = \begin{bmatrix} \cos(-\theta_3) & -\sin(-\theta_3) & 0 \\ \sin(-\theta_3) & \cos(-\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-8)$$

Further, \mathbf{u}_2 is the unit vector corresponding to \mathbf{r}_2 .

$$\mathbf{u}_2 = \frac{(p_2 - p_3)}{|p_2 - p_3|} \quad (3-9)$$

The pose determination for vector \mathbf{r}_5 is more involved than the initial analysis. Due to the spatial nature of the link, the point p_5 can only be defined based on the two critical points in the loop, p_4 and p_6 . The pose of \mathbf{r}_5 can be established using 2 relations, shown in Equations (3-10) and (3-11). The two equations below both involve the use of p_5 , along with pre-defined points p_4 and p_6 to define the link lengths r_6 and r_5 . At this point in the analysis it is relevant to mention that the Y coordinate of p_5 is constant as this point moves similar to a slider in the XZ- plane. This simplifies the relations below, making a solution for the remaining coordinates of p_5 obtainable. Solving the above relations simultaneously provides the x and z coordinates for p_5 .

$$r_6 = |p_6 - p_5| \quad (3-10)$$

$$r_5 = |p_4 - p_5| \quad (3-11)$$

$$p_{5,y} = \text{constant} \quad (3-12)$$

The above relations can be solved by expanding the two equations, as shown in Equations (3-13) and (3-14). These equations can then be solved simultaneously using the symbolic computation toolbox in MATLAB. The resultant explicit expression is a complex relation providing the pose of \mathbf{r}_5 in terms of p_4 and p_6 . Once the coordinates of p_5 are determined, the vector \mathbf{r}_6 can be defined, the pose of this vector provides the output, Z-axis rotation θ_6 .

$$r_6 = \sqrt{(p_{6x} - p_{5x})^2 + (p_{6y} - p_{5y})^2 + (p_{6z} - p_{5z})^2} \quad (3-13)$$

$$r_5 = \sqrt{(p_{4x} - p_{5x})^2 + (p_{4y} - p_{5y})^2 + (p_{4z} - p_{5z})^2} \quad (3-14)$$

Likewise, the aft linkage forward kinematics is determined using the linear actuator length as the joint variable input. Due to the difference in geometry of the chain, the definition of the angle θ_7 is more involved than for θ_2 . The equations below outline this procedure. The constant angle ϕ_2 is found using the length d_9 and the X-axis coordinate of \mathbf{d}_9 . Applying the cosine law to the triangle formed by points p_7 , p_8 and p_9 yields variable angle ϕ_3 . Once these intermediary angles are found the relevant joint variable θ_7 is established.

$$\theta_7 = \phi_2 - \phi_3 \quad (3-15)$$

$$\phi_2 = \arccos\left(\frac{d_9}{d_{9x}}\right) \quad (3-16)$$

$$\phi_3 = \arccos\left[\frac{d_9^2 + r_7^2 - r_8^2}{2d_9 r_7}\right] \quad (3-17)$$

$$\mathbf{p}_8 = \mathbf{r}_7 \begin{bmatrix} \cos\theta_7 & -\sin\theta_7 & 0 \\ \sin\theta_7 & \cos\theta_7 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-18)$$

The point p_9 is a fixed position revolute joint, and thus is used for the definition of point p_{10} . The vector \mathbf{r}_9 is determined by applying a rotation to the unit vector \mathbf{u}_8 , corresponding to vector \mathbf{r}_8 , thus establishing the orientation for the required vector. The rotated unit vector is then

scaled by the length of the link, r_9 . These transformations provide the pose of \mathbf{r}_9 and are shown in Equation (3-19).

$$\mathbf{p}_{10} = \mathbf{p}_9 + R_8 \mathbf{u}_8 r_9 \quad (3-19)$$

The rotation matrix R_8 is composed to represent a Z-axis rotation by an angle of θ_9 . The angle is positive in this case as the difference in orientation of \mathbf{r}_8 and \mathbf{r}_9 is a θ_9 counter clockwise rotation about the Z-axis.

The pose of vector \mathbf{r}_{10} is defined based on the location of \mathbf{p}_{10} and \mathbf{p}_{11} . The location of \mathbf{p}_{11} is determined using a similar analysis to \mathbf{p}_5 . The two relations involving \mathbf{p}_5 are generated using the vector norms of \mathbf{r}_{10} and \mathbf{r}_{11} .

$$r_{10} = |\mathbf{p}_{11} - \mathbf{p}_{10}| \quad (3-20)$$

$$r_9 = |\mathbf{p}_{10} - \mathbf{p}_9| \quad (3-21)$$

$$p_{10y} = \text{constant} \quad (3-22)$$

Following the procedure for joint \mathbf{p}_5 in the fwd linkage, expanding the equations above yields the following system of equations.

$$r_{10} = \sqrt{(p_{11x} - p_{10x})^2 + (p_{11y} - p_{10y})^2 + (p_{11z} - p_{10z})^2} \quad (3-23)$$

$$r_9 = \sqrt{(p_{10x} - p_{9x})^2 + (p_{10y} - p_{9y})^2 + (p_{10z} - p_{9z})^2} \quad (3-24)$$

Solving these equations simultaneously for the X and Z coordinates of p_{10} provides the explicit solution for the pose of r_{10} . The explicit solution is found using the symbolic computation toolbox in MATLAB, due to the complexity of the equations and the solution.

3.6 Workspace Definition

Upon establishing the forward kinematic solutions, the workspace of the initial proposed design must be evaluated. This is determined by tracking the motion of the end effector through the maximum design stroke range of the actuator. The resultant workspace of the system is shown in the figure below. As can be seen in the figure the range of motion of the aft linkage is lower than that of the fwd linkage. This presents a problem as both the linkages must have the same output link angle, as this angle corresponds to the winglet interface – a solid platform. Without synchronized motion of the 2 output links, the configuration causes undesired resistance for the actuators, and twisting in the winglet interface. It is, however, a design confirmation to notice that the motion of the two output joints is solely in the XZ – plane, shown in Figure 3-10 (b). Furthermore, it can be seen from Figure 3-10 (a) that the motion of the bell crank joint p_4 and p_{10} is relatively small along the Y-axis. The fwd linkage has the largest displacement for this joint, which is under 1 inch. This limits the amount of lateral twisting in link r_5 and r_{11} . The Z axis translation of joints p_5 and p_{11} shown in Figure 3-10 (a). The goal of the design is to limit the protrusion of the mechanism outside the winglet frame. This Z axis motion is representative of such motion. It is shown in the figure that this motion is also fairly small, the largest value being about 1 inch below the starting position. The next stage involves the inverse kinematics to establish the required stroke lengths for each chain to reach the desired output angular range, in a synchronous manner.

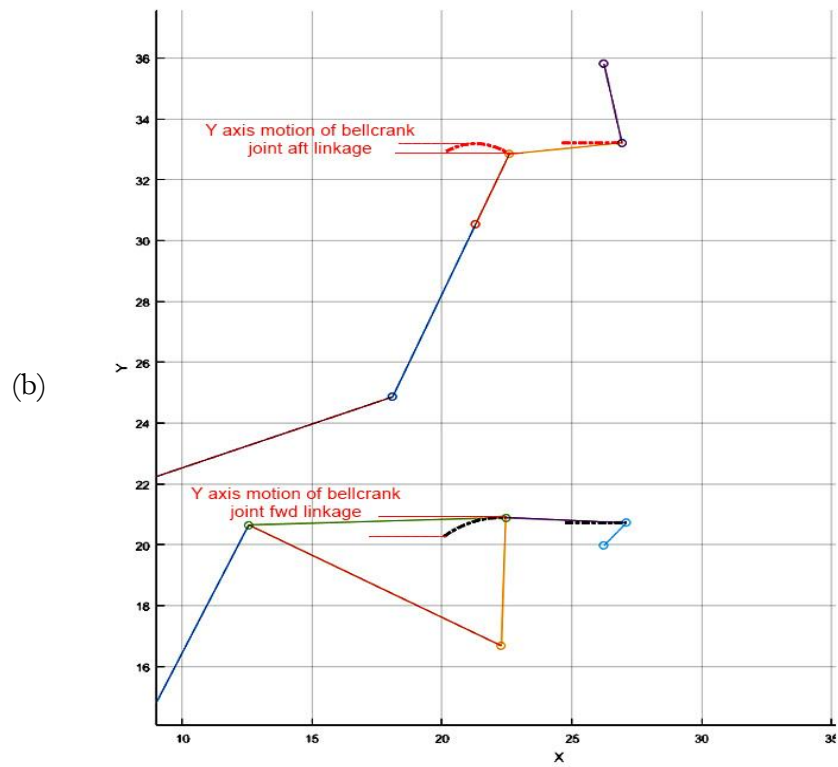
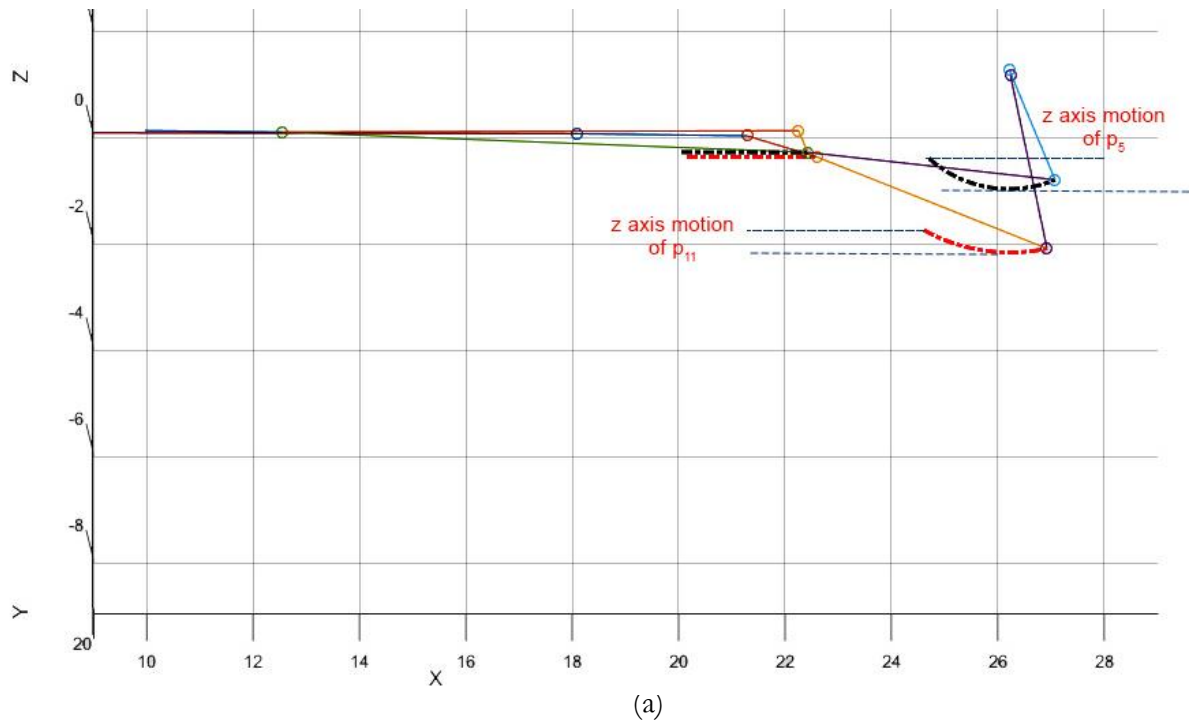


Figure 3-10: Preliminary Workspace (a) XZ-plane, (b) XY-plane

3.7 Inverse Kinematics

The inverse kinematics provides the relations for determining the required actuator length for given winglet angle. The output for the fwd linkage was selected as the Y-axis orientation angle of link \mathbf{r}_6 . Using this angle and the length of the link it is possible to determine the pose of link \mathbf{r}_6 . This is represented in Equation (3-28). It is important to recall that the Y coordinate of \mathbf{p}_5 is constant, and thus the vector can be modeled as a Y-axis rotation by θ_6 of intermediary vector \mathbf{q}_5 . This intermediary vector is defined as a vector of length r_6 , between \mathbf{p}_6 and point \mathbf{q}_6 . This vector is defined in Equation (3-25). The pose of \mathbf{p}_5 is then found by rotating the vector \mathbf{q}_5 by ϕ_6 about the Y-axis and translating from \mathbf{p}_6 by the rotated vector of length r_6 .

$$\mathbf{q}_5 = [r_6, 0, 0] \quad (3-25)$$

$$R_6 = \begin{bmatrix} \cos(\phi_6) & 0 & \sin(\phi_6) \\ 0 & 1 & 0 \\ -\sin(\phi_6) & 0 & \cos(\phi_6) \end{bmatrix} \quad (3-26)$$

$$\phi_6 = -\theta_6 \quad (3-27)$$

$$\mathbf{p}_5 = \mathbf{p}_6 - R_6 \mathbf{q}_5 \quad (3-28)$$

The pose of \mathbf{r}_4 is similar in complexity to the determination of the pose of \mathbf{r}_5 in the forward kinematics. The 2 relations needed are shown in the equations below. The vector norms of \mathbf{r}_5 and \mathbf{r}_3 provide the 2 equations required to determine an explicit solution for the pose of \mathbf{r}_5 . The Z coordinate of \mathbf{p}_4 is a design variable and is constant for the purpose of establishing the kinematics and inverse kinematics.

$$\mathbf{r}_5 = |\mathbf{p}_5 - \mathbf{p}_4| \quad (3-29)$$

$$\mathbf{r}_3 = |\mathbf{p}_3 - \mathbf{p}_4| \quad (3-30)$$

$$p_{4_z} = \text{constant} \quad (3-31)$$

As with the forward kinematics case, the equations were solved explicitly using MATLAB's symbolic computation toolbox. The expanded form of the above equations is shown below.

$$r_5 = \sqrt{(p_{5x} - p_{4x})^2 + (p_{5y} - p_{4y})^2 + (p_{5z} - p_{4z})^2} \quad (3-32)$$

$$r_3 = \sqrt{(p_{3x} - p_{4x})^2 + (p_{3y} - p_{4y})^2 + (p_{3z} - p_{4z})^2} \quad (3-33)$$

The pose of \mathbf{r}_3 is known, as \mathbf{p}_3 is a fixed point. The remaining vector \mathbf{r}_2 is defined using the inverse form of Equations (3-7) – (3-9) from the forward kinematics section. The rotation in this case is a positive θ_3 rotation about the Z-axis. The unit vector, \mathbf{u}_4 , representing the orientation of \mathbf{r}_4 is rotated using the rotation matrix in Equation (3-34), scaled to the length of r_2 , and translated relative to \mathbf{p}_3 . This process is outlined in the equations below.

$$R_2 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-34)$$

$$\mathbf{u}_4 = \frac{(\mathbf{p}_3 - \mathbf{p}_4)}{|\mathbf{p}_3 - \mathbf{p}_4|} = \frac{(\mathbf{p}_3 - \mathbf{p}_4)}{r_3} \quad (3-35)$$

$$\mathbf{p}_2 = \mathbf{p}_3 - R_2 \mathbf{u}_4 r_2 \quad (3-36)$$

Taking the Euclidean norm of \mathbf{p}_2 yields the linear actuator length as the output of the inverse kinematics for the fwd linkage.

Similarly, the aft linkage inverse kinematics are determined starting with the pose of link \mathbf{r}_6 . The orientation of this link is the input for the inverse kinematics, θ_{11} . The coordinates of \mathbf{p}_{11} are established in the same manner as those of \mathbf{p}_5 . The corresponding equations are given below.

$$\mathbf{q}_{11} = [r_{11}, 0, 0] \quad (3-37)$$

$$R_{11} = \begin{bmatrix} \cos(\phi_{11}) & 0 & \sin(\phi_{11}) \\ 0 & 1 & 0 \\ -\sin(\phi_{11}) & 0 & \cos(\phi_{11}) \end{bmatrix} \quad (3-38)$$

$$\phi_{11} = -\theta_{11} \quad (3-39)$$

$$\mathbf{p}_{11} = \mathbf{p}_{12} - R_{11}(\mathbf{q}_{11}) \quad (3-40)$$

The location of \mathbf{p}_{10} is determined using the two relations below, involving the two vector norms of \mathbf{r}_{10} and \mathbf{r}_9 . The Z coordinate of \mathbf{p}_{10} is constant. The expanded forms of the relations are provided in Equations (3-45) and (3-46). The expressions are explicitly solved using the symbolic computation Toolbox in MATLAB.

$$\mathbf{r}_{10} = |\mathbf{p}_{11} - \mathbf{p}_{10}| \quad (3-41)$$

$$\mathbf{r}_9 = |\mathbf{p}_9 - \mathbf{p}_{10}| \quad (3-42)$$

$$p_{10z} = \text{constant} \quad (3-43)$$

$$r_{10} = \sqrt{(p_{11x} - p_{10x})^2 + (p_{11y} - p_{10y})^2 + (p_{11z} - p_{10z})^2} \quad (3-44)$$

$$r_9 = \sqrt{(p_{10x} - p_{9x})^2 + (p_{10y} - p_{9y})^2 + (p_{10z} - p_{9z})^2} \quad (3-45)$$

Point \mathbf{p}_9 is fixed, and point \mathbf{p}_8 is found using the inverse of Equations (3-15) – (3-18). This process is outlined in the equations below.

$$R_8 = \begin{bmatrix} \cos(\theta_8) & -\sin(\theta_8) & 0 \\ \sin(\theta_8) & \cos(\theta_8) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-46)$$

$$\mathbf{u}_{10} = \frac{(\mathbf{p}_9 - \mathbf{p}_{10})}{|\mathbf{p}_9 - \mathbf{p}_{10}|} = \frac{(\mathbf{p}_9 - \mathbf{p}_{10})}{r_9} \quad (3-47)$$

$$\mathbf{p}_8 = \mathbf{p}_9 - R_8 \mathbf{u}_{10} r_8 \quad (3-48)$$

With the pose of \mathbf{r}_8 determined, the output of the inverse kinematics of the aft linkage is established by taking the Euclidean norm of link \mathbf{r}_7 .

Using the inverse kinematics results it is possible to determine the required stroke length to achieve synchronized motion of the two chains. Additionally, the target angular range for the output platform (winglet interface) can be accurately achieved once the required stroke length is determined.

3.8 Stroke Length Determination for Target Workspace

Using the inverse kinematics relations, it is possible to determine the required input actuator range for both, the fwd and aft, linkages in order to reach the required target output angular range in a synchronous manner. This value can be found by inputting the desired angular range into the inverse kinematics relations to output the minimum and maximum length of the linear actuator. The resultant new stroke lengths as a function of the winglet angle are shown in Figure 3-11.

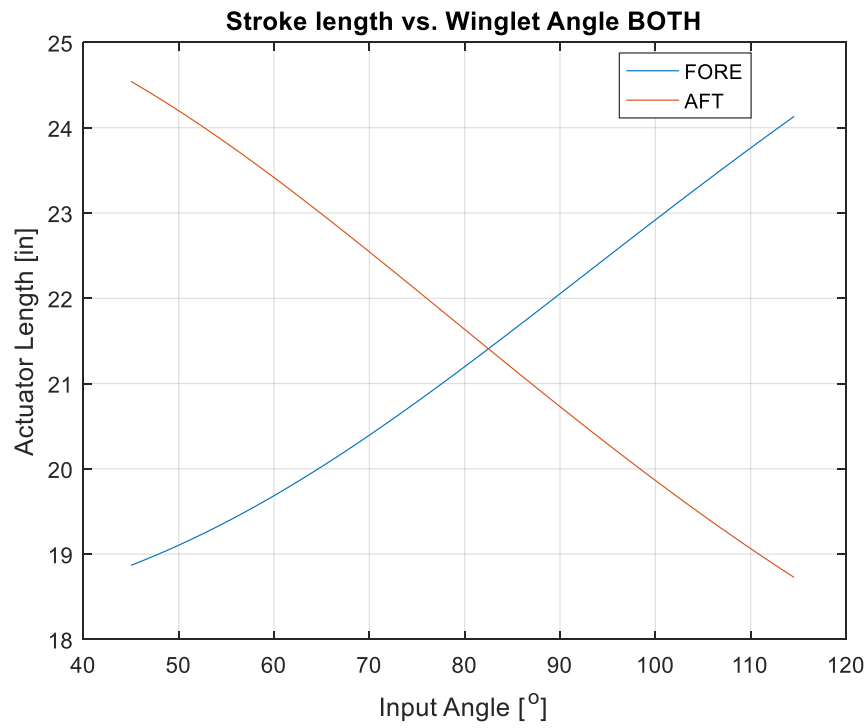


Figure 3-11: Relation between Winglet interface angle and linear actuator stroke length

CHAPTER 4 Velocity Analysis

The velocity relations are calculated for each chain by differentiating the forward and inverse kinematic equations. The following sections outline the process in detail. The velocity relations are then used to determine the synchronized actuator velocities. Further, the system is studied to determine the optimized actuator velocity, the impact of design values on the system velocity, and the optimal manner to achieve and maintain the target velocity.

4.1 Fwd linkage Forward Velocity

The forward velocity of the fwd linkage is calculated by successively differentiating the corresponding forward kinematics equations. The input is the linear velocity of the prismatic link, \mathbf{r}_1 , in the form of the stroke speed - referred to as s_2 . Differentiating Equation (3-6) leads to:

$$\mathbf{v}_2 = s_2 \mathbf{R}_2 + r_2 \dot{\mathbf{R}}_2 \quad (4-1)$$

$$\dot{\mathbf{R}}_2 = \begin{bmatrix} -\sin(\theta_2) & -\cos(\theta_2) & 0 \\ \cos(\theta_2) & -\sin(\theta_2) & 0 \\ 0 & 0 & 0 \end{bmatrix} \omega_2 \quad (4-2)$$

$$\omega_2 = \frac{d}{dt}(\theta_1 + \phi_1) \quad (4-3)$$

$$\omega_2 = -\frac{1}{\sqrt{\left(1 - \left(\frac{r_1^2 + d_3^2 - r_2^2}{2r_1 d_3}\right)^2\right)}} \quad (4-4)$$

Equations (4-5) – (4-7) show the velocity derivation for point p_4 . Recall that p_3 is a fixed point and thus the velocity at this point is 0. The velocity is found by differentiating Equations (3-7) – (3-9) with respect to time. The time varying component at this point is the direction of the velocity vector, as the angle θ_3 is a fixed design constraint, as is the link length r_3 . The unit vector representing the orientation of the velocity vector of point p_2 is $\dot{\mathbf{u}}_2$.

$$\mathbf{v}_4 = \frac{d}{dt}(\mathbf{p}_3 + R_4 \mathbf{u}_2 r_3) \quad (4-5)$$

$$\mathbf{v}_4 = R_4 \dot{\mathbf{u}}_2 r_3 \quad (4-6)$$

$$\dot{\mathbf{u}}_2 = \frac{\dot{\mathbf{p}}_2}{r_2} = \frac{\mathbf{v}_2}{r_2} \quad (4-7)$$

Ultimately, the velocity of \mathbf{p}_5 is found by differentiating Equations (3-10) – (3-11). The resulting equations are shown below. The two equations must be solved simultaneously for an explicit solution for \mathbf{v}_5 . The Y coordinates of \mathbf{p}_6 and \mathbf{p}_5 are constant, along with the Z coordinates of \mathbf{p}_4 , thus the corresponding velocities are 0.

$$0 = 2(p_{4_x} - p_{5_x})(-v_{5_x}) + 2(p_{4_z} - p_{5_z})(-v_{5_z}) \quad (4-8)$$

$$0 = 2(p_{4_x} - p_{5_x})(v_{4_x} - v_{5_x}) + 2(p_{4_y} - p_{5_y})(v_{4_y}) + 2(p_{4_z} - p_{5_z})(-v_{5_z}) \quad (4-9)$$

4.2 Aft linkage Forward Velocity

Following earlier methods, the aft linkage velocity is calculated similar in manner to the fwd linkage. The velocity of \mathbf{p}_8 is found by differentiating Equation (3-18). The equation below is the result. The input is s_7 , the stroke length of the linear actuator.

$$\mathbf{v}_8 = s_7 \mathbf{R}_7 + r_7 \dot{\mathbf{R}}_7 \quad (4-10)$$

$$\dot{\mathbf{R}}_7 = \begin{bmatrix} -\sin(\theta_7) & -\cos(\theta_7) & 0 \\ \cos(\theta_7) & -\sin(\theta_7) & 0 \\ 0 & 0 & 0 \end{bmatrix} \omega_7 \quad (4-11)$$

$$\omega_7 = \frac{d}{dt}(\phi_2 - \phi_3) \quad (4-12)$$

The time varying component in Equation (4-12) is ϕ_3 . The derivative of θ_7 is ω_7 the solution for which is shown in Equations (4-13) – (4-14).

$$\omega_7 = -\dot{\phi}_3 \quad (4-13)$$

$$\omega_7 = \frac{1}{\sqrt{1 - \left(\frac{d_9^2 + r_7^2 - r_8^2}{2d_9r_7} \right)^2}} \quad (4-14)$$

Point p_9 is fixed meaning the corresponding velocity is 0. The velocity of p_{10} is found by differentiating Equation (3-19). The resultant velocity equation is given below.

$$\mathbf{v}_{10} = R_8 \dot{\mathbf{u}}_8 r_{89} \quad (4-15)$$

The time varying component in Equation (4-15) is \mathbf{u}_8 . The derivative of this is found using the equation below.

$$\dot{\mathbf{u}}_8 = -\frac{\mathbf{v}_8}{r_8} \quad (4-16)$$

Equations (3-20) – (3-21) are differentiated to find the velocity of p_{11} . The result is two equations, given below, that need to be solved simultaneously for the explicit solution of the velocity of p_{11} .

The Y component of the velocity vector for p_{11} is 0 as the motion is in the XZ plane.

$$0 = 2(p_{11x} - p_{10x})(v_{11x} - v_{10x}) + 2(p_{11y} - p_{10y})(-v_{10y}) + 2(p_{11z} - p_{10z})(v_{11z}) \quad (4-17)$$

$$0 = 2(p_{12x} - p_{11x})(-p_{11x}) + 2(p_{12z} - p_{11z})(-p_{11z}) \quad (4-18)$$

4.3 Fwd linkage Inverse Velocity

The inverse velocity of the system must be modeled in order to maintain the required rate of output motion in a synchronous manner. The velocity of p_5 was determined first, by differentiating Equation (3-28). The equations below indicate the velocity for joint p_5 . It is relevant to note here that this velocity vector comprises both the angular velocity and linear velocity of the joint.

$$\dot{R}_6 = \begin{bmatrix} -\sin(\phi_6) & 0 & \cos(\phi_6) \\ 0 & 0 & 0 \\ -\cos(\phi_6) & 0 & -\sin(\phi_6) \end{bmatrix} \omega_6 \quad (4-19)$$

$$\phi_6 = \theta_6 \quad (4-20)$$

$$\mathbf{v}_5 = -\dot{R}_6 \omega_6 (p_6 - q_5) \quad (4-21)$$

The velocity of joint p_4 is next found by differentiating Equations (3-29) – (3-30). The equations below must be solved simultaneously for the velocity of p_4 . The Z velocity of this joint is 0 as the motion is in the XY-plane.

$$0 = 2(p_{5_x} - p_{4_x})(v_{5_x} - v_{4_x}) + 2(p_{5_y} - p_{4_y})(-v_{4_y}) + 2(p_{5_z} - p_{4_z})(v_{5_z}) \quad (4-22)$$

$$0 = 2(p_{3_x} - p_{4_x})(-v_{4_x}) + 2(p_{3_y} - p_{4_y})(-v_{4_y}) \quad (4-23)$$

The velocity of joint p_3 is 0, given that this is a fixed joint. The velocity of joint p_2 is found by differentiating Equations (3-34) – (3-35). The resultant equation is shown below.

$$\mathbf{v}_2 = -R_2 \dot{\mathbf{u}}_4 r_2 \quad (4-24)$$

Only the unit vector \mathbf{u}_4 is changing with time as θ_3 is a constant design constraint, indicating that the magnitude of the velocity of p_2 and p_4 are equal, simply the direction changes with time. This change in direction can be found with $\dot{\mathbf{u}}_4$ as follows:

$$\dot{\mathbf{u}}_4 = -\frac{\mathbf{v}_4}{r_3} \quad (4-25)$$

where \mathbf{v}_4 is determined by simultaneously solving Equations (4-22) and (4-23).

Once the velocity of \mathbf{p}_2 is determined, the linear actuator velocity can be found, by dividing the dot product of the velocity and position vector by the stroke length, and then multiplying this by the unit vector to get the direction and magnitude of the linear velocity.

$$\mathbf{S}_2 = \frac{\mathbf{v}_2 \cdot \mathbf{p}_2}{|\mathbf{p}_2|} \cdot \frac{\mathbf{p}_2}{|\mathbf{p}_2|} \quad (4-26)$$

The actuator stroke speed is determined by then taking the Euclidean norm of the \mathbf{S}_2 vector.

4.4 Aft Linkage Inverse Velocity

Due to the similarity of the two linkages the process of generating the velocity functions for the aft linkage is not much different than the fwd linkage. The velocity equations for each of the joints are determined by differentiating Equations (3-41) – (3-49). Starting with joint \mathbf{p}_{11} the velocity is given below.

$$\dot{\mathbf{R}}_{11} = \begin{bmatrix} -\sin(\phi_{11}) & 0 & \cos(\phi_{11}) \\ 0 & 0 & 0 \\ -\cos(\phi_{11}) & 0 & -\sin(\phi_{11}) \end{bmatrix} \omega_{11} \quad (4-27)$$

$$\phi_{11} = \theta_{11} \quad (4-28)$$

$$\mathbf{v}_{11} = -\dot{\mathbf{R}}_{11} \omega_{11} (\mathbf{p}_{12} - \mathbf{q}_{11}) \quad (4-29)$$

Next, the velocity of joint \mathbf{p}_{10} is established using the derivative of Equations (3-42) – (3-43). The resulting equations are given below. The two equations need to be solved simultaneously as with Equations (4-22) – (4-23).

$$0 = 2(\mathbf{p}_{11x} - \mathbf{p}_{10x})(v_{11x} - v_{10x}) + 2(\mathbf{p}_{11y} - \mathbf{p}_{10y})(-v_{10y}) + 2(\mathbf{p}_{11z} - \mathbf{p}_{10z})(v_{11z}) \quad (4-30)$$

$$0 = 2(\mathbf{p}_{9x} - \mathbf{p}_{10x})(-v_{10x}) + 2(\mathbf{p}_{9y} - \mathbf{p}_{10y})(-v_{10y}) \quad (4-31)$$

Point p_9 is fixed and thus the corresponding velocity is 0. The velocity for point p_8 is found by differentiating Equations (3-47) – (3-49). The resultant velocity equations are found below.

$$\mathbf{v}_8 = -R_8 \dot{u}_9 r_8 \quad (4-32)$$

In this case, similar to joint p_2 , the time varying component is the direction of the velocity vector. This is found by taking the derivative of the unit vector \mathbf{u}_9 .

$$\dot{\mathbf{u}}_9 = -\frac{\mathbf{v}_{10}}{r_9} \quad (4-33)$$

Once the velocity of p_8 is determined, the linear component of the velocity is found with the equation below, similar to the fwd linkage case, this provides the actuator linear velocity. The Euclidean norm can be taken of the \mathbf{s}_8 vector to determine the stroke speed.

$$\mathbf{s}_8 = \frac{\mathbf{v}_8 \cdot \mathbf{p}_8}{|\mathbf{p}_8|} \cdot \frac{\mathbf{p}_8}{|\mathbf{p}_8|} \quad (4-34)$$

4.5 Velocity Synchronization

The velocity solutions can now be used to synchronize the motion of both the linkages. This is necessary to prevent clashing of the two chains. Additionally, this ensures the output motion is equal for both chains, such that there is no actuator pushback or twisting of the platform. The resultant actuator velocities are plotted as a function of winglet interface angle in Figure 4-1 and Figure 4-2. The synchronization is achieved by defining the input angular speed, thus ensuring the velocities of p_{11} and p_5 remain equal. This was checked by monitoring \mathbf{v}_{11} and \mathbf{v}_5 . The corresponding script can be found the Appendix [A]. As can be seen in Figure 4-1 the velocity increases instantly between the 45°-46° increment. This is due to the results being the theoretical instantaneous velocity at each position. In order to accommodate the realistic operation of this system velocity tuning would be required. This is addressed in Section 5.5.1.

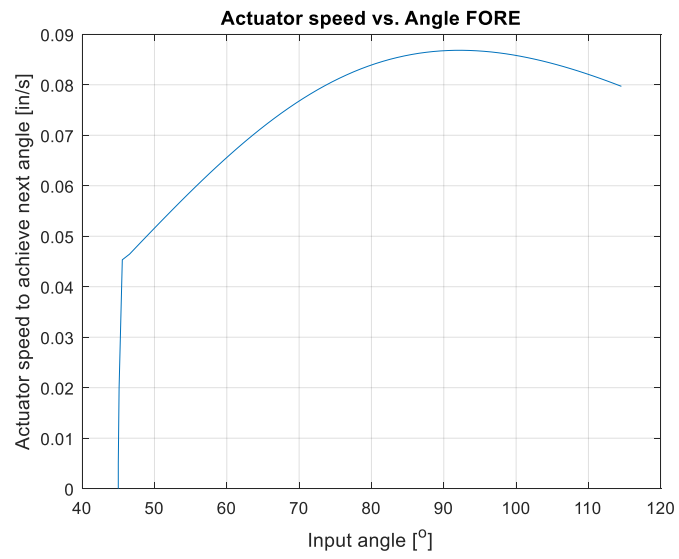


Figure 4-1: Fwd linkage actuator speed vs. Input angle

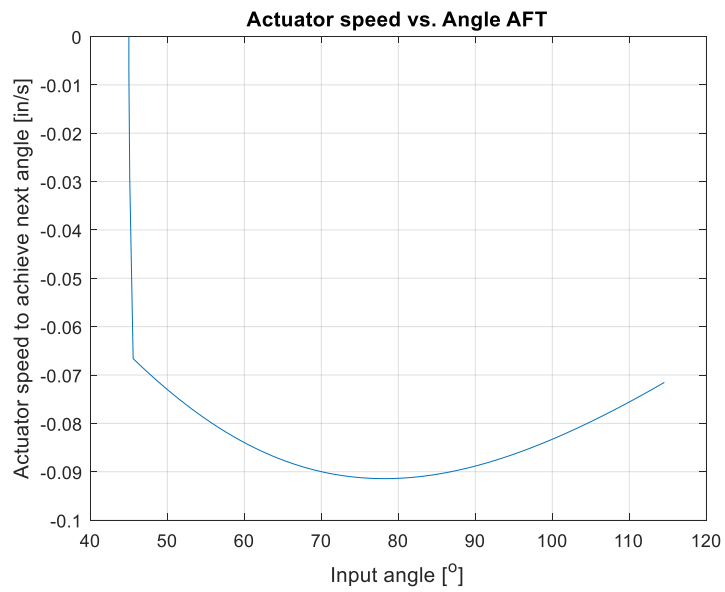


Figure 4-2: Aft Linkage Actuator speed vs. Input angle

CHAPTER 5 Implementation

The stages of implementation involve theoretical testing of the system via simulation, design of the controller, prototyping of scaled test model, preliminary testing, and full-scale prototyping and testing. The scope of this thesis covers the implementation from the simulation phase to the preliminary test phase. The sections below cover this process in detail.

5.1 Simulation

The simulation of the system was conducted for the forward kinematics, inverse kinematics, and inverse velocity relations. The forward kinematics simulation was conducted to determine the system workspace. This stage is outlined in detail in Section 3.6. The simulation of the inverse kinematics provided the stroke lengths required to attain the desired target winglet angles. This process is outlined in Section 3.8. These results were used for the position control component of the controller design. The inverse velocity simulation provided the relation between a target winglet angular rate, and the required linear velocity of the actuator. These results were then used for the velocity control component of the system. The detailed process is shown in Section 4.5. Further the control optimized velocity relation is used to reduce the error seen in the test system. The optimization process is given in detail in Section 5.5.1. The corresponding scripts for each stage of simulation are given in the Appendix. Using these simulations, the theoretical behaviour of the system was analysed, and the synchronization requirements established, allowing the testing phase to proceed to the development of a controller.

5.2 Control system set-up

The preliminary controller was designed, by Gabriel H Campos, for position control of the actuator stroke length. The initial design was based on the Inverse Kinematics results. The final

version of the controller included the velocity control components, as well as the IK results, and angle feedback via a rotary potentiometer. This controller is shown in Figure 5-1. The controller uses a ‘master-slave’ relationship to ensure synchronicity of the two sides of the system. It is vital to the operation of the system for the two actuators to remain in synchronous operation to prevent fighting between the two linkages, to this end the forces of the actuators are monitored as a component to the feedback of the system. The primary feedback for the system is a rotary potentiometer mounted on the winglet pivot axis, used to return the winglet angle for system feedback.

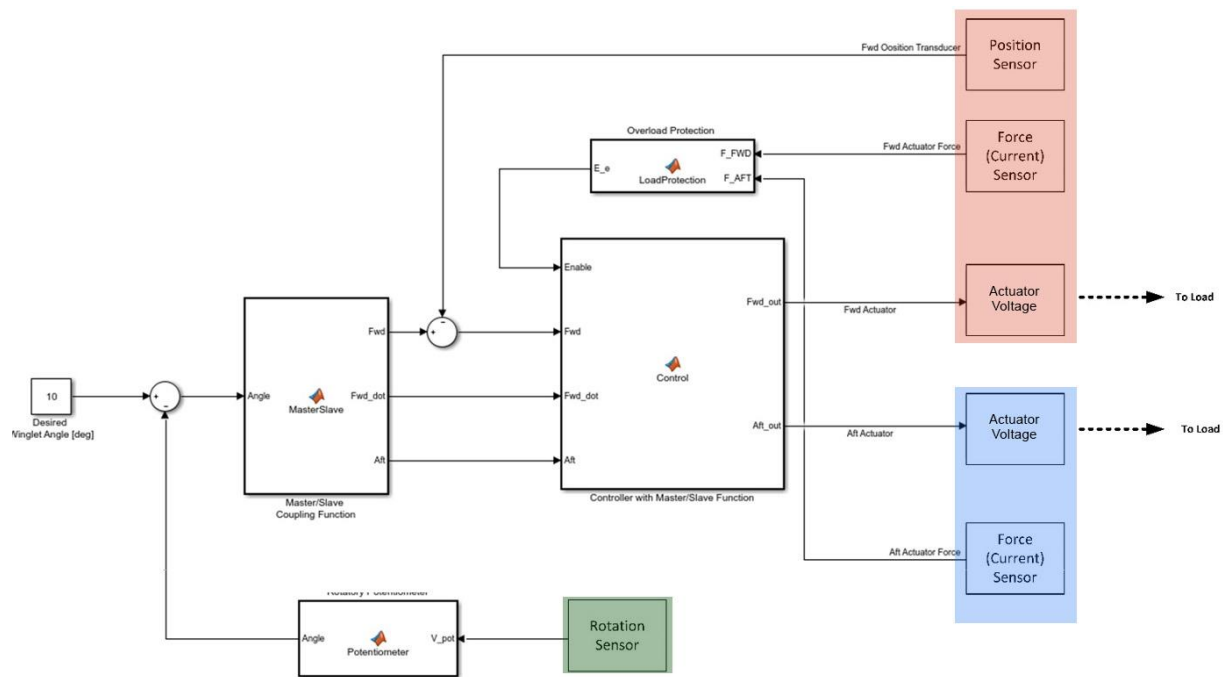


Figure 5-1: Final Controller design

5.3 Design Prototyping

The prototype was created by scaling the design to 50%, and 3D printing the components. The Markforged Mark Two 3D printer was used, with carbon fibre reinforced ONYX. The parts were mounted on a frame of extruded aluminum spars. The winglet attachment piece was printed, along with a foam scale replica of the winglet. The image below shows the test setup and printed prototype.

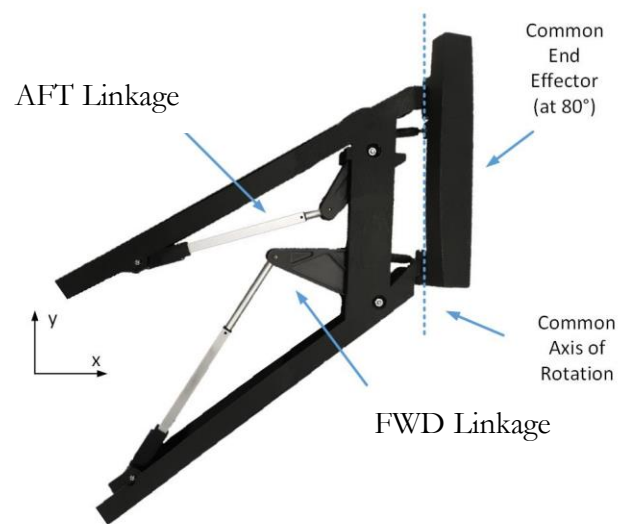


Figure 5-2: Prototype Image

5.4 Implementation Issues

There were issues that arose during the implementation of the system. The preliminary model was created by 3D printing the CAD design. The initial design had to be scaled to 50% to account for the size of the test actuators. This scaling and printing required minor modification of the design, while maintaining the position of the key points of the mechanism, to ensure consistency in the linkage behavior. It was further noticed at this stage that the original CAD design contained clashing components these were modified to prevent issues during operation. The axes had to be aligned and component structure adjusted prior to printing. The image of the printed prototype is shown in

Figure 5-2. Due to these inaccuracies, and the need for force optimization for the system to account for aerodynamic loads, an optimized design was developed. The control results and implementation details are given in the next section for the optimized prototype.

Furthermore, the complexity of the IK equations proved difficult to compute online on the current system hardware. As such, two proposed solutions were evaluated, the first option being, to store the required stroke lengths and velocities for certain key angles – in a lookup table, along with a polynomial interpolation function for intermediary points. However, this would be inefficient from a data storage point of view. An alternative approach involves fitting a quadratic curve to the relation between stroke length and target winglet angle, and a cubic polynomial for the relationship between stroke velocity and target winglet angle. In this manner the simplified curve can be stored, and the required velocities and stroke lengths can be determined online. As such this second method was selected for the controller. The general algorithm for this approach is shown in Figure 5-3 below. The specific curve fitting results are given in the next section with the optimized prototype details.

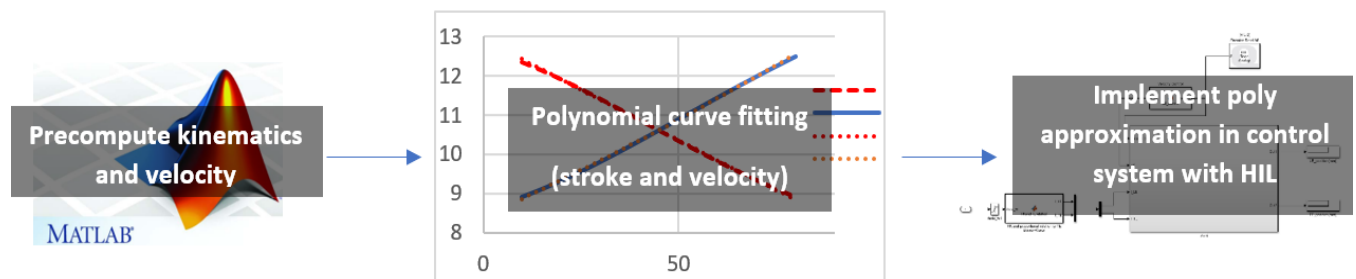


Figure 5-3: General controller design algorithm

The following was the input used for the controller in the initial prototype, found by fitting a cubic curve to the results of the velocity simulation:

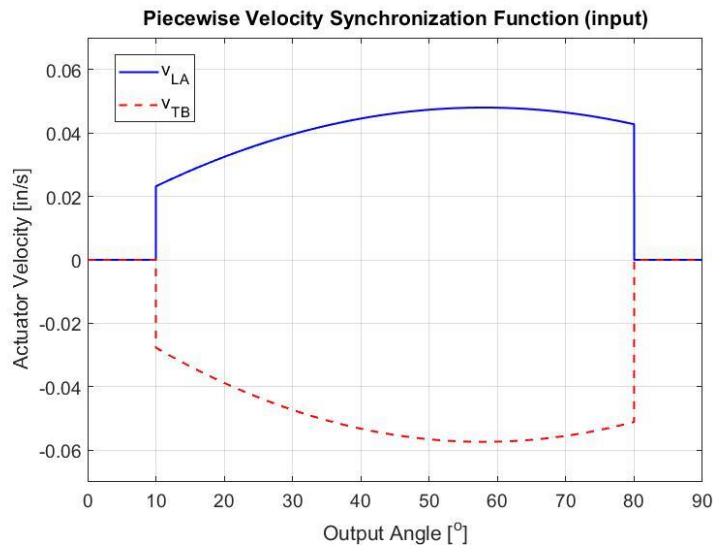


Figure 5-4: Controller input

The figure below shows the resulting actuator lengths as a function of output angle, found by fitting a quadratic curve to the results of the IK simulation:

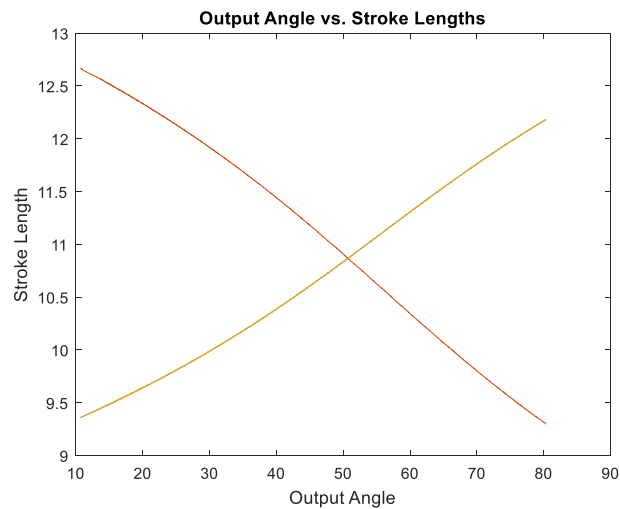


Figure 5-5: Actuator Lengths vs. Time

5.4.1 Error Evaluation

The first prototype was operated using the controller, and the Velocity and IK results. The test results are summarized in the table below. There was a considerable error noticed in this initial prototype. This is likely due to the modifications made, as well as inexperience with the operation of the new printer, resulting in larger than required tolerances allowing for play in the joints. This was done as a precaution to prevent the tolerances from being excessively low, increasing joint friction and contributing to impact forces on the actuators due to unexpected behavior at the joints. The figures below show the error after calibration, as can be seen the actual system lags initially, resulting in the output angle being higher/lower than required. This lag is particularly prevalent with increasing angles as opposed to decreasing. This is likely due to the weight of winglet, and components of the system allowing for greater bending due to inaccurate fits. Due to the stiffness of the printer material, the large tolerances allowed a high amount of bending and twisting in the system.

5.4.1.1 Calibration

Due to the discrepancies in the preliminary design prototype there was calibration required to operate the mechanism as specified. This calibration was carried out by measuring the winglet angle via a precision optical digitizer. The process involved using the 3D digitizer target to set points for setting up a reference frame, the points were measured from the prototype surface shown in Figure 5-6. With the reference plane defined, the axis of rotation was specified between the two points shown in Figure 5-7. The rotation arc was then defined by measuring the center of the winglet interface at 3 key positions. This arc is shown in Figure 5-8. Using this arc, and the reference plane the 0° angle reference was set. Once the reference position is determined, this is used to determine the actual prototype angle, at 10° increments. Once the measured data was compared to the

theoretical target, the error was used to modify the controller to adjust for the offsets. The results of this calibration are shown in Figure 5-9 and Figure 5-10. [24]



Figure 5-6: Reference plane for calibration

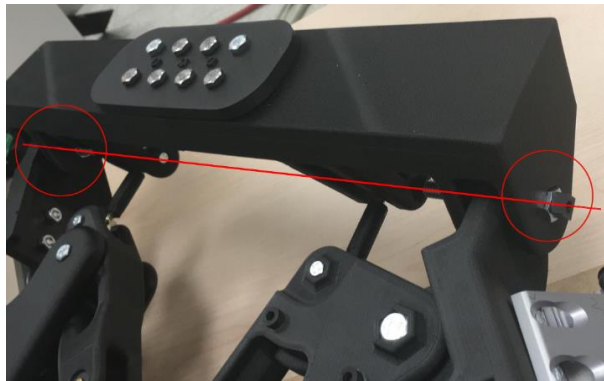


Figure 5-7: Points used to generate rotation axis

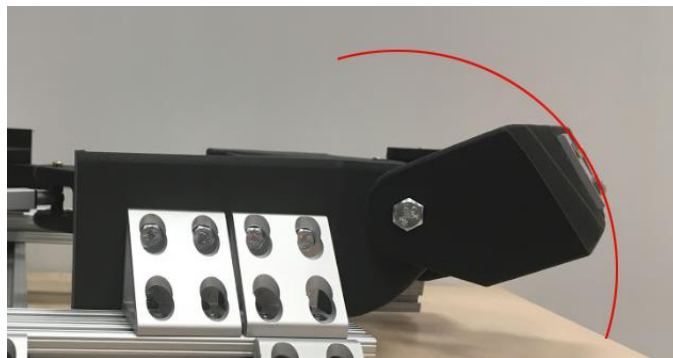


Figure 5-8: Rotation arc definition

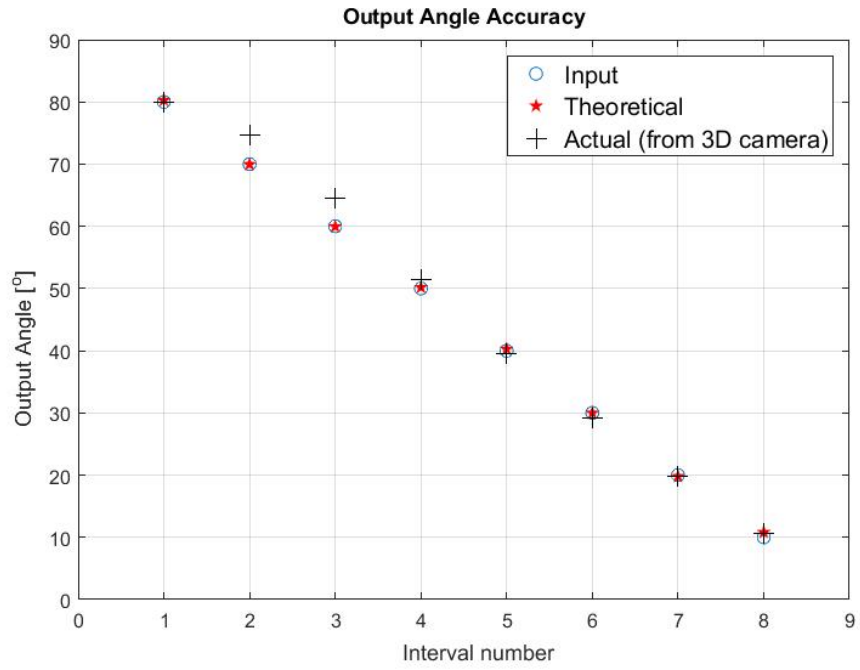


Figure 5-9: System error for decreasing winglet cant angles

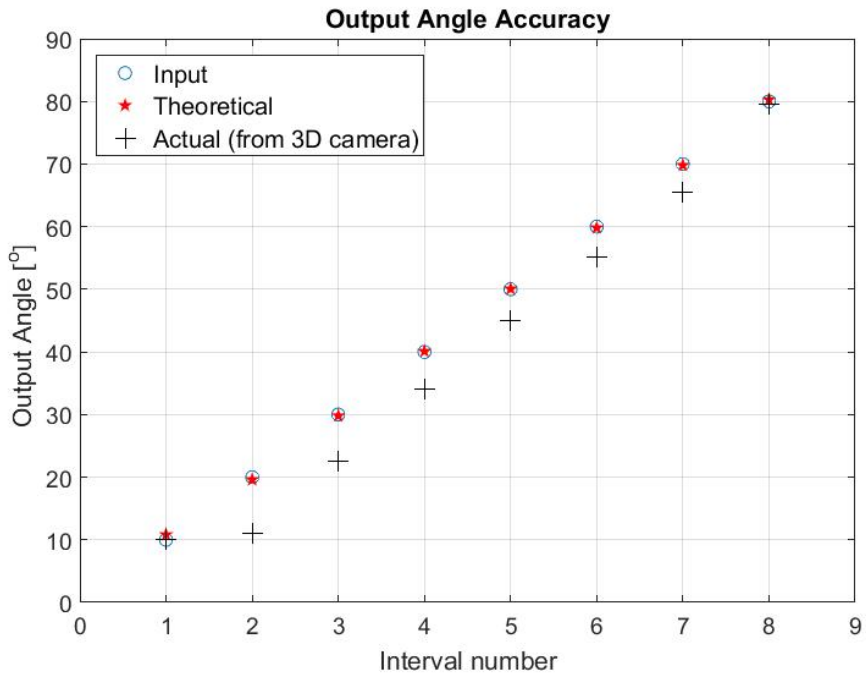


Figure 5-10: System error for increase cant angles

5.5 Optimized Design

The optimized design was developed to maximize the load bearing capabilities of the system and eliminate the structural design issues of the initial design. The optimized system design is shown in Figure 5-11 and Figure 5-12, the major changes to the design involved reposition of the frame points and resizing of the links. As a result of some structural changes, the simulation of the system kinematics and velocity had to be modified to account for the spatial nature of the system. While the mechanism sizing and positioning was modified, the kinematic and velocity equations remain unchanged. The script provided in the Appendix is the modified simulation script for the optimized system. The results of the inverse kinematics and inverse velocity simulations are summarized in the figures below.

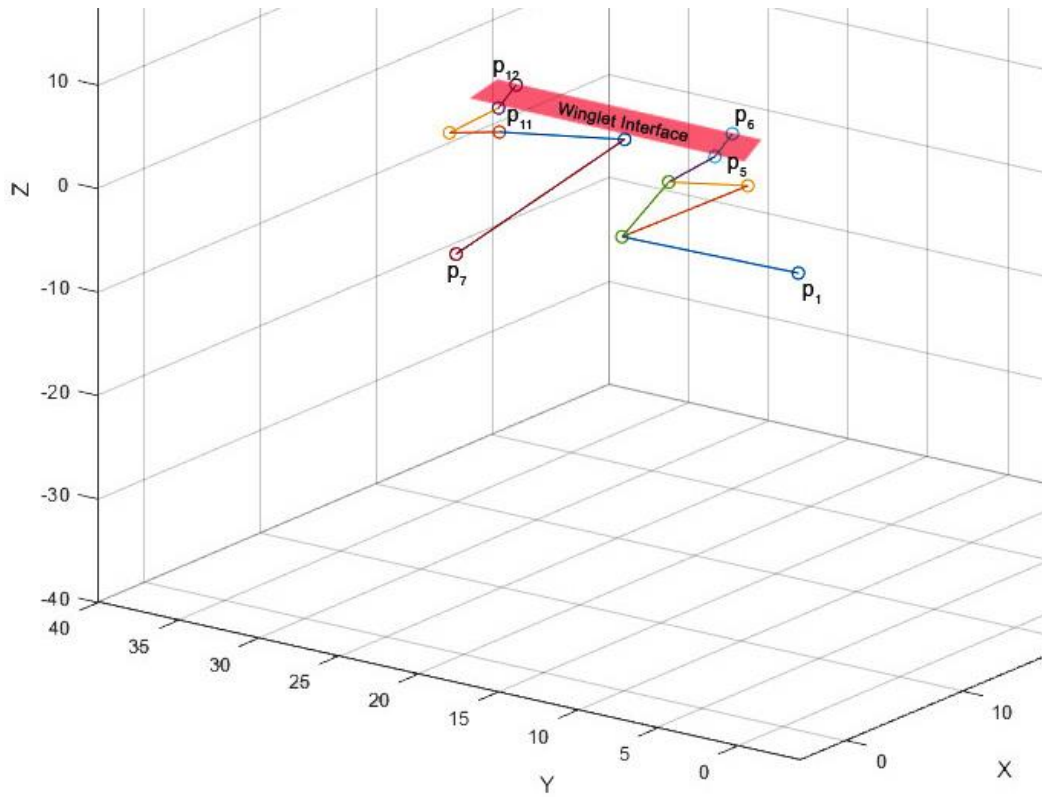


Figure 5-11: Optimized design simulation at 80° configuration

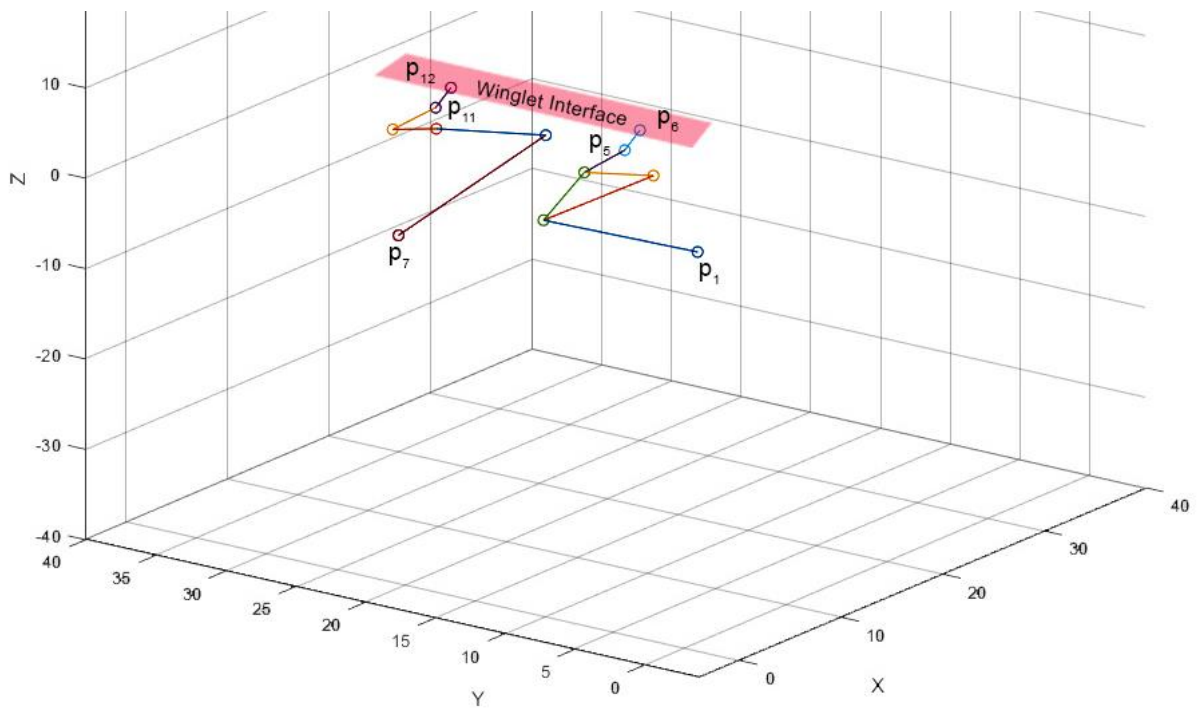


Figure 5-12: Optimized design simulation at 10° configuration

The following figures show the results of the simulations with the optimized design. Figure 5-14 shows the stroke lengths corresponding to each target angle. It is clearly shown that the two linkages operate inversely to one another. Here it is shown that the stroke length displacement over the full operational range for the fwd linkage is 5.97 inches. The stroke length displacement for the Aft linkage is -5.76 inches. As such the range of motion of the two sides is not entirely equal. The intersection of the two plots occurring between 80° - 81° . As can be seen from Figure 5-13 and Figure 5-14 there is a minor error in the results of the velocity analysis. This is likely due to computational error. The results shown in Figure 5-18 and Figure 5-19 are the instantaneous velocity results to the inverse velocity formulations. The results from these curves are not ideal for real operation of the system, and the design of a controller. In order to establish a velocity curve better suited to real applications motion planning using the acceleration of the actuators is required. Due to the limitations of the prototype equipment, this is not possible with the test model. As such Section 5.5.1 outlines a rudimentary method to accomplish this.

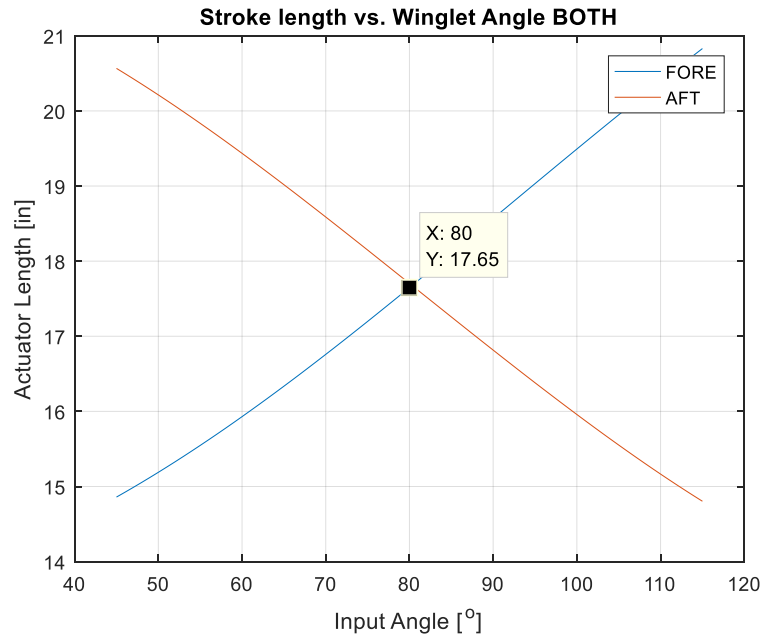


Figure 5-13: IK results for optimized design

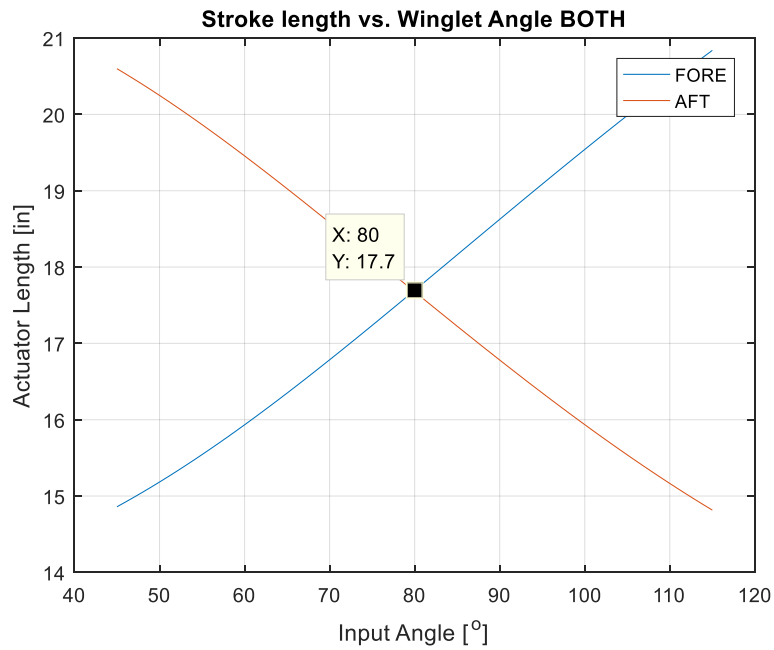


Figure 5-14: Stroke length vs. Target angle for inverse velocity results

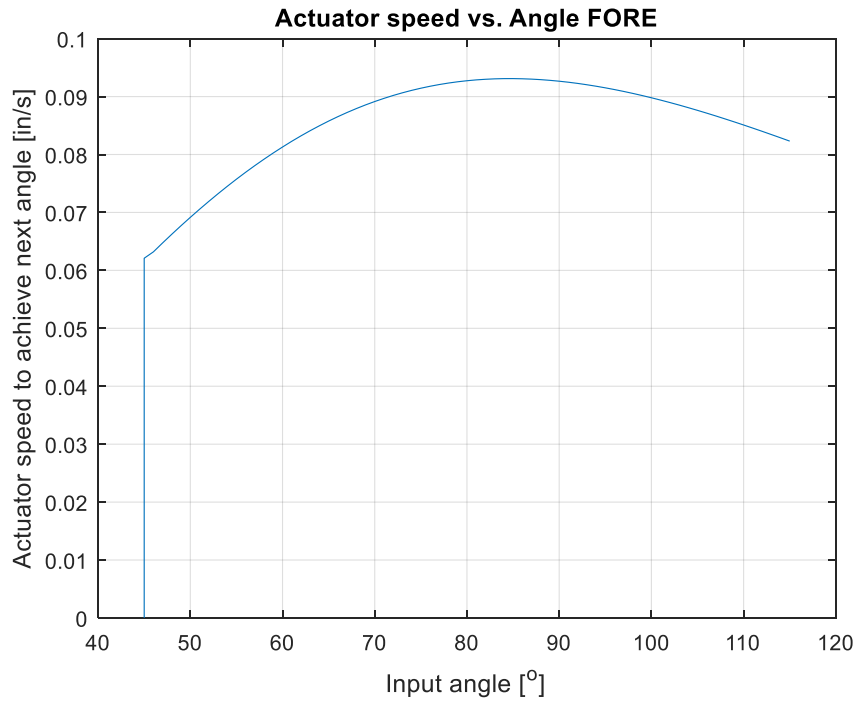


Figure 5-15: Fwd linkage actuator speed vs. current winglet angle

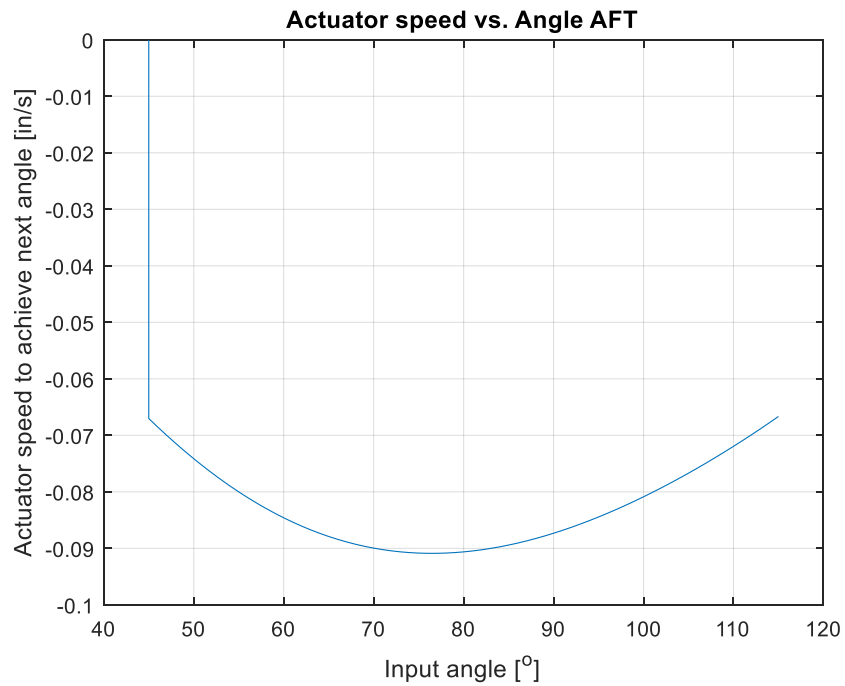


Figure 5-16: Aft linkage actuator speed vs. current winglet angle

5.5.1 Velocity Tuning for Control

The error in the test prototype was affected by the lack of a ramped transition to the target velocity being included in the velocity profile. The velocity response of the system was analysed to determine the capability of the mechanism. The error in the velocity results was determined by comparing the total stroke displacement (10° - 80°) to the IK results. It was observed, as can be seen in Figure 5-17, that higher target angular velocities resulted in greater error in the system. The target angular velocity was varied as a function of time to generate an actuator velocity profile better suited for the design of a controller. Due to the lack of information regarding the actuators used in the test prototype, this presented a unique challenge. The traditional method of adjusting a ramp according to the acceleration capability of the actuators was not applicable for the test model. As such an alternate method was selected. This involved setting a ‘delay’ time, over which the system would increase the velocity until reaching the target output velocity. This was done by exponentially increasing the input angular velocity over the “delay” time. This delay time was varied to assess the effect on the system error. In order to determine the optimal angular velocity, and delay, the MATLAB multi-objective optimization functions were evaluated. The goal attainment function was determined to be best suited to this type problem. Due to the complexity of the velocity equations the computation times were larger than expected. It was observed that the selected function provided an optimal delay time, yet the angular rate was not significantly modified from the initial guess parameters. This is likely indicative of further study being required for the system. The parameters selected for the goal attainment optimization algorithm are given in Table 5-1. As such, the initial guess for angular rate was manually varied within a narrow range, from $0.1^{\circ}/s$ to $5^{\circ}/s$, the results of these preliminary tests are summarized below. It was noticed that the delay factor optimal value consistently remained near 2.8 seconds. Given the realistic constraints of the hardware, the delay was selected as 2.8 seconds without considering the microsecond variations per angular rate.

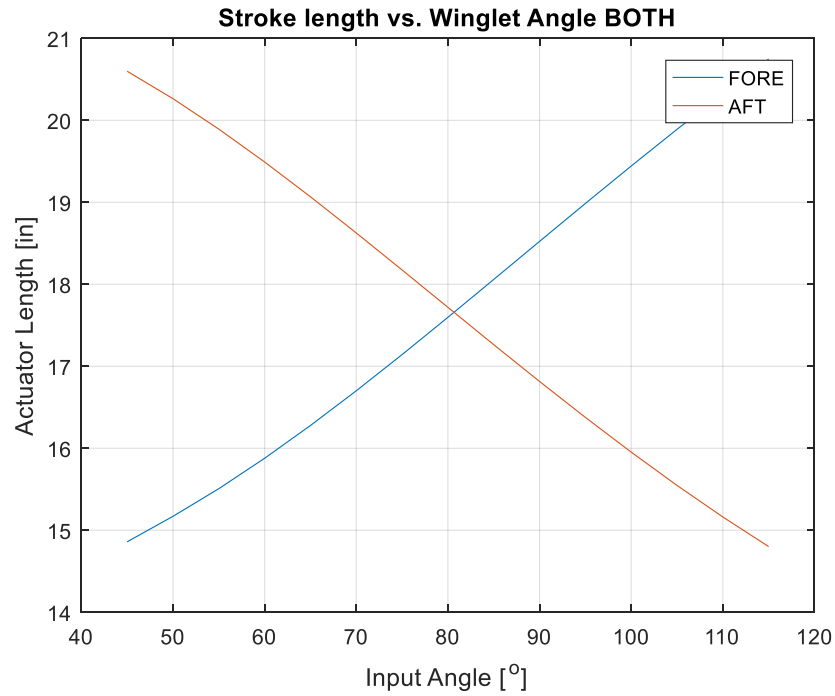


Figure 5-17: Stroke length vs. winglet angle output for $\omega = 5^\circ/s$

Table 5-1: Goal Attainment Initializing Parameters

Parameter	Value
Goal (target error for fwd linkage and aft linkage)	$\leq 0.1\%$
Lower bound Angular velocity	$0.1^\circ/s$
Lower bound Delay factor	1 s (corresponds to instantaneous increase no ramp)
Upper bound Angular velocity	$10^\circ/s$
Upper bound Delay factor	5 s (any longer was considered unreasonable)
Maximum Iterations	30
Finite Difference Step Size	0.1

Table 5-2: Delay Results for various initial inputs

Initial Guess [angular rate, delay time]	Final Optimized Value [angular rate, delay time]	Error [Fwd, Aft]
[0.5,1]	[0.500000008887608,2.857418539447498]	[0.0745,0.0464]
[0.1,1]	[0.100000030638924,2.857127485719390]	[0.0548,0.0298]
[10,2]	[10.000000035274110,2.857420861231254]	[0.0746,0.0465]

In spite of lower angular rates providing better error results, the target angular rate was maintained at $1^\circ/\text{s}$ due to the project requirements. The delay was set to 2.8 seconds and the following results were obtained from the simulation.

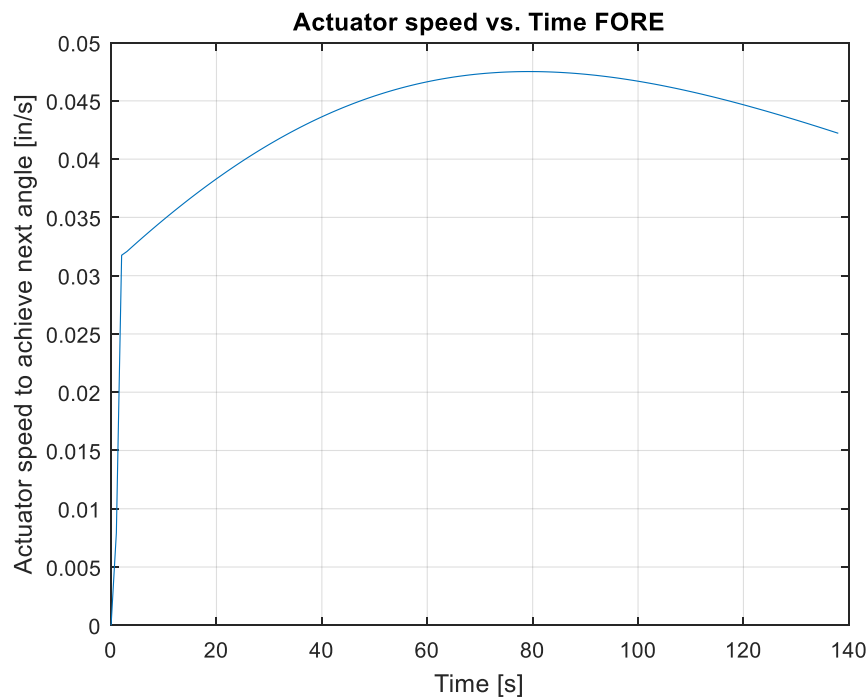


Figure 5-18: Fwd linkage Actuator speed vs Time with delay of 2.8 seconds

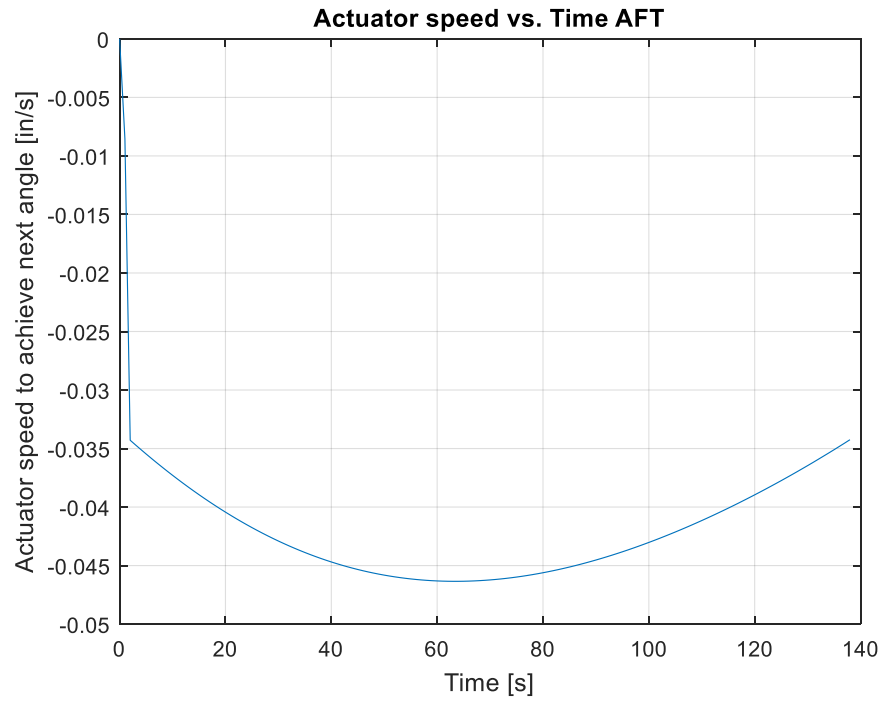


Figure 5-19: Aft linkage Actuator speed vs Time with delay of 2.8 seconds

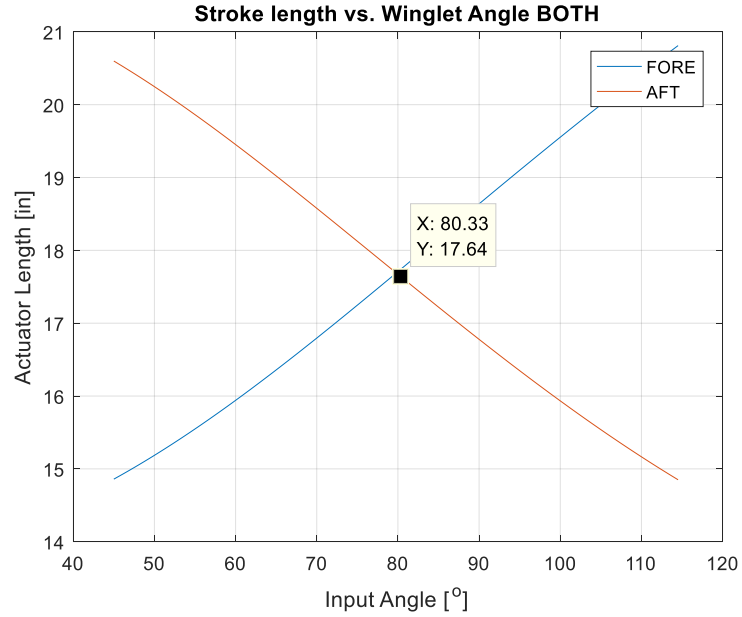


Figure 5-20: Stroke length vs. Input angle with delay

It is relevant at this stage to discuss the justification for approaching the tuning and system capability study from the inverse velocity perspective. The forward velocity profile of the system could be analysed and used in tuning, however, due to the requirement for synchronicity in the system this is a significantly involved process and could result in minor misalignments of the two linkage outputs. As such it was determined that tuning was best conducted on a synchronized system using the inverse velocity, to ensure aligned motion of the outputs, preventing any input related force fighting in the actuators.

5.5.2 Model Set-up

The new prototype was printed at a 60% scale, due to the finalized version of the design no significant modifications were required. The prototype was again printed in ONYX for the custom components. Similar actuators were used as for the first prototype, along with the aluminum frames for mounting. The winglet itself was also the same component as for the initial prototype. The images below show the test set-up for the optimized prototype.

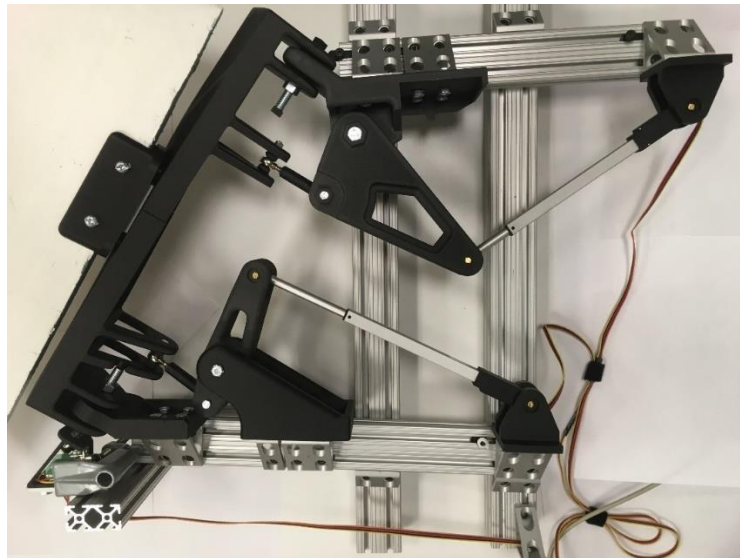


Figure 5-21: Prototype test setup Top view



Figure 5-22: Prototype test setup with Winglet at 10°



Figure 5-23: Prototype test setup with Winglet at 80°

The figures below show the polynomial and quadratic fit used in the controller. The final controller is shown in Figure 5-26.

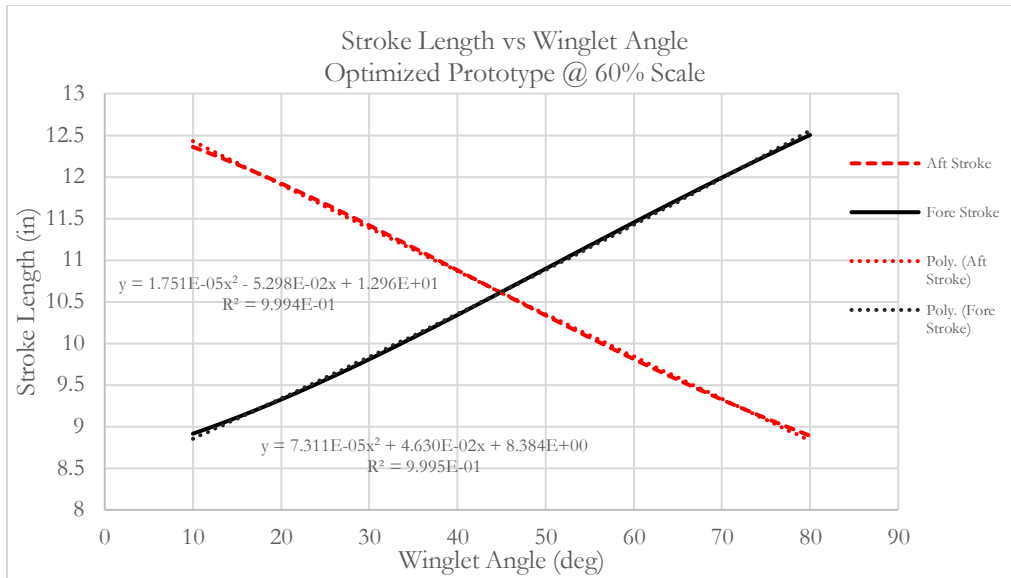


Figure 5-24: Polynomial fit used for estimation of IK relationship for input to controller

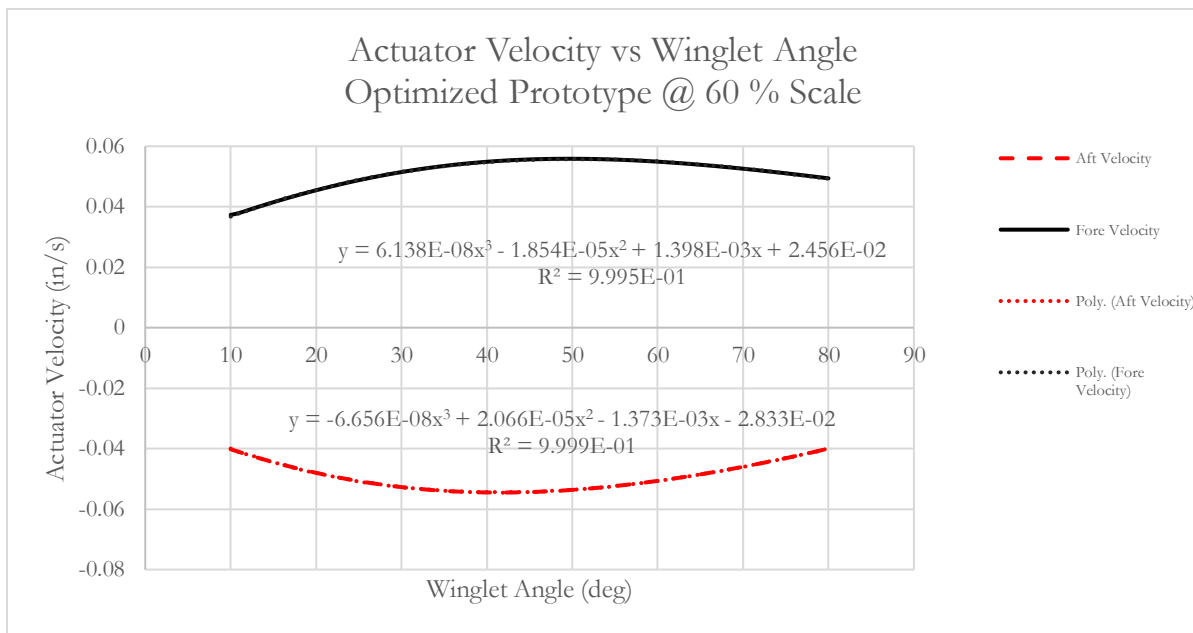


Figure 5-25: Quadratic fit used for Inverse Velocity relation estimation for input to the controller

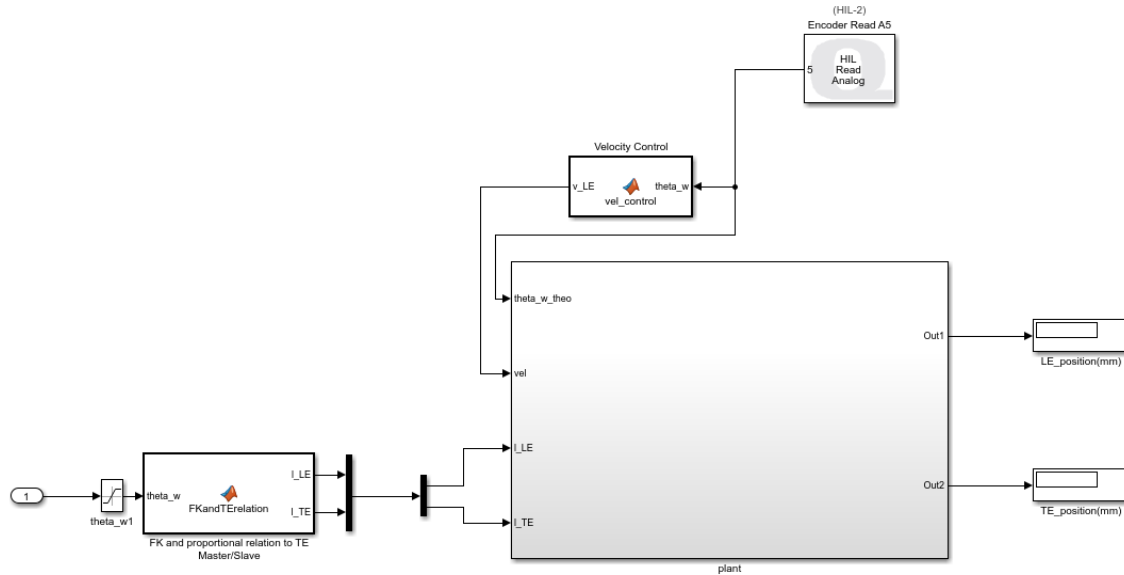


Figure 5-26: Final controller design

5.5.3 Error Analysis

The results of the optimized model are shown in Figure 5-27 to Figure 5-36. Aside from the standard causes of error in the physical implementation of systems, i.e. minor parasitic motion in components, effects of friction forces and unaccounted forces in the system itself, there was an overshoot in the stroke length and target angle. This overshoot was likely a response to the instantaneous velocity increase in the actuators to meet the target angular velocity. Given that the actuators cannot instantaneously match the required velocity, there is additional unaccounted for motion while the actuators accelerate from static to the required velocity. To ascertain whether this was indeed the source of the error the ramped optimized velocity profile was tested with the prototype. The results showed an elimination of the overshoot, a match of the target angles, and target angular rate after the ramp delay. These results are summarized in the figures below.

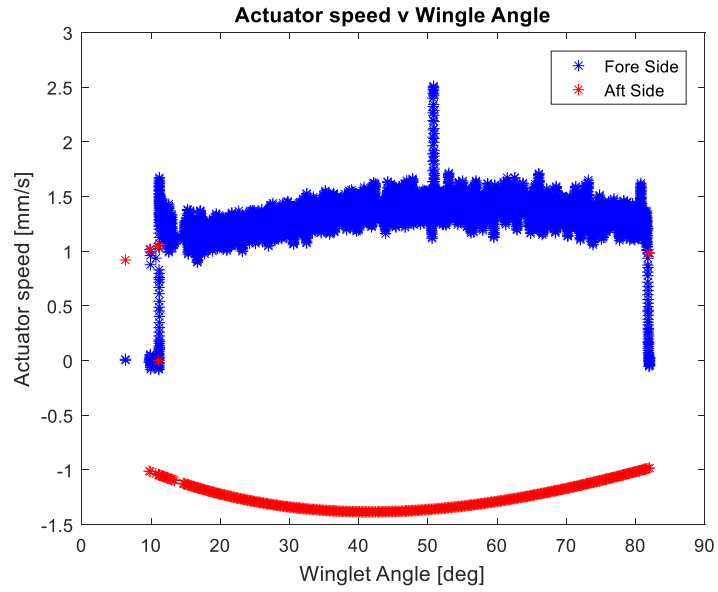


Figure 5-27: Actuator speed vs. winglet angle without ramp up

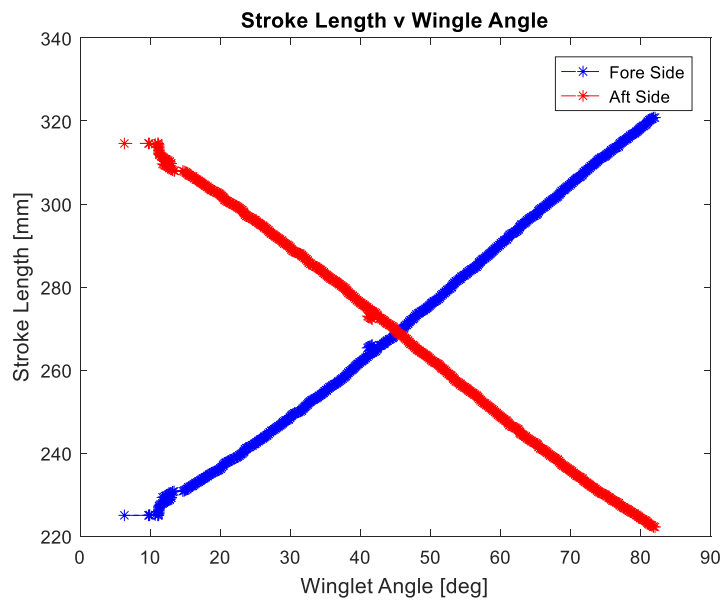


Figure 5-28: Stroke length vs. Winglet angle without ramp up

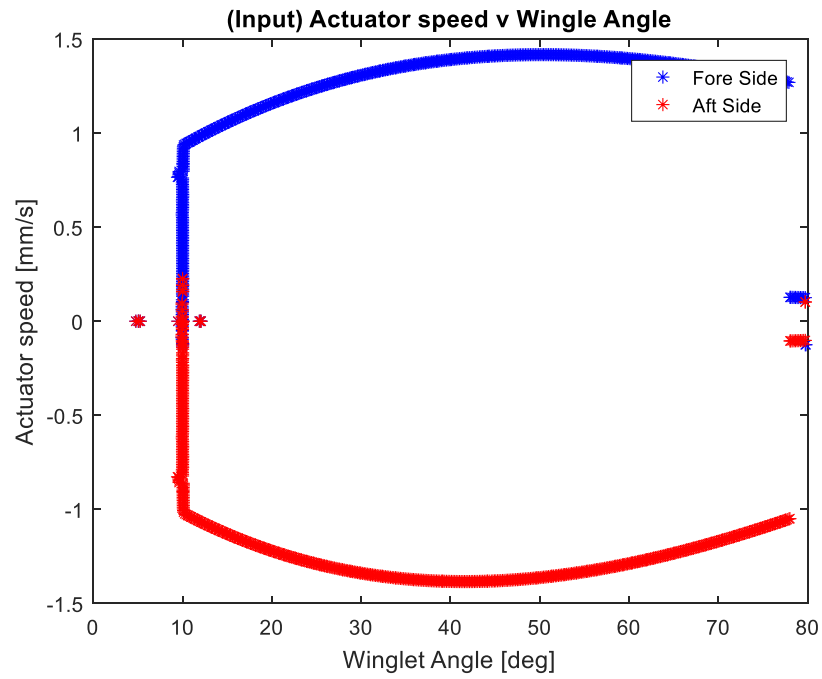


Figure 5-29: System input for case with ramp up to target angular rate

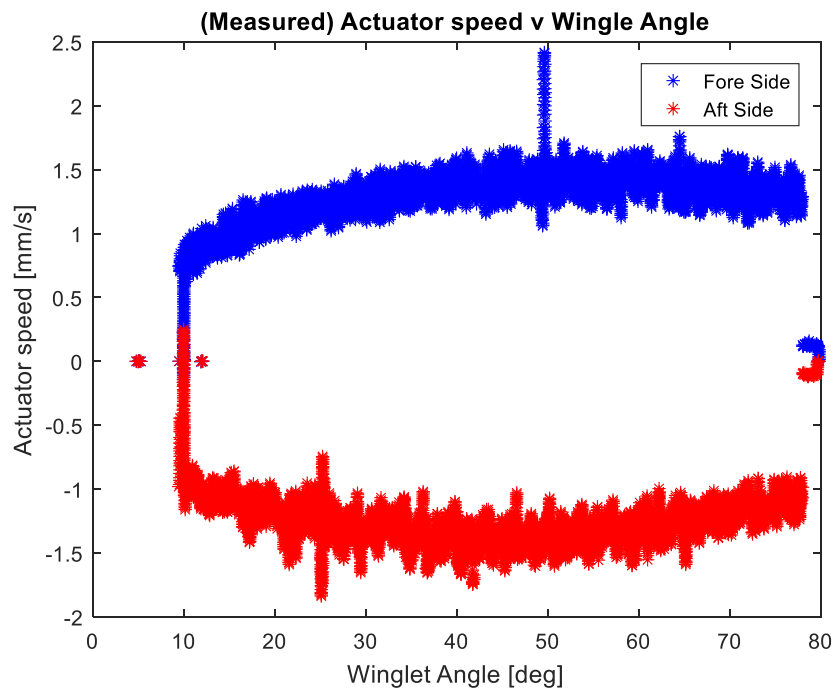


Figure 5-30: System response to ramped case

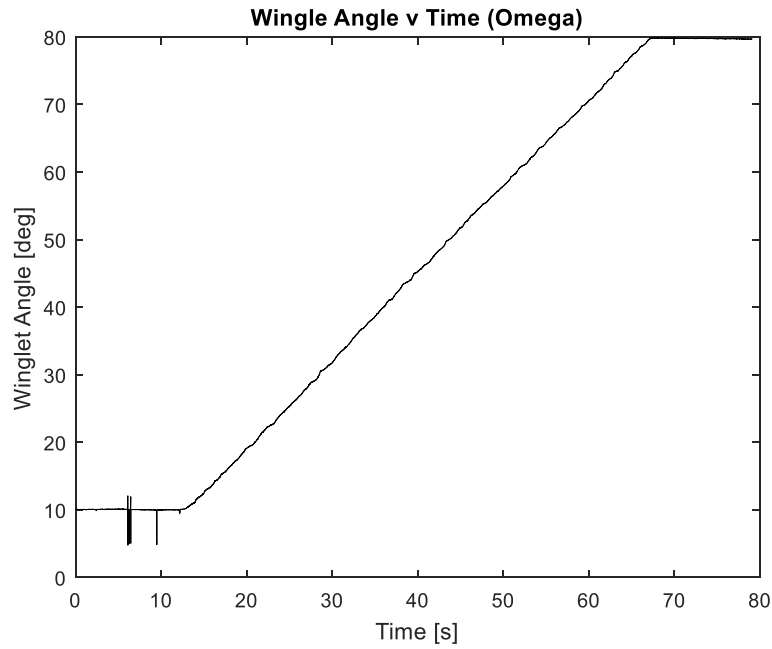


Figure 5-31: Resultant angular velocity of wingle for ramped case

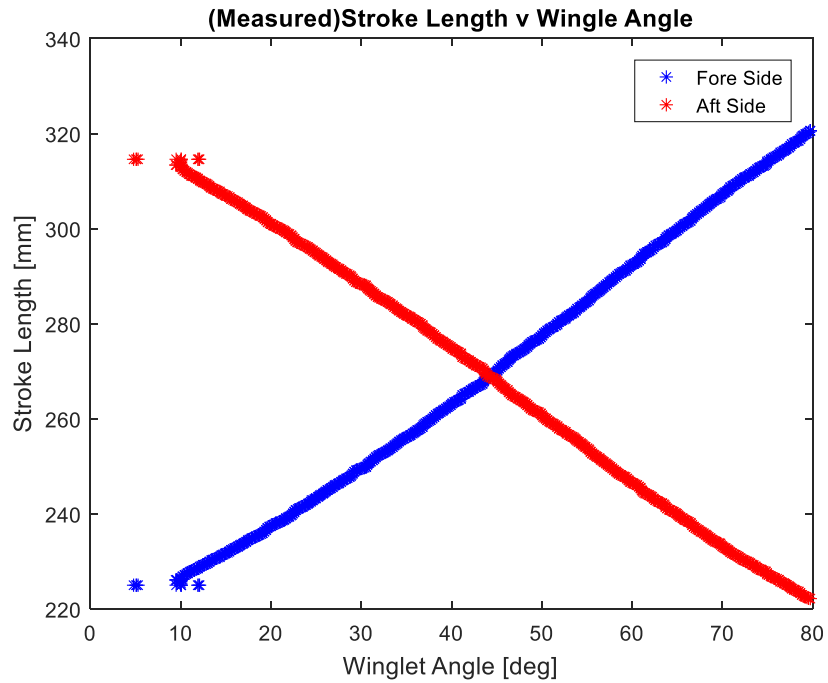


Figure 5-32: Stroke length vs. Wingle angle ramped case

Further the error after mounting the winglet was evaluated. The results indicated an overshoot of target angle when the winglet was at 10° and an underachievement when at 80° . Upon observation of the system this was due to the winglet weight creating a moment about the mount and affecting the output angle. This effect is likely compounded in the prototype due to the manufacturing material being prone to bending and twisting effects. It is relevant to consider this effect however, as inflight aerodynamic impacts on the winglet would certainly affect the motion of the mechanism. The overshoot/undershoot issue would be a minor response, given that twisting and imbalanced aerodynamic forces could impact the synchronicity of the system. The controller has been designed to account for some of these effects in the form of force feedback. It is vital that a detailed force analysis of the system be conducted, as well as the system response to various loads. The results of the winglet loaded system are given in the figures below.

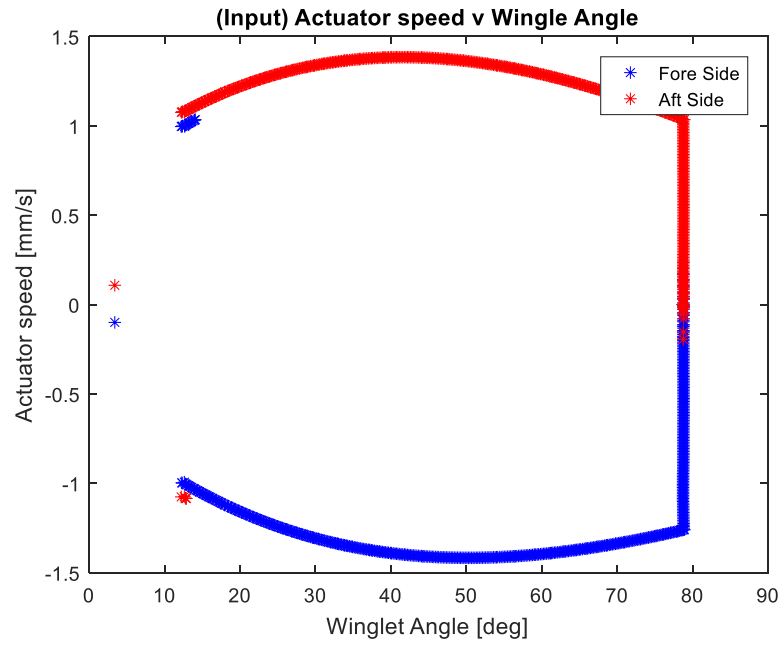


Figure 5-33: System input with Winglet load

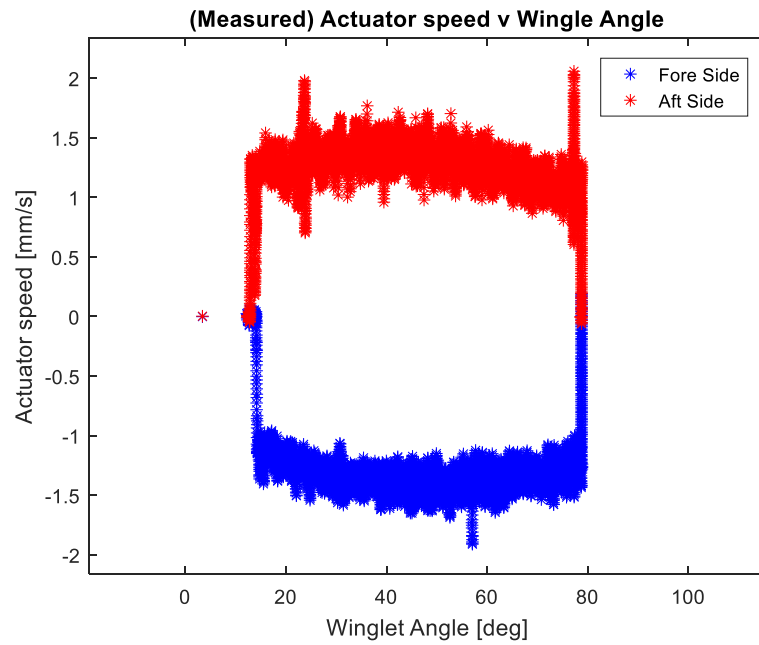


Figure 5-34: System response with Winglet load

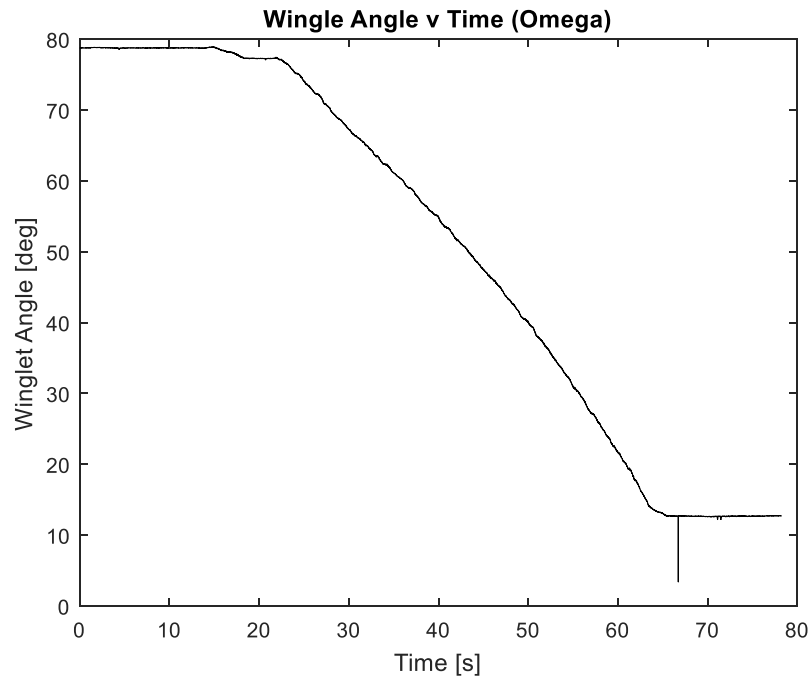


Figure 5-35: Output Wingle angular velocity with Wingle load

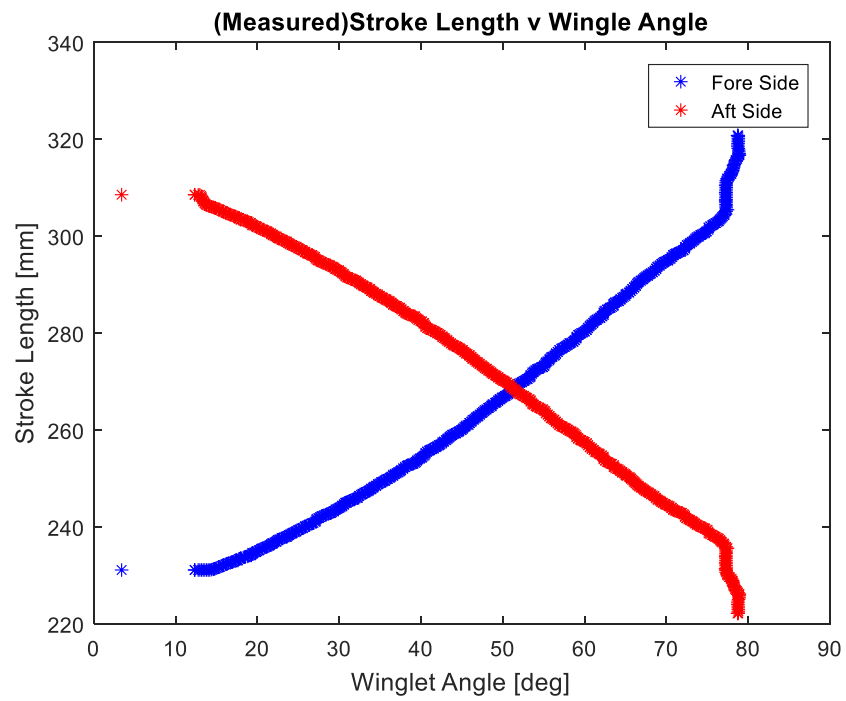


Figure 5-36: Stroke length vs. Wingle angle for loaded case

CHAPTER 6 Conclusion

6.1 Contributions

This thesis provided the analysis and results required to progress the modeling of a proposed morphing winglet design from the design concept stage to prototype development and testing. The system was identified as a redundantly actuated parallel mechanism due to the secondary actuator required for the force constraints of the system. This results in actuator redundancy for a 1DOF system. The kinematic and dynamic modeling of this mechanism was conducted. These results were used to validate the use of the design in a morphing winglet project with strictly defined operational requirements. The workspace of the mechanism was evaluated, and used to confirm design constraints, as well as establish the capability of the design as per the proposed starting configuration. The inverse kinematic relations were used to determine the required stroke lengths and stroke range to allow the system to meet the design criteria. The velocity relations of the system were developed, and the inverse velocity models used to ensure synchronous operation of the mechanism linkages. The system motion was simulated to ensure desired operation. The design was prototyped and tested by use of a velocity controller designed by a project member as per the operational requirements, using the results of the inverse kinematic and velocity simulations. The large discrepancy between the theoretical and experimental data was determined to be caused by the design issues discussed in Section 5.4. The motion of the optimized model was validated via simulation using the IK and inverse velocity relations developed. The velocity profile was further tuned for ideal input to the control in the prototype phase. The prototype was developed and tested against both the standard velocity profile and the tuned case. The improvement in performance in the tuned version was considerable and warrants implementation of the optimized profile.

6.2 Future Work

This thesis encompasses the preliminary design stages of a mechanism development. Full scale prototyping and testing, as well as further analysis of the system remain to be conducted. The velocity profile of the system can be tuned further for the experimental case, using true motion planning methods. The force interactions of the system should be evaluated, along with the effect of various loads on the mechanism motion. Additionally, the optimization of the system can continue in the form a linkage synthesis to eliminate experimental errors. Furthermore, the accuracy of the prototype can be increased by increasing the density of the material used. The current printer density was set to 40%, increasing this would increase the stiffness of the components, reducing bending and twisting effects.

Appendix A

Forward Kinematics

```
clear;
clc;
format long

p{1} = [0;0;0];
p{2} = [12.0055;14.5559;0];
p{3} = [22.2887;16.6786;0];
p{4} = [20.084;20.2534;-0.3787];
p{5} = [24.7385;20.7213;-0.3191];
p{6} = [26.2385;19.9713;1.1809];

p_2_final = [12.5893;20.7;0];

l_1_min = norm(p{2});
l_1_max = norm(p_2_final);

l_2 = norm(p{3}-p{2});
l_3 = norm(p{4}-p{3});
l_4 = norm(p{4}-p{2});
l_5 = norm(p{5}-p{4});
l_6 = norm(p{6}-p{5});
d_1 = norm(p{3});
l_1 = l_1_min;

theta_3 = 70.08;
p_4_z = p{4}(3);

increment = (l_1_max-l_1_min)/100;
actuator_length = [];
cangle = [];
pd = zeros(3,100);

%side 2
p{7} = [0.0746;19.6652;0];
p{8} = p{7}+[24.0096;5.0881;0];
p{9} = p{7}+[21.2221;10.862;0];
p{10} = p{7}+[19.9996;13.1877;-0.3787];
p{11} = p{7}+[24.6639;13.544;-0.3191];
p{12} = p{7}+[26.1639;16.144;1.1809];

l_7_min = 18.7085;
l_7_max = norm(p{8}-p{7});
l_8 = norm(p{9}-p{8});
l_9 = norm(p{10}-p{9});
l_11 = norm(p{11}-p{10});
l_12 = norm(p{12}-p{11});
d_7 = norm(p{9}-p{7});
l_7 = l_7_max;
```

```

increment_2 = (l_7_max-l_7_min)/100;
actuator_length_2 = [];
cangle_2 = [];
pd_2 = zeros(3,100);

ende = zeros(3,100);
ende_2 = zeros(3,100);
% sim
for i = 1:100 %forward kinematics

    theta_1 = acosd((l_1^2+d_1^2-l_2^2)/(2*l_1*d_1));
    phi_1 = atand(p{3}(2)/p{3}(1));
    R_1 = [cosd(theta_1+phi_1) -sind(theta_1+phi_1) 0; sind(theta_1+phi_1)
cosd(theta_1+phi_1) 0; 0 0 1];

    theta_2 = asind(d_1*(sind(theta_1)/l_2));
    theta_3 = 70.08;

    R_4 = [cosd(-theta_3) -sind(-theta_3) 0; sind(-theta_3) cosd(-theta_3) 0;
0 0 1];

    p{2} = R_1*[l_1;0;0];

    chck = norm(p{2}-p{3});

    p{4} = p{3}+R_4*((p{2}-p{3})/chck)*l_3;
    p{4}(3) = p_4_z;

    t = l_5^2-(p{5}(2)-p{4}(2))^2;
    m = p{6}(1)-p{4}(1);
    n = p{6}(3)-p{4}(3);
    k = (t-m^2-n^2-l_6^2)/(-2*l_6);
    r = sqrt(m^2+n^2);
    alpha_sim = atan2d(-n,m);
    theta_6 = (acosd(k/r))-alpha_sim;
    phi_6 = -(90-theta_6);

%
    R_6 = [cosd(phi_6) 0 -sind(phi_6); 0 1 0; sind(phi_6) 0 cosd(phi_6)];
    p5_y = p{5}(2);
%
    p{5} = p{6}-R_6*[0;0;l_6];
    p{5}(2) = p5_y;
    p{5}(1) = p{6}(1)-l_6*cosd(theta_6);
    p{5}(3) = p{6}(3)-l_6*sind(theta_6);

% winglet interface plane
    wing{1}(1) = (p{6}(1))-(1.5*l_6)*cosd(theta_6-35+90);
    wing{1}(3) = p{6}(3)-(1.5*l_6)*sind(theta_6-35+90);
    wing{1}(2) = p5_y-1;

    wing{2}(1) = (p{6}(1))+(0.5*l_6)*cosd(theta_6-35+90);
    wing{2}(3) = p{6}(3)+(0.5*l_6)*sind(theta_6-35+90);

```



```

wing{2}(2) = p5_y-1;

wing{3}(1) = (p{6}(1))+(0.5*l_6)*cosd(theta_6-35+90);
wing{3}(3) = p{6}(3)+(0.5*l_6)*sind(theta_6-35+90);
wing{3}(2) = p5_y+17;

wing{4}(1) = (p{6}(1))-(1.5*l_6)*cosd(theta_6-35+90);
wing{4}(3) = p{6}(3)-(1.5*l_6)*sind(theta_6-35+90);
wing{4}(2) = p5_y+17;

wing{5}(1) = (p{6}(1))-(1.5*l_6)*cosd(theta_6-35+90);
wing{5}(3) = p{6}(3)-(1.5*l_6)*sind(theta_6-35+90);
wing{5}(2) = p5_y-1;
%check
%check
ck = norm(wing{2}-wing{1})

% winglet plane
winglet{1}(1) = (p{6}(1))+((1.5*l_6)*cosd(theta_6-35));
winglet{1}(3) = p{6}(3)+((1.5*l_6)*sind(theta_6-35));
winglet{1}(2) = p5_y-1;

winglet{2}(1) = (p{6}(1));
winglet{2}(3) = p{6}(3);
winglet{2}(2) = p5_y-1;

winglet{3}(1) = (p{6}(1));
winglet{3}(3) = p{6}(3);
winglet{3}(2) = p5_y+17;

winglet{4}(1) = (p{6}(1))+((1.5*l_6)*cosd(theta_6-35));
winglet{4}(3) = p{6}(3)+((1.5*l_6)*sind(theta_6-35));
winglet{4}(2) = p5_y+17;

winglet{5}(1) = (p{6}(1))+((1.5*l_6)*cosd(theta_6-35));
winglet{5}(3) = p{6}(3)+((1.5*l_6)*sind(theta_6-35));
winglet{5}(2) = p5_y-1;

if i==1
    theta_6_initial=theta_6;
    pd(1,i) = p{4}(1);
    pd(2,i) = p{4}(2);
    pd(3,i) = p{4}(3);
elseif isreal(theta_6)
    theta_6_final = theta_6;
    cangle(i-1) = theta_6;
    actuator_length(i-1) = l_1;
    pd(1,i) = p{4}(1);
    pd(2,i) = p{4}(2);
    pd(3,i) = p{4}(3);
    range = theta_6_final-theta_6_initial;
end

% side 2
theta_8 = acosd((l_7^2+l_8^2-d_7^2)/(2*l_7*l_8));

```

```

phi_7 = asind((sind(theta_8)*l_8/d_7));
alpha_7 = acosd(p{9}(1)/d_7);
theta_7 = alpha_7-phi_7;
R_7 = [cosd(theta_7) -sind(theta_7) 0; sind(theta_7) cosd(theta_7) 0; 0 0
1];

R_10 = [cosd(180) -sind(180) 0; sind(180) cosd(180) 0; 0 0 1];

p{8} = p{7}+R_7*[l_7;0;0];

chck = norm(p{8}-p{9});

p{10} = p{9}+R_10*((p{8}-p{9})/chck)*l_9;
p{10}(3) = -0.3787;

t = l_11^2-(p{11}(2)-p{10}(2))^2;
m = p{12}(1)-p{10}(1);
n = p{12}(3)-p{10}(3);
k = (t-m^2-n^2-l_12^2)/(-2*l_12);
r = sqrt(m^2+n^2);
alpha_sim = atan2d(-n,m);
theta_12 = (acosd(k/r))-alpha_sim;
phi_12 = -(90-theta_12);
%
R_12 = [cosd(phi_12) 0 -sind(phi_12); 0 1 0; sind(phi_12) 0
cosd(phi_12)];
p11_y = p{11}(2);
p{11} = p{12}-R_12*[0;0;l_12];
p{11}(2) = p11_y;
% p{5}(1) = p{6}(1)-l_6*cosd(theta_6)
% p{5}(3) = p{6}(3)-l_6*sind(theta_6)

if i==1
    theta_12_initial=theta_12;
    pd_2(1,i) = p{10}(1);
    pd_2(2,i) = p{10}(2);
    pd_2(3,i) = p{10}(3);
elseif isreal(theta_12)
    theta_12_final = theta_12;
    cangle_2(i-1) = theta_12;
    actuator_length_2(i-1) = l_7;
    pd_2(1,i) = p{10}(1);
    pd_2(2,i) = p{10}(2);
    pd_2(3,i) = p{10}(3);
    range = theta_12_final-theta_12_initial;
end

%plotting
if isreal(theta_6) && isreal(theta_12)
% plotting
figure (1)
for j = 1:5

```

```

        plot3([p{j}(1) p{j+1}(1)], [p{j}(2) p{j+1}(2)], [p{j}(3) p{j+1}(3)],
'0-')
        hold on

if j==4
        plot3([p{j}(1) p{j-2}(1)], [p{j}(2) p{j-2}(2)], [p{j}(3) p{j-
2}(3)], '0-')
end

end

for j = 7:11

        plot3([p{j}(1) p{j+1}(1)], [p{j}(2) p{j+1}(2)], [p{j}(3) p{j+1}(3)],
'0-')
        hold on

end

        grid on
        axis([-5 40 -5 40 -20 25]);
        view([0 -1 0])
        axis square
        xlabel('X')
        ylabel('Y')
        zlabel('Z')
        drawnow
        xlabel('X')
        ylabel('Y')
        zlabel('Z')
        hold off

        l_1 = l_1+increment;
        l_7 = l_7-increment_2;

elseif ~isreal(theta_6) || ~isreal(theta_12)
        disp('out of range')
break

end

ende(:,i) = p{5};
ende_2(:,i) = p{11};

end
hold on
plot3(ende(1,:),ende(2,:),ende(3,:), '-.k', 'LineWidth',2)
plot3(ende_2(1,:),ende_2(2,:),ende_2(3,:), '-.r', 'LineWidth',2)

```

```

%% input for IK
in_theta_6 = 45;
p_4_z = p{4}(3);
%
% IK loop
for k = 1:70
    i_p{6} = p{6};
    phi_6 = -(90-in_theta_6);
    R_6 = [cosd(phi_6) 0 -sind(phi_6); 0 1 0; sind(phi_6) 0 cosd(phi_6)];
    i_p5_y = p{5}(2);

    i_p{5} = i_p{6}-R_6*[0;0;l_6];
    i_p{5}(2) = i_p5_y;

    c = l_5^2 - (p_4_z-i_p{5}(3))^2;
    e = p{3}(1)-i_p{5}(1);
    g = p{3}(2)-i_p{5}(2);
    h = (c-e^2-g^2-l_3^2)/(2*l_3);
    r_i = sqrt(e^2+g^2);
    beta = atan2d(g,-e);
    theta_5 = 180-(acosd(h/r_i)+beta);

    i_p{4}(1) = p{3}(1)+l_3*cosd(theta_5);
    i_p{4}(2) = p{3}(2)+l_3*sind(theta_5);
    i_p{4}(3) = p_4_z;

    i_p{3} = p{3};
    i_R_3 = [cosd(theta_3) -sind(theta_3) 0; sind(theta_3) cosd(theta_3) 0; 0 0 1];
    u_34 = ([i_p{3}(1); i_p{3}(2)]-[i_p{4}(1);i_p{4}(2)])/norm([i_p{3}(1);
i_p{3}(2)]-[i_p{4}(1);i_p{4}(2)]);
    u_34(3) = 0;
    i_p{2} = i_p{3}-i_R_3*u_34*l_2;

    i_p{1} = p{1};
    check = norm(i_p{3}-i_p{2});

    ang_act(k,1) = in_theta_6;
    ang_act(k,2) = norm(i_p{2});

    if k>1
        ang_act(k-1,3) = (ang_act(k,1)-ang_act(k-1,1));
        ang_act(k-1,4) = (ang_act(k,2)-ang_act(k-1,2));
    elseif k==70
        ang_act(k,3) = 0;
        ang_act(k,4) = 0;
    end

    figure(3)
    % plotting
    for j = 1:5

```

```

        plot3([i_p{j}(1) i_p{j+1}(1)], [i_p{j}(2) i_p{j+1}(2)], [i_p{j}(3)
i_p{j+1}(3)], 'o-')
        hold on

if j==4
        plot3([i_p{j}(1) i_p{j-2}(1)], [i_p{j}(2) i_p{j-2}(2)],
[i_p{j}(3) i_p{j-2}(3)], 'o-')
end
end

        grid on
        axis([-4 40 -4 40 -5 5]);
%axis square
%view([0 -1 0])
drawnow

        in_theta_6 = in_theta_6+1;

        xlabel('X')
        ylabel('Y')
        zlabel('Z')

        hold off
end

```

Inverse Kinematics

```

clear;
clc;
format long

p{1} = [144.327;80.845;-0.026];
p{2} = [277.496;434.372;-0.026]-p{1};
p{3} = [544.11;427.621;-0.005]-p{1};
p{4} = [510.174;528.759;-9.626]-p{1};
p{5} = [628.358;541.028;-8.105]-p{1};
p{6} = [666.46;541.028;30]-p{1};

phi6 = atan2d(sqrt((p{6}(2)-p{5}(2))^2+(p{6}(3)-p{5}(3))^2),(p{6}(1)-
p{5}(1)));

% converting to inches for scale clarity
for k = 1:6
    p{k} = (p{k}/10)*(0.393701);
end

l_2 = norm(p{3}-p{2});
l_3 = norm(p{4}-p{3});
l_4 = norm(p{4}-p{2});
l_5 = norm(p{5}-p{4});
l_6 = norm(p{6}-p{5});
d_1 = norm(p{3});

theta_3 = 70.08;

%% input for IK
in_theta_6 = 45;
p_4_z = p{4}(3);
%
input_ang = zeros(1,71);
output_stroke = zeros(1,71);

% IK loop
for k = 1:71

    input_ang(k) = in_theta_6;

    i_p{6} = p{6};
    i_p{5} = [p{6}(1)-sqrt(l_6^2-(p{6}(2)-p{5}(2))^2);p{5}(2);p{6}(3)];
    phi_6 = (-in_theta_6);
    R_6 = [cosd(phi_6) 0 sind(phi_6); 0 -1 0; -sind(phi_6) 0 cosd(phi_6)];

    i_p{5} = i_p{6}- R_6*(i_p{6}-i_p{5});

    chck = norm(i_p{6}-i_p{5})

    syms rcxrcyrbxrby
    eqn1= l_5^2==(rbx-rcx)^2+((rby-rcy)^2)+((p{5}(3)-p{4}(3))^2);

```

```

eqn2= l_3^2==(p{3}(1)-rcx)^2+(p{3}(2)-rcy)^2+(p{3}(3)-p{4}(3))^2;
var = [rcx, rcy];
[solx, soly]=solve([eqn1,eqn2],var);
rbx = i_p{5}(1);
rby = i_p{5}(2);
r_cx = vpa(solx(2));
r_cy = vpa(soly(2));

rbx = i_p{5}(1);
rby = i_p{5}(2);
rcx = subs(r_cx);
rcy = subs(r_cy);
i_p{4} = [double(rcx);double(rcy);p_4_z];

chk2 = norm(i_p{5}-i_p{4})

i_p{3} = p{3};
i_R_3 = [cosd(theta_3) -sind(theta_3) 0; sind(theta_3) cosd(theta_3) 0; 0 0 1];
u_34 = ([i_p{3}(1); i_p{3}(2)]-[i_p{4}(1);i_p{4}(2)]) / norm([i_p{3}(1);
i_p{3}(2)]-[i_p{4}(1);i_p{4}(2)]);
u_34(3) = 0;
i_p{2} = i_p{3}-i_R_3*u_34*l_2;

i_p{1} = p{1};
check = norm(i_p{3}-i_p{2});

ang_act(k,1) = in_theta_6;
ang_act(k,2) = norm(i_p{2});

if k>1
    ang_act(k-1,3) = (ang_act(k,1)-ang_act(k-1,1));
    ang_act(k-1,4) = (ang_act(k,2)-ang_act(k-1,2));
elseif k==70
    ang_act(k,3) = 0;
    ang_act(k,4) = 0;
end

figure(3)
% plotting
for j = 1:5

    plot3([i_p{j}(1) i_p{j+1}(1)], [i_p{j}(2) i_p{j+1}(2)], [i_p{j}(3)
i_p{j+1}(3)], 'o-')
    hold on

if j==4
    plot3([i_p{j}(1) i_p{j-2}(1)], [i_p{j}(2) i_p{j-2}(2)],
[i_p{j}(3) i_p{j-2}(3)], 'o-')
end

end

```

```

        grid on
        axis([-4 40 -4 40 -5 5]);
%axis square
        view([0 -1 0])
        drawnow

        in_theta_6 = in_theta_6+1;
        output_stroke(k) = norm(i_p{2});
%         disp(num2str(i_p{5}));
%         disp(num2str(norm(i_p{2})));

        xlabel('X')
        ylabel('Y')
        zlabel('Z')

        hold off
end
%%
plot(input_ang(1,:),output_stroke(1,:));
xlabel('Output Angle(^o)');
ylabel('Actuator Stroke length (in)');
p = polyfit(input_ang(1,:),output_stroke(1,:),3);
x1 = 45:115;
y1 = polyval(p,x1);
figure(2)
plot(x1,y1);

```


Inverse Kinematics Aft Linkage

```
clear;
clc;
format long

p{1} = [144.327;80.845;-0.026];
p{7} = [93.827;564.518;-0.044]-p{1};
p{8} = [606.491;669.115;-0.006]-p{7};
p{9} = [536.828;817.556;-0.017]-p{7};
p{10} = [509.797;877.321;-9.641]-p{7};
p{11} = [628.358;885.15;-8.105]-p{7};
p{12} = [666.46;885.15;30]-p{7};

phi6 = atan2d(sqrt((p{12}(1)-p{11}(1))^2+(p{12}(3)-p{11}(3))^2),(p{12}(2)-p{11}(2)));

% converting to inches for scale clarity
for k = 1:12
    p{k} = (p{k}/10)*(0.393701);
end

% l_7_min = norm(p{8}-p{7});
l_7_max = norm(p{8});
l_8 = norm(p{9}-p{8});
l_9 = norm(p{10}-p{9});
l_11 = norm(p{11}-p{10});
l_12 = norm(p{12}-p{11});
d_7 = norm(p{9});

theta_3 = 180.5;

%% input for IK
in_theta_6 = 45;
p_10_z = p{10}(3);
%
input_ang = zeros(1,71);
output_stroke = zeros(1,71);

% IK loop
for k = 1:71

    input_ang(k) = in_theta_6;

    i_p{12} = p{12};
    i_p{11} = [p{12}(1)-sqrt(l_12^2);p{11}(2);p{12}(3)];
    phi_6 = (-in_theta_6);
    R_6 = [cosd(phi_6) 0 sind(phi_6); 0 1 0; -sind(phi_6) 0 cosd(phi_6)];

    i_p{11} = i_p{12}- R_6*(i_p{12}-i_p{11});

    chck = norm(i_p{12}-i_p{11})
```

```

syms r11xr11yr10xr10y
eqn1= l_11^2==(r11x-r10x)^2+(r11y-r10y)^2+(p{11}(3)-p{10}(3))^2;
eqn2= l_9^2==(r10x-p{9}(1))^2+(r10y-p{9}(2))^2+(p{10}(3)-
p{9}(3))^2);
var = [r10x, r10y];
[solx, soly]=solve([eqn1,eqn2],var);
r11x = i_p{11}(1);
r11y = i_p{11}(2);
r_10x = vpa(solx(1));
r_10y = vpa(soly(1));

r11x = i_p{11}(1);
r11y = i_p{11}(2);
r10x = subs(r_10x);
r10y = subs(r_10y);
i_p{10} = [double(r10x);double(r10y);p_10_z];

chk2 = norm(i_p{11}-i_p{10})

i_p{9} = p{9};

chk3 = norm(i_p{10}-i_p{9})

i_R_9 = [cosd(theta_3) -sind(theta_3) 0; sind(theta_3) cosd(theta_3) 0; 0 0 1];
u_910 = ([i_p{9}(1); i_p{9}(2)]-[i_p{10}(1);i_p{10}(2)]) / norm([i_p{9}(1);
i_p{9}(2)]-[i_p{10}(1);i_p{10}(2)]);
u_910(3) = 0;
i_p{8} = i_p{9}-i_R_9*u_910*1_8;

i_p{7} = p{7};

chk4 = norm(i_p{9}-i_p{8});

ang_act(k,1) = in_theta_6;
ang_act(k,2) = norm(i_p{8});

if k>1
    ang_act(k-1,3) = (ang_act(k,1)-ang_act(k-1,1));
    ang_act(k-1,4) = (ang_act(k,2)-ang_act(k-1,2));
elseif k==70
    ang_act(k,3) = 0;
    ang_act(k,4) = 0;
end

figure(1)
% plotting
for j = 7:11

    plot3([i_p{j}(1) i_p{j+1}(1)], [i_p{j}(2) i_p{j+1}(2)], [i_p{j}(3)
i_p{j+1}(3)], 'o-')

```

```

        hold on

end

    grid on
    axis([-4 40 -4 40 -5 5]);
%axis square
    view([0 -1 0])
    drawnow

    in_theta_6 = in_theta_6+1;
    output_stroke(k) = norm(i_p{8});
%    disp(num2str(i_p{5}));
%    disp(num2str(norm(i_p{2})));

    xlabel('X')
    ylabel('Y')
    zlabel('Z')

    hold off
    test_phi6 = atan2d(sqrt((i_p{12}(2)-i_p{11}(2))^2+(i_p{12}(3)-
i_p{11}(3))^2),(i_p{12}(1)-i_p{11}(1)))

end
%%
figure(2)
plot(input_ang(1,:),output_stroke(1,:));
xlabel('Output Angle(^o)');
ylabel('Actuator Stroke length (in)');
% p = polyfit(input_ang(1,:),output_stroke(1,:),3);
% x1 = 45:115;
% y1 = polyval(p,x1);
% figure(3)
% plot(x1,y1);

```

Velocity Fwd Linkage

```
clear;
clc;
format long

p{1} = [144.327;80.845;-0.026];
p{2} = [324.871;579.485;-0.026]-p{1};
p{3} = [544.11;427.621;-0.005]-p{1};
p{4} = [571.2;530.804;-9.626]-p{1};
p{5} = [689.229;541.028;-18.838]-p{1};
p{6} = [666.46;541.028;30]-p{1};

% converting to inches for scale clarity
for k = 1:6
    p{k} = (p{k}/10)*(0.393701);
end

%%%%%%%%%%correct p{2} from IK%%%%%%%%%%
p{2} = [5.242515700126356;13.903727145708107;0.000826772100000];
%%%%%%%%%%

l_1_min = norm(p{2});
l_2 = norm(p{3}-p{2});
l_3 = norm([p{4}(1);p{4}(2);0]-p{3});
l_4 = norm(p{4}-p{2});
l_5 = norm(p{5}-p{4});
l_6 = norm(p{6}-p{5});
d_1 = norm(p{3});
l_1 = l_1_min;
theta_3 = 70.08;
p_4_z = p{4}(3);

actuator_length = [];
cangle = [];
pd = zeros(3,100);

% sim
theta_6 = 45;
p_5_i = p{5};
p_4_i = p{4};
p_2_i = p{2};

ds1=zeros(71,1);
fktheta6=zeros(71,1);
s_2 = zeros(71,3);
s2 = zeros(71,1);
dt = 5;

% fk sim
syms frcxfrcyfrbxfrbzfry
eqn5= l_6^2==(p{6}(1)-frbx)^2+(p{6}(2)-frby)^2+(p{6}(3)-frbz)^2;
```

```

eqn6= l_5^2==((frcx-frbx)^2)+((frcy-p{5}(2))^2)+((p{4}(3)-frbz)^2);
var3 = [frbx, frbz];
[solx, soly]=solve([eqn5,eqn6],var3);
fr_bx = vpa(solx(1));
fr_bz = vpa(soly(1));

% velocity sim
syms v4xv4yv5xv5zr4xr4yr5xr5yr5zr4z
eqn3 = 0==2*(r5x-r4x)*(v5x-v4x)+2*(r5y-r4y)*(-v4y)+2*(r5z-r4z)*(v5z);
eqn4 = 0==2*(p{3}(1)-r4x)*(-v4x)+2*(p{3}(2)-r4y)*(-v4y);
var2 = [v4x, v4y];
[solvx, solvy]=solve([eqn3,eqn4],var2);
d_4x = (vpa(solvx));
d_4y = vpa(solvy);

for i = 1:round((70/dt)+1)

    format shortg

    ratio = pi/180;
    phi_6 = ratio*(theta_6); % relation of theta_6 to phi_6
    omega_6 = ratio*dt; %d/s

    v{5} = l_6*(omega_6)*[sin(phi_6);0;-cos(phi_6)];
    v_5(i,:) = v{5};

    r5x = p{5}(1);
    r5y = p{5}(2);
    r5z = p{5}(3);
    r4z = p{4}(3);

    r4x = p{4}(1);
    r4y = p{4}(2);
    v5x = v{5}(1);
    v5z = v{5}(3);
    d4x = double(subs(d_4x));
    d4y = double(subs(d_4y));

    v{4} = [(d4x);(d4y);0];

    R_3 = [cos(ratio*theta_3) -sin(ratio*theta_3) 0; sin(ratio*theta_3)
cos(ratio*theta_3) 0; 0 0 1]; % rotation matrix for p{2} in IK using constant
design angle theta_3
    d_u34 = -[v{4}(1);v{4}(2)]/l_3; % derivative of the unit vector from p{3}
to p{4} this is the component that changes with time for v{2}
    d_u34(3) = 0;
    v{2} = -R_3*d_u34*l_2; % derivative of p{2}

    s_2(i,:) = (dot(p{2},v{2})/norm(p{2}))* (p{2}/norm(p{2}));
    s2(i,1) = norm(s_2(i,:));
    %% FK simulation

    theta_1 = acosd((l_1^2+d_1^2-l_2^2)/(2*l_1*d_1));
    phi_1 = atan2d(sqrt((p{3}(2))^2+(p{3}(3))^2),(p{3}(1)));
    theta_2 = (theta_1+phi_1);

```

```

R_1 = [cosd(theta_2) -sind(theta_2) 0; sind(theta_2) cosd(theta_2) 0; 0 0
1];

theta_3 = 70;

R_4 = [cosd(-theta_3) -sind(-theta_3) 0; sind(-theta_3) cosd(-theta_3) 0;
0 0 1];

fk_p{2} = R_1*[l_1;0;0];
chck = norm(fk_p{2}-p{3});

fk_p{4} = p{3}+R_4*((fk_p{2}-p{3})/chck)*l_3;
fk_p{4}(3) = p_4_z;

frcx = fk_p{4}(1);
frcy = fk_p{4}(2);
frby = p{5}(2);
frbx = subs(fr_bx);
frbz = subs(fr_bz);
fk_p{5} = [double(frbx);p{5}(2);double(frbz)];
fk_p{1} = p{1};
fk_p{3} = p{3};
fk_p{6} = p{6};
fk_theta_6 = atan2d(sqrt((fk_p{6}(2)-fk_p{5}(2))^2+(fk_p{6}(3)-
fk_p{5}(3))^2),(fk_p{6}(1)-fk_p{5}(1)));
fktheta6(i,1) = ceil(fk_theta_6);

% generating matrix for output plot
ang_act(i,1) = theta_6;
ang_act(i,2) = l_1;
ang_act(i,3) = s2(i,1);

figure(1)
% plotting
for j = 1:5

    plot3([fk_p{j}(1) fk_p{j+1}(1)], [fk_p{j}(2) fk_p{j+1}(2)],
[fk_p{j}(3) fk_p{j+1}(3)], 'o-')
    hold on

if j==4
    plot3([fk_p{j}(1) fk_p{j-2}(1)], [fk_p{j}(2) fk_p{j-2}(2)],
[fk_p{j}(3) fk_p{j-2}(3)], 'o-')
end
end

grid on
axis([-4 40 -4 40 -22 22]);
axis square
% view([0 -1 0])
drawnow

xlabel('X')

```

```

ylabel('Y')
xlabel('Z')

hold off

theta_6 = theta_6+dt;
l_1 = l_1+s2(i,1);

p{5} = fk_p{5};
p{4} = fk_p{4};
p{2} = fk_p{2};

ds1(i,1) = s2(i,1);
end
%%
figure(2)
plot(ang_act(:,1),ang_act(:,3))
grid on
xlabel('Input angle [^\circ]')
ylabel('Actuator speed to achieve next angle [in/s]')

figure(3)
plot(ang_act(:,2),ang_act(:,3))
grid on
xlabel('Actuator Length [in]')
ylabel('Actuator speed to achieve next angle [in/s]')

figure(4)
plot(ang_act(:,1),ang_act(:,2))
grid on
xlabel('Input Angle [^\circ]')
ylabel('Actuator Length [in]')

```

Velocity Aft Linkage

```

clear;
clc;
format long

p{1} = [144.327;80.845;-0.026];
p{7} = [93.827;564.518;-0.044]-p{1};
p{8} = [606.491;669.115;-0.006]-p{7};
p{9} = [536.828;817.556;-0.017]-p{7};
p{10} = [509.797;877.321;-9.641]-p{7};
p{11} = [628.358;885.15;-8.105]-p{7};
p{12} = [666.46;885.15;30]-p{7};

% converting to inches for scale clarity
for k = 1:12
    p{k} = round((p{k}/10)*(0.393701),4);
end

%%%%%%%%%%correct p{8} from IK%%%%%%%%%%
p{8} = [ 25.8349;7.2864;0.00003937];
%%%%%%%%%%

phi6 = atan2d(sqrt((p{12}(2)-p{11}(2))^2+(p{12}(3)-p{11}(3))^2),(p{12}(1)-
p{11}(1)));

% l_7_min = norm(p{8}-p{7});
l_7_max = round(norm(p{8}),3);
l_8 = round(norm(p{9}-p{8}),3);
l_9 = round(norm(p{10}-p{9}),3);
l_11 = round(norm(p{11}-p{10}),3);
l_12 = round(norm(p{12}-p{11}),3);
d_7 = round(norm(p{9}),3);
l_7 = l_7_max;

theta_3 = 180.5;

p_10_z = p{10}(3);

% sim
theta_6 = 45;

ds1=zeros(71,1);
fktheta12=zeros(71,1);
s_2 = zeros(71,3);
s2 = zeros(71,1);
dt = 5; % delta_theta_per_second

% fk sim
syms frcxfrcyfrbxfrbzfrbyfrcz
eqn5= l_12^2==(p{12}(1)-frbx)^2+(p{12}(2)-frby)^2+(p{12}(3)-frbz)^2;

```



```

eqn6= l_11^2==((frcx-frbx)^2)+((frcy-p{11}(2))^2)+((frcz-frbz)^2);
var3 = [frbx, frbz];
[solx, soly]=solve([eqn5,eqn6],var3);
fr_bx = vpa(solx(1));
fr_bz = vpa(soly(1));

syms v10xv10yv11xv11zr11xr11yr10xr10yr11zr10z
eqn3 = 0==(2*(r11x-r10x)*(v11x-v10x))+(2*(r11y-r10y)*(-v10y))+(2*(r11z-
r10z)*(v11z));
eqn4 = 0==(2*(p{9}(1)-r10x)*(-v10x))+(2*(p{9}(2)-r10y)*(-v10y));
var2 = [v10x, v10y];
[solvx, solvy]=solve([eqn3,eqn4],var2);
d_10x = vpa(solvx);
d_10y = vpa(solvy);

for i = 1:round((70/dt)+1)

    format shortg

    ratio = pi/180;
    phi_6 = ratio*(theta_6); % relation of theta_6 to phi_6
    omega_6 = ratio*dt; %d/s

    v{11} = l_12*omega_6*[sin(phi_6);0;-cos(phi_6)];
    v_11(i,:) = v{11};

    v11x = v{11}(1);
    v11z = v{11}(3);
    r11x = p{11}(1);
    r11y = p{11}(2);
    r11z = p{11}(3);
    r10x = p{10}(1);
    r10y = p{10}(2);
    r10z = p{10}(3);

    d10x = (double(subs(d_10x)));
    d10y = (double(subs(d_10y)));

    v{10} = [(d10x);(d10y);0];

    R_3 = [cos(ratio*theta_3) -sin(ratio*theta_3) 0; sin(ratio*theta_3)
cos(ratio*theta_3) 0; 0 0 1]; % rotation matrix for p{2} in IK using constant
design angle theta_3
    d_u109 = -v{10}/(norm([p{10}(1);p{10}(2);0]-[p{9}(1);p{9}(2);0])); %
derivative of the unit vector from p{3} to p{4} this is the component that
changes with ti
    v{8} = -(R_3*d_u109*norm([p{9}(1);p{9}(2);0]-[p{8}(1);p{8}(2);0])); %
derivative of p{2}

    s_2(i,:) = (dot(p{8},v{8})/l_7)*(p{8}/l_7);
    s2(i,1) = -(norm([s_2(i,1);s_2(i,2)]));

```

```

%% FK to simulate

theta_3 = 180.5;
R_10 = [cosd(-theta_3) -sind(-theta_3) 0; sind(-theta_3) cosd(-theta_3)
0; 0 0 1];

fk_p{8} = p{8};
chck = norm(fk_p{8}-p{9});

fk_p{10} = ((p{9}+R_10*([fk_p{8}(1);fk_p{8}(2);0]-
[p{9}(1);p{9}(2);0])/norm([fk_p{8}(1);fk_p{8}(2);0]-
[p{9}(1);p{9}(2);0]))*(norm([p{10}(1);p{10}(2);0]-[p{9}(1);p{9}(2);0]))));
fk_p{10}(3) = (p_10_z);

frcx = fk_p{10}(1);
frcy = fk_p{10}(2);
frcz = fk_p{10}(3);
frby = p{11}(2);
frbx = (subs(fr_bx));
frbz = (subs(fr_bz));
fk_p{11} = [double(frbx);p{11}(2);double(frbz)];
fk_p{7} = p{7};
fk_p{9} = p{9};
fk_p{12} = p{12};

fk_theta_12 = atan2d(sqrt((fk_p{12}(2)-fk_p{11}(2))^2+(fk_p{12}(3)-
fk_p{11}(3))^2),(fk_p{12}(1)-fk_p{11}(1)));
fktheta12(i,1) = (fk_theta_12);
fktheta12(i,1) = ceil(fk_theta_12);

% generating matrix for output plot
ang_act(i,1) = theta_6;
ang_act(i,2) = l_7;
ang_act(i,3) = s2(i,1);

figure(1)
% plotting
for j = 7:11

    plot3([fk_p{j}(1) fk_p{j+1}(1)], [fk_p{j}(2) fk_p{j+1}(2)],
[fk_p{j}(3) fk_p{j+1}(3)], 'o-')
    hold on

end

grid on
axis([-4 40 -4 40 -5 5]);
%axis square
% view([0 -1 0])
drawnow

xlabel('X')
ylabel('Y')

```

```

        zlabel('Z')

        hold off

        theta_6 = theta_6+(dt);
        l_7 = l_7+s2(i,1);

        p{8} = p{8}+(v{8});

        p{10} = fk_p{10};
        p{11} = fk_p{11};
        %     p{8} = fk_p{8};

        ds1(i,1) = s2(i,1);
    end
    %%
    figure(2)
    plot(ang_act(:,1),ang_act(:,3))
    grid on
    xlabel('Input angle [^o]')
    ylabel('Actuator speed to achieve next angle [in/s]')

    figure(3)
    plot(ang_act(:,2),ang_act(:,3))
    grid on
    xlabel('Actuator Length [in]')
    ylabel('Actuator speed to achieve next angle [in/s]')

    figure(4)
    plot(ang_act(:,1),ang_act(:,2))
    grid on
    xlabel('Input Angle [^o]')
    ylabel('Actuator Length [in]')

```

Velocity Tuning

Goal Attainment

```

clear;
clc;
x_0 = [1,1];
goal = [1e-1,1e-1];
weight = abs(goal);
lb = [0.1,1];
ub = [10,5];
options =
optimoptions('fgoalattain','Display','iter','MaxIterations',30,'FiniteDifferenceStepSize',1e-1);
[x,fval,attainfactor] =
fgoalattain(@velocity_optimization,x_0,goal,weight,[],[],[],[],lb,[],[],options)

```

Velocity Function used for Tuning

```
function F = velocity_optimization(x)
format long
omega = x(1);
delay = x(2);

p{1} = [144.327;80.845;-0.026];
p{2} = [324.871;579.485;-0.026]-p{1};
p{3} = [544.11;427.621;-0.005]-p{1};
p{4} = [510.174;528.759;-9.626]-p{1};
p{5} = [628.358;541.028;-8.105]-p{1};
p{6} = [666.46;541.028;30]-p{1};

% side 2
p{7} = [93.827;564.518;-0.044]-p{1};
p{8} = [606.491;669.115;-0.006]-p{1};
p{9} = [536.828;817.556;-0.017]-p{1};
p{10} = [509.797;877.321;-9.641]-p{1};
p{11} = [628.358;885.15;-8.105]-p{1};
p{12} = [666.46;885.15;30]-p{1};

% IK delta strokes
fore_dstroke = 5.970684494864495;
aft_dstroke = 5.762707551016634;

% converting to inches for scale clarity
for k = 1:12
    p{k} = (p{k}/10)*(0.393701);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
p{2} = [5.242515700126356;13.903727145708107;0.000826772100000];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% side 2

l_1_min = norm(p{2});
l_2 = norm(p{3}-p{2});
l_3 = norm([p{4}(1);p{4}(2);0]-p{3});
l_4 = norm(p{4}-p{2});
l_5 = norm(p{5}-p{4});
l_6 = norm(p{6}-p{5});
d_1 = norm(p{3});
l_1 = l_1_min;
theta_3 = 70.08;
p_4_z = p{4}(3);

pp{7} = [93.827;564.518;-0.044];
pp{8} = [606.491;669.115;-0.006]-pp{7};
pp{9} = [536.828;817.556;-0.017]-pp{7};
pp{10} = [509.797;877.321;-9.641]-pp{7};
```

```

pp{11} = [628.358;885.15;-8.105]-pp{7};
pp{12} = [666.46;885.15;30]-pp{7};

for k = 1:12
    pp{k} = (pp{k}/10)*(0.393701);
end
% pp{8} = [25.8349;7.2864;0.00003937];

% side 2
l_7_max = round(norm(pp{8}),3);
l_8 = round(norm(pp{9}-pp{8}),3);
l_9 = round(norm(pp{10}-pp{9}),3);
l_11 = round(norm(pp{11}-pp{10}),3);
l_12 = round(norm(pp{12}-pp{11}),3);
d_7 = round(norm(pp{9}),3);
l_7 = l_7_max;
theta_9 = 180.5;
p_10_z = pp{10}(3);

% sim
theta_6 = 45;

% side 1
% fktheta6=zeros(71,1);
% s_1 = zeros(71,3);
% s1 = zeros(71,1);

% side 2
% fktheta12=zeros(71,1);
% s_2 = zeros(71,3);
% s2 = zeros(71,1);

% velocity omega definition
inc = 1/delay;
omega = 1;
t = (0:inc:1)*omega;
unitstep = t>=0;
ramp = t.^2.*unitstep;

% fk sim
syms fr4xfr4yfr5xfr5zfr5y
eqn5= l_6^2==(p{6}(1)-fr5x)^2+(p{6}(2)-fr5y)^2+(p{6}(3)-fr5z)^2;
eqn6= l_5^2==(fr4x-fr5x)^2+(fr4y-p{5}(2))^2+(p{4}(3)-fr5z)^2;
var3 = [fr5x, fr5z];
[solx, soly]=solve([eqn5,eqn6],var3);
fr_5x = vpa(solx(1));
fr_5z = vpa(soly(1));

syms frcxfrcyfrbxfrbzfrbyfrcz
eqn5= l_12^2==(pp{12}(1)-frbx)^2+(pp{12}(2)-frby)^2+(pp{12}(3)-frbz)^2;
eqn6= l_11^2==(frcx-frbx)^2+(frcy-pp{11}(2))^2+(frcz-frbz)^2;
var3 = [frbx, frbz];
[solx, soly]=solve([eqn5,eqn6],var3);
fr_bx = vpa(solx(2));
fr_bz = vpa(soly(2));

```

```

% velocity sim
syms v4xv4yv5xv5zr4xr4yr5xr5yr5zr4z
eqn3 = 0==2*(r5x-r4x)*(v5x-v4x)+2*(r5y-r4y)*(-v4y)+2*(r5z-r4z)*(v5z);
eqn4 = 0==2*(p{3}(1)-r4x)*(-v4x)+2*(p{3}(2)-r4y)*(-v4y);
var2 = [v4x, v4y];
[solvx, solvy]=solve([eqn3,eqn4],var2);
d_4x = (vpa(solvx));
d_4y = vpa(solvy);

syms v10xv10yv11xv11zr11xr11yr10xr10yr11zr10z
eqn3 = 0==(2*(r11x-r10x)*(v11x-v10x))+(2*(r11y-r10y)*(-v10y))+(2*(r11z-
r10z)*(v11z));
eqn4 = 0==(2*(pp{9}(1)-r10x)*(-v10x))+(2*(pp{9}(2)-r10y)*(-v10y));
var2 = [v10x, v10y];
[solvx, solvy]=solve([eqn3,eqn4],var2);
d_10x = vpa(solvx);
d_10y = vpa(solvy);
i = 1;

while theta_6<=115

if i<size(ramp,2)
    dt = (ramp(i));
else
    dt = ramp(end);
end

format longg

ratio = pi/180;
phi_6 = ratio*(theta_6); % relation of theta_6 to phi_6
omega_6 = ratio*dt; %d/s

v{5} = l_6*(omega_6)*[sin(phi_6);0;-cos(phi_6)];
v_5(i,:) = v{5};

r5x = p{5}(1);
r5y = p{5}(2);
r5z = p{5}(3);
r4z = p{4}(3);

r4x = p{4}(1);
r4y = p{4}(2);
v5x = v{5}(1);
v5z = v{5}(3);
d4x = double(subs(d_4x));
d4y = double(subs(d_4y));

v{4} = [(d4x);(d4y);0];

R_3 = [cos(ratio*theta_3) -sin(ratio*theta_3) 0; sin(ratio*theta_3)
cos(ratio*theta_3) 0; 0 0 1]; % rotation matrix for p{2} in IK using constant
design angle theta_3

```

```

    d_u34 = -[v{4}(1);v{4}(2)]/l_3; % derivative of the unit vector from p{3}
to p{4} this is the component that changes with time for v{2}
    d_u34(3) = 0;
    v{2} = -R_3*d_u34*l_2; % derivative of p{2}

    s_1(i,:) = (dot(p{2},v{2})/norm(p{2}))* (p{2}/norm(p{2}));
    s1(i,1) = norm(s_1(i,:));

% side 2
    v{11} = l_12*omega_6*[sin(phi_6);0;-cos(phi_6)];
    v_11(i,:) = v{11};

    v11x = v{11}(1);
    v11z = v{11}(3);
    r11x = pp{11}(1);
    r11y = pp{11}(2);
    r11z = pp{11}(3);
    r10x = pp{10}(1);
    r10y = pp{10}(2);
    r10z = pp{10}(3);

    d10x = (double(subs(d_10x)));
    d10y = (double(subs(d_10y)));

    v{10} = [(d10x);(d10y);0];

    R_9 = [cos(ratio*theta_9) -sin(ratio*theta_9) 0; sin(ratio*theta_9)
cos(ratio*theta_9) 0; 0 0 1]; % rotation matrix for p{2} in IK using constant
design angle theta_3
    d_u109 = -v{10}/(norm([pp{10}(1);pp{10}(2);0]-[pp{9}(1);pp{9}(2);0])); %
derivative of the unit vector from p{3} to p{4} this is the component that
changes with ti
    v{8} = -(R_9*d_u109*norm([pp{9}(1);pp{9}(2);0]-[pp{8}(1);pp{8}(2);0])); %
derivative of p{2}

    s_2(i,:) = (dot(pp{8},v{8})/l_7)*(pp{8}/l_7);
    s2(i,1) = -(norm([s_2(i,1);s_2(i,2)]));

%% FK simulation

    theta_1 = acosd((l_1^2+d_1^2-l_2^2)/(2*l_1*d_1));
    phi_1 = atan2d(sqrt((p{3}(2))^2+(p{3}(3))^2),(p{3}(1)));
    theta_2 = (theta_1+phi_1);
    R_1 = [cosd(theta_2) -sind(theta_2) 0; sind(theta_2) cosd(theta_2) 0; 0 0
1];

    theta_3 = 70;

    R_4 = [cosd(-theta_3) -sind(-theta_3) 0; sind(-theta_3) cosd(-theta_3) 0;
0 0 1];

```

```

fk_p{2} = R_1*[1_1;0;0];
chck = norm(fk_p{2}-p{3});

fk_p{4} = p{3}+R_4*((fk_p{2}-p{3})/chck)*1_3;
fk_p{4}(3) = p_4_z;

fr4x = fk_p{4}(1);
fr4y = fk_p{4}(2);
fr5y = p{5}(2);
fr5x = subs(fr_5x);
fr5z = subs(fr_5z);
fk_p{5} = [double(fr5x);p{5}(2);double(fr5z)];
fk_p{1} = p{1};
fk_p{3} = p{3};
fk_p{6} = p{6};
fk_theta_6 = atan2d(sqrt((fk_p{6}(2)-fk_p{5}(2))^2+(fk_p{6}(3)-
fk_p{5}(3))^2),(fk_p{6}(1)-fk_p{5}(1)));
fktheta6(i,1) = ceil(fk_theta_6);

% generating matrix for output plot
ang_act(i,1) = theta_6;
ang_act(i,2) = 1_1;
ang_act(i,3) = s1(i,1);

% side 2
theta_9 = 180.5;
R_10 = [cosd(-theta_9) -sind(-theta_9) 0; sind(-theta_9) cosd(-theta_9)
0; 0 0 1];

fk_p{8} = pp{8};
chck2 = norm(fk_p{8}-pp{9});
%%%%

fk_p{10} = ((pp{9}+R_10*([fk_p{8}(1);fk_p{8}(2);0]-
[pp{9}(1);pp{9}(2);0])/norm([fk_p{8}(1);fk_p{8}(2);0]-
[pp{9}(1);pp{9}(2);0]))*(norm([pp{10}(1);pp{10}(2);0]-
[pp{9}(1);pp{9}(2);0]))));
fk_p{10}(3) = (p_10_z);

frcx = fk_p{10}(1);
frcy = fk_p{10}(2);
frcz = fk_p{10}(3);
frby = pp{11}(2);
frbx = (subs(fr_bx));
frbz = (subs(fr_bz));
fk_p{11} = [double(frbx);pp{11}(2);double(frbz)];

fk_p{11} = fk_p{11}+pp{7}-p{1};
fk_p{10} = fk_p{10}+pp{7}-p{1};
fk_p{8} = fk_p{8}+pp{7}-p{1};
fk_p{7} = p{7};
fk_p{9} = p{9};
fk_p{12} = p{12};

```



```

    fk_theta_12 = atan2d(sqrt((fk_p{12}(2)-fk_p{11}(2))^2+(fk_p{12}(3)-
fk_p{11}(3))^2),(fk_p{12}(1)-fk_p{11}(1)));
    fktheta12(i,1) = (fk_theta_12);
    fktheta12(i,1) = ceil(fk_theta_12);

% generating matrix for output plot
    ang_act2(i,1) = theta_6;
    ang_act2(i,2) = l_7;
    ang_act2(i,3) = s2(i,1);

    theta_6 = theta_6+dt;
    l_1 = l_1+s1(i,1);
    l_7 = l_7+s2(i,1);

    p{5} = fk_p{5};
    p{4} = fk_p{4};
    p{2} = fk_p{2};

% side 2
    pp{8} = pp{8}+(v{8});
    pp{10} = pp{10}+v{10};
    pp{11} = pp{11}+v{11};
    i =i+1;
end

%% for optimization
dstroke1 = ang_act(i-1,2)-ang_act(1,2);
dstroke2 = ang_act2(1,2)-ang_act2(i-1,2);
e1 = (abs(fore_dstroke-dstroke1)/fore_dstroke)*100;
e2 = (abs(aft_dstroke-dstroke2)/aft_dstroke)*100;
F = [e1,e2];
figure(2)
plot(ang_act(:,1),ang_act(:,3))
grid on
xlabel('Input angle [^\circ]')
ylabel('Actuator speed to achieve next angle [in/s]')
title('Actuator speed vs. Angle FORE')

figure(3)
plot(ang_act(:,2),ang_act(:,3))
grid on
xlabel('Actuator Length [in]')
ylabel('Actuator speed to achieve next angle [in/s]')
title('Actuator speed vs. Length FORE')

figure(4)
plot(ang_act(:,1),ang_act(:,2))
grid on
xlabel('Input Angle [^\circ]')
ylabel('Actuator Length [in]')
title('Stroke length vs. Winglet Angle FORE')

%% side 2

```

```

figure(5)
plot(ang_act2(:,1),ang_act2(:,3))
grid on
xlabel('Input angle [^\circ]')
ylabel('Actuator speed to achieve next angle [in/s]')
title('Actuator speed vs. Angle AFT')

figure(6)
plot(ang_act2(:,2),ang_act2(:,3))
grid on
xlabel('Actuator Length [in]')
ylabel('Actuator speed to achieve next angle [in/s]')
title('Actuator speed vs. Length AFT')

figure(7)
plot(ang_act2(:,1),ang_act2(:,2))
grid on
xlabel('Input Angle [^\circ]')
ylabel('Actuator Length [in]')
title('Stroke length vs. Winglet Angle AFT')

figure(8)
plot(ang_act(:,1),ang_act(:,2),ang_act2(:,1),ang_act2(:,2))
grid on
xlabel('Input Angle [^\circ]')
ylabel('Actuator Length [in]')
title('Stroke length vs. Winglet Angle BOTH')
legend('FORE','AFT')

end

```

Appendix B

The figure below shows the cant and toe axis on a wing. The angle indicated by θ is the cant angle, with the axis coming out of the plane representing the cant axis. The toe axis the axis perpendicular to that, along the wingspan.



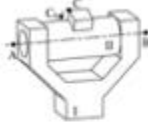


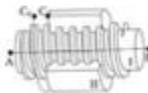
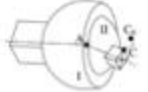
Figure 2. Winglet Dimensions

Figure B-1: Wing cant and toe axis diagram

Appendix C

The table below details the individual kinematic equations for the lower pairs of the spatial parallel linkage for the Gao and Wu solution process.

Table C-1: Constraint equations for lower kinematic pairs

Name of the kinematics pairs	mechanism	Non-active pairs constraint equations	Active pairs constraint equations
Rotational pair		$\ P_A - P_C\ = \ P_A - P_{C0}\ $ $\ P_B - P_C\ = \ P_B - P_{C0}\ $	$P_C - P_B = R(P_{C0} - P_B)$
Cylindrical pair		$\ (P_A - P_C) \times e\ = \ (P_A - P_{C0}) \times e\ $	$P_C - P_B = R(P_{C0} - P_B) + xe$
Prismatic pair		$(P_C - P_{C0}) \times e = 0$	$P_C - P_{C0} = xe$
Helical pair		$\theta = \frac{(P_C - P_{C0}) \cdot e}{P_h}$ $P_C - P_B = R(P_{C0} - P_B) + \frac{\theta P_h}{2\pi} e$	$P_C - P_B = R(P_{C0} - P_B) + \frac{\theta P_h}{2\pi} e$
Spherical pair		$\ P_A - P_C\ = \ P_A - P_{C0}\ $	$P_C - P_A = R_{tpy}(P_{C0} - P_A)$

Appendix D

Non-computational analytic solution for FK

A non-computational analytical solution was attempted for the forward kinematics. This solution involved simplifying assumptions of the system geometry and eventually proved to be prone to error. The equations below outline the solution for the fwd and aft linkages. The error when compared to the computational results is shown in Figure ().

$$k = \frac{r_5^2 - (p_{5y} - p_{4y})^2 - (p_{6x} - p_{4x})^2 - (p_{6z} - p_{4z})^2 - r_6^2}{-2r_6} \quad (\text{D-1})$$

$$m = \sqrt{((p_{6x} - p_{4x})^2 + (p_{6z} - p_{4z})^2)} \quad (\text{D-2})$$

$$\alpha = \text{atan2}\left(-\frac{(p_{6z} - p_{4z})^2}{(p_{6x} - p_{4x})^2}\right) \quad (\text{D-3})$$

$$\theta_6 = \cos^{-1}\left(\frac{k}{m}\right) - \alpha \quad (\text{D-4})$$

$$k2 = \frac{r_{11}^2 - (p_{11y} - p_{10})^2 - (p_{12x} - p_{10x})^2 - (p_{12z} - p_{10z})^2 - r_{12}^2}{-2r_6} \quad (\text{D-5})$$

$$m2 = \sqrt{((p_{12x} - p_{10x})^2 + (p_{12z} - p_{10z})^2)} \quad (\text{D-6})$$

$$\alpha2 = \text{atan2}\left(-\frac{(p_{12z} - p_{10z})^2}{(p_{12x} - p_{10x})^2}\right) \quad (\text{D-7})$$

$$\theta_{12} = \cos^{-1}\left(\frac{k}{m}\right) - \alpha2 \quad (\text{D-8})$$

Non-computational Analytic Solution Error

The figure below displays the error of the non-computational solution for the FK simulation. While the error may seem low, this is significant for the theoretical results, and due to the sensitivity of the system accuracy of the analytic solution is vital. This is the cause for continuing with the computational analytic solution, despite its complexity.

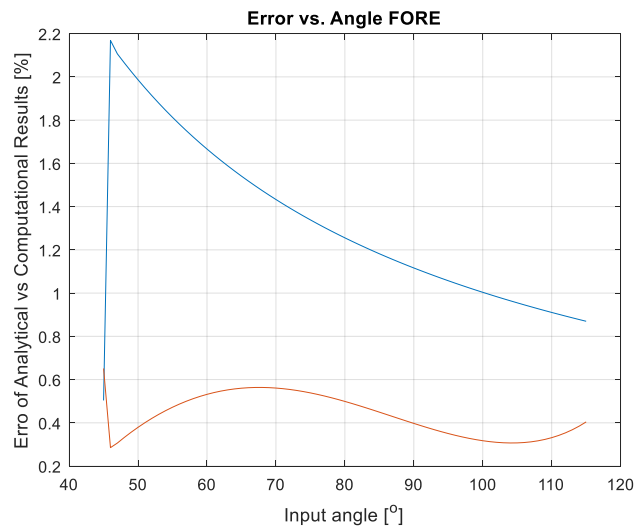


Figure D-1: Error of non-computational solution

References

- [1] J. D. Anderson, *Introduction to flight*, 7th ed. New York: McGraw Hill, 2012.
- [2] T. K. Guha, W. S. Oates, and R. Kumar, "Characterization of piezoelectric macrofiber composite actuated winglets," *Smart Mater. Struct.*, vol. 24, no. 6, p. 065043, Jun. 2015.
- [3] A. Beehook and J. Wang, "Aerodynamic analysis of variable cant angle winglets for improved aircraft performance," in *Automation and Computing (ICAC), 2013 19th International Conference on*, 2013, pp. 1–6.
- [4] "Airbus - Corporate information - Innovation & technology," 21-Apr-2009. [Online]. Available: <https://web.archive.org/web/20090421092103/http://www.airbus.com:80/en/corporate/innovation/>.
- [5] "Wing View," *Private Jets Charter*. [Online]. Available: <http://www.privatejetscharter.com/photos/photo/67/>. [Accessed: 11-Dec-2017].
- [6] C. Brady, *The Boeing 737 Technical Guide*. Lulu Press, Inc, 2017.
- [7] P. Paudel, "Aerodynamic Aspects in the Development of Morphing Winglet for a Regional Aircraft," *Dept App Sci Aerosp. Eng Ryerson Univ.*, 2013.
- [8] P. Bourdin, A. Gatto, and M. I. Friswell, "Aircraft Control via Variable Cant-Angle Winglets," *J. Airvr.*, vol. 45, no. 2, pp. 414–423, Mar. 2008.
- [9] L. Falcão, A. A. Gomes, and A. Suleman, "Aero-structural Design Optimization of a Morphing Wingtip," *J. Intell. Mater. Syst. Struct.*, vol. 22, no. 10, pp. 1113–1124, Jul. 2011.
- [10] S. W. Kim, J. Y. Lee, and K. J. Cho, "Towards a bistable morphing winglet for unmanned aerial vehicle(UAV)," in *IEEE ISR 2013*, 2013, pp. 1–3.
- [11] F. Xi, "ME8138 Computational Dynamics: Lecture Notes." Ryerson University, 2008.
- [12] B. Siciliano, Ed., *Robotics: modelling, planning and control*. London: Springer, 2009.
- [13] L. X. Xie, Z. X. Tian, J. Y. Kong, G. Z. Jiang, and G. F. Li, "Kinematic Analysis of a Spatial 6-Bar Linkage Mechanism," *Appl. Mech. Mater.*, vol. 496–500, pp. 831–835, Jan. 2014.
- [14] Y. Liu and S. Yang, "Kinematic solution of spherical Stephenson-III six-bar mechanism," *Chin. J. Mech. Eng.*, vol. 26, no. 5, pp. 851–860, Sep. 2013.
- [15] L. Gao and W. Wu, "Forward kinematics modeling of spatial parallel linkage mechanisms based on constraint equations and the numerical solving method," *Robotica*, vol. 35, no. 02, pp. 293–309, Feb. 2017.
- [16] X. Kong, "Kinematic analysis of a 6R single-loop overconstrained spatial mechanism for circular translation," *Mech. Mach. Theory*, vol. 93, pp. 163–174, Nov. 2015.
- [17] X. Wei-Liang and Z. Qi-Xian, "Probabilistic analysis and Monte Carlo simulation of the kinematic error in a spatial linkage," *Mech. Mach. Theory*, vol. 24, no. 1, pp. 19–27, 1989.
- [18] S. B. Nokleby, R. Fisher, R. P. Podhorodeski, and F. Firmani, "Force capabilities of redundantly-actuated parallel manipulators," *Mech. Mach. Theory*, vol. 40, no. 5, pp. 578–599, May 2005.
- [19] D. Zhang and J. Lei, "Kinematic analysis of a novel 3-DOF actuation redundant parallel manipulator using artificial intelligence approach," *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 1, pp. 157–163, Feb. 2011.
- [20] Chinmay S. Ukidve, Farhad Jafari, and John E. McInroy, *Quantifying and Optimizing Failure Tolerance of a Class of Parallel Manipulators*. INTECH Open Access Publisher, 2008.
- [21] J. F. O'Brien and J. T. Wen, "Redundant actuation for improving kinematic manipulability," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, 1999, vol. 2, pp. 1520–1525.

- [22] A. Müller and P. Maisser, “Generation and application of prestress in redundantly full-actuated parallel manipulators,” *Multibody Syst. Dyn.*, vol. 18, no. 2, pp. 259–275, Aug. 2007.
- [23] H. Cheng, G. F. Liu, Y. K. Yiu, Z. H. Xiong, and Z. X. Li, “Advantages and dynamics of parallel manipulators with redundant actuation,” in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 2001, vol. 1, pp. 171–176.