Kinematic Optimization of a Four-Bar Linkage for Continuous Eggshell Membrane Extraction Using Multi-Objective Optimization and Reinforcement Learning

Vincent Robin Patrick Kelly

A thesis submitted to the University of Ottawa in partial fulfillment of the requirements for the degree of Master of Applied Science in Mechanical Engineering

Department of Mechanical Engineering
Faculty of Engineering
University of Ottawa

uOttawa

Abstract

Eggshell membrane is a valuable by-product of the egg-breaking industry, rich in bioactive proteins and polysaccharides used in pharmaceutical and cosmetic products. Conventional separation methods for extracting membranes from eggshells using water or acid baths are inefficient and unsuitable for continuous operations. To address this, a mechanical extraction method can be used for direct integration into egg-breaking machines. A design was proposed to use a four-bar linkage based on the Chebyshev–Lambda mechanism to insert the extraction tool into eggs moving on constant-velocity conveyors typically found on egg-breaking machines. This thesis focuses on optimizing the prototype's linkages through a structured study with five objectives: minimizing velocity fluctuations of the output link effector, achieving path-generation using Fourier harmonic decomposition, optimizing link length magnitudes, and improving the insertion-to-exit velocity ratio of the extraction tool on the effector. Reinforcement learning was evaluated alongside classical global optimization for path-generation. The multi-objective study produced results that included a configuration with an 87% reduction in velocity variation and a 40% increase in theoretical extraction rate. The results were evaluated in a correlation study to assess the relative competitiveness of the different objectives with respect to velocity fluctuation.

Acknowledgements

I would like to thank my supervisors, Dr. Eric Lanteigne and Dr. Maxwell Hincke, for their continued support, guidance, and patience throughout this research. They have inspired me to be more inquisitive and ambitious. I would like to thank my parents for their love and encouragement throughout my academic career; without them, I would not be in this fortunate position. I would also like to thank my partner for always being there for me, my friends for their support, and everyone who patiently listened to my discussions about optimization models and chicken eggs.

Table of Contents

List of Figures

List of Tables

Abbreviations

CMA-ES	Covariance Matrix Adaptation Evolution Strategy
DDPG	Deep Delayed Deterministic Policy Gradient
DE	Differential Evolution
ESM	Eggshell Membrane
FD	Fourier Descriptor
FFNN	Feed-Forward Neural Network
GA	Genetic Algorithm
GS	GlobalSearch
ML	Machine Learning
MOGA	Multi-Objective Genetic Algorithm
NN	Neural Network
PBO	Policy-Based Optimization
PP	Prismatic-Prismatic
PS	Pattern Search
PSO	Particle Swarm Optimization
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
RP	Revolute-Prismatic
RR	Revolute-Revolute
RMS	Root-Mean-Square
SA	Simulated Annealing
TD3	Twin Deep Delayed Deterministic Policy Gradient
VAE	Variational Autoencoder

Chapter 1

1 Introduction

Linkage-based mechanisms have been used for centuries, and their design is among the oldest problems in mathematics and mechanical engineering. Linkage mechanisms can transform simple angular motions into complex linear motions. For example, they can be powered by a single motor and convert the motor's rotary motion into elliptical curves containing straight-line segments, such as in the Chebyshev-Lambda linkage [1]. These elegant systems are mechanically simple, but the highly nonlinear equations governing their kinematics complicate their design.

This research aims to define the parameters and study the kinematics of a mechanism for use in the chicken egg-breaking industry. Multiple researchers and companies have identified that eggshell membranes (ESM) contain essential compounds for biomedical applications, nutraceutical supplements, skincare, and cosmetics [2]. Furthermore, a technological gap in the current ESM extraction processes used in industry has motivated Dr. Hincke and Dr. Lanteigne of the University of Ottawa to design and patent a technology to extract ESM mechanically [3].

1.1 Motivation and Research Problem

ESM is a paper-thin proteinaceous membrane between the eggshell and albumen of all bird eggs. The chicken ESM, shown in Figure 1, is made up of more than 500 different types of highly cross-linked proteins, such as collagen [4]. It also contains other bioactive compounds, such as hyaluronic acid, chondroitin sulphate, and glucosamine, and forms an antimicrobial protective layer to shield the egg's contents. The bioactive compounds contained in ESM are highly sought after. They are used for various biomedical applications, including health supplements, wound-healing dressings, tissue-regeneration scaffolds, antimicrobials, anti-inflammatories, biosensors, and cosmeceuticals. Furthermore, ESM is one of the only vegetarian collagen sources, which opens a significant untapped market for oral collagen supplements [2].

Figure 1. Example of ESM being manually extracted [5]

Currently, ESM is mainly a waste by-product of the poultry egg industry, which is a massive global industry with a production scale of over 90 million metric tonnes [6] and over $220 billion USD [7]. Nearly 30% of chicken eggs are processed by breaking machines in egg-breaking facilities, where machines crack and empty the eggs and then separate the yolks from the albumen [8]. Cracked eggshells with the attached ESM are usually discarded and ground up to be used as livestock feed additives and fertilizers or disposed of in landfills, which are practices that largely ignore the ESM's valuable biomedical potential. Systematic extraction of the membrane's bioactive compounds could yield significant economic benefits. The recovery and use of ESM in biomedical applications can thus contribute to a circular and environmentally sustainable economy.

The economic viability of ESM as a biomedical ingredient relies on the quality and yield of its separation techniques. The most well-established separation techniques operate on pulverized eggshell and ESM from egg-breaking machines. The eggshell and ESM are then washed, suspended in a fluid, and subjected to various techniques to induce separation, such as cavitation of their suspension fluid, being passed through a cyclone tank, vacuum-induced flash-steaming of the suspension fluid, heat treatment and grinding, and suspension in specific gravity water baths [9–16]. These techniques share one commonality: they are separate from the egg-breaking and liquid-extraction processes, which can be a drawback when aiming to maximize the efficiency of ESM and liquid egg harvesting. Furthermore, these processes require a significant amount of process water, generate chemical waste, employ heating and/or pH adjustment techniques that may alter the chemistry of the ESM, and involve multiple complex fluid-dynamic or chemical processes that must be tightly controlled to prevent the destruction ESM and achieve maximal yield.

These disadvantages present a technological gap, which motivated Dr. Lanteigne and Dr. Hincke of the University of Ottawa to form a research group to pursue the design and patent a novel technology to extract ESM mechanically [3]. A vacuum ESM extraction head was prototyped and tested on prepared and emptied eggshells. To mechanically extract the ESM, the sharp end of the egg is removed, the broad end is perforated to create a pressure relief hole, and the vacuum tool is inserted into the egg until it reaches the base, where the membrane latches onto the vacuum tool due to the induced pressure difference. The membrane is then extracted by pulling it out vertically at a constant vertical velocity to prevent tearing.

The proposed process of mechanically separating the eggshell membrane from the eggshell using a specially designed vacuum head is delicate, and the team needed to validate whether the extraction could be performed using an automated machine. To achieve this, the low-cost proof-of-concept prototype, shown in Figure 2, was designed with a constant-velocity conveyor and a planar four-bar Chebyshev-lambda style linkage mechanism. This type of linkage was selected because of its quasi-constant velocity path portion, which was synchronized with the motion of the egg, such that the linkage inserts the extraction tool into the egg, matches the egg and conveyor speed, contacts the bottom where the membrane is latched on to and pulled out, all while maintaining near-constant horizontal velocity, and returns to the starting position. This single-degree-of-freedom planar linkage mechanism was also chosen because it is low-cost, requires only a single motor to drive it, and is highly repeatable [17].

It was specified that this extraction process could occur within an egg-breaking machine, which typically uses open-loop-controlled constant-velocity carrier systems under a continuous duty cycle; therefore, a constant-velocity conveyor was chosen to handle the prepared eggs. The constant-velocity conveyor and linkage are driven by two motors, each controlled by a manually synchronized open-loop motor driver.

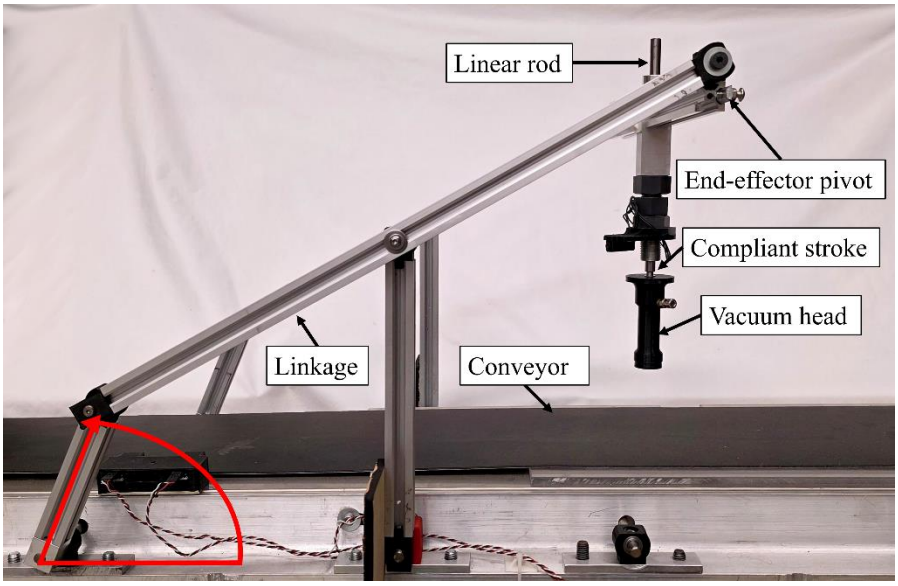

Figure 2. Mechanical ESM extraction prototype built using a four-bar linkage and constant velocity conveyor, with the principal elements and vacuum tool components labelled

Dimensional synthesis of the proposed linkage was initially conducted using a function minimizer algorithm in MATLAB and heuristics to develop an adequate linkage design for the proof-of-concept. Ultimately, the linkage design was suboptimal, and two degrees of adjustability were added to the vacuum tool using a linear rod with bushings to improve vertical alignment and timing. An end-effector pivot, annotated in Figure 2, was also added to help keep the tool within the egg in the event of misalignment.

The toolhead A), aperture cutting tool B), and their major dimensions are shown in Figure 3. The outer diameter of the tool head is 26.4mm, and the inner diameter of the cutter is 32 mm. Therefore, if it is perfectly aligned with the center axis of the egg, there is a 2.8 mm gap on either side of the toolhead as it enters the egg. This translates to a maximum permissible relative center-axis offset at insertion of 2.8 mm between the tool and egg. During the insertion phase, if the offset exceeds the 2.8 mm threshold, the toolhead may become stuck outside the egg, and the cycle fails.

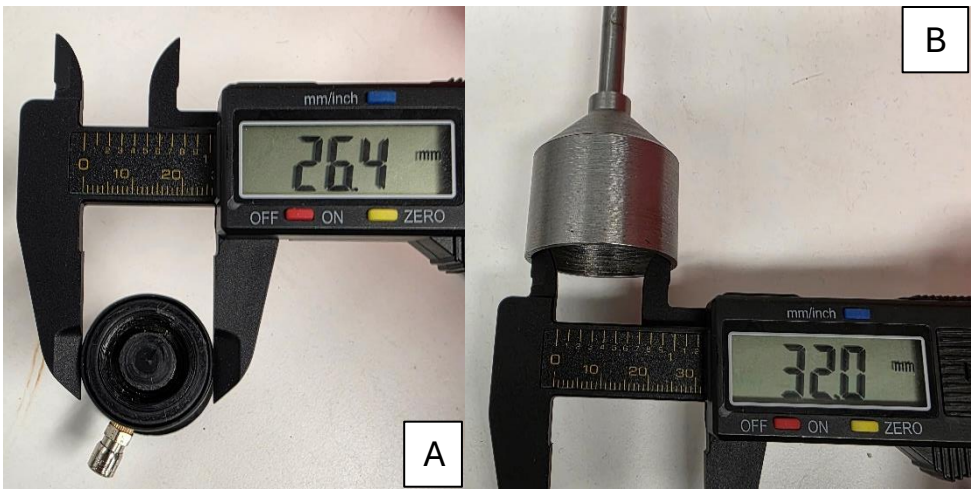

Figure 3. A) Toolhead and B) aperture cutter, and their major dimensions

This misalignment offset is also significant during extraction. It was observed that if this offset is too large at extraction, the membrane can tear, as shown in Figure 4.B, which is also a failed cycle. The effects of misalignment during extraction are highlighted in Figure 4.A by the blue arrows, which highlights the gap caused by the misalignment and the tool exiting the egg near its leading edge rather than being aligned with the egg's vertical axis. The green and red dotted lines denote the tool's angle upon exit and demonstrate the importance of the end-effector pivot. This misalignment results from the tool's horizontal velocity fluctuating relative to the egg's constant horizontal velocity and represents a significant factor limiting system reliability.

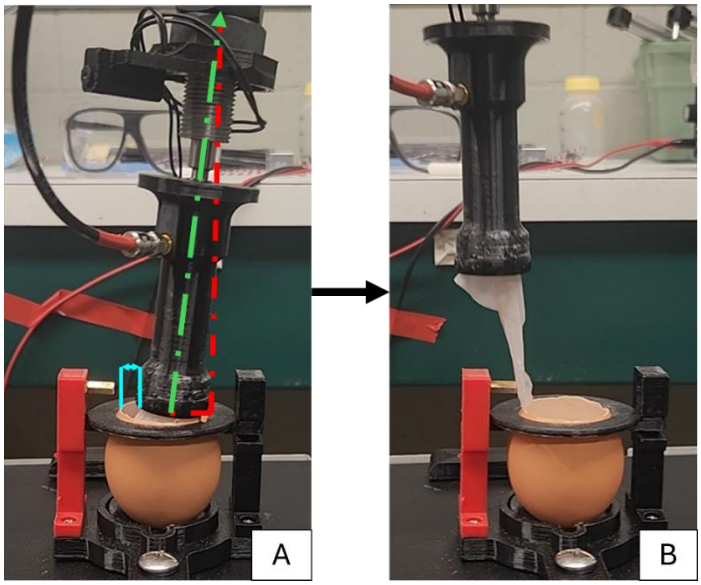

Figure 4. A) Tool misalignment at exit and B) membrane tear

Furthermore, during the development of the proof-of-concept linkage mechanism, it became evident that real-world applications, particularly in industrial machinery, require consideration of manufacturing and assembly constraints, as well as dynamic effects, when defining optimization objectives. For instance, counterweights were added to the prototype to mitigate the negative torque acting on the drive motor. Although dynamic metrics such as input torque, shaking moment, and shaking force are well established in the literature and have been extensively optimized, they are not addressed in the present work. Instead, this thesis focuses on kinematic optimization objectives described in the following sections, while dynamic optimization is reserved for future work.

The proof-of-concept prototype's Chebyshev-lambda linkage, shown in Figure 2, is referred to as the "*Baseline*" linkage and is graphed in Figure 5. The red link is the crank arm, the turquoise link is part of the coupler–rocker–end-effector triangle that connects from the extremity of the crank arm to the end-effector and is covering the coupler in this configuration, the blue link is the rocker, the black link is the ground link and extends to the origin, and the pink link is the other extremity of the coupler–rocker–end-effector triangle that connects from the rocker to the end-effector. The coupler link, which connects from the extremity of the crank arm to the rocker, is green and appears in linkage configurations in which the coupler–rocker–end-effector triangle has all non-zero internal angles, which is not the case for the Chebyshev-lambda linkage. The *Baseline* linkage coupler curve is traced in orange, and the new desired path for this thesis' trajectory optimization is overlaid as blue dots. The new desired path will be defined in detail in section 3.2 of the methodology.

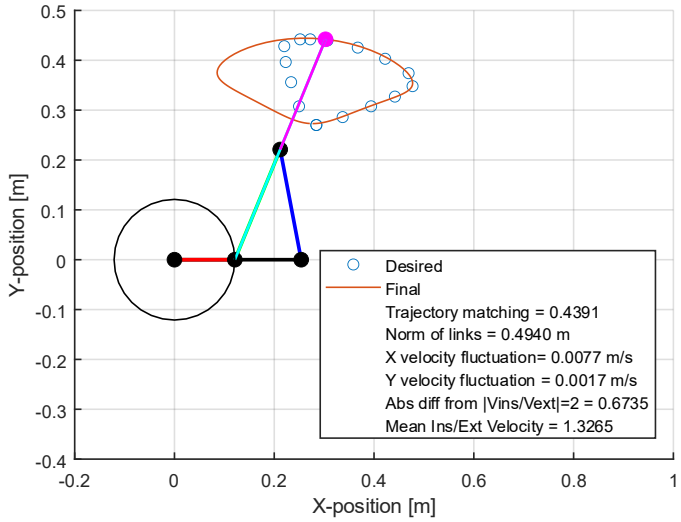

Figure 5. The Baseline linkage with performance metrics shown

The figure also displays the *Baseline* linkage's five different objective function values. The *Baseline* has a low desired path trajectory matching of 0.4391, high X-velocity fluctuations, which contribute to the tool misalignment mentioned above, moderate Y-velocity fluctuations, and a low insertion-to-extraction velocity ratio. However, it benefits from short links. The path's efficiency, defined as the rate of ESM extraction per second, was not considered during the initial linkage design and remains an area for improvement addressed in this thesis.

1.2 Objectives and Contributions

Improving the system reliability limitations observed in the prototype linkage design constitutes the primary objective of this research. These design limitations will be addressed in a multi-modal optimization study based on these five optimization objectives, grouped under two design goals:

ESM extraction reliability

- Minimize horizontal velocity fluctuation during membrane extraction
- Minimize vertical velocity fluctuation during retraction

Device efficiency

- Match a shorter and more efficient desired coupler curve shape
- Minimize the total link lengths
- Maximize the ratio of vertical insertion to retraction velocities during extraction

Reliability is improved by reducing fluctuations in end-effector velocity when the tool is within the egg. Specifically, horizontal velocity variation is minimized to maintain alignment with the conveyor, which has been shown to reduce the risk of membrane tearing during extraction, and vertical velocity variation is minimized to reduce unnecessary loads on the membrane during retraction. Efficiency is enhanced by matching the linkage trajectory to a shorter, more effective path, thereby reducing overall link lengths to decrease mass and improve dynamic performance, and by increasing the insertion-to-retraction velocity ratio to minimize cycle time.

This thesis will investigate multiple optimization methods for meeting these objectives. First, classical global optimization will be compared with Reinforcement Learning (RL) to determine whether neural networks (NNs) can perform path-generation optimization corresponding to the first device efficiency objective listed above.

Second, a multi-objective optimization study will evaluate the objective functions in progressively increasing combinations to determine whether all five design objectives can be simultaneously optimized, and to identify a solution that maximizes both ESM extraction reliability and device efficiency. The results will also be evaluated in a correlation study to determine the relative competitiveness of the different objectives with respect to X-direction velocity variation.

The scientific contributions of this thesis include:

• Evaluated the feasibility of reinforcement learning as an alternative optimization approach for linkage synthesis by comparing its performance with traditional local and global optimization methods.

• Developed a multi-start reinforcement learning framework aiming to address global optimization challenges in linkage design.

• Demonstrated a strong relationship between X-direction velocity fluctuation and RMS acceleration for the lower portion of the four-bar linkage coupler curve and established the redundancy of including both metrics as simultaneous optimization objectives for constant-velocity motion-based operations.

• Quantified the extent to which the kinematics of a four-bar linkage can be optimized to achieve reliable and efficient motion-based operations on objects moving at a constant velocity.

• Quantified the sensitivity and the relative competitiveness of multiple optimization objectives with respect to X-direction velocity variation, providing guidance for objective selection in future four-bar linkage kinematic optimizations.

• Identified a connection between the vertical-axis symmetry of the coupler curve and the attainment of near-constant velocity in the lower portion of the four-bar linkage coupler trajectory.

Chapter 2

2 Literature review

2.1 Background on Dimensional Synthesis of Linkages

Dimensional synthesis is a subcategory of kinematic synthesis in which a mechanism's dimensions are determined to achieve a specified motion. In contrast, type synthesis identifies the type of mechanism, including linkage, cam, gear, or joint types, while number synthesis defines the number of links or joints needed to accomplish a motion task [18]. This research concentrates on the dimensional synthesis of planar four-bar linkages. Furthermore, dimensional synthesis problems can be classified into three groups: function generation, motion generation, and path generation.

2.1.1 Function Generation

Function generation aims to synthesize linkages to generate a specific input-output relationship between the crank and follower. Ferdinand Freudenstein's research, published in his 1954 Ph.D. dissertation, outlines the first attempt to solve the function generation problem analytically for planar four-bar mechanisms [19]. He developed the Freudenstein Equation to synthesize function-generating four-bar linkages for three input-output angle pairs.

Function generation problems can also be solved for more than three angle pairs but become overconstrained and require optimization techniques. These optimization problems often attempt to reduce two key factors: structural error — the difference between desired and generated output motion [30], and design error — the arithmetic residual obtained when evaluating the function-generation synthesis equations [20], [21]. These key factors are considered in the research on multi-objective optimization of function generators, where additional factors and constraints are considered, such as mechanical error related to coupler to output link transmission angle, relative link length, machining tolerance, velocity and acceleration, linkage defect mitigation [22–28]. Algorithms and formulations for dimensional synthesis have introduced refinements but have largely remained consistent with the classical Freudenstein approach [29–32].

However, recently, kinematic mapping in the Study's image space has been used to develop algebraic input-output [18,33–35] equations between link angles, facilitating novel continuous approximate synthesis methods for function generation of planar four-bar mechanisms, as well as spherical and spatial four-bar mechanisms [21].

2.1.2 Motion Generation

Motion generation, or rigid-body guidance, was first defined and investigated by Ludwig Burmester in his 1888 work *Lehrbuch der Kinematik* [36]. In motion generation, the geometric parameters of four-bar linkages are calculated such that the synthesized linkage's coupler attains a set of finitely separated poses [37]. A pose consists of the position and orientation of a rigid body, or in this case, a rigid body attached to a coupler link [38]. This problem was initially solved graphically for up to three poses. It can also be solved graphically for four and five poses, but the problem becomes much more complicated and advanced analytical techniques are required.

Sandor and Erdman developed synthesis equations using loop-closure and complex number representation to solve the Burmester problem for two to five poses [39]. These equations can yield closed-form solutions to the five-pose problem, yielding zero, two, or four real revolute-revolute (RR) dyads. Dyads consist of two links connected by a revolute or prismatic joint, with each dyad representing half of a four-bar linkage. For four-pose problems, these synthesis equations produce an infinite number of solutions, whereas for six or more poses, the problem requires approximate solutions. The four-pose problem is solved using cubic centerpoint and circlepoint curves. These are formed by parameterizing the solution space of the synthesis equation and represent the loci of fixed ground-joint locations for the dyads. Any corresponding pair of points from the centerpoint curve and circlepoint curve can be used to form a dyad of the linkage.

Since then, considerable efforts have been made to enhance understanding of the Burmester and approximate Burmester problems by finding solutions for special cases of the four and five-pose problems, resulting in prismatic-revolute (PR) dyads [40] and subsequently extending the solution to RP and PP dyads, thus forming a comprehensive solution that encompasses all four planar dyads [37]. Furthermore, kinematic mapping has proven to be a powerful tool for solving synthesis equations by forming hyperboloids in the image space generated by the five-pose quadratic constraint equations [41]. It has also been employed to identify potential linkage defects in the solutions [42], to lay the groundwork for solving the approximate Burmester problem for all 27

types of planar 4R linkages [43], and to provide an approach for addressing the extended Burmester problem for six or more poses [44].

2.1.3 Path Generation

Path generation is the third form of dimensional synthesis and is the primary focus of this research. This synthesis aims to design a linkage to guide a point on the linkage through a set of precision points or along a curve in the linkage plane. The specific point is often located on the coupler link since points on the coupler trace curves in the fixed frame as the linkage moves. The basic form of the coupler curve is an ellipse [45] but it has been shown that almost any algebraic curve can be formed by the coupler curve [46] and that coupler curves are sextic curves whose shape is related to link dimensions and assembly method [45,47,48]. Furthermore, the Rober-Chebyshev theorem states that three linkage cognates of different geometries produce the same coupler curve [47], further complicating the definition of exact solutions to dimensional synthesis problems.

The planar four-bar linkage path generation problem for five precision points was first analytically solved in closed form by G.N. Sandor [49]. His method uses complex-number representation, modelling each link as a vector in the complex plane. A vector loop closure equation is formulated for the linkage's configuration. The loop-closure equation is then formulated for each of the five precision points, yielding a system of nonlinear algebraic equations. These equations are reduced to a single univariate polynomial of up to 36 degrees through systematic algebraic manipulation and can be solved using analytical and, more frequently, numerical root-finding techniques. The real roots of this polynomial represent link lengths that can then be back-substituted into the loop closure equations to determine the remaining real-valued linkage parameters.

Polynomial continuation has also been used to solve the five-point high-degree loop closure polynomial [50,51]. Recently, the complex number method has been expanded upon by using harmonic analysis of the loop closure equations with and without prescribing timing [52,53]. Thus, the linkage parameters correspond to the Fourier coefficients of the closed path. This results in two low-complexity polynomials that can be solved for the linkage parameters.

Another analytical method for solving the path generation problem is presented in [54] and further improved upon in [48]. These methods utilize the tricircular sextic form of the parameterized four-bar coupler curve equation, a novel algebraic formulation of the equations derived from its coefficients, and a sequential calculation to address the path-generation problem. The sextic

coupler curve equation comprises 15 terms and is derived in [45]. Conversely, a four-bar linkage has nine design parameters. Therefore, the problem has historically been deemed over-determined. These two novel methods [48,54] first demonstrated the system's determinacy, then simplified the system of equations obtained from the coupler curve equation's coefficients to a system of four equations and four variables. The solution to this system represents the coordinates of the linkage's ground link. The remaining five parameters are then determined by back-substituting these four coordinates into the coefficient equations. This novel formulation and sequential calculation thus represent an exact analytical solution to the path-generation problem, provided that the desired coupler path is in the tricircular sextic algebraic form.

Furthermore, Hayes and Rotzoll recently derived a novel method for deriving an algebraic parametric coupler curve for planar four-bar revolute linkages [55]. Their method is based on their novel IO linkage angle equations [18,34], which are derived using Study's kinematic mapping, and the homogeneous transformation matrix of the open 2R kinematic chain formed by the crank and coupler. Since this method utilizes an algebraic and analytical framework provided by homogeneous transformation matrices, Study's kinematic mapping, and associated algebraic geometry tools, it may be feasible to invert the problem and solve for linkage dimensions and design parameters from desired path or coupler curve equations. This would thus be another exact solution to the path generation problem, provided that the desired coupler path is in the same algebraic form as their novel algebraic parametric coupler curve.

2.2 Local and Global Optimization

Optimization techniques have been widely studied for their ability to handle path-generation problems with more than five precision points, manage complex design formulations, and avoid the need for high-order polynomial root-finding or fitting tricircular sextics to precision points. This is because local and global optimization techniques are well-suited to the nonconvex, nonlinear, and continuous nature of this type of problem.

Optimization techniques fall under two categories: local and global optimization. Local optimization techniques, such as MATLAB's fmincon [56], are gradient-based and best suited to smooth problems with a single well-defined maximum. These techniques have been used to find optimal linkage designs [57–61]; however, research is increasingly using multi-start or metaheuristic global optimization methods. Metaheuristic techniques are better suited for optimal

approximate path-generation problems because they can handle the problem's nonlinear, implicit objective functions, which often contain multiple local minima [62]. These techniques allow the designer to confidently determine a near-globally optimal solution without evaluating the search space's continuity and convexity.

Metaheuristic global optimization can find optimal solutions to nonconvex problems by leveraging advanced search space exploration techniques often inspired by nature. Population-based techniques such as genetic algorithm (GA), differential evolution, particle swarm optimization (PSO), and simulated annealing (SA) are some of the most popular nature-inspired metaheuristic optimization algorithms.

2.2.1 Multi-Start Algorithms

Multi-start algorithms extend local optimization techniques by sampling the search space from multiple starting points. This reduces the likelihood of becoming trapped in local a minimum by introducing stochasticity into an otherwise purely gradient-based algorithm. MATLAB's GlobalSearch (GS) algorithm is an example of a multi-start algorithm that uses scatter-search to explore the search space, rather than uniformly sampling the search space or using user-defined start points [63]. Scatter-search combines and refines diverse sample points to identify promising regions and better explore the search space. These techniques excel in solving locally smooth, non-convex problems with continuous variables and medium to low dimensionality.

2.2.2 Genetic Algorithms

GAs leverage elements of population genetics and evolution to solve combinatorial and continuous optimization problems characterized by large, computationally expensive search spaces. Thus, they are well equipped to solve multi-parameter design problems like four-bar linkage synthesis [64–68].

2.2.3 Differential Evolution

Differential evolution is another global optimization method that draws on elements of a population's genetic evolution but differs from GA in how it performs genetic operations, such as reproduction and mutation. Storn and Price first presented differential evolution as a simple yet effective global optimizer that was more robust and less sensitive to hyperparameter tuning than GA and other evolutionary algorithms [69,70]. Differential evolution is widely used in linkage design [71–76]. Comparative studies by Acharyya and Mandal, and by Kang et al. have shown that

DEs produce better optimal multi-precision point path generation linkage synthesis results than other population-based algorithms [77,78].

2.2.4 Particle Swarm Optimization

PSO, pioneered by Kennerdy, Eberhart, and Shi [79–81], has also been used to solve the four-bar linkage path generation problem [82–84]. This method mimics insect swarm behaviour and relies on particle velocity to explore the search space. It also records the swarm's collective memory, storing information about the best results.

2.2.5 Simulated Annealing

SA was first described and applied to NP-hard problems by Kirkpatrick et al. [85], Burkard and Rendl [86], and Cerny [86]. It is a metaheuristic optimization algorithm based on the physical process of metal annealing, which involves slowly cooling metals from high temperatures, thereby bringing their atoms to a more stable, lower-energy crystalline state. This optimization technique has also been successfully applied to the planar four-bar linkage optimal path generation problem [87–89].

2.3 Fourier Descriptors

The traditional objective function used for path generation and employed by most optimization techniques above is the sum of squares of the Euclidean distance between desired (target or precision) points and points produced by the generated coupler curve [58,73,76,82,88], also known as the structural error function or tracking error [90]. The structural error equation is shown in (1) where $\boldsymbol{x}$ is the vector of design variables, x_{di} and y_{di} are the ith desired points, x_{gi} and y_{gi} are the ith generated points, and N is the number of points [73,82,91].

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \sqrt{\left(x_{gi} - x_{di}\right)^2 + \left(y_{gi} - y_{di}\right)^2} \qquad (1)$$

The structural error function compares corresponding points on the desired and generated curves solely by their indices, thereby implicitly assuming that equal indices correspond to the same geometric location along the path. As a result, it can attribute error to differences in parametrization or timing rather than to the actual geometric discrepancy between the two trajectories, and it offers no straightforward way to normalize the paths consistently before comparison, which further limits its effectiveness as an objective function [91].

Devising a more effective method for comparing coupler curves has been the subject of multiple studies. One method is to use Fourier descriptors (FD) to describe, normalize, and compare curves for use in optimization algorithms. McGarva and Mullineux first introduced Fourier coefficients to describe closed planar coupler curves [92]. FD can be used to compare the similarity of curve geometry by applying the normalization techniques described by McGarva and Mullineux, then summing the squared Euclidean distances between the Fourier coefficients, which are complex numbers. The smaller the sum of the distances, the better the match. By using this normalized curve comparison technique, the design space of the optimization problem is reduced from nine linkage parameters to five, thereby removing the influence of linkage pose and simplifying the design space [91]. FD have also been used in more recent works in combination with various optimization techniques [93–95].

FD have also been used to formulate novel linkage-synthesis problems for path generation. Wu employed FD to create a novel formulation that solves the problem for both open and closed paths using a least-squares curve fitting scheme [90]. Sharma improved upon this technique by using an optimal time parameter matching scheme, rather than the traditional uniform time parameter values [96]. Li and Qian both formulated analytical techniques using FD to create synthesis polynomials that can be solved to find the linkage parameters [52,53]. Other methods can also be found in the literature, such as using differential objective functions, where the first and second derivatives of the desired and resultant paths are compared [97,98], as well as an objective function based on the normalized Hausdorff distance between paths [99].

2.4 Machine Learning

The normalized FD of coupler curves, mapped to their linkage parameters and generated through systematic variations in link length and coupler point position, creates an ideal dataset for machine learning (ML). Vasiliu and Yannou employed a dataset of 30,000 FD-linkage parameter pairs to train a feedforward neural network (FFNN) with two hidden layers and a sigmoid activation function [100]. Once trained, one could input a desired coupler curve's FD into the FFNN, which would then output the five corresponding linkage parameters. Khan et al. used a similar approach, where they used a 5000 FD-linkage parameter dataset to train a three-hidden-layer tan-sigmoid FFNN [101]. Li and Chen further improved upon the method using a modified arc-length parameterization for their FD, as well as a 101,700-pair dataset to train their three-hidden-layer

FFNN [102]. Other coupler curve parameterizations have also been used to train FFNNs, such as the use of a power spectrum and a radial basis function FFNN in [103], and the wavelet transform of coupler curves and a two-hidden-layer log-sigmoid FFNN [104].

Recently, advanced NN techniques and architectures have been employed to solve increasingly complex path synthesis problem formulations, such as those involving more precise solutions, multi-bar linkages, and multi-joint type linkage synthesis. These techniques included deep NNs, combined ML and optimization techniques, deep generative models, and variational autoencoder-based approaches.

Chen (2023) utilized the FD representation, 1.6M data pairs, and combined multiple deep NNs and the MFQuery method to produce a model capable of multiple synthesis solutions, which are filtered to select the best one(s) [105]. Wang et al. (2024) used a combination of an extreme gradient boosting ML (XGBoost) and genetic algorithms, as well as a relative slope coupler curve representation, for their high-precision synthesis operation [106].

Furthermore, advancements in linkage datasets have been at the forefront of state-of-the-art data-driven kinematic synthesis, as the quality and scale of the data are among the most critical factors for ML. Notably, Nobari et al. published the largest known dataset of linkage mechanisms, called LINKS, containing nearly 100 million one-degree-of-freedom planar linkage mechanisms, ranging from 4 to 20 revolute joints [107]. These linkages are expressed as undirected topology graphs, whose information is encoded into adjacency matrices, as well as an $n \times 3$ matrix (where n represents the number of joints) describing the initial position and type of each joint. Furthermore, for each linkage, the coupler curve path is simulated at 200 equally spaced crank-arm positions, normalized, and stored along with the corresponding matrices. With a streamlined version of this dataset, they created an advanced multi-bar linkage synthesis framework called LInK, which leverages contrastive learning for rapid linkage retrieval, enabling the warm start of a hierarchical unconstrained nonlinear optimization algorithm [108]. Using their massive dataset and their combined ML and optimization techniques, LInK outperformed other state-of-the-art multi-bar synthesis models in tracing advanced paths, including the eight test curves from Pan et al. [109] and tracing the 26 capital letters of the English alphabet.

Nurizada et al. have also made significant advancements in creating and publishing kinematic synthesis datasets, as well as in utilizing image data, convolutional networks, and variational

autoencoders (VAE) for kinematic synthesis. In 2025, they published a dataset of 3 million single-degree-of-freedom planar 4-, 6-, and 8-bar revolute joint linkage mechanisms, as well as 4-bar revolute-prismatic joint linkage mechanisms with both open and closed coupler curves [110]. Their dataset consisted of normalized, digitized black-and-white coupler-curve images, along with their corresponding linkage lengths and joint parameters. They then used this data to train a VAE-based model to generate multiple solutions for 4-, 6-, and 8-bar revolute linkages, as well as 4-bar linkages with both revolute and prismatic joints, to best match the desired complex open and closed paths.

Later in 2025, they expanded their initial dataset to create a 4-bar revolute-prismatic linkage dataset for open and closed path tracing [111]. This dataset contains over 12 million linkages, comprising 4,346,626 RRRR, 4,211,930 RRRP, and 4,619,098 RRPR linkages. They then used this dataset to train a novel conditional β-variational autoencoder. This ML model then generates a diverse set of revolute- and prismatic-planar four-bar mechanisms that match the given open or closed coupler curve.

2.5 Reinforcement Learning

RL has emerged as an interesting approach to kinematic synthesis, complementing traditional and advanced ML techniques. Unlike methods that depend on large, pre-labelled datasets, RL trains NNs through direct interaction with an environment, using simulated or real-world data via online learning, to tackle complex pattern recognition and sequential decision-making tasks. RL is characterized by agents, states, actions, policies, and reward signals. The agent learns to optimize its actions in a given state by repeatedly refining its decision-making policy, typically represented as an actor network. At the same time, a value function, normally also approximated by a NN, evaluates the desirability of state-action pairs by estimating the expected cumulative future rewards. This evaluation, driven by the reward signal, the scalar feedback from the environment indicating the immediate desirability of an action, directs policy and network updates [112].

Recently, RL has shown promise in solving engineering design optimization problems by learning how to navigate the search space. This is evident in Viquerat et al.'s work on the design of optimized shapes for fluid dynamics [113]. In their work, they trained a policy-based optimization (PBO) RL agent with a single-step optimization framework (degenerate RL) to create optimal

airfoil-like shapes, with the model having no prior knowledge of fluid dynamics, solely by employing a CFD-based reward function to guide training.

Gallego et al. also used Viquerat et al.'s PBO degenerate RL method for an optimal design problem involving a revolute four-bar linkage, creating mechanisms capable of tracing straight-line segments [114]. They accomplished this by employing a reward function based on the curvature of the coupler curve, whose value increases as the coupler curve segments become straighter, and an action policy based on a multivariate normal distribution, which outputs the link lengths of the linkages. Three actor networks are trained to update the policy's mean vector and covariance matrix to generate optimal linkage designs.

RL has been effectively combined with graph grammar formalisms, notably inspired by Lipson et al.'s generative design principles for kinematic synthesis [115]. In 2018, Vermeer et al. applied this graph grammar approach, using feature-based RL, to generate and expand four-bar linkages through the sequential addition of nodes and edges, demonstrating the technique's ability in synthesizing mechanisms for tasks such as tracing straight-line and figure-eight paths [116]. Building upon this foundation, in 2022, Fogelson et al. advanced this method for multi-bar mechanism synthesis by introducing the ability to incorporate 0-DOF Assur groups onto the linkages, thereby enabling the creation of designs capable of producing complex paths [117]. Fogelson's algorithm employs a graph-convolution-based deep RL approach that leverages graph convolutional networks to extract relevant topological features and learn an optimal sequential generative decision-making policy that matches complex paths.

2.6 Multi-objective optimization

Multi-objective optimization produces a Pareto front of Pareto-optimal design configurations when the objectives are conflicting, which is frequently the case in engineering design. A configuration is said to be Pareto-optimal if no other configuration can improve one objective without worsening at least one other objective [118]. The Pareto-optimal set comprises all Pareto-optimal solutions and is defined by their corresponding design variable values. When these solutions are mapped into the objective space, they form the Pareto front, which consists of the objective function values associated with each Pareto-optimal solution and can be graphically represented. The Pareto front typically appears as a curve or surface and illustrates the trade-offs among conflicting objectives. A feasible design space $S \in R^3$, its objective value space $Z \in R^2$, and

its Pareto front, which contains the Pareto optimal objective vector z*, are shown in Figure 6. One can observe that the left extremity of the Pareto front of Z occurs when worsening (i.e., increasing) the z_2 objective; any further worsening of z_2 would also worsen z_1. The right extremity ends because further worsening the z_1 objective would also worsen z_2. Designers can then use the Pareto front graph to pinpoint the design that best suits their needs, rather than relying on heuristics or weighted-sum optimizations.

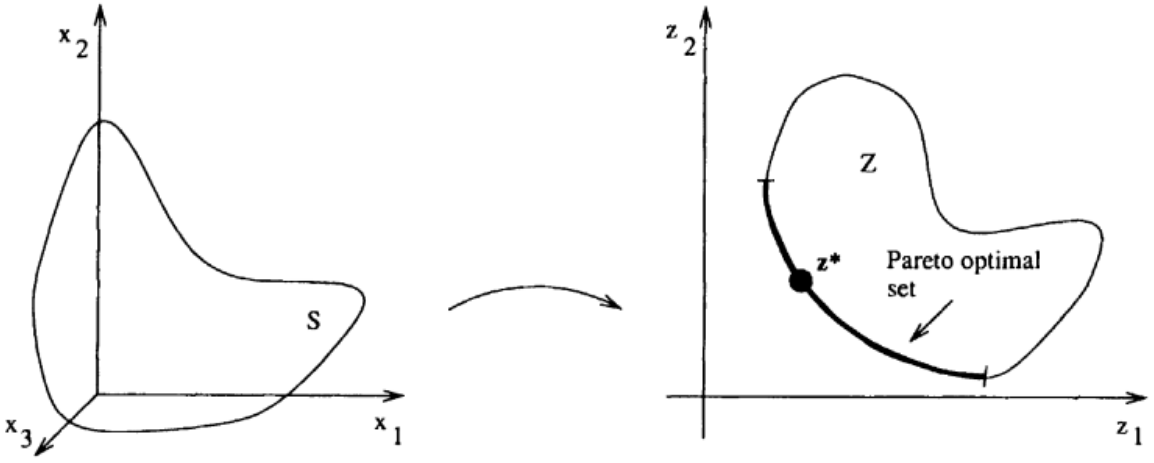

Figure 6. Pareto optimal set S, Pareto front Z, and Pareto Optimal vector z^*[118]

Multi-objective optimization has been employed to optimize four-bar linkages for various applications. These optimizations have focused on path-following and torque transmission problems, where tracking error, transmission angle, and torque input-output ratios have been conflicting objectives that were balanced using multi-objective genetic algorithms (MOGA) [119] and MOGA combined with local search [120].

Four-bar linkage shaking-force and shaking-moment balancing problems have also been solved using MOGA, as well as multi-objective differential evolution and multi-objective particle swarm algorithms [121,122]. These problems have been addressed by introducing and optimizing counterweights for established linkage parameters [123,121], as well as by directly optimizing the links' inertial parameters by adjusting their shape [122,124].

Multi-objective optimization has also been used to optimize Chebyshev-lambda linkages for walking robots by optimizing step length and height, as well as reducing step speed fluctuation and maximum acceleration using MOGA [125].

2.7 Summary and Connection to the Proposed Methodology

The optimization and machine learning techniques discussed in the literature review establish the foundation for this thesis's methodology. Initially, local and global optimization algorithms from MATLAB's local and global optimization toolboxes will be compared to identify the most effective for the four-bar path matching problem. These include GS, a multi-start method; fmincon, a local constrained optimizer; and metaheuristic algorithms such as GA, PSO, SA, and Pattern Search. Furthermore, the path-matching optimizations will utilize a Fourier Descriptor-based objective function.

Secondly, the most effective classical global optimization technique will be compared with Reinforcement Learning (RL) to assess whether neural networks (NNs) can perform path-matching and whether RL provides an effective optimization approach. The agents employed will be Twin Delayed Deep Deterministic Policy Gradient (TD3) and Delayed Deep Deterministic Policy Gradient (DDPG) from MATLAB's Reinforcement Learning toolbox.

Finally, a multi-objective optimization study will be conducted using MATLAB's MOGA from the Multi-Objective Optimization Toolbox. This will evaluate the objective functions across progressively combined sets to determine whether all five design objectives can be optimized simultaneously and to identify a solution that maximizes both ESM extraction reliability and device efficiency.

Chapter 3

3 Methodology

3.1 Planar Four-Bar Linkages Kinematic Model

This chapter begins with kinematic modelling of the planar four-bar linkage, followed by the formulation of global and multi-objective optimization problems, including the general structure, constraints, objectives, and an overview of the optimization procedure. The RL framework is then described, outlining the definitions of the state, action, and reward, the NN architecture, and the overall training procedure. Finally, the post-optimization geometric transformations applied to the linkage, including scaling, rotation, and translation, are explained.

The planar four-bar linkage model forming the basis of this optimization consists of a crank arm vector $\boldsymbol{R}_{12}$ of length r_{12} between node 1 and 2, a coupler vector $\boldsymbol{R}_{23}$ of length r_{23} from node 2 to 3, a rocker $\boldsymbol{R}_{34}$ of length r_{34} from node 3 to 4, and a fixed link $\boldsymbol{R}_{41}$ of length r_{41} from node 4 to 1. Additionally, there is the coupler triangle ($d_{35x'}$, $d_{35y'}$) that connects node 2 to the end effector node 5 and is formed by vector $\boldsymbol{R}_{25}$ of length r_{25}.

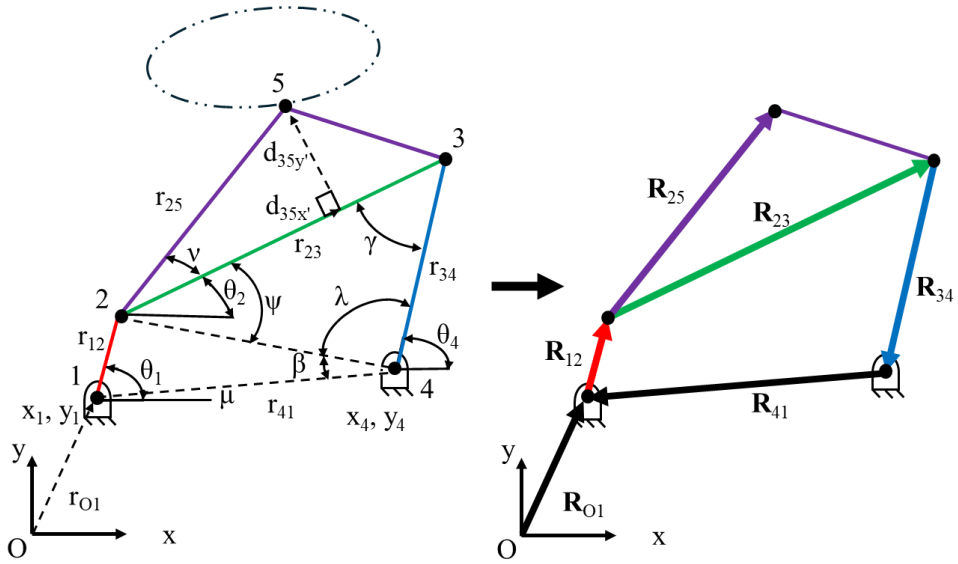

Figure 7. Four-bar linkage position analysis (left) and vector (right) diagrams

Using these vectors, the loop closure equations (2), (3) and (4), between the origin and point 5, are established.

$$\boldsymbol{R}_{O1} + \boldsymbol{R}_{12} + \boldsymbol{R}_{25} - \boldsymbol{R}_{O5} = 0 \tag{2}$$

$$r_{O1x} + r_{12}\cos\theta_1 + r_{25}\cos(\theta_2 + \nu) = r_{O5x} \tag{3}$$

$$r_{O1y} + r_{12}\sin\theta_1 + r_{25}\sin(\theta_2 + \nu) = r_{O5y} \tag{4}$$

Equations (3) and (4) can be rewritten as a function of only the crank angle θ_1 using the joint angle relationships **Error! Reference source not found.** to (9) below:

$$\theta_2 = \Psi - \beta + \mu \tag{5}$$

where θ_2 is the angle between $\boldsymbol{R}_{23}$ and the horizontal, Ψ is the angle between $\boldsymbol{R}_{23}$ and the fictitious link $\boldsymbol{R}_{24}$, β is the angle between $\boldsymbol{R}_{24}$ and $\boldsymbol{R}_{41}$, and μ is the static angle between $\boldsymbol{R}_{41}$ and the horizontal.

The joint angle equations, (6) to (9), are formed using the cosine law of both internal triangles of the linkage's lower trapezoid [126]. $\mu = \tan\frac{y_4 - y_1}{x_4 - x_1}$ *(6)*

$$\beta = \cos^{-1}\frac{{r_{41}}^2 + {r_{24}}^2 - {r_{12}}^2}{2r_{41}r_{24}} \tag{7}$$

$$\Psi = \cos^{-1}\frac{{r_{23}}^2 + {r_{24}}^2 - {r_{34}}^2}{2r_{23}r_{24}} \tag{8}$$

$$r_{24} = \sqrt{{r_{41}}^2 + {r_{12}}^2 - 2r_{41}r_{12}\cos(\theta_1 - \mu)} \tag{9}$$

Once (3) and (4) are reduced to a function of θ_1 only, the equations become nonlinear. However, it allows for the straightforward calculation of the end-effectors' complete closed path by inputting a set of crank angles between 0 and 2π, thereby outputting the end-effector's trajectory.

3.2 Optimization Formulation

3.2.1 Objective Functions

The main goal is to improve the system's reliability and efficiency through five objectives: 1) the path shape, 2) the tool's x-velocity fluctuations, 3) the tool's y-velocity fluctuations within the egg,

4) the total link length magnitude, and 5) the insertion-to-extraction velocity ratio. Efficiency is primarily addressed by objectives 1, 4, and 5, while reliability is targeted through objectives 2 and 3.

The current prototype's tool path is shown in blue in Figure 8. This path's length is 0.89 m**,** and a large portion is outside the extraction zone, which is defined as the area in the purple dashed box. The tool must be raised high above the egg to ensure that the ESM, which inverts upon extraction, is completely removed from the egg. Ideally, the tool would have a short, steep path into the egg with a high vertical insertion velocity and would spend most of its time in the extraction zone, steadily removing the ESM along the egg's vertical axis with minimal acceleration.

The extraction zone lies between the height of the prepared egg's aperture (0.065 m from the base of the egg) and the height at which the compliant stroke begins (0.013 m from the base of the egg), as dimensioned in Figure 8. The compliant stroke itself is excluded from this zone because, once engaged, the toolhead's vertical velocity is effectively zero. At the compliant stroke engagement stage, the toolhead latches onto the membrane, and the compliant mechanism allows the remaining portion of the end effector to move vertically freely until the stroke is fully extended again, at which point the tool begins separating the membrane. The new desired path, shown in Figure 8 (orange asterisks), was designed based on insights gained from extraction testing conducted with the *Baseline* prototype and reflects the geometry described above.

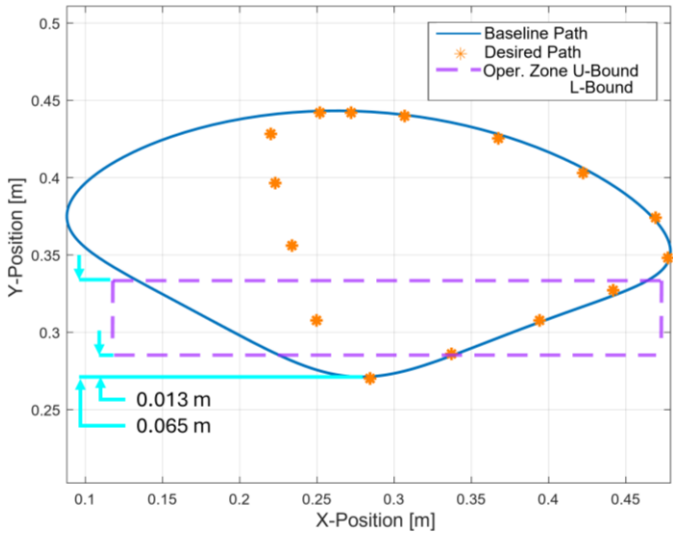

Figure 8. Plot of Baseline's path (dark blue), the desired path (orange asterisks), the extraction zone (dashed purple), and extraction zone dimensions (light blue)

This problem will first be solved using single-objective constrained optimization for trajectory matching and subsequently using multi-objective constrained optimization. In this study's single-objective optimization, the trajectory-matching objective function $f_1(\boldsymbol{x})$ will be minimized, as shown in (10) where $\boldsymbol{x}$ is the vector of design variables, $\boldsymbol{x}_1^*$ denotes the vector of optimal design variables obtained by the Argmin function of the optimization with respect to the first objective function. It will be minimized while adhering to the equality and inequality constraints presented in 3.2.3. $f_1(\boldsymbol{x})$ as well as the rest of the objective functions will be presented at length in the subsequent sections.

$$\boldsymbol{x}_1^* = \text{Argmin}\, f_1(\boldsymbol{x}) \tag{10}$$

In this study's multi-objective constrained optimizations, four cases are evaluated where a set of up to five objective functions, $f_1(\boldsymbol{x})$ to $f_5(\boldsymbol{x})$, is minimized and the Argmin is taken to form the Pareto sets of each case, from $\boldsymbol{X}^*_{Case\ 1}$ to $\boldsymbol{X}^*_{Case\ 4}$, as shown in (11) to (14). The objective functions $f_1(\boldsymbol{x})$ through $f_5(\boldsymbol{x})$, are presented in the next sections 3.2.1.1 to 3.2.1.5, respectively.

$$\boldsymbol{X}^*_{Case\ 1} = \text{Argmin}\{\frac{f_1(\boldsymbol{x})}{s_1}, \frac{f_2(\boldsymbol{x})}{s_2}\}^{\boldsymbol{T}} \tag{11}$$

$$\boldsymbol{X}^*_{Case\ 2} = \text{Argmin}\{\frac{f_1(\boldsymbol{x})}{s_1}, \frac{f_2(\boldsymbol{x})}{s_2}, \frac{f_3(\boldsymbol{x})}{s_3}\}^{\boldsymbol{T}} \tag{12}$$

$$\boldsymbol{X}^*_{Case\ 3} = \text{Argmin}\{\frac{f_1(\boldsymbol{x})}{s_1}, \frac{f_2(\boldsymbol{x})}{s_2}, \frac{f_3(\boldsymbol{x})}{s_3}, \frac{f_4(\boldsymbol{x})}{s_4}\}^{\boldsymbol{T}} \tag{13}$$

$$\boldsymbol{X}^*_{Case\ 4} = \text{Argmin}\{\frac{f_1(\boldsymbol{x})}{s_1}, \frac{f_2(\boldsymbol{x})}{s_2}, \frac{f_3(\boldsymbol{x})}{s_3}, \frac{f_4(\boldsymbol{x})}{s_4}, \frac{f_5(\boldsymbol{x})}{s_5}\}^{\boldsymbol{T}} \tag{14}$$

In these equations, the objective function values of the *Baseline* configuration, s_1 to s_5, serve as normalizing factors for scaling the objective functions relative to the *Baseline* and normalizing the values to 1. Thus, ensuring they share the same order of magnitude during the optimization. Their values are shown in Table 1, and earlier in Figure 5.

Table 1: Objective function scaling factors via the Baseline's objective values

Normalization factor	***Objective Function***	***Value***

s_1	Trajectory matching	0.4391
s_2	Link length norm	0.4940 m
s_3	X-velocity fluctuation	0.077 m/s
s_4	Y-velocity fluctuation	0.0017 m/s
s_5	Velocity Ratio	1.3265

Furthermore, since single- and multi-objective optimizations aim to minimize the objective functions, smaller objective values are preferable, and the goal is to find a global minimum.

3.2.1.1 Trajectory Matching Objective Function

To mitigate issues arising from structural errors and interpolation, FDs were employed to form the trajectory-matching objective function. In this study, the crank angle is sampled uniformly but the end-effector moves at varying speeds along the path and forms a coupler curve of 101 points generated by the kinematic simulation. However, the desired path was entered manually as a series of 16 points. This opens the possibility of an indexing mismatch that would cause a structural-error objective function to reflect differences in indexing and motion speed between the manually entered points and the points generated by the linkage, rather than genuine geometric differences between the paths, making it unsuitable for this problem. Interpolation was considered, but it smooths out important local features and may exaggerate minor deviations and bias comparisons depending on the chosen interpolation method.

Conversely, FDs compare the overall shape of paths in the frequency domain. They can be easily normalized in space, making path comparisons independent of point count, path traversal speed, scale, and pose. This study employs the FD-based objective function $f_1(\boldsymbol{x})$, expressed as:

$$f_1(\boldsymbol{x}) = \sqrt{\sum (\hat{\boldsymbol{c}}(\boldsymbol{x}) - \hat{\boldsymbol{c}}_{desired})^2} \tag{15}$$

and corresponds to the norm of the difference between the Fourier coefficients of the current and desired paths. It is a unitless metric. Furthermore, an in-depth mathematical derivation of (15) can be found in Appendix A.

The number of coefficients is a function of the number of harmonics j selected to best match the set of points, as shown in (16):

$$\hat{\boldsymbol{c}} = \left[\hat{c}_{-j}, \hat{c}_{-j-1}, \ldots, \hat{c}_0, \ldots, \hat{c}_{j-1}, \hat{c}_j\right]^T, \hat{\boldsymbol{c}} \in \mathbb{C}^{2j+1}. \tag{16}$$

In this case, five harmonics best represent the desired path; therefore, there are 11 coefficients [92]. The selection of five harmonics will be elaborated on later in this section. The Fourier coefficients are calculated using (17), where $\boldsymbol{z}_k$ is the representation of the path points $\boldsymbol{x}_{O5,k}$, $\boldsymbol{iy}_{O5,k}$, $\boldsymbol{x}_{desired,k}$, and $\boldsymbol{iy}_{desired,k}$ in the complex plane, as seen in (18), N is the number of path points, and m is the coefficient index.

$$c_m = \frac{1}{N}\sum_{k=0}^{N-1} z_k e^{-\frac{2\pi i m k}{N}},\ m = [-j, \ldots, 0, \ldots, j] \tag{17}$$

$$\boldsymbol{z}_k = \boldsymbol{x}_{O5,k} + \boldsymbol{iy}_{O5,k} = \boldsymbol{x}_{desired,k} + \boldsymbol{iy}_{desired,k} \tag{18}$$

To normalize the Fourier coefficients such that they become independent of their pose and scale, two shifting factors α and β, and the inverse of the magnitude of the first harmonic $|c_1|^{-1}$ are used to form:

$$\hat{c}_m = c_m\, e^{-i\alpha} e^{-im\beta} \frac{1}{|c_1|}, m = [-j, \ldots, 0, \ldots, j] \tag{19}$$

which yields the normalized coefficients $\hat{c}_m$, which are then used in the objective function. The fundamental coefficient, c_0, which represents the centroid of the set of points, is also set to 0 to center the normalized path about the origin. The coefficients can also be used to graph the closed Fourier harmonic curve that represents the set of points using (20):

$$z_t = \sum_{k=0}^{N-1} c_m\, e^{2\pi i m t}, m = [-u, \ldots, 0, \ldots, u] \tag{20}$$

where u represents the number of harmonics selected to be part of the partial sum from the first to the u^{th} harmonic.

This is evident in Figure 9 where the partial sums of the first to the fifth harmonics are graphed. The approximation improves as additional harmonics are included, with the five-harmonic

representation closely reproducing the target point set. This observation is consistent with the findings of McGarva and Mullineux, who reported that only a limited number of low-order harmonics are required to achieve an effective approximation of curve shapes [92].

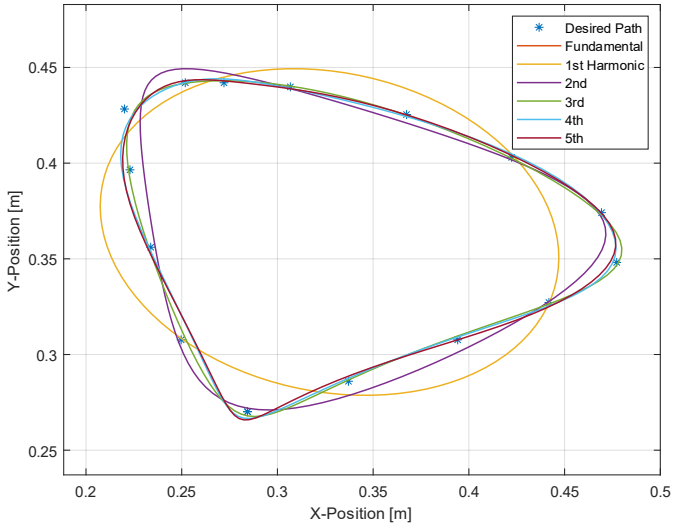

Figure 9. Plot of the desired path points and the closed curves formed using the path points' fundamental harmonic and first through fifth Fourier harmonics

The improvement in the approximation is also observed in Figure 10 where one can see the structural error between the desired path and its Fourier decomposition as a function of the number of harmonics used of the calculation. The 5th harmonic is thus a good truncation point since the error plateaus near that point. Structural error is used in this case instead of (15) because it is showcasing the error between a set of points and its own Fourier decomposition instead against a linkage's coupler curve, therefore, there are no scale or pose issues.

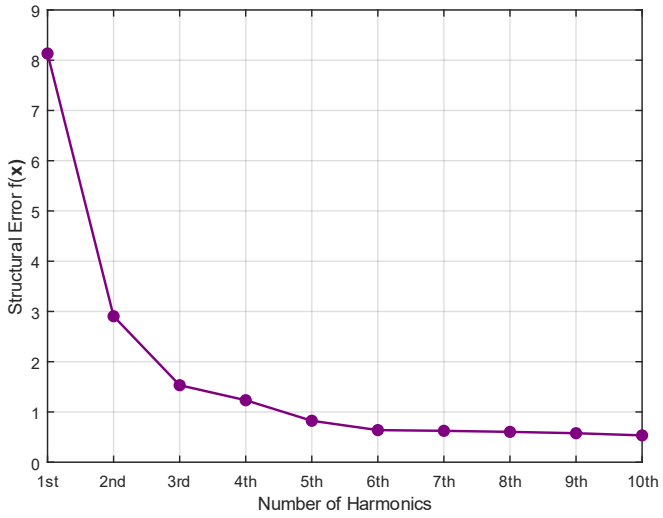

Figure 10. Plot of the structural error between the desired path and its Fourier harmonic decomposition as a function of the number of harmonics used

3.2.1.2 X-Velocity Fluctuation Objective Function

The X-velocity fluctuation objective function $f_2(\boldsymbol{x})$ is expressed as:

$$f_2(\boldsymbol{x}) = \frac{1}{n_{oper.}} \sum_{i=0}^{n_{oper.}} |v_{x,i}(\boldsymbol{x}) - \bar{v}_x(\boldsymbol{x})| \left[\frac{m}{s}\right] \tag{21}$$

and is based on [125]. This equation represents the average absolute difference between the tool's instantaneous $v_{x,i}$ and average horizontal $\bar{v}_x$ velocities within the egg in meters per second, where $n_{oper.}$ is the number of points in time inside operating zone. This corresponds to the position of the tool under the purple dashed line in Figure 11. Thus, it quantifies the extent to which the tool's horizontal velocity deviates from the average. This value would be zero if the instantaneous velocity was constant. The absolute value function is employed rather than a smooth positive-

valued alternative, such as the quadratic function, as the latter never reaches zero during the optimizations and thus avoids the associated discontinuity and differentiability concerns.

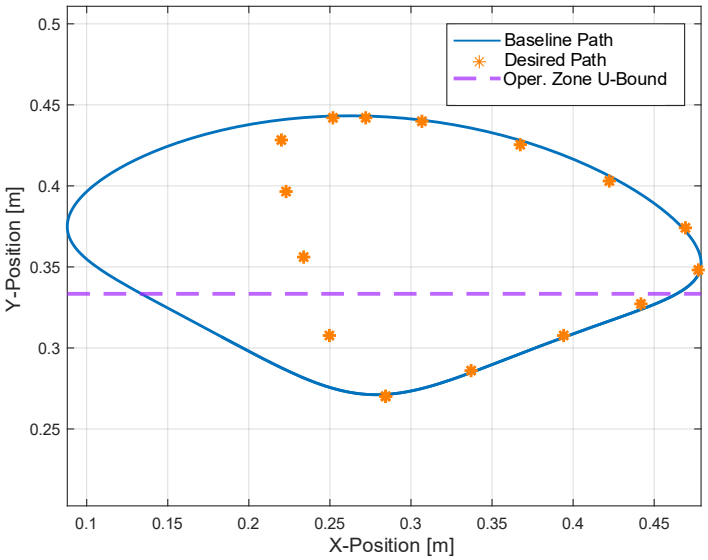

Figure 11. Operation zone for X-Velocity fluctuation calculation

3.2.1.3 Y-Velocity Fluctuation Objective Function

The Y-velocity fluctuation objective function $f_3(\boldsymbol{x})$, based on [125], is given by:

$$f_3(\boldsymbol{x}) = \frac{1}{n_{ext.}} \sum_{i=0}^{n_{ext.}} |v_{y,i}(\boldsymbol{x}) - \bar{v}_y(\boldsymbol{x})| \left[\frac{m}{s}\right]. \tag{22}$$

This equation is similar to (21) but the velocity fluctuation is calculated as the tool exits the egg, represented by the part of the trajectory inside the green dashed box shown in Figure 12 and where

$n_{ext.}$ is the number of points in time inside extraction zone. This function also never reaches zero during the optimizations and avoids the associated discontinuity and differentiability concerns.

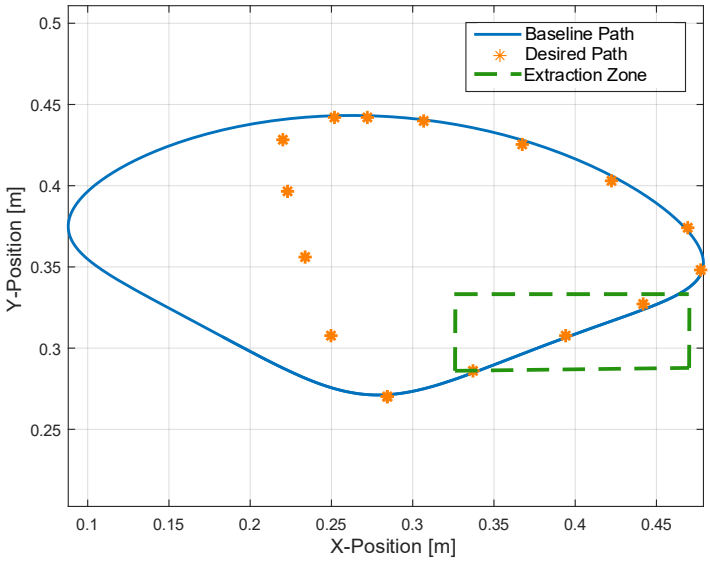

Figure 12. Extraction zone for Y-Velocity fluctuation calculation

3.2.1.4 Link Length Norm Objective Function

The link length objective function $f_4(\boldsymbol{x})$ measures the norm of all design parameters in meters after undergoing the three pose and scale linear transformations post shape-optimization, described in detail in 3.4, and is defined as:

$$f_4(\boldsymbol{x}) = \sqrt{\sum (\boldsymbol{x}_{9-param}{}')^2}\ [m] \tag{23}$$

where the tilde on the parameter vector, $\tilde{\boldsymbol{x}}_{9-param}$, signifies that the parameters have undergone the affine post shape-optimization transformations. It contains all 9 design parameters as shown in (24).

$$\tilde{\boldsymbol{x}}_{9-param} = [r_{12}, r_{23}, r_{34}, d_{35x'}, d_{35y'}, x_1, x_1, y_4, y_4]_{rotated,scaled,translated} \tag{24}$$

This objective helps to further control the size of the links during the multi-objective optimization. For context, the *Baseline* linkage link lengths can be seen in Table 9 in Appendix B.

3.2.1.5 Velocity Ratio Objective Function

Finally, the Y-velocity ratio objective function $f_5(\boldsymbol{x})$ corresponds to the absolute difference between the target velocity ratio R_{target} and the ratio of average insertion to extraction velocity of

the current configuration $R_v(\boldsymbol{x})$, as shown in (25), and (26), and is a unitless metric as stated by the $[-]$. These functions are composed of a rational function that may cause a singularity if $\bar{v}_{y\,ext.}(\boldsymbol{x})$ is ever zero. However, $\bar{v}_{y\,ext.}(\boldsymbol{x})$ being zero is discouraged by the objectives and thus never reaches zero during the optimizations and avoids the associated singularity concerns.

$$f_5(\boldsymbol{x}) = \left|R_{target} - R_v(\boldsymbol{x})\right| [-] \tag{25}$$

$$R_v(\boldsymbol{x}) = \left|\frac{\bar{v}_{y\ in.}(\boldsymbol{x})}{\bar{v}_{y\,ext.}(\boldsymbol{x})}\right| [-] \tag{26}$$

The average insertion velocity $\bar{v}_{y\,in.}(\boldsymbol{x})$ is taken inside the purple dashed box in Figure 13 and the average extraction velocity $\bar{v}_{y\,ext.}(\boldsymbol{x})$ inside the green dashed box. The path between the two sets of dashed lines is not essential since the end-effector's compliant stroke is activated and the tool head is in contact with the base of the egg. The target ratio for this optimization is 2, meaning the end effector ideally enters twice as fast as it exits. This value was set to 2 heuristically since it was observed that coupler curves with velocity ratios higher than 2 had shapes that were irregular and were not pertinent to this optimization.

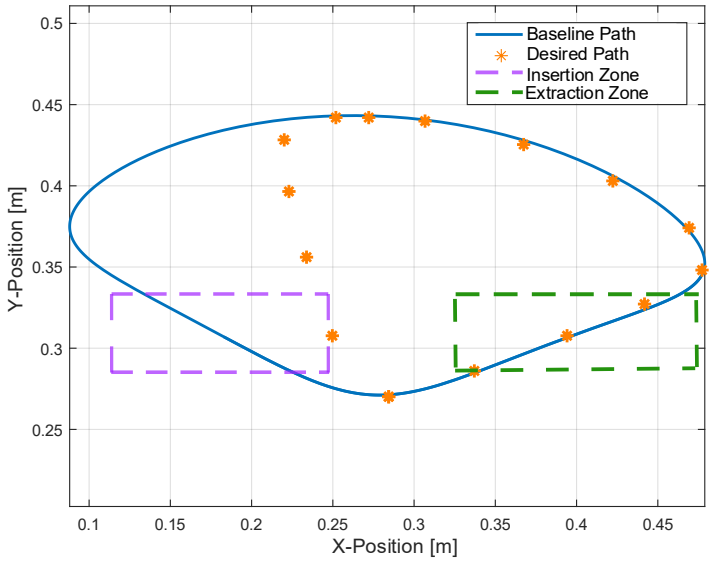

Figure 13. Insertion and extraction zones for velocity ratio calculation

3.2.2 Pareto Front Formulation

A multi-objective optimization with k objective functions yields a Pareto set $\boldsymbol{X}_k^*$ consisting of the Pareto-optimal configurations' vectors, $\boldsymbol{x}_{k,1}^*, \boldsymbol{x}_{k,2}^*, \dots, \boldsymbol{x}_{k,n}^*$, where n is the number of Pareto optimal configurations as seen in **Error! Reference source not found.** and earlier in (11) to (14).

$$\boldsymbol{X}_k^* = [\boldsymbol{x}_{k,1}^*, \boldsymbol{x}_{k,2}^*, \dots, \boldsymbol{x}_{k,n}^*] \in \mathbb{R}^{p \times n} \tag{27}$$

The Pareto set thus forms a $p \times n$ matrix, where p is the number of design variables in each configuration vector. The Pareto set yields a Pareto front $\boldsymbol{F}_k^*$, composed of the objective-function value vectors of the associate configurations, $\boldsymbol{f_k}(\boldsymbol{x}_{k,1}^*), \boldsymbol{f_k}(\boldsymbol{x}_{k,2}^*), \dots, \boldsymbol{f_k}(\boldsymbol{x}_{k,n}^*)$ formulated as:

$$\boldsymbol{F}_k^* = \left[\boldsymbol{f_k}(\boldsymbol{x}_{k,1}^*), \boldsymbol{f_k}(\boldsymbol{x}_{k,2}^*), \dots, \boldsymbol{f_k}(\boldsymbol{x}_{k,n}^*)\right] \in \mathbb{R}^{k \times n}. \tag{28}$$

The Pareto front thus forms a $k \times n$ matrix where k is the number of simultaneously optimized objective functions.

For a single objective optimization, $\boldsymbol{x}_1^*$ would be a single vector of optimal design variable values associated with a single optimal objective function value. Depending on the dimensionality of the Pareto front, it can be visualized as either a 2D curve, a 3D surface, or a set of curves. If it is a $2 \times n$ matrix via a two-objective optimization and then visualized as a curve. If it is a $3 \times n$ matrix via a three-objective optimization and then visualized as a surface. For dimensionalities higher than $3 \times n$, the Pareto front must be visualized as a set of curves or surfaces. Such visualizations can then be used as a design tool to help the designer select the optimal design, since they represent the trade-offs among design objectives.

3.2.3 Constraints

All optimizations conducted in this research are constrained problems; therefore, they must respect a set of inequality g constraints, as shown in (29), where m is the number of inequality constraints, a set of equality constraints h, as shown in (30), where q is the number of equality constraints, and design variable value constraints (31) where $\boldsymbol{x}_L$ and $\boldsymbol{x}_U$ are vectors of the lower and upper bounds of the design variables. For this problem, the link length and the Grashof inequality constraints must be satisfied.

$$g_i(\boldsymbol{x}) \leq 0,\ i = 1,2,\dots,m \tag{29}$$

$$h_j(\boldsymbol{x}) = 0,\ j = 1,2,\dots,q \tag{30}$$

$$\boldsymbol{x}_L \leq \boldsymbol{x} \leq \boldsymbol{x}_U \tag{31}$$

3.2.3.1 Link length constraints

As shown in (32), only five link lengths are optimized. This is because the ground-link length r_{41} is fixed at 0.5 m, thereby scaling the other link dimensions relative to it. This is a common practice in the linkage synthesis literature when using normalized FD for pose-invariant coupler curves [91,94,101].

$$\boldsymbol{x} = [r_{12}, r_{23}, r_{34}, d_{35x'}, d_{35y'}] \tag{32}$$

The linkage's position relative to the origin (x_1, y_1, and y_4) is determined via linear transformations following optimization. The five design variables and r_{41} together govern the path's shape, while the position of the fixed link relative to the origin determines its pose. The complete set of nine design parameters is then scaled proportionally to match the desired path's area.

The link length constraints, shown in (33) to (37), are based on measurements taken from industrial egg-breaking machines during a site visit [127], the proof-of-concept prototype, and four-bar linkages documented in literature. These are hard constraints but had no limiting effect on the optimization, as the resulting optimized link lengths remained well within these bounds, suggesting that the target path geometry is well-represented by link proportions that naturally satisfy these bounds.

$$0.01\ m \leq \mathrm{r}_{12} \leq 0.25\ m \tag{33}$$

$$0.1\ m \leq \mathrm{r}_{23} \leq 0.5\ m \tag{34}$$

$$0.1\ m \leq \mathrm{r}_{34} \leq 0.5\ m \tag{35}$$

$$-0.5\ m \leq \mathrm{d}_{35x'} \leq 0.5\ m \tag{36}$$

$$-0.5\ m \leq \mathrm{d}_{35y'} \leq 0.5\ m \tag{37}$$

3.2.3.2 Grashof Constraint

The Grashof constraint g_1, the problem's first and only non-design-variable inequality constraint, is given in (38) where s is the length of the shortest link (red crank arm), l the longest link (black ground link), and p and q are the lengths of the remaining two links [126].

$$g_1: (s + l) - (p + q) \leq 0 \tag{38}$$

The associated link length ordering requirement is:

$$s \leq p \leq q \leq l \tag{39}$$

The Grahsof constraint allows the complete rotation of the shortest link. In this case, the crank arm is fixed to be the shortest link, thereby producing a closed coupler curve for a full rotation of the crank.

3.2.4 Single and Multi-Objective Optimization Procedure

The flowchart representing the optimization procedure for single and multi-objective optimizations is shown in Figure 14. The single- and multi-objective optimization algorithms in this study begin with the *Baseline* linkage configuration as the initial guess. A kinematic simulation is then performed to generate the end-effector trajectory in the Fourier domain, from which the objective functions are evaluated. The FD-based trajectory-matching objective function (15) is evaluated for the single-objective optimization case. The X- and Y-direction velocity fluctuation objective functions (22) and (23), the link-length magnitude objective function (24), and the insertion-to-extraction velocity ratio objective function (25) are all are evaluated for the multi-objective cases.

The algorithms subsequently determine whether convergence to a global optimum has been achieved. If not, the linkage parameters are updated using methods such as Sequential Quadratic Programming (SQP) within the GlobalSearch framework, or genetic mutation in Genetic Algorithms, and the optimization loop continues. Multi-start algorithms, such as GS, periodically resample new starting points within the design space to mitigate the risk of local minima.

Once an optimal solution is identified in the Fourier domain, the resulting configuration is scaled, rotated, and translated to best align with the target path in the spatial (X–Y) domain. At this stage, a single-objective algorithm terminates with a single optimal configuration. In contrast, a multi-

objective algorithm continues to iterate and update the Pareto front until it completes the specified number of iterations or reaches the maximum run time. For this study, the MOGA was constrained to a maximum runtime of five hours and a maximum of 1,000,000 generations.

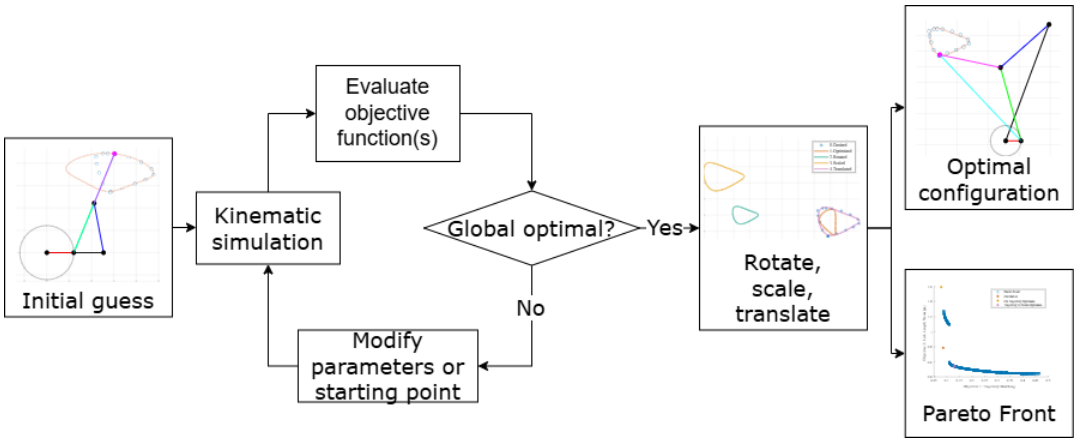

Figure 14. Optimization flowchart

3.3 Reinforcement Learning

In RL, the problem must be formulated as a Markov decision process. In this case, the environment comprises the kinematic simulation and the state-action transition mechanics; the states are the link lengths (design variables), the actions are small changes to these variables, and the reward function is a modified version of the trajectory-matching Fourier-coefficient-difference function used in single-objective optimization. The Fourier-coefficient-difference function is adapted to the reward-maximization framework of RL by changing the sign of the expression and by using the Argmax operator, as shown by:

$$\boldsymbol{s_t}^* = \text{Argmax}\left(-\sqrt{\sum(\hat{\boldsymbol{c}}(\boldsymbol{s_t}) - \hat{\boldsymbol{c}}_{desired})^2}\right). \tag{40}$$

Consequently, the agent maximizes a negative quantity, thereby minimizing the coefficient-difference norm between the current state's coefficients $\hat{\boldsymbol{c}}(\boldsymbol{s_t})$ and the desired path coefficients $\hat{\boldsymbol{c}}_{desired}$, promoting smaller discrepancies, and improving path conformity. Furthermore, to enforce the Grashof condition, a soft constraint was used, penalizing the reward by a large number when the Grashof condition is not satisfied at the current state. This penalty was set to -20 following testing and tuning.

The RL algorithm's states are bound to the same values as the optimization formulation, as shown in (33) to (37). However, the actions are bound according to **Error! Reference source not found.**, where $\boldsymbol{a}_L$ and $\boldsymbol{a}_U$ are the lower and upper bounds, and $\boldsymbol{s}_1$ represents the initial state of the training session and first episode, which is equal to the linkage parameters of the *Baseline* configuration.

$$\boldsymbol{a}_L, \boldsymbol{a}_U = \pm 0.25([0.1, 0.1, 0.1, 0.1, -0.1] - \boldsymbol{s}_1)\ [m] \tag{41}$$

The upper bound is set by multiplying the difference between the bias vector and the initial state by -0.25 ,and the lower bound by multiplying this difference by $+0.25$. These bounds were set heuristically after testing and tuning the coefficient (± 0.25) and bias vector to constrain action magnitudes, which were found to improve training stability. The final term of the bias vector is negative to deal with the small magnitude of the initial value of $d_{35y'}$, which is 1×10^{-4}, and prevents its upper bound from being negative. Action clipping was used as a failsafe to keep the action and state values within their bounds. Furthermore, the state and action spaces are continuous, meaning that both can take on any real-valued combination within their defined bounds rather than being limited to discrete values.

The state-action transition mechanics are linear, as shown in (42), and the algorithms operate in an episodic manner, with a fixed number of episodes and iterations per episode, as shown in Table 2, before training terminates. The action is sampled from the multivariate normal distribution (43) formed by the mean vector $\boldsymbol{\mu}_t$ and the covariance matrix $\boldsymbol{\Sigma}_t$.

$$\boldsymbol{s}_{t+1} = \boldsymbol{s}_t + \boldsymbol{a}_t \tag{42}$$

$$\boldsymbol{a}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \tag{43}$$

The covariance matrix is formed using the Cholesky method (44), and the lower-triangle matrix $\boldsymbol{L}_t$. This lower-triangle matrix is formed by the standard deviation vector $\boldsymbol{\sigma}_t$ and the strictly lower-triangle matrix $\boldsymbol{L}_{t-}$ (45). The mean vector $\boldsymbol{\mu}_t$, the standard deviation vector $\boldsymbol{\sigma}_t$, and the strictly lower-triangle matrix $\boldsymbol{L}_{t-}$ are all output by the actor network.

$$\boldsymbol{L}_t \boldsymbol{L}_t{}^T = \boldsymbol{\Sigma}_t \tag{44}$$

$$\boldsymbol{L_t = \sigma_t + L_{t-}} \tag{45}$$

The state-action transition mechanics for the degenerate RL formulation that will also be explored in this research are slightly different than (42). Instead, it is a single-step mapping in which the next state equals the action, as shown in (46). The actor, therefore, seeks the optimal policy that maps an initial state directly to the optimal state.

$$\boldsymbol{s_{t+1} = a_t} \tag{46}$$

In RL, NNs are trained to approximate the optimal policy through repeated interaction with the environment, rather than by directly adjusting the current state using gradients of the objective function, as is done in GS. Instead, the reward signal and the sequence of state–action transitions influence the weight updates of the actor and critic networks. The actor learns to select optimal actions for given states, while the critic evaluates these actions by estimating the corresponding Q-values, defined as the predicted long-term reward associated with the state–action pair (s_t, a_t). In this study, the networks were trained using the Adam optimizer, an adaptive learning rate algorithm is common in RL and ML. The initial learning rates used in the continuous control and degenerate formulations of the agents used in this research are shown in Table 2 as well as the remaining hyperparameters used.

Table 2. Hyperparameters and corresponding values for both TD3 and DDPG formulations

	TD3		***DDPG***	
Hyperparameter	***Cont***	***Deg***	***Cont***	***Deg***
Number of episodes	50	500	50	500
Steps per episode	100	1	100	1
Actor learning rate	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Critic learning rate	1×10^{-3}	1.0 e-3	1.0 e-3	1.0 e-3
Mini Batch Size	256	256	256	256
Buffer size	1 e6	1 e6	1 e6	1 e6
Discount factor	0.99	1	0.99	1
Target smooth factor	0.005	0.01	0.005	0.01
Exploration noise std. dev.	0.05	0.25	0.05	0.25
Exploration noise decay rate	0.088	0.0122	0.088	0.0122
Learning frequency	End of episode	End of episode	End of episode	End of episode

The agents used in this study are the TD3 and DDPG agents from MATLAB's Reinforcement Learning Toolbox. These agents were selected for their high training stability, which facilitates exploration and promotes convergence to the global optimum over the short event horizon.

To help prevent convergence to local minima, RL agents, such as the TD3 and DDPG, incorporate exploratory noise into the action space and utilize stochastic updates via an experience buffer. In this study, each action was characterized by a multivariate normal distribution generated by the actor network, adding action-level stochasticity to agents that are otherwise deterministic in their action outputs. This also allows the actor network to learn the influence of each design variable on trajectory matching via the covariance matrix [113], [114].

Additionally, stochasticity was added through the training protocol: during the first 50% of episodes, each episode began from a new initial point sampled uniformly from the design space, whereas in the remaining episodes, the starting point corresponded to the terminal state of the previous episode. The first half mirrors the design-space sampling of a multi-start global optimization algorithm, whereas the latter half models a continuous control problem with sequential states. This was not done to the degenerate formulation since it is based on identical episode initial states. Ideally, the stochasticity introduced by the multi-start formulation, the covariance matrix, and the action noise is sufficient to escape local minima without impeding convergence to the global maximum. The RL optimization and network training loop are shown

in Figure 15. One can see that it is similar to Figure 14 except for adopting Markov terminology and adding the network training loop.

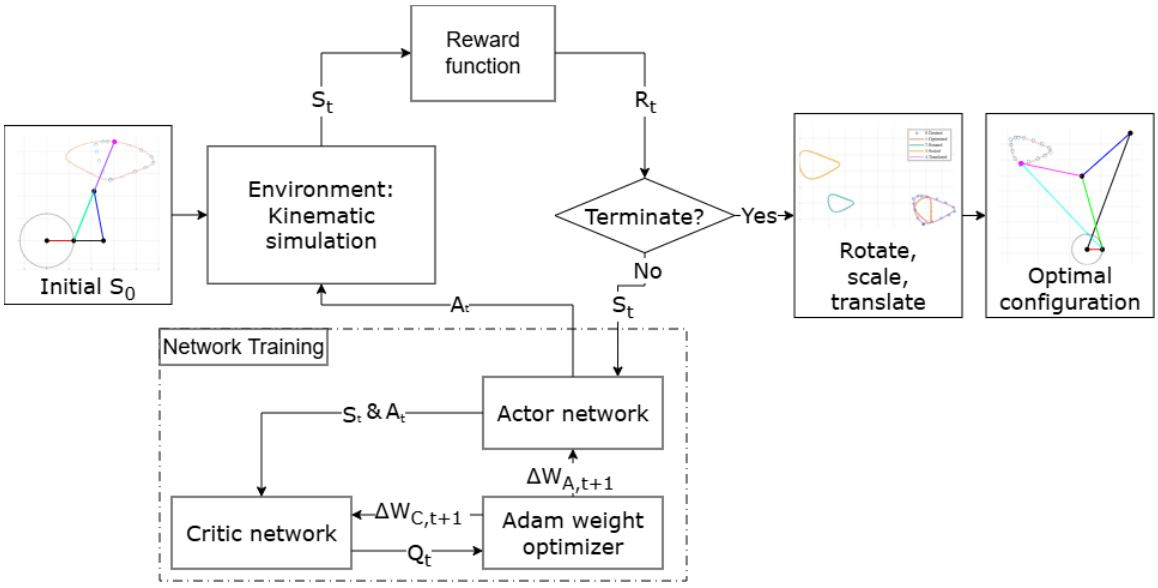

Figure 15. RL flowchart

3.3.1 Neural Network Architecture

The actor network consists of three subnetworks: the mean, standard deviation, and covariance networks. The outputs of these subnetworks are then concatenated to create the output vector. The action is subsequently sampled from the covariance matrix generated in the environment. All these subnetworks have three hidden layers, each with 256 neurons.

The critic network also comprises three hidden layers of 256 neurons each, but it outputs only a single value: the Q-value. A diagram showing the network architecture is shown in Figure 16 below, where N represents the number of neurons in each layer.

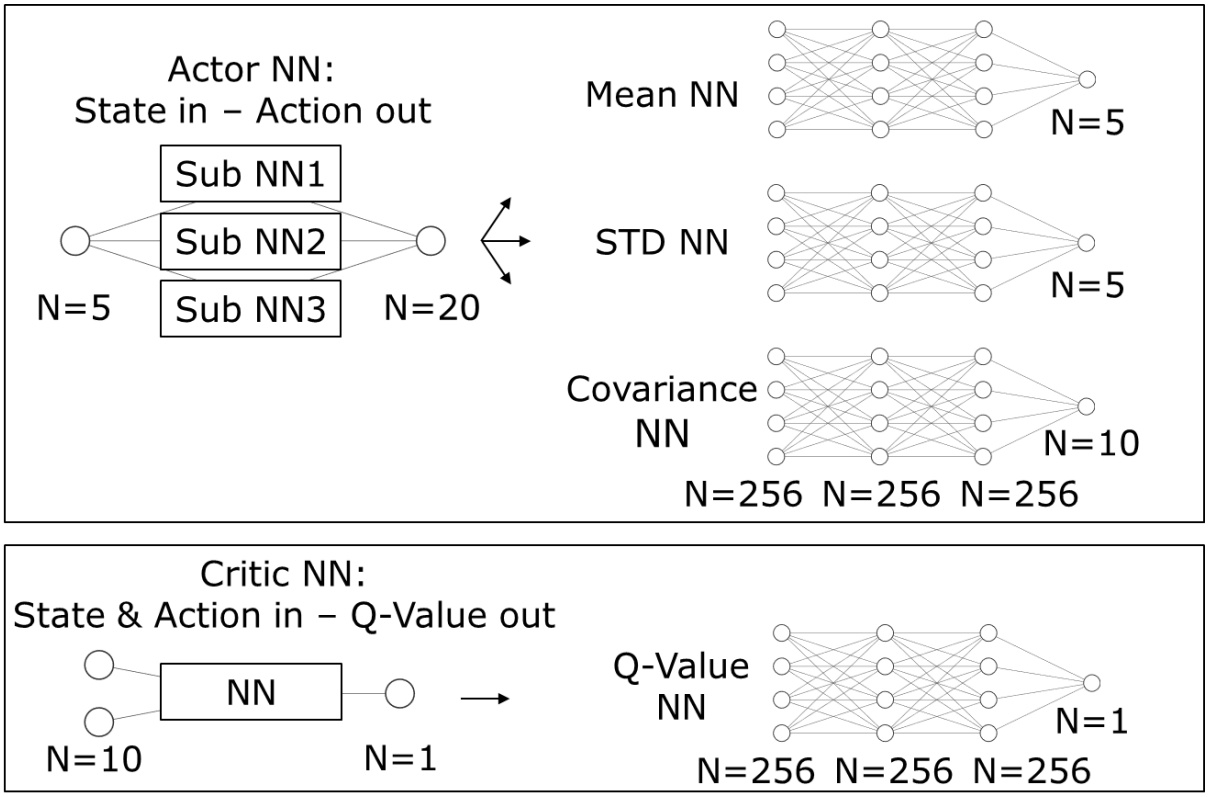

Figure 16. Actor (top) and critic (bottom) NN diagrams, and N is the number of neurons in each layer

The Rectified Linear Unit (ReLU) activation function was used in both networks, as it is widely adopted in deep ML and deep RL for its efficiency and ability to mitigate vanishing gradients. Figure 17 illustrates how a simple NN composed of ReLU units can approximate the function $\sin x \cos y$. The close agreement between the NN's output and the true function

demonstrates the network's ability to approximate complex nonlinear relationships, which is an ability leveraged in the RL framework to model and optimize the objective functions.

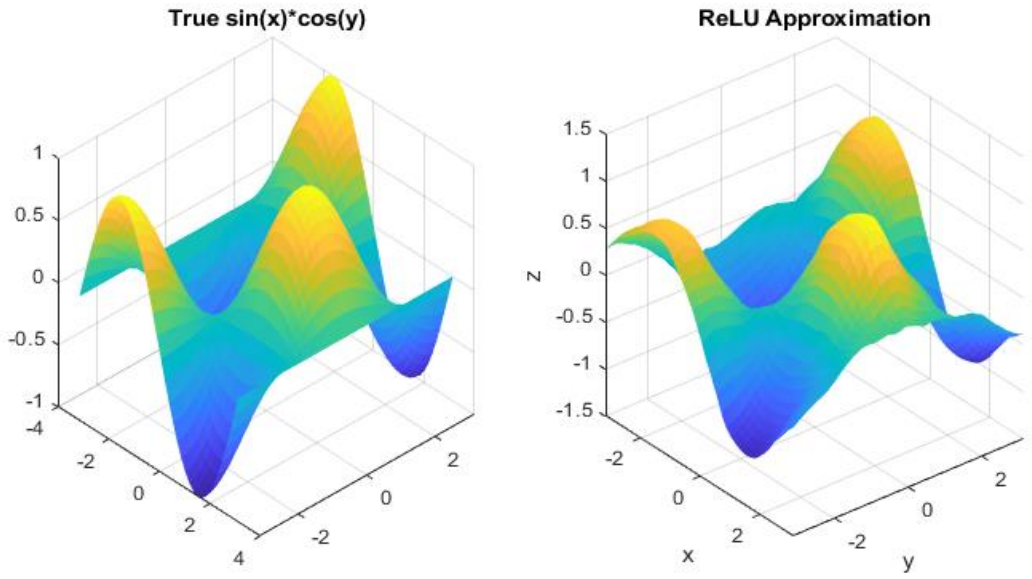

Figure 17. Comparison of a true sine-cosine 3D graph (left) and a function approximation using a trained ReLU NN (right)

3.4 Linear Transformation of Linkage Parameters for Pose and Scale Matching

Linkage optimization and normalization of its coupler curve are performed in the Fourier domain. To match the pose and scale of the desired path in the spatial domain (X–Y), three linear transformations are applied to the linkage parameters.

3.4.1 Rotation

The rotation of the linkage and coupler curve is achieved using a 2D rotation matrix and the Fourier shifting factors $\alpha_{\boldsymbol{desired}}$ and $\alpha(\boldsymbol{x}^*)$ from the Fourier coefficient normalization of both the desired path and the shape-optimized configuration's path via (19). These shifting factors represent the angle by which the paths were rotated to normalize them in the Fourier domain; this angle reflects the extent to which the main diagonal of the first harmonics deviates from the horizontal axis. By taking the difference between $\alpha_{\boldsymbol{x}^*}$ and $\alpha_{\boldsymbol{desired}}$, the rotation matrix in (47) is formed. It can then be multiplied by the vectors containing x_1 and y_1 or x_4 and y_4 to calculate the new position of the optimized linkage's fixed link $[x_1, y_1, x_4, y_4]_{\text{rotated}}$.

$$\begin{bmatrix} x_{1,4} \\ y_{1,4} \end{bmatrix}_{rotated} = \begin{bmatrix} \cos(\alpha_{\boldsymbol{desired}} - \alpha_{x^*}) & -\sin(\alpha_{\boldsymbol{desired}} - \alpha(\boldsymbol{x}^*)) \\ \sin(\alpha_{\boldsymbol{desired}} - \alpha_{x^*}) & \cos(\alpha_{\boldsymbol{desired}} - \alpha(\boldsymbol{x}^*)) \end{bmatrix} \begin{bmatrix} x_{1,4} \\ y_{1,4} \end{bmatrix} \quad (47)$$

3.4.2 Scaling

To match the scale of the desired path, the areas of the rotated and desired paths are calculated. A scaling factor is then formed by taking the square root of the ratio of the desired path's area $A_{\boldsymbol{desired}}$ over the optimized path's area $A(\boldsymbol{x}^*)$. This factor is then multiplied by the nine linkage design parameters of the rotated linkage which include $[x_1, y_1, x_4, y_4]_{\text{rotated}}$ and form $\boldsymbol{x}^*_{9,rotated}$. The result is in a new configuration $\boldsymbol{x}^*_{9,rotated,scaled}$ that produces a path with the same area as the desired path.

$$\boldsymbol{x}^*_{9,rotated,scaled} = \boldsymbol{x}^*_{9,rotated} \sqrt{\frac{A_{\boldsymbol{desired}}}{A(\boldsymbol{x}^*)}} \quad (48)$$

3.4.3 Translation

The centroid of the rotated and scaled linkage is translated to match the desired path's position. This is done by adding the difference between the desired path's centroid, $\bar{x}_{\boldsymbol{desired}}$ and $\bar{y}_{\boldsymbol{desired}}$, and the optimized linkage's centroid, $\bar{x}(\boldsymbol{x}^*)$ and $\bar{y}(\boldsymbol{x}^*)$, to the coordinates of the fixed link of the scaled and rotated linkage, as shown in:

$$\begin{bmatrix} x_1^* & x_4^* \\ y_1^* & y_4^* \end{bmatrix}_{rotated,scaled,translated} = \begin{bmatrix} x_1^* & x_4^* \\ y_1^* & y_4^* \end{bmatrix}_{rotated,scaled} + \begin{bmatrix} \bar{x}_{\boldsymbol{desired}} - \bar{x}(\boldsymbol{x}^*) \\ \bar{y}_{\boldsymbol{desired}} - \bar{y}(\boldsymbol{x}^*) \end{bmatrix}. \quad (49)$$

This results in the ground link coordinates $\begin{bmatrix} x_1^* & x_4^* \\ y_1^* & y_4^* \end{bmatrix}_{rotated,scaled,translated}$ of the fully transformed linkage parameters $\tilde{\boldsymbol{x}}_{9-param}$, also used in (23).The cumulative effect of the three-step linear transformation is shown in Figure 18.

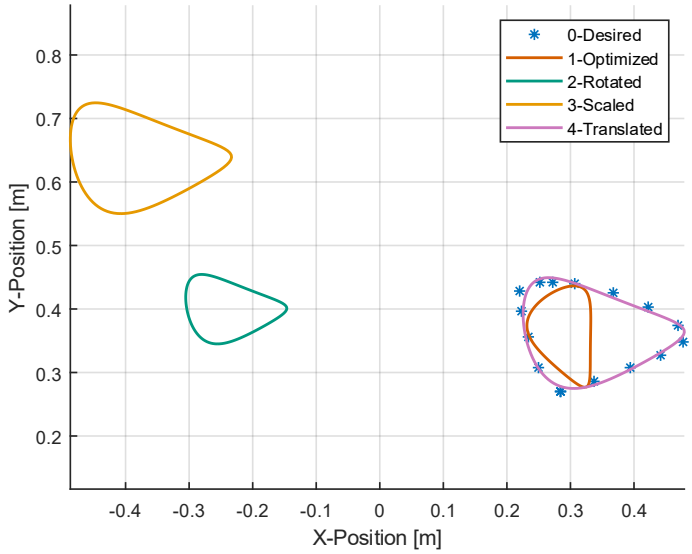

Figure 18. Desired path points and a step-by-step illustration of the three-step linear transformation post-processing

3.5 Constraint-Based Outlier Filtering of the Pareto Fronts

In this research, constraint-based outlier filtering is employed to improve the interpretability of the Pareto fronts for Cases 2, 3, and 4 of the multi-objective optimization study, which involve the simultaneous optimization of three, four, and five objective functions, respectively. In contrast, an unfiltered Pareto front is presented for Case 1 in Figure 29 on page 57, which shows the results of a two-objective optimization and yields a dense and complete Pareto front with all solutions lying within a physically meaningful range. The constraints presented in this section were therefore established to bound the Pareto fronts of the subsequent case to Case 1's objective range and ensure all the results are physically meaningful. These constraints are all upper bounds applied to the minimization problems in this thesis; therefore, their effect on Pareto optimality is minimal.

In the following equations, k denotes the number of simultaneously optimized objective functions for each case, i indexes the Pareto-optimal solutions, and n represents the total number of Pareto-optimal solutions.

$$k \in \{3, 4, 5\} \tag{50}$$

$$i \in [1, 2, \dots, n] \tag{51}$$

L_1 and L_4 in (52) and (53) correspond to the upper bounds of the dense Pareto front range observed in Case 1 and the upper range of the trajectory-matching, a unitless measure, and link-length norm, measured in meters, objective values of the filtered Pareto fronts.

$$f_1(\boldsymbol{x}^*_{k,i}) < L_1 = 0.5\ [-] \tag{52}$$

$$f_4(\boldsymbol{x}^*_{k,i}) < L_4 = 2\ m \tag{53}$$

Chapter 4

4 Results

4.1 Single-objective Trajectory-Matching Optimization

The results of the single-objective trajectory-matching optimization are presented below. Traditional global optimization techniques, such as metaheuristic and multi-start methods, and non-traditional optimization techniques, such as RL, were utilized to obtain these results. These methods, along with all other optimizations in this thesis, were implemented on a personal computer with a 4-core Intel Core i7-8550U processor and 8 GB of RAM.

Table 3 shows the performance metrics for the various traditional local and global optimization algorithms, available in MATLAB's global and local optimization toolboxes, used to solve the trajectory-matching optimization. These algorithms include GS, Pattern Search (PS), PSO, GA, SA, and the local solver FMINCON. The *iteration criterion type,* number of *maximum iterations*, *average necessary iterations*, and the *average % used* of the maximum number of iterations are shown in the table for 50 runs of each algorithm. The *average clock time* and the *average terminal* $f_1(\boldsymbol{x})$ are also displayed.

Table 3. Trajectory-matching optimization algorithm performance metric summary

	Iteration criterion type	***Maximum iterations***	***Average necessary iterations***	***Average % used***	***Average clock time [s]***	***Average terminal*** $\boldsymbol{f_1(x)}$
GlobalSearch	*NumTrialPoints*	1000	31.36	3.14	11.90	0.0782
Genetic Algorithm	*MaxGenerations*	150	5	3.33	44.89	0.0826
FMINCON	*MaxIterations*	400	28	7	0.31	0.0826
Pattern Search	*MaxIterations*	500	4	0.8	8.70	0.0785
Particle Swarm Optimization	*MaxIterations*	500	373.32	74.66	14.95	0.0825
Simulated Annealing	*MaxIterations*	10000	5001.74	50.00	12.86	0.1263

From this table, one can see that none of the algorithms reach 100% of their maximum number of iterations. This is because they were terminated by their adaptive termination strategies. In fact, a function tolerance criterion was set to a constant 1×10^{-6} across all solvers. This is used as a stalling criterion to terminate GA, FMINCON, PSO, and SA when improvement in the objective function stagnates across consecutive generations and iterations.

GS interprets the function tolerance criterion slightly differently because it performs multiple FMINCON runs from different starting points. The function tolerance is applied at the local solver level to terminate individual runs when the objective function no longer improves significantly. The overall maximum iteration criterion value is governed by *NumTrialPoints*, which specifies the number of candidates starting points to evaluate. GS employs a scatter-search strategy that filters trial points during optimization, initiating local solver runs only for the most promising candidates. This combination of local solver stopping criteria and strategic start-point selection makes GS a highly resource-efficient solver for continuous search-space problems.

Pattern Search (PS) is a mesh-based optimization algorithm that uses both a mesh tolerance and a function tolerance, each set to 1×10^{-6}. The mesh tolerance terminates the algorithm when the search mesh contracts below the threshold, while the function tolerance stops the search when improvements in the objective function become negligible. Termination occurs when either criterion is met, ensuring efficient convergence.

It is evident that there is no direct correlation between average % used, average clock time, and average terminal $f_1(x)$. However, the values for average terminal $f_1(x)$ and for average clock time are generally of the same order of magnitude across solvers, except for FMINCON and GA. The shorter clock time for FMINCON is expected, as it is purely a local solver. GA, on the other hand, exhibits a longer average clock time because population-based genetic operations are applied to the design variable vectors, including selection, crossover, and mutation. This increased computational cost is inherent to the algorithm rather than a limitation, reflecting its emphasis on broad exploration of the search space. As a result, GA remains a robust global optimization method, particularly well-suited to problems with complex or discrete design spaces.

Figure 19 shows the results of the average instantaneous $f_1(\mathbf{x})$ across n=50 iterations of each algorithm, represented as solid lines and their objective value distribution represented as bands. The results are shown as a function of percent completeness of the optimizations (0-100%) since

they all have different clock times and numbers of steps and iterations. The average terminal objective values and wall-clock times are shown in the figure legend. This comparison is motivated by the no-free-lunch theorem [128], which states that no universal optimization algorithm outperforms all others across all possible problems.

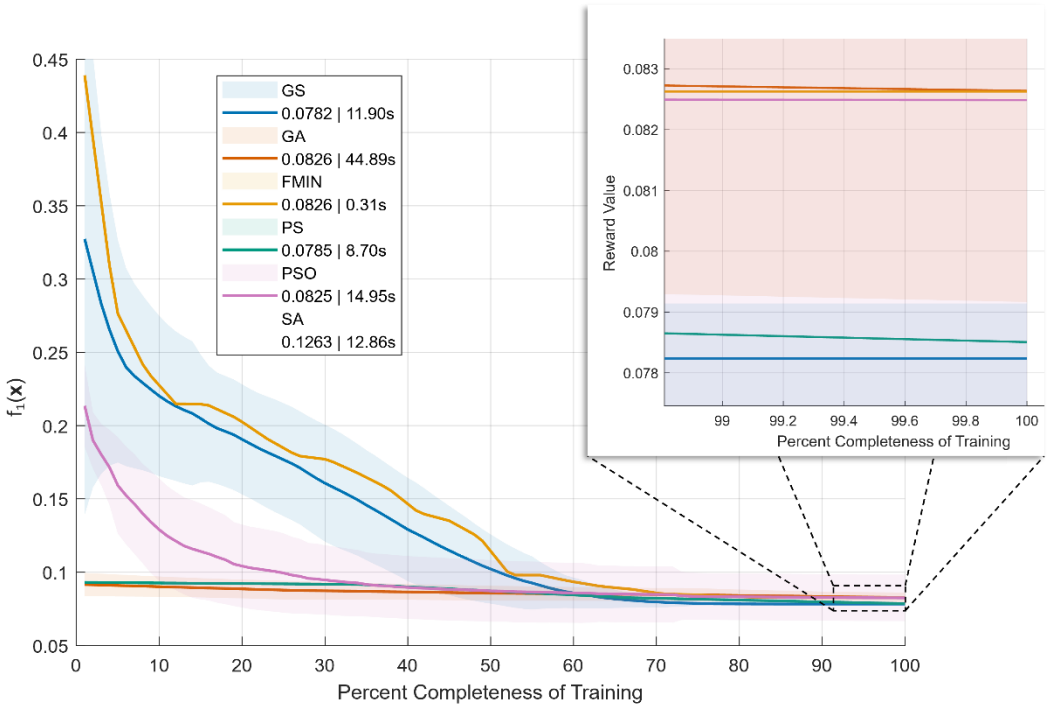

Figure 19. Comparison of the reward value as a function of the global optimization algorithms' training session percent completeness, showing the mean (solid line) and standard deviation (shaded area), n=50.

The blue line representing GS achieves a lower terminal objective value than the other solvers. It has a narrow terminal objective value distribution , indicating that it consistently converges to the optimal solution and best matches the desired trajectory. PS and PSO achieve near-optimal results quickly during training but consistently miss the global optimum. Furthermore, when examining the objective function convergence behaviour of PS, the comparatively less flat curve relative to GS and the others suggests that improvement had not fully stalled at termination, indicating that PS may have stopped due to the mesh tolerance criterion before complete objective function convergence was achieved. This will be elaborated on in the Future Work – Optimization and Implementation section.

FMINCON, the local optimizer, exhibits a training progress curve similar to GS but does not reach the global optimum either. GA has a progress curve shape that lies between FMINCON and GS,

and PS and PSO, with relatively high standard deviations and, on average, terminal results similar to PSO. Ultimately, all these optimizers are metaheuristic or gradient-based and thus have no formal proof certifying their solutions as globally optimal and can only be considered a strong candidate for the global optimum rather than a certified one. This reinforces the fact that a comparative analysis is necessary when evaluating for a global optimum.

As shown in Table 3 and Figure 19, MATLAB's GS and its SQP solver produced the best linkage configuration with a trajectory-matching value of $R_1(\boldsymbol{x}) = 0.0781$. Its output path is shown in orange in Figure 20 alongside the desired trajectory marked by blue asterisks. The *Optimized* path corresponds to the shape-optimized and linearly transformed version of the optimal linkage. In this figure, the output path closely aligns with the desired path, although some points are less well matched, particularly the points at the left, right, and bottom.

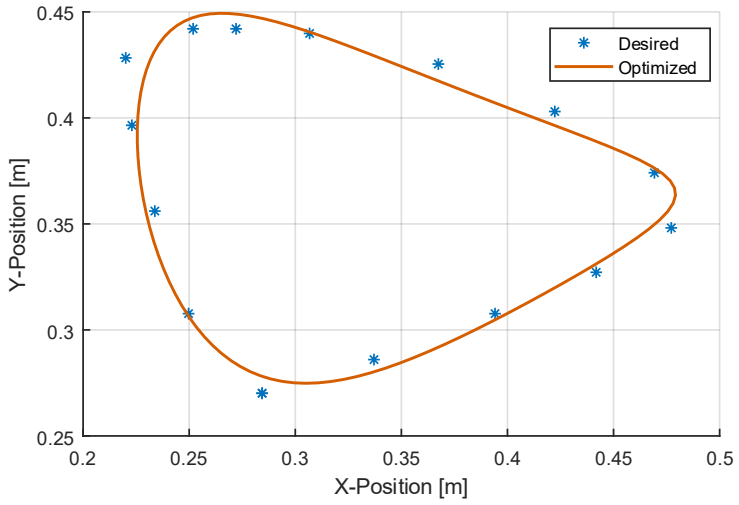

Figure 20. Desired path points and linearly transformed optimal path (Final) in the XY plane

In Figure 21, one can see the normalized Fourier coefficient approximations for the desired (*Desired*) and shape-optimal (*Optimized*) paths produced during the path-shape optimization before the linear transformations. Some of the same path-shape discrepancies (furthest left and bottom) seen in Figure 20 are also observed in Figure 21.

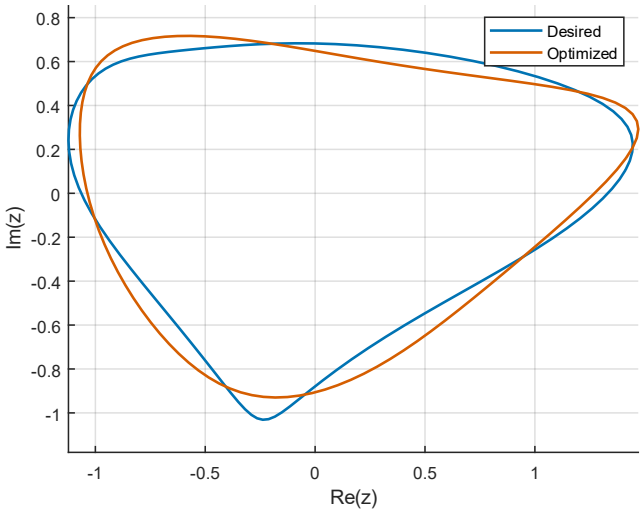

Figure 21. Normalized Fourier-transformed desired path and optimal path generated using GS

GS is a gradient-based algorithm that uses SQP and scatter-search design-space sampling to improve its starting points [63]. It excels at solving locally smooth, non-convex problems with moderate dimensionality. To investigate the smoothness and non-convexity of the trajectory-matching problem, the r_{12}-r_{23} hyperplane in the design space was generated and plotted in Figure 22. The hyperplane shows the trajectory-matching objective value $f_1(\boldsymbol{x})$ on the z-axis and r_{12} and r_{23} on the x and y axes. It was created by calculating $f_1(\boldsymbol{x})$ for each combination of r_{12} and r_{23} that covers the extent of their length constraints, while keeping all other design parameters identical to those in row "*Trajectory*" in Table 9 in Appendix B. The values of $f_1(\boldsymbol{x})$ were all sign-changed to negative to reflect global maxima rather than minima, which better showcase the nonconvexity to the reader.

In the 3D plot, two local maxima are clearly visible. This graph represents only two of the five design variables involved in the trajectory-matching optimization; therefore, if two local maxima exist within this segment of the design space, it is reasonable to assume many more exist across the entire combinatorial space. Consequently, the optimization algorithms presented above can become trapped in local maxima and fail to reach the global optimum consistently.

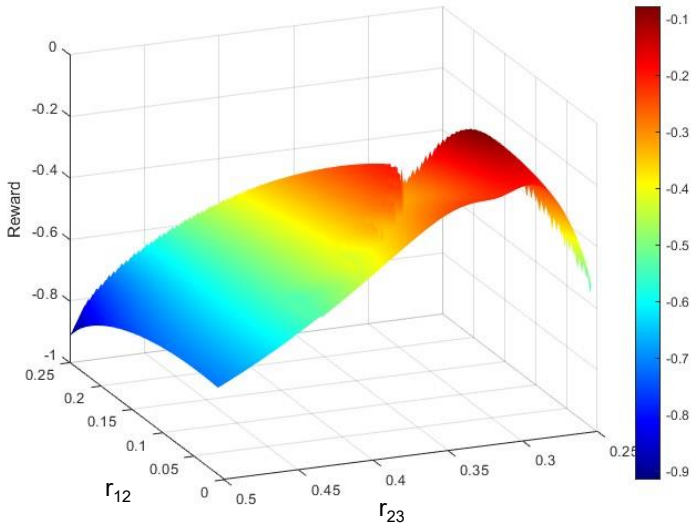

Figure 22. Solution space hyperplane formed with r_{12} and r_{23} varied between [0.01, 0.25] and [0.1,0.5] respectively, while all other design parameters are identical to the GS trajectory-optimal result

Moreover, by looking at Figure 22, one can see that this problem is locally smooth and globally nonconvex due to the presence of multiple local maxima.

4.1.1 Training Comparison: RL vs Global Optimization

A comparison was made between the GS and the continuous control TD3 and DDPG RL algorithms, where the instantaneous best reward value is graphed as a function of percent completeness of the training (0-100%) in Figure 23. GS's objective values are negated and treated as reward values for ease of comparison between the GS and RL. The solid lines show the average instantaneous best reward across 50 runs, while the bands indicate the distribution of reward values. The average completion times and average final best reward values of each algorithm are displayed in the legend. The number of steps and episodes for the TD3 and DDPG algorithms was selected to match GS's computational time.

One can see that GS exhibits gradual improvement, with an ever-narrowing objective value distribution band, until almost all runs converge to the same optimal result near the 70% completion mark, with an average final reward of R = -0.0782. This behaviour is near-deterministic, given the minimal terminal reward value distribution, indicating that GS efficiently explores the design space and converges to the same maximum in nearly every run, rarely becoming trapped in local maximum.

Conversely, the continuous-control TD3 and DDPG algorithms converge quickly, completing most of their rapid convergence within the first 30% of training, after which they significantly slowed their progress for the remainder of training. Early convergence suggests that the agent is getting stuck in a local maximum. Notably, the high terminal objective value distribution indicates that the design space contains multiple local maxima that traps the TD3 and DDPG agents. This early convergence could be due to actor network overfitting, which occurs when the network learns too quickly from early states, reducing exploration and leading to poor performance. This issue was observed across all RL algorithms and was difficult to mitigate. Attempts to minimize policy overfitting included increasing exploration noise, reducing learning rates, employing L2 regularization to promote smaller network weights early on, and experimenting with different network architectures. Presented are the most successful outcomes obtained using these overfitting mitigation strategies.

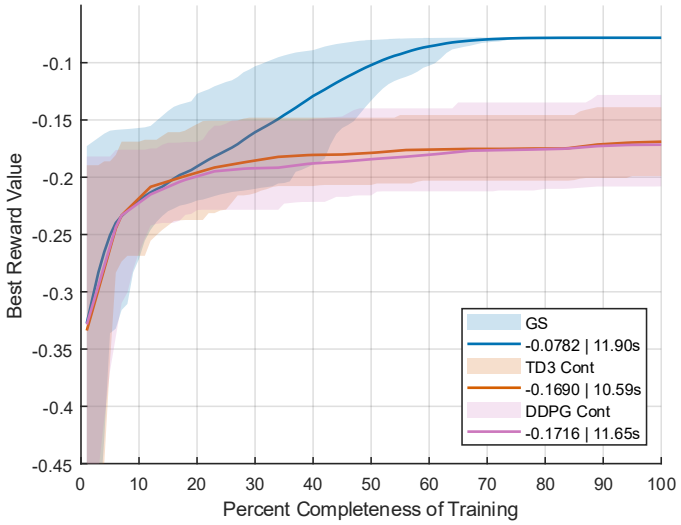

Figure 23. Comparison of the reward value as a function of training session percent completeness of GS, and continuous control TD3 and DDPG, showcasing the mean (solid line) and reward value distribution (shaded area), n=50.

Next, the algorithms from Figure 23 are compared with a modified version of the TD3 and DDPG algorithms that follow the principles of degenerate RL [113,114] where the state is optimized in a single step, and at each episode of this problem, the state is reset to the initial *Baseline* state before an action is taken. Therefore, in degenerate RL, the actor NN aims to find the optimal one-shot action policy that maps the initial state to the optimal state in a single step. This comparison is shown in Figure 24, and one can see that single-step degenerate RL algorithms achieve lower

terminal rewards than continuous algorithms; thus, they get stuck earlier in even lower-reward local minima.

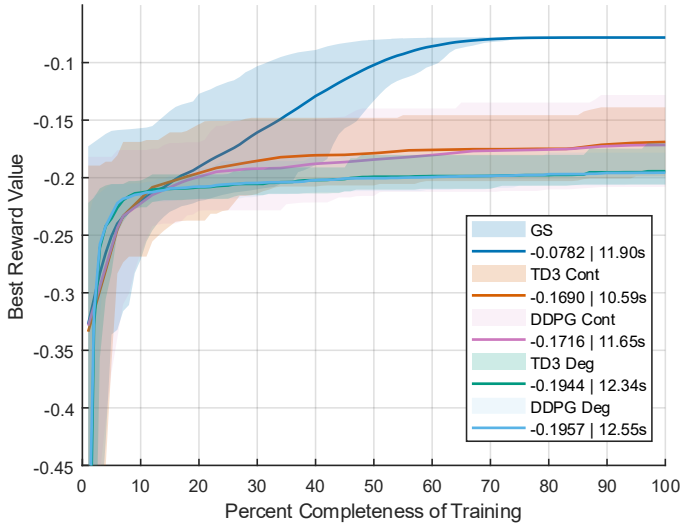

Figure 24. Comparison of the reward value as a function of training session percent completeness of GS, and continuous control and single-step degenerate TD3 and DDPG, showcasing the mean (solid line) and reward value distribution (shaded area), n=50.

In a final attempt to improve TD3 and DDPG, the algorithms are hybridized with FMINCON to enhance training after 50% of the training is complete, aiming to boost performance after they plateau. This result is shown in Figure 25 and again, one can see that they are outperformed by GS and exhibit a large reward value distribution at the end of training. This final attempt confirms that

GS and classical global optimization are better suited to this locally smooth, non-convex combinatorial optimization problem.

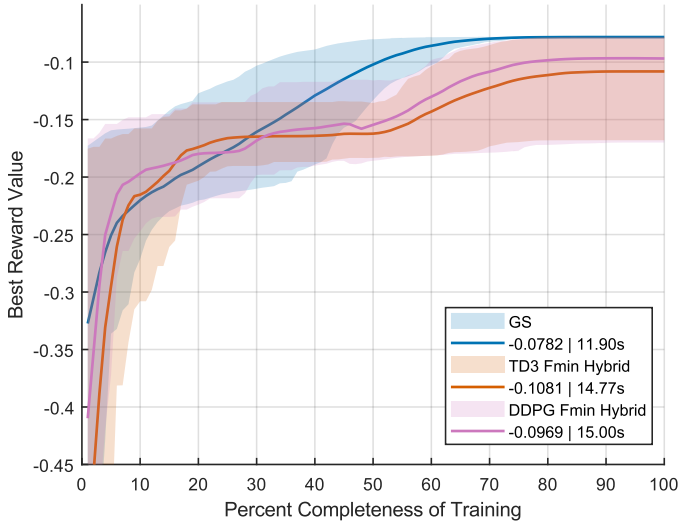

Figure 25. Comparison of reward value as a function of training session percent completeness of GS, and continuous control TD3 and DDPG fmincon hybrid, showcasing the mean (solid line) and distribution bounds (shaded area), n=50.

4.2 Multi-objective optimization

The single-objective trajectory-matching optimization to improve efficiency was successful, but an interesting phenomenon emerged in the GS results, which catalyzed multi-objective optimization research.

A boxplot in Figure 26 shows the distributions of the final reward values for GS, and TD3 and DDPG FMIN Hybrids. GS exhibits near-deterministic behaviour, with only a single outlier, while the hybrid TD3 and DDPG algorithms encounter a few local minima near -0.17 and -0.08. Additionally, the hybrid algorithms achieve results close to the optimal point of -0.781, but this occurs much less frequently than with GS.

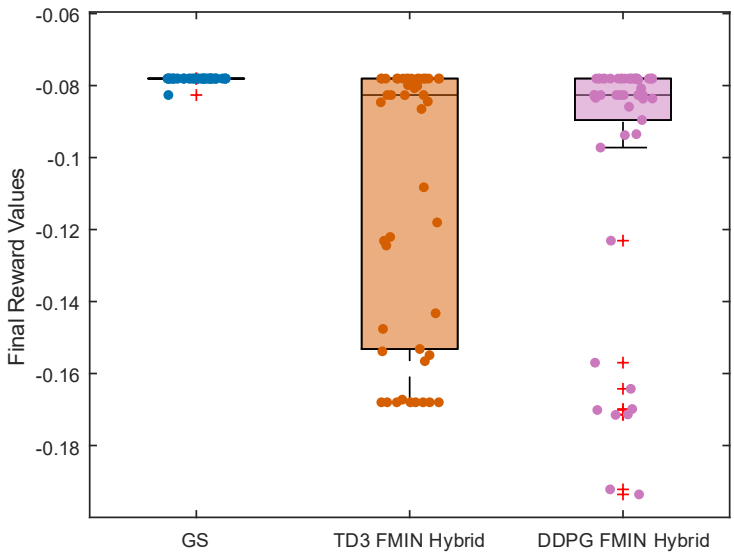

Figure 26. Boxplot of the final reward values of the GS and RL FMIN hybrid algorithms

A distinct outlier is evident in the distribution of the GS final reward value shown in this figure. This outlier has a trajectory-matching value of -0.0826, whereas the others have a value of -0.0781. This outlier occurred in 4% of runs (2 out of 50). GS's two configurations, which will be referred to as "*Trajectory*" and "*Outlier*", are graphed in Figure 27 and Figure 28.

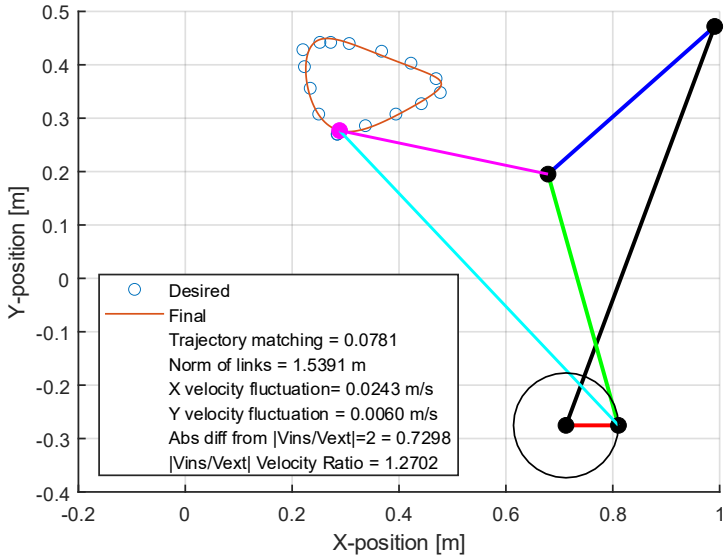

Figure 27. Linkage corresponding to the Trajectory configuration

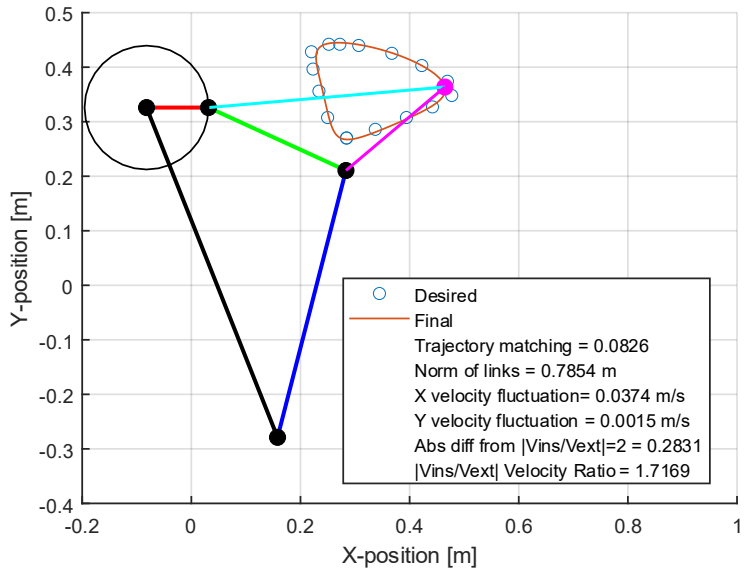

Figure 28. Linkage corresponding to the trajectory-optimized Outlier configuration

By examining the resulting linkages, their trajectory-matching values, and the norm of their link lengths, it becomes clear that the *Outlier* configuration has a slightly lower trajectory-matching value but a much smaller norm of link lengths. Link lengths are directly correlated with material cost and link mass. Link mass, in turn, influences the dynamics of the entire system, as higher mass requires more robust components and a larger driving motor. The following section aims to assess whether it is possible to improve upon the *Outlier* configuration's trajectory results with minimal link length increase.

4.2.1 Case 1: Trajectory Matching and Norm of Link Lengths

A two-objective MOGA was set up to further explore the possibility of finding a more optimal configuration in terms of both trajectory matching and link length norm than the *Trajectory* and *Outlier* configurations. This algorithm ran for five hours and produced a Pareto front of 1121 configurations, where n is the number of configurations in the Pareto front. The Pareto front is illustrated in Figure 29 and contains the *Trajectory* (yellow cross) and *Outlier* (orange cross) configurations displayed above. It is a well-defined, dense Pareto front, consisting of a main and a secondary front. The main front trends towards the *Outlier* configuration, and the other front trends towards the *Trajectory* configuration. The MOGA produced multiple configurations with slightly poorer trajectory matching than the *Outlier* configuration but considerably shorter links.

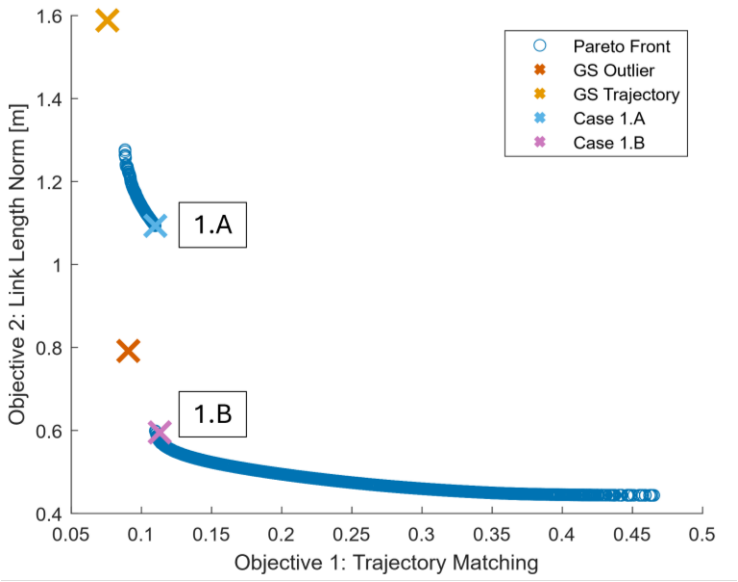

Figure 29. Pareto front of trajectory matching versus link length norm, showing the Outlier, Trajectory, 1.A, and 1.B configurations, and m, n = 1121, t =5 h

The link length norm minimum of the secondary front is highlighted by a light blue cross and labelled as 1.A, and the trajectory matching minimum of the main front is highlighted by the purple cross and labelled as 1.B. These configurations are significant because they lie on either side of a discontinuity in the Pareto front and represent a large improvement in link length norm from 1.A to 1.B.

Configuration 1.A and 1.B's linkages are shown in Figure 30 and Figure 31, respectively, and they are compared to the *Baseline*, *Trajectory*, and *Outlier* configurations in Table 4. A comparison of their objective values shows that configuration 1.B exhibits only a negligible decrease in trajectory-matching performance (0.1099 versus 0.1098 for configuration 1.A). In contrast, 1.B achieves a substantially lower link-length norm (0.5989 m compared with 1.0932 m for 1.A). Thus, configuration 1.B offers a major improvement in form factor and mass for a marginal loss in trajectory accuracy. This makes this Pareto optimal configuration highly attractive for implementation, if trajectory matching and link length norm were the only design objectives. There also exists a significant difference in their fixed link orientation, therefore the discontinuity in the Pareto front could represent the transition from orientation to the other. This also showcases the optimization formulation's ability to produce a variety of 4R linkage designs and fixed link orientations.

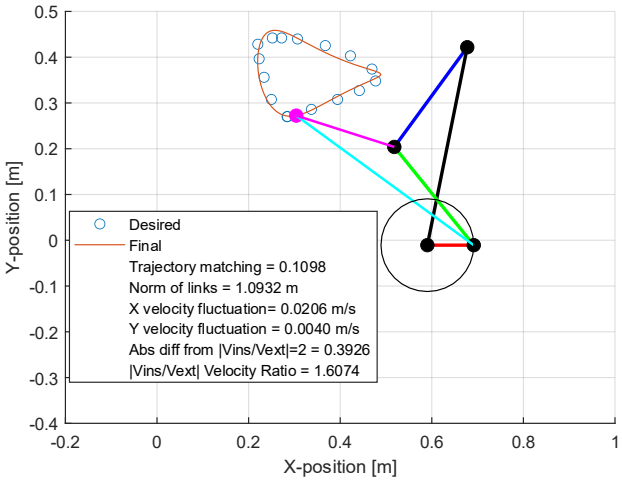

Figure 30. Linkage corresponding to the 1.A configuration

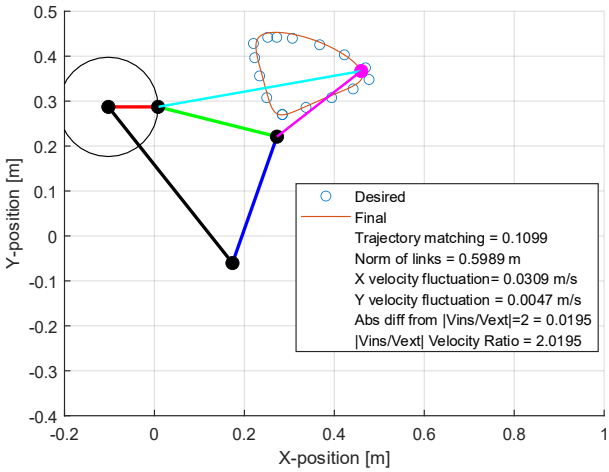

Figure 31. Linkage corresponding to the 1.B configuration

Configuration 1.B may initially appear attractive, as its trajectory-matching performance is approximately 75% better than the *Baseline* and its link-length norm is only about 20% higher. However, it exhibits a 301% increase in X-direction velocity fluctuation relative to the *Baseline*. This outcome is expected, as the MOGA used for Case 1.A and Case 1.B did not include velocity fluctuation as an optimization objective. Because X-direction velocity has been empirically shown to be critical to the membrane-extraction process, the three subsequent multi-objective optimizations will explicitly target its improvement.

Table 4. Performance comparison of the Baseline and optimized configurations: Trajectory, Outlier, 1.A, and 1.B

	Trajectory Matching	***% Diff1***	***Norm of Links [m]***	***% Diff***	***X Velocity Fluctuation [mm/s]***	***% Diff2***	***Y Velocity Fluctuation [mm/s]***	***% Diff3***	***|Vins/Vext| Velocity Ratio***	***% Diff4***
Baseline	0.4391	—	0.4940	—	0.0077	—	0.0017	—	1.3265	—
Trajectory	0.0781	-82.21	1.5391	211.56	0.0243	215.58	0.0060	252.94	1.2702	-4.24
Outlier	0.0826	-81.19	0.7854	58.99	0.0374	385.71	0.0015	-11.76	1.7169	29.43
Case 1.A	0.1098	-74.99	1.0392	110.36	0.0206	167.53	0.0040	135.29	1.6074	21.18
Case 1.B	0.1099	-74.97	0.5989	21.23	0.0309	301.30	0.0047	176.47	2.0195	52.24

4.2.2 Case 2: Trajectory, X- and Y-Direction Velocity Fluctuation

Before proceeding to optimize for velocity fluctuations, the correlation between velocity fluctuations and acceleration was examined. Velocity fluctuations and acceleration were identified early on as key kinematic performance metrics. In Miler et al. [125], a multi-objective optimization of a Chebyshev–Lambda linkage for the design of a walking robot targeted horizontal velocity fluctuation and maximum acceleration as independent objectives. To evaluate whether optimizing the velocity fluctuations and accelerations is redundant for this case, two three-objective MOGA optimizations were conducted: one for the X-direction and one for the Y-direction. They had the following objectives: trajectory matching, velocity fluctuations, and RMS (root-mean-square) acceleration. The first optimization, shown in Figure 32 for X-velocity and X-direction RMS acceleration did not yield a trade-off surface but rather a line segment in the velocity fluctuation–acceleration plane, demonstrating that as X-velocity fluctuations improve, so does X-direction RMS acceleration, confirming their correlation.

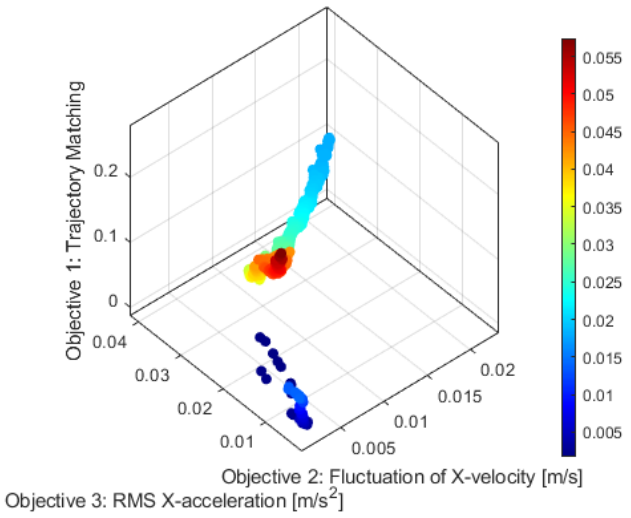

Figure 32. Three-objective Pareto front: trajectory matching vs. X-velocity fluctuation vs. RMS X-acceleration, with point colours representing trajectory matching, n=1767, t =1 h.

The second optimization is shown in Figure 33. For Y-velocity and Y-direction RMS acceleration, a line segment in the velocity fluctuation–RMS acceleration plane was also produced, demonstrating that these objectives are also correlated. For the remainder of the research, only the velocity fluctuation will be optimized, assuming the acceleration is sufficiently reduced. This correlation study did not take place in [125] , because the formulation targeted maximum acceleration, and velocity fluctuation cannot be minimized in parallel to directly reduce the maximum acceleration. However, their velocity fluctuations and maximum accelerations appeared to be correlated based on visual inspection of the plots.

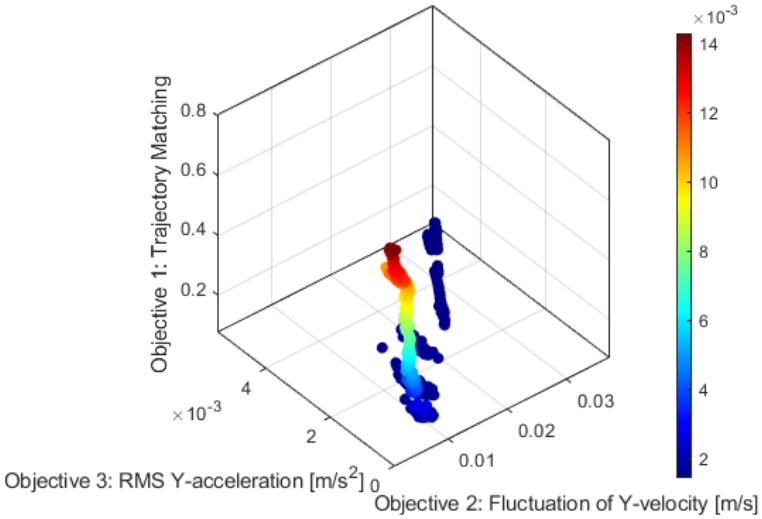

Figure 33. Three-objective Pareto front: trajectory matching vs. Y-velocity fluctuation vs. RMS Y-acceleration, with point colours representing trajectory matching, n=1121, t =1 h.

Following the correlation study, the Case 2 three-objective MOGA simultaneously optimized the linkage for trajectory matching and X- and Y-direction velocity fluctuations and ran for five hours. It yielded 889 configurations after outlier removal, representing 20.7% fewer than in Case 1. The resulting Pareto front, shown in Figure 34, contains a dense region but also a significant number of configurations that do not lie on the trade-off surface. This non-uniformity, together with the proportion of removed outliers, suggests that among the three objectives there may be a highly conflicting pair, contributing to increased optimization complexity and a less uniform Pareto front compared to Case 1.

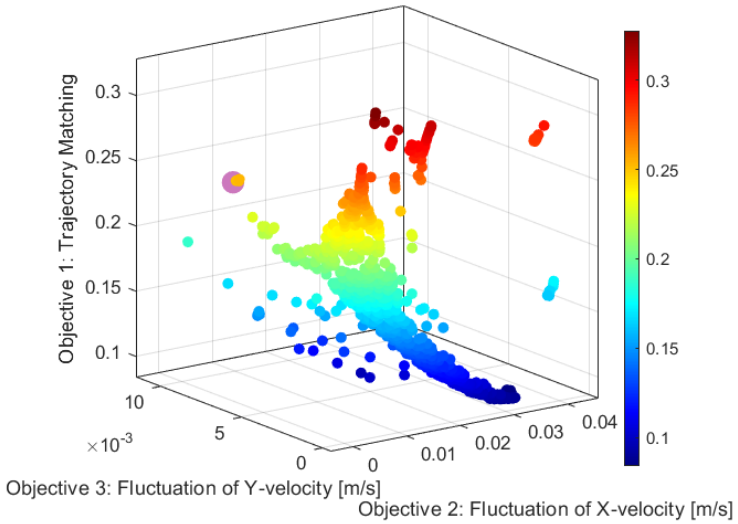

Figure 34. Case 2's Pareto front, with point colours representing trajectory matching, and the point with the lowest X-direction velocity fluctuation highlighted in purple, n=889, t =5 h.

Ultimately, minimizing X-direction velocity fluctuations is the most important objective of these optimizations. Therefore, the configuration with the lowest X-direction velocity fluctuation was selected as an optimal configuration from the Pareto set. It is highlighted by the purple marker in the 3D Pareto front. One can see that the configuration sits quite high on the Trajectory Matching axis of Figure 34, thus showing lower trajectory-matching accuracy and indicating a trade-off in trajectory matching to minimize velocity fluctuations.

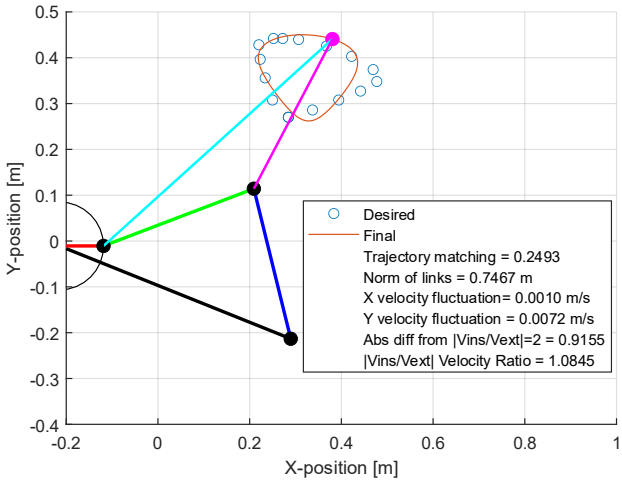

Figure 35. Case 2 lowest X-direction velocity fluctuation configuration

One can also see from the link-length norm metric that the link lengths are significantly longer than those of Case 1.B. Therefore, in the next section, the link length will be optimized alongside trajectory matching and X- and Y-direction velocity fluctuations to address this increase in link length.

4.2.3 Case 3: Trajectory, X- and Y-Direction Velocity Fluctuation, and Norm of Links

The link length norm objective was combined with the objectives of Case 2; thus, the algorithm was run to simultaneously optimize trajectory matching, X-velocity and Y-velocity fluctuations, and the link length norm. The five-hour runtime yielded 406 configurations after outlier removal, substantially fewer than in Cases 1 and 2. However, the resulting Pareto fronts, seen in Figure 36 remain well populated. The Z axis of the Pareto front on the left depicts the link length norm objective, and the Z axis on the right depicts the trajectory-matching objective values. In both graphs, the X and Y axes denote the objective results for velocity fluctuation. Both 3D plots are

necessary to represent the full extent of Case 3's four-objective Pareto front since only three objectives can be graphed at once.

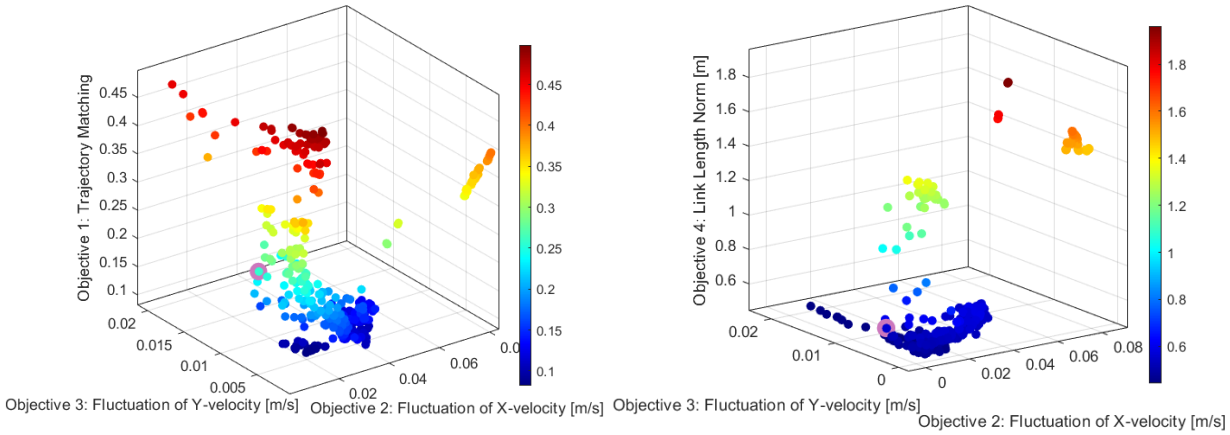

Figure 36. Case 3's Pareto fronts, with point colours representing the Z axis, and the point with the lowest X-direction velocity fluctuation highlighted in purple, n=406, t =5 h.

The Pareto fronts in Figure 36 are sparser than Case 2's Pareto front, indicating that the optimization solver had more difficulty optimizing across all four objectives. This observation suggests that the inclusion of all four objectives may introduce at least one strongly conflicting objective pair, contributing to the increased difficulty of the optimization.

Furthermore, since minimizing X-direction velocity fluctuations is the most important objective of these optimizations the Case 3 optimal was selected by selecting the configuration with lowest X-direction velocity fluctuation. This optimum is highlighted in the 3D Pareto front of Figure 36 by the purple marker, and its linkage is shown in Figure 37.

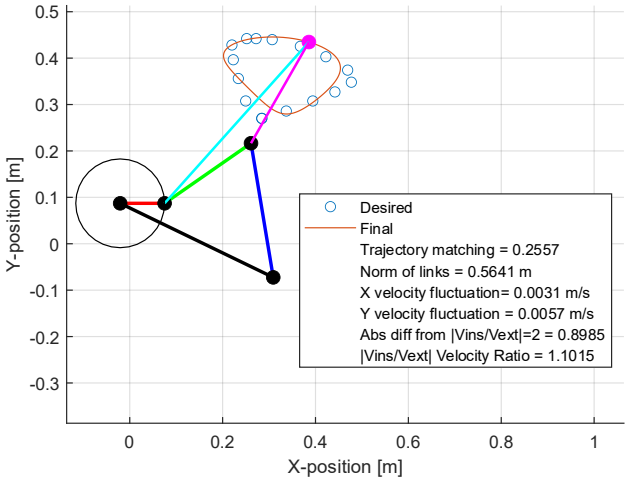

Figure 37. Case 3 lowest X-direction velocity fluctuation configuration

As expected, the performance metrics shown in Figure 37 indicate that Case 3 optimal has shorter links than Case 2, but this comes at the cost of a significant increase in X-direction velocity fluctuation. The Case 2 and 3 X-direction velocity fluctuation optimal configurations have all also had velocity ratio values that are lower than the *Baseline*. The velocity ratio objective is thus be added to the Case 4 optimization.

4.2.4 Case 4: Trajectory, X- and Y-Direction Velocity Fluctuation and Absolute Velocity Ratio Difference

Case 4's five-objective MOGA had a five-hour runtime yielding 90 configurations after outlier removal, substantially fewer than in the three previous cases, and resulted in sparse Pareto fronts shown in Figure 38, Figure 39, and Figure 40. The Z axis in Figure 38 corresponds to the trajectory-matching objective, the Z axis of Figure 39 corresponds to the link length norm objective, and the Z axis of Figure 40 corresponds to the velocity ratio objective. The X and Y axes of all three figures denote the objective results for velocity fluctuation.

The noticeable sparsity of the Pareto fronts indicates that the solver struggled to optimize all five objectives within the 5-hour timeframe. This outcome suggests that incorporating the velocity ratio objective into the algorithm, already challenged by the simultaneous optimization of four objectives and showing signs of pairs of highly conflicting objectives, rendered the problem excessively complex for the time constraint.

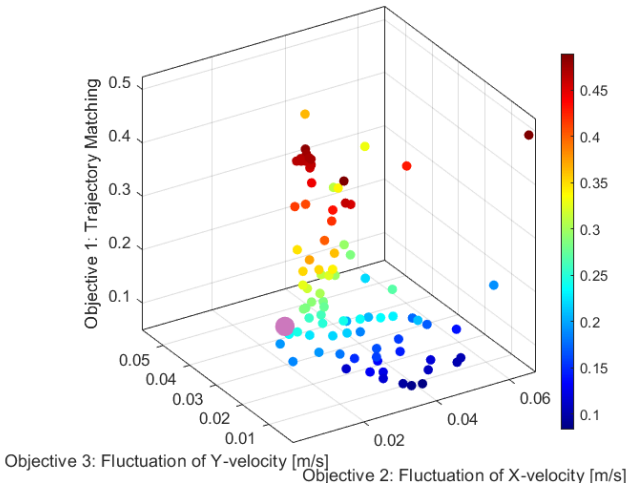

Figure 38. Case 4's trajectory matching and X- and Y-direction velocity fluctuation Pareto front, with point colours representing trajectory matching and the point with the lowest X-direction velocity fluctuation highlighted in purple, n=90, t = 5 h.

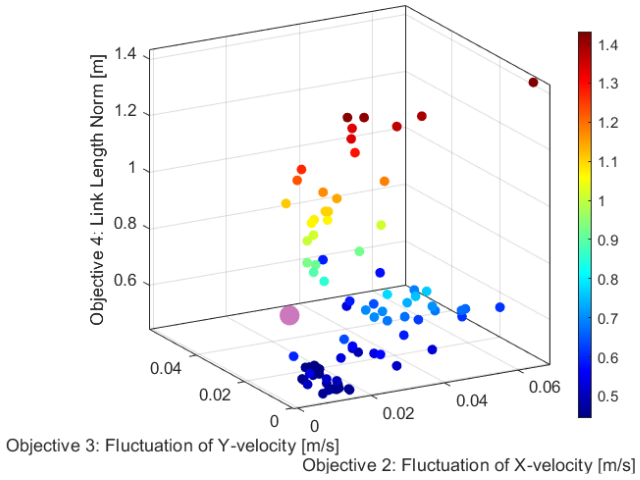

Figure 39. Case 4's link length and X- and Y-direction velocity fluctuation Pareto front, with point colours representing link length norm and the point with the lowest X-direction velocity fluctuation highlighted in purple, n=90, t = 5 h.

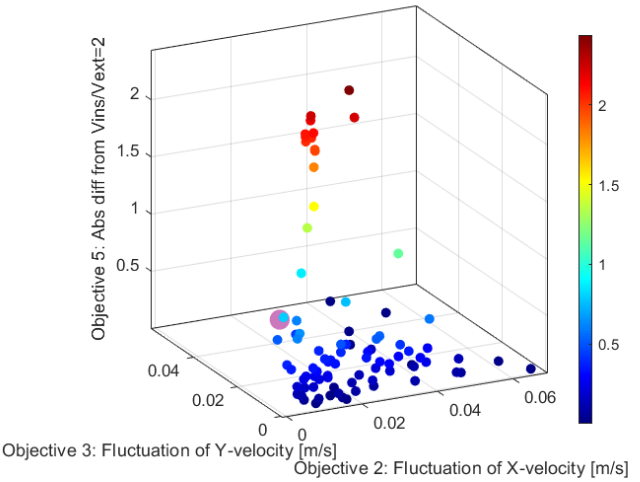

Figure 40. Case 4's absolute velocity difference and X- and Y-direction velocity fluctuation Pareto front, with point colours representing absolute velocity ratio difference and the point with the lowest X-direction velocity fluctuation highlighted in purple, n=90, t = 5 h.

Once again, the case four optimal corresponds to the configuration with lowest X-direction velocity fluctuation. This optimum is highlighted in the 3D Pareto front of Figure 38 , Figure 39, and Figure 40 by the purple marker, and its linkage can be seen in Figure 41. This configuration exhibits low x-velocity fluctuations and a higher insertion-to-extraction velocity ratio than the Case 2 and 3 optimal configurations; however, it also features longer links and elevated Y-direction velocity fluctuations.

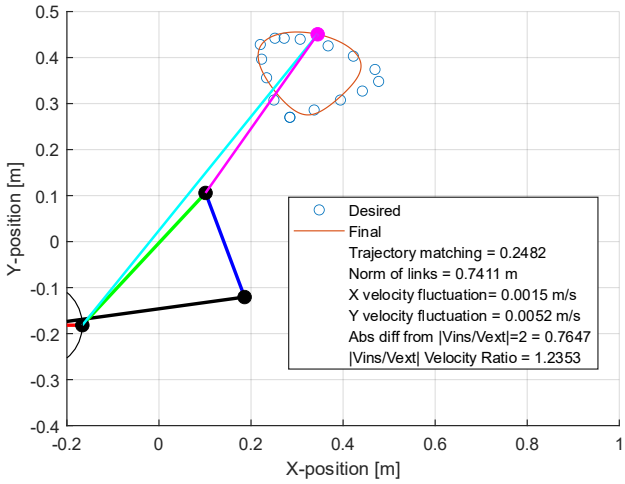

Figure 41. Case 4 lowest X-direction velocity fluctuation configuration

Chapter 5

5 Discussion and Future Work

The research presented in this thesis optimized the reliability and efficiency of a four-bar linkage for mechanically extracting eggshell membranes from prepared eggs on a constant-velocity conveyor under a continuous duty cycle. This section provides a detailed discussion of the comparison between classical global optimization and reinforcement learning for trajectory-matching linkage synthesis, the multi-objective optimization, and an in-depth analysis of the resulting velocity fluctuations and efficiency gains. Future work is also discussed, focusing on reformulating the reinforcement learning approach, extending the multi-objective optimization framework, and on the next steps to assess the real-world feasibility of the proposed linkage configuration.

5.1 Classical Global Optimization vs. Reinforcement Learning

Ultimately, classical global optimization techniques outperformed the proposed RL formulations in solving the constrained combinatorial optimization problem. The five-degree nonlinear problem is locally smooth and globally nonconvex and was solved most efficiently by MATLAB's GS algorithm, averaging 11.9 seconds. This algorithm uses SQP for objective-function gradient analysis and uses the gradients to interact directly with the design variables, rather than indirectly using NNs as in RL. SQP's direct use of the objective function's gradient also helped GS handle the continuous nature of the design space and objective function. GS demonstrated superior performance compared to all three formulations, including the continuous, degenerate, and continuous fmincon hybrid formulations implemented for the TD3 and DDPG RL agents.

Meticulous effort went into designing the environment, reward function, policy, and NN architecture, as well as tuning the hyperparameters. However, the system still converged prematurely to local minima. Early NN convergence is often attributed to NN overestimation of Q-values or to vanishing gradients arising from an overly large or complex network architecture. This may have been the case here, as the problem involves only five design variables, but three actor sub-networks with three hidden layers, and 256 neurons each were used. Although smaller covariance-matrix-based architectures were tested during refinement, the design presented remains

the most effective. Still, the covariance matrix formulation itself, using three sub-networks, may be inherently too complex for this problem. The stochasticity of the covariance matrix coupled with the multi-start formulation and the action exploration noise may have ultimately inhibited the agent from consistently converging on the optimal result. However, the multi-start formulation is promising since continuous formulation outperformed the degenerate formulation which was lacking multi-start.

Furthermore, this problem may have been affected by the curse of dimensionality, given its continuous design space and reward function, making it difficult for RL agents to sufficiently explore and generalize across the state–action space. In fact, RL is particularly well-suited to optimizing discrete decision-making policies. This approach is illustrated in Fogelson's work [117], where RL is used to learn a discrete decision policy for constructing N-bar (multi-bar) linkages. Conversely, the covariance matrix adaptation evolutionary strategy (CMA-ES), a classical global optimization algorithm, is used to optimize link lengths and help the RL agent solve hypercomplex trajectory-matching problems. His work extends Lipson's work on N-bar linkages, which originally used evolutionary algorithms [115].

Furthermore, RL excels at model-free adaptive control, where it can use online learning and a longer timeframe to optimize a system's control policy by interacting with its environment via an array of sensors. Using a model of the system dynamics and environment, an RL agent can pre-train the actor and critic networks to enable smoother deployment in the real environment. This pre-training resembles the approach applied in the four-bar linkage optimization conducted in this study. At its core, applying RL to combinatorial optimization in a static environment underutilizes its capabilities.

Finally, classical supervised and unsupervised ML are better suited for linkage synthesis problems than RL. This is supported by Nobari's success in using a dataset of 100 million 4- to 20-bar planar linkages to train NNs for multi-bar trajectory-matching linkage synthesis [107]. Or Nurizada's research at Stony Brook University, where a dataset of over 12 million linkages containing both revolute and prismatic joints was used to train NNs for open and closed curve synthesis [111]. Since large datasets and deep networks require substantial computational resources, this research aimed to explore whether RL could leverage NNs in a more resource-efficient manner for a static combinatorial task.

Conversely, the positive results of the classical global optimization techniques helped demonstrate that the FD-based objective function was well-suited to the trajectory-matching problem between the four-bar's end-effector and the manually entered desired path. The Fourier-domain normalization produced results independent of point count, path-traversal speed, scale, and pose, as demonstrated by the wide range of resulting configuration orientations. Furthermore, the efficiency of classical global optimization in solving the trajectory-matching problem supported the feasibility of conducting a systematic multi-objective optimization study incorporating the four remaining design objectives.

5.2 Multi-Objective Optimization Case Comparison

Table 5 compares performance metrics across Cases 2, 3 and 4, relative to the *Baseline*, including percentage differences.

Table 5. Performance comparison of the Baseline and the lowest X-direction velocity fluctuation configurations from Cases 2, 3, and 4

	Trajectory Matching	***% Diff1***	***Norm of Links [m]***	***% Diff2***	***X Velocity Fluctuation [mm/s]***	***% Diff3***	***Y Velocity Fluctuation [mm/s]***	***% Diff4***	***|Vins/Vext| Velocity Ratio***	***% Diff5***
Baseline	0.4391	—	0.4940	—	7.6540	—	1.6690	—	1.3265	—
Case 2 XOpp	0.2493	-43.22	0.7467	51.15	1.0080	-86.83	7.2052	331.71	1.0845	-18.24
Case 3 XOpp	0.2557	-41.77	0.5641	14.19	3.0520	-60.13	5.6971	241.35	1.1015	-16.96
Case 4 XOpp	0.2482	-43.48	0.7411	50.02	1.5359	-79.93	5.1762	210.14	1.2353	-6.88

This comparison of the optimal X-direction velocity-fluctuation configurations shows that Case 2 exhibits significantly better trajectory matching than the *Baseline* and achieves the lowest

combined X- and Y-velocity fluctuations among all four cases. However, it has the longest links and the lowest velocity ratio, which is significantly lower than the *Baseline*.

Case 3 yields the shortest links of the three cases, as expected, since the link length norm was included as an optimization objective. It exhibits higher X-velocity fluctuations and slightly worse trajectory matching than Case 2, and a lower velocity ratio than the *Baseline*.

Finally, Case 4, which simultaneously optimizes all five objectives, achieves the best trajectory matching, the second-best X-velocity fluctuation, and the most favourable insertion-to-extraction velocity ratio among the last three cases. However, it has the second-longest links, and its velocity ratio remains lower than the *Baseline*.

Overall, the optimizations generated a diverse set of linkage configurations that improved the design objectives to varying degrees, and each case produced at least one design with substantially lower X-direction velocity fluctuation than the *Baseline*. However, the relative center-axis offset between the tool and the egg during extraction, as well as the correlations among the optimization objectives, have yet to be examined. Moreover, all three configurations that minimize X-direction velocity fluctuations exhibit velocity ratios lower than that of the *Baseline*. The analyses of the tool and egg offset, correlations between optimization objectives, and path symmetry are presented in the following sections of this chapter.

5.2.1 Tool and Egg Offset Analysis

Figure 42 illustrates the relative offset between the tool and the egg's center axis over time as the tool is within the egg. It compares the relative motion of the *Baseline* and the optimal configurations for X-direction velocity fluctuations across cases 2-4 (cases optimized for velocity fluctuations). This graph was generated by synchronizing the tool and egg motion such that their vertical axes are aligned at the first point of contact during the compliant stroke and by equating the egg's horizontal velocity to the average horizontal velocity of the end-effector when it is within the egg. The results from this graph are also tabulated in Table 6 where the maximum absolute offset, offset at entry, and absolute offset at exit are shown.

Figure 42 shows that the *Baseline* has a significant oscillating center-axis offset throughout its time in the egg. Conversely, the three optimal cases have significantly flatter curves, with Cases 2 and

4 being the flattest. This indicates that they have a much more constant horizontal velocity while the tool is within the egg.

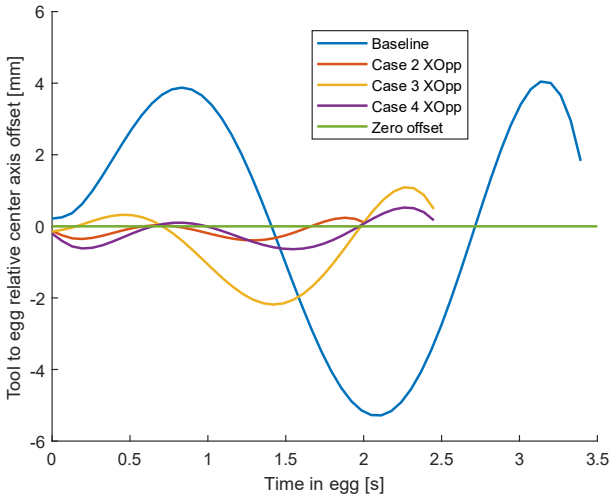

Figure 42. Relative center axis offset between the tool and egg as a function of time in the egg for the Baseline and the lowest X-direction velocity fluctuation configurations of Case 2, 3, and 4

The maximum absolute offset of the *Baseline* is 5.28 mm, which is much higher than Cases 2, 3, and 4 with 0.39 mm, 2.19 mm, and 0.64 mm, respectively. One can also see that the *Baseline* exits near the egg's leading edge, with an offset of 1.83mm. This exit offset is similar, but not as significant as the offset experienced by the prototype and shown in Figure 4, meaning the alignment and timing of the prototype linkage weren't optimal. The offsets at exit of Cases 2, 3, and 4 are 0.09 mm, 0.49 mm, and 0.17 mm, respectively. Finally, the *Baseline* has an absolute offset at entry of 0.22 mm, whereas Cases 2, 3, and 4 have offsets of 0.13 mm, 0.14 mm, and 0.20 mm, respectively. These results show a substantial improvement in tool offset over the *Baseline*.

Table 6. Relative tool offset comparison for the Baseline and the lowest X-direction velocity fluctuation configurations of Cases 2, 3, and 4

Configuration	***Maximum Absolute Offset [mm]***	***% Diff***	***Absolute Offset at Entry [mm]***	***% Diff***	***Absolute Offset at Exit [mm]***	***% Diff***
Baseline	5.28	—	0.22	—	1.83	—
Case 2 XOpp	0.39	-92.57	0.13	-93.00	0.09	-94.89
Case 3 XOpp	2.19	-58.63	0.14	-92.15	0.49	-73.44
Case 4 XOpp	0.64	-87.92	0.20	-88.91	0.17	-90.79

Figure 43 shows the Y-direction velocity plotted against membrane extraction time, indicating that Cases 2, 3, and 4 exhibit smooth acceleration during extraction compared to the flat but slightly oscillating profile of the *Baseline*. The significance of the Y-velocity fluctuation has not yet been experimentally determined.

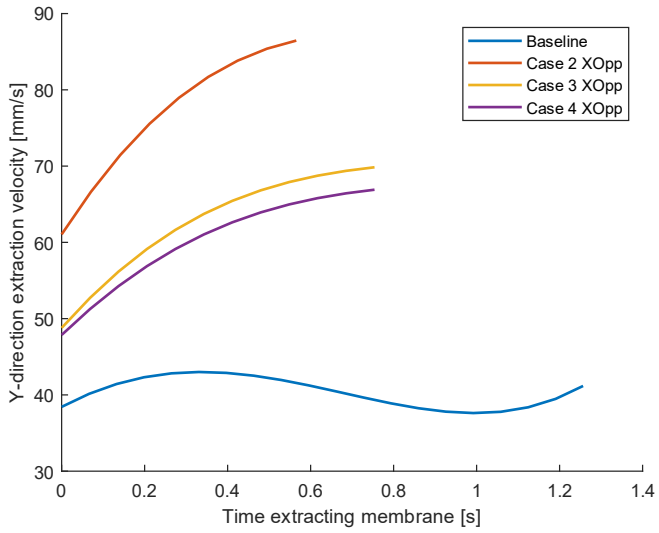

Figure 43. Y-direction velocity as a function of extraction time egg for the Baseline and the lowest X-direction velocity fluctuation configurations of Case 2, 3, and 4

Furthermore, the *Baseline* has a path length of 0.8647 m, whereas Case 2's X-direction velocity fluctuation optimal has a path length of 0.6162 m. Case 2's path length is thus 29% shorter, resulting in a rate of eggs processed per second that is 40% higher than the baseline (in-depth calculation shown in Appendix C). This is a collateral benefit of the trajectory-matching objective, since the desired path shape, which was derived directly from empirical evidence gathered during prototype testing, is substantially shorter than the baseline path. This marks a significant improvement in system efficiency attributed to the optimization study.

5.3 Optimization Objective Correlation Analysis

The multi-objective optimization study yielded multiple Pareto fronts, three of which were generated by distinct MOGA formulations aimed at minimizing the tool's X-direction velocity fluctuation within the egg. Following the initial outlier filtering, these Pareto fronts exhibited progressively increasing sparsity as the number of objectives increased, suggesting heightened competition among the objectives. In the following section, the trajectory matching, Y-direction velocity fluctuation, link length norm, velocity ratio, and maximum absolute tool-to-egg offset

linkage properties are plotted as functions of the X-direction velocity fluctuation to examine their correlations with this primary objective and to extract meaningful insights into objective interactions and trade-off mechanisms that may inform and improve future optimization formulations.

To ensure that outliers or suboptimal configurations are excluded from the correlation graphs and to focus exclusively on the best X-direction velocity-fluctuation configurations, the constraint-based outlier filtering described in 3.5 is further refined. Specifically, the constraint on X-direction velocity fluctuation $f_2(\boldsymbol{x}_{k,i}^*)$ is restricted to less than 2.5 mm/s, as shown in (54), and the constraint on the link length norm $f_4(\boldsymbol{x}_{k,i}^*)$ is limited to 0.9 m, as shown in (55). Here, k denotes the number of simultaneously optimized objective functions for each case, and i indexes the Pareto-optimal solutions.

$$f_2(\boldsymbol{x}_{k,i}^*) < L_2 = 2.5\ [mm/s] \tag{54}$$

$$f_4(\boldsymbol{x}_{k,i}^*) < L_4 = 0.9\ [m] \tag{55}$$

To quantify the correlation between the linkage properties and the X-direction velocity fluctuation, linear regression models are fitted to the data. The X-direction velocity fluctuation is treated as the independent variable X and the linkage properties are represented by the dependent variables Y. In the regression model, β_0 denotes the y-intercept and β_1 denotes the slope of the regression line, as shown in (56) [129].

$$Y = \beta_1 X + \beta_0 \tag{56}$$

The Pearson correlation coefficient R is used to measure the strength of each correlation. It is calculated by dividing the covariance of the dependent and independent data, $cov(X, Y)$, by the product of their standard deviations, σ_X and σ_Y [130].

$$R = \frac{cov(X, Y)}{\sigma_X \sigma_Y} \tag{57}$$

Figure 44 presents Cases 2 and 4's trajectory-matching objective as a function of the X-direction velocity fluctuation objective, filtered with the updated constraints. The linear regression model fit to the data has a slope of -0.020 $[\frac{\mathrm{mm}^{-1}}{\mathrm{s}}]$ and a regression coefficient of -0.595. The correlation

coefficient and the spread of the data indicate that the trajectory-matching objective is moderately negatively correlated with the X-direction velocity fluctuation objective.

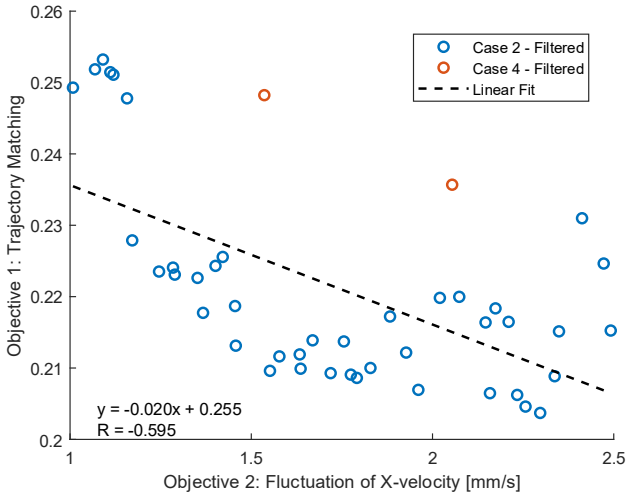

Figure 44. Trajectory matching objective value as a function of X-direction velocity fluctuation following secondary filtering

Figure 45 shows the Y-direction velocity fluctuation objective as a function of X-direction velocity fluctuation for the filtered data from Cases 2 and 4. The linear regression model fit to the data has a slope of -3.168 $[\frac{\text{mm}}{\text{s}}\frac{\text{mm}^{-1}}{\text{s}}]$ and a regression coefficient of -0.889. This indicates that minimizing the tool's X- and Y-direction velocity fluctuations in the lower portion of this path shape are closely negatively correlated objectives.

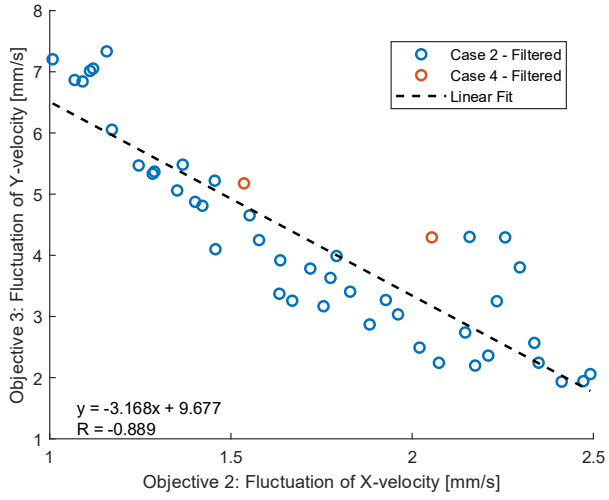

Figure 45. Y-direction velocity fluctuation as a function of X-direction velocity fluctuation following secondary filtering

Figure 46 shows the link length norm objective as a function of the X-direction velocity fluctuation for the filtered data from Cases 2 and 4. The linear regression model fitted to the data has a slope

of -0.051 $[m\frac{\mathrm{mm}^{-1}}{\mathrm{s}}]$ and a regression coefficient of -0.586. This graph and correlation coefficient show a moderate negative correlation between link length norm and X-direction velocity fluctuation.

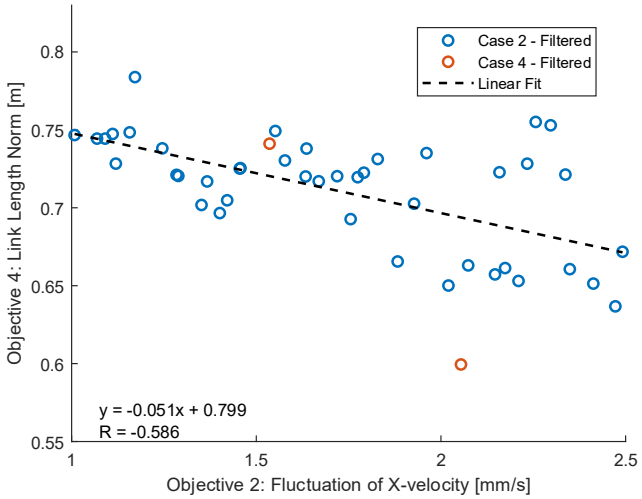

Figure 46. Link length norm as a function of X-direction velocity fluctuation following secondary filtering

Figure 47 shows the velocity ratio as a function of X-direction velocity fluctuation for the filtered data. The linear regression model fitted to the data has a slope of 0.209 $[\frac{\mathrm{mm}^{-1}}{\mathrm{s}}]$ and a regression coefficient of 0.905. This is the highest correlation coefficient across the five studies and indicates a strong correlation. However, this correlation is not helpful for optimization, as the goal is to minimize X-direction velocity fluctuations and maximize the velocity ratio.

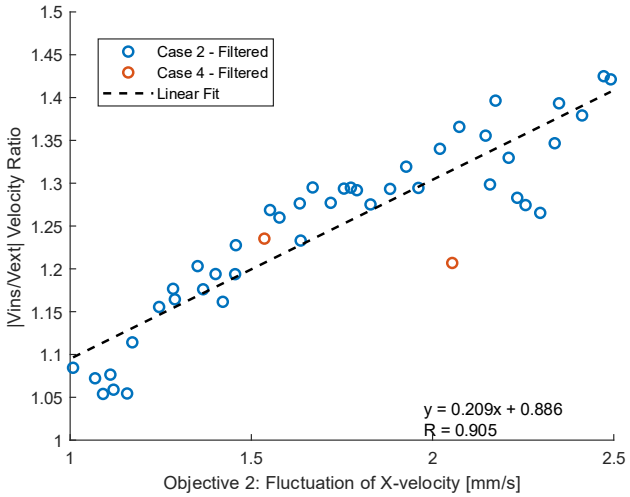

Figure 47. Velocity ratio as a function of X-direction velocity fluctuation following secondary filtering

Figure 48 shows the maximum absolute offset between the tool and egg during extraction as a function of X-direction velocity fluctuation. The linear regression model fitted to the data has a slope of 0.954 $[mm\frac{\mathrm{mm}^{-1}}{\mathrm{s}}]$ and a regression coefficient of 0.867. These two quantities exhibit a strong positive correlation, supporting the conclusion that minimizing X-direction velocity fluctuation also improves the tool-to-egg offset.

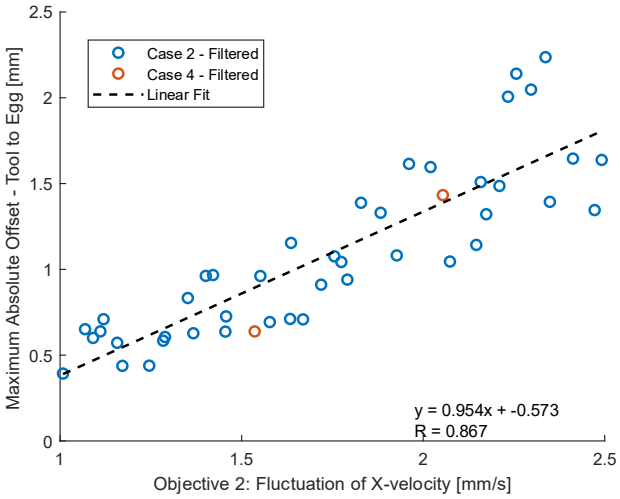

Figure 48. Absolute tool-to-egg offset as a function of X-direction velocity fluctuation following secondary filtering

Table 7 reports the Pearson correlation coefficients R for the relationships shown above, with correlation strengths classified according to the guidelines presented in [129]. In addition, the table identifies both the desired and observed directions of R and the corresponding regression slope β_1 for each linkage-property correlation.

The desired direction represents the correlation orientation that renders the objectives compatible with minimizing X-direction velocity fluctuations, thereby enabling parallel optimization. The observed direction corresponds to the actual polarity of the estimated R and β_1. Agreement between the desired and observed directions, therefore, indicates objective compatibility, whereas disagreement implies a conflicting trade-off.

For example, both trajectory matching error and X-direction velocity fluctuation are objectives to be minimized; consequently, a positive correlation between these quantities would be desirable. However, the observed correlation is negative, indicating that reducing X-direction velocity fluctuations increases trajectory matching error. As a result, the desired and observed directions are inconsistent, and the objectives are conflicting.

In contrast, the maximum absolute tool-to-egg offset is ideally expected to decrease as X-direction velocity fluctuation decreases. Since this behaviour is also observed in the data, the desired and observed directions agree, indicating that these objectives are compatible. One can see that four of the five linkage parameters are negatively correlated with the objective of minimizing the tool's X-direction velocity fluctuation within the egg.

Table 7. Correlation analysis of kinematic properties as a function of X-direction velocity fluctuation

Linkage property	***R***	***Correlation strength***	***Desired direction of R and β_1***	***Observed direction of R and β_1***	***Desired-actual sign agreement***
Trajectory matching	-0.595	Moderate	Positive	Negative	Disagree
Y-direction velocity fluctuation	-0.889	High	Positive	Negative	Disagree
Link length norm	-0.586	Moderate	Positive	Negative	Disagree
Velocity ratio	0.905	Very high	Negative	Positive	Disagree
Maximum absolute offset	0.867	High	Positive	Positive	Agree

5.3.1 Sensitivity Analysis

While the correlation coefficient characterizes the direction and strength of association between correlated data, it does not quantify the magnitude of the trade-off or the sensitivity of the dependent variable to changes in the independent variable. However, the sensitivity of correlated data addresses this limitation by quantifying the relative magnitude of change in the dependent variable in response to variations in the independent variable. Thus, in terms of multi-objective optimization, high sensitivity and a correlation direction that disagrees with the desired correlation direction result in severe trade-offs, indicating that the objectives are strongly conflicting. Conversely, high sensitivity and positive correlation enable the objectives to be improved in parallel and are therefore strongly compatible objectives.

The slopes of the linear regression models will be used to compare the sensitivity of different linkage properties to the correlation between X-direction velocity fluctuations for the best X-direction velocity-fluctuation configurations. These slopes must first be normalized to have the same units, be independent of the magnitude of the dependent variable, and isolate the relative rate of change of each linkage property with respect to X-direction velocity fluctuation. This will be done by normalizing the linear fit slopes β_1 by the mean of each set of dependent data μ_Y. This will transform the slopes into mean-normalized slopes β_{1,μ_Y-norm} with units of $[\frac{1}{X\ unit}]$ as shown in (58).

$$\beta_{1,\mu_Y-norm}\left[\frac{1}{X\ unit}\right] = \frac{\beta_1\ \left[\frac{Y\ unit}{X\ unit}\right]}{\mu_Y[Y\ unit]} \tag{58}$$

To further enable the direct comparison across linkage parameters, the absolute Z-score of the mean normalized slopes is computed, as shown in (59). The Z-score quantifies how many standard deviations a value lies from the population mean, and taking its absolute value removes dependence on the sign of the slope. Consequently, the absolute Z-score of the mean-normalized slopes provides a scalar, sign-independent measure of sensitivity, enabling direct comparison of how strongly each linkage property responds to changes in X-direction velocity fluctuation. The results of this sensitivity analysis are summarized in Table 8.

$$Z_{Y,abs} = \left|\frac{\beta_{1,\mu_Y-norm} - \mu_{\beta 1,\mu_Y-norm}}{\sigma_{\beta 1,\mu_Y-norm}}\right| \tag{59}$$

Table 8. Sensitivity analysis of kinematic properties as a function of X-direction velocity fluctuation

Linkage property	β_1	*Y-axis mean*	μ_Y*-norm* β_1 $[\frac{mm^{-1}}{s}]$	*Absolute* μ_Y*-norm* β_1 *Z-score [-]*	*Z-Score sensitivity rank*
Trajectory matching	-0.020 $[\frac{\mathrm{mm}^{-1}}{\mathrm{s}}]$	0.221 [-]	-0.088	0.195	4th
Y-direction velocity fluctuation	-3.168 $[\frac{\mathrm{mm}}{\mathrm{s}}\frac{\mathrm{mm}^{-1}}{\mathrm{s}}]$	4.190 $[\frac{\mathrm{mm}}{\mathrm{s}}]$	-0.756	1.328	2nd
Link length norm	-0.051 $[m\frac{\mathrm{mm}^{-1}}{\mathrm{s}}]$	0.710 $[m]$	-0.072	0.168	5th
Velocity ratio	0.209 $[\frac{\mathrm{mm}^{-1}}{\mathrm{s}}]$	1.248 [-]	0.168	0.239	3rd
Maximum absolute offset	0.954 $[mm\frac{\mathrm{mm}^{-1}}{\mathrm{s}}]$	1.080 $[mm]$	0.884	1.453	1st

The sensitivity analysis reveals that the maximum absolute tool-to-egg offset is the most sensitive linkage property, with an absolute mean-normalized z-score of 1.453. This result is physically intuitive, as the maximum absolute tool-to-egg offset represents the time integral of instantaneous velocity fluctuation between the tool and the egg. The combination of high sensitivity, a strong positive correlation, and a desired positive correlation indicates that these linkage parameters are compatible and can be optimized in parallel.

The second most sensitive linkage property is the Y-direction velocity fluctuation with an absolute mean-normalized z-score of 1.328. In conjunction with its strong negative correlation with X-direction velocity fluctuations, a desired positive correlation indicates that minimizing velocity fluctuations of the tool in both Cartesian directions while it is in the egg constitutes a highly conflicting objective pair. This objective pair was present in Cases 2, 3, and 4.

The remaining three linkage properties exhibit sensitivity that is approximately one order of magnitude lower than that of the two most sensitive linkage parameters. They are also negatively correlated with X-direction velocity fluctuation.

In particular, the velocity ratio exhibits a strong positive correlation with maximizing X-direction velocity fluctuation but low sensitivity. However, because the desired correlation direction is negative, these objectives are conflicting. The low sensitivity suggests that, while the velocity ratio closely tracks changes in X-direction velocity fluctuations, it does so with a shallow gradient. This suggests a shallow trade-off between high velocity ratios and ultra-low X-direction velocity fluctuations.

Additionally, the trajectory-matching objective and the link length norm exhibit low sensitivity and a moderate negative correlation with X-direction velocity fluctuation. Because a positive correlation is desired, the two objectives are therefore in conflict with respect to X-direction velocity fluctuation. The moderate correlation implies that the objectives are only partially coupled, resulting in greater variability in the spread of data. Meanwhile, the low sensitivity indicates that reductions in X-direction velocity fluctuation led to relatively small degradation in trajectory matching and link length norm. Consequently, the trade-offs are limited but still present. This suggests that the desired trajectory and short link lengths are not well-suited for ultra-low X-direction velocity fluctuations of the tool within the egg.

Overall, this sensitivity analysis demonstrates that Cases 2, 3, and 4 of this study contain at least one pair of highly competitive optimization objectives, which explains the relative sparsity of their Pareto fronts compared to Case 1. This analysis also highlights that objectives with negative correlations do not impose equivalent trade-offs in multi-objective optimization, and that sensitivity analysis is valuable for improving understanding of the relationships among optimization objectives.

5.3.2 Path Symmetry Analysis

Furthermore, it was observed that Case 2 exhibited both the lowest X-direction velocity fluctuation and the most vertically symmetrical trajectory among the three optimal configurations analyzed in section 5.2.1. Therefore, the two configurations A and B with the next lowest X-direction velocity fluctuation were selected from Figure 49 and compared in Figure 50 to determine whether they also exhibit a vertically symmetric trajectory.

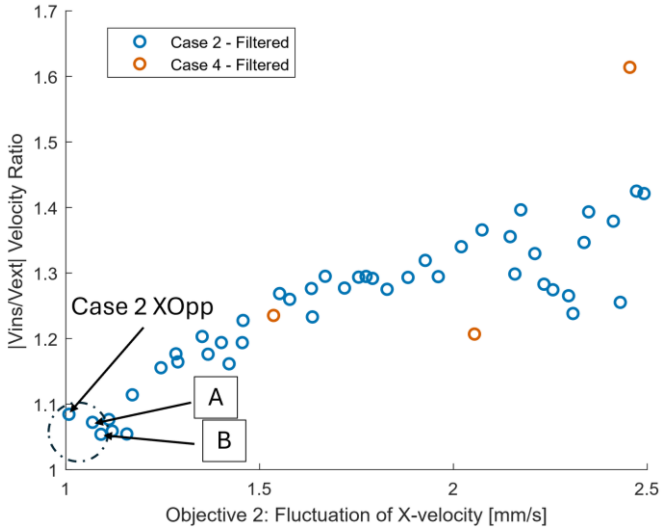

Figure 49. Velocity ratio as a function of X-direction velocity fluctuation following secondary filtering, highlighting the three configurations with the lowest X-direction velocity fluctuation

As shown in Figure 50, Case 2's optimal and A and B from Figure 49 have the lowest X-direction velocity fluctuation values of all the configurations, at 1.084 mm/s, 1.054 mm/s, and 1.072 mm/s, respectively. Furthermore, all three exhibit strong vertical symmetry in their trajectories. This indicates that vertical path symmetry is positively correlated with reduced X-direction velocity fluctuation in the lower portion of the coupler curve. This also supports the conclusion that this thesis's desired trajectory shape is not well-suited for ultra-low X-direction velocity fluctuations of the tool within the egg.

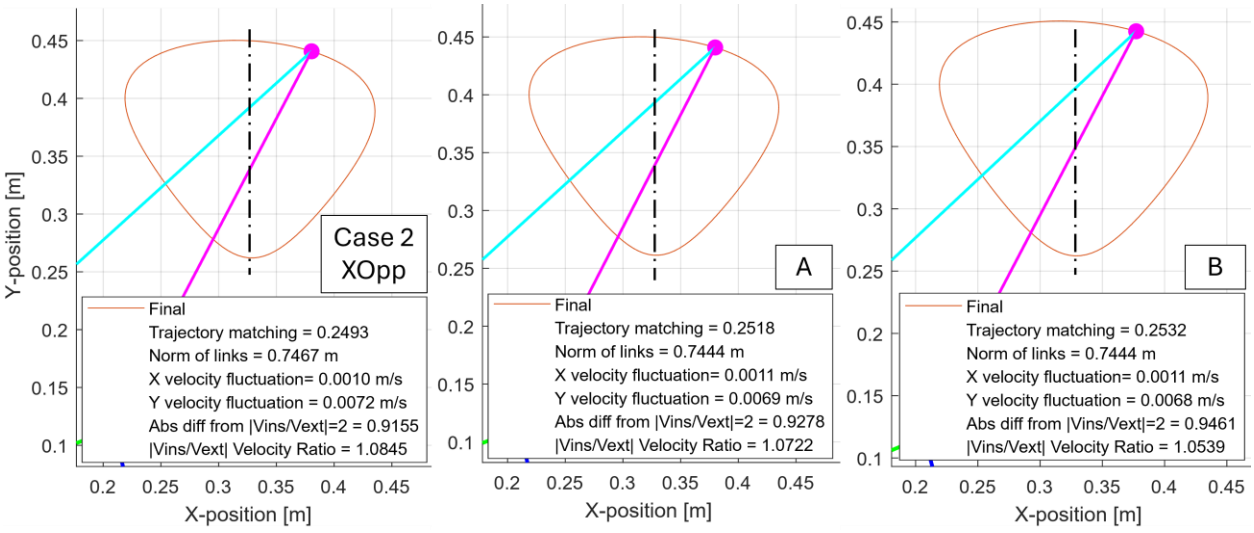

Figure 50. Lowest X-direction velocity fluctuation configuration path symmetry comparison

5.4 Future Work – Reinforcement Learning

Future work on the RL multi-start framework used in this thesis should focus on simplifying the NN architecture to prevent overfitting while still utilizing the multi-start formulation. Multi-start was employed by the continuous TD3 and DDPG agents, which outperformed their degenerate counterparts. However, the covariance matrix action structure was used by both, and the use was influenced by [113] and [114], but may have been a source of excessive stochasticity, preventing consistent convergence to the global optimal. The usefulness of the covariance-matrix action structure, coupled with TD3 and DDPG, should thus be evaluated.

Eliminating the covariance matrix in the actor allows evaluation of a simpler, single-network architecture with reduced computational cost. Other continuous-action RL agents with stochastic action structures, such as Proximal Policy Optimization and Soft Actor-Critic, could also be considered.

The discretization of the problem's design space using tile coding or another discretization method, combined with a straightforward discrete-environment RL agent such as Deep Q-Learning, should also be evaluated. Ideally, this would help mitigate the curse of dimensionality and allow the agent to better avoid local minima. This approach could then be integrated with a local optimization solver, as demonstrated in this thesis using TD3 and DDPG with fmincon. Such a combination would aid in fine-tuning the results and capturing the subtleties lost through the discretization of the design space.

5.5 Future Work – Optimization and Implementation

The correlation between the linkage parameter and the X-direction velocity fluctuations, based on the best X-direction velocity fluctuation configurations identified, points to several promising guidelines for future multi-objective optimisation studies of four-bar linkages designed for motion-based operations on objects moving at a constant velocity. Notably, Y-direction velocity fluctuation was found to be the most competitive objective, among the analyzed design objectives, relative to X-direction velocity fluctuation. This provides strong evidence that Y-direction velocity fluctuation should be omitted as an explicit optimisation objective in future studies to reduce the overall competitiveness of the trade-off space. Instead, it should be treated as a constraint, with a maximum permissible value determined empirically and enforced through an inequality constraint.

Trajectory matching, link length norm, and velocity ratio were all observed to be less sensitive than the Y-direction velocity fluctuation objective. Consequently, a new optimization study should be conducted using X-direction velocity fluctuation alongside these three objectives, while constraining Y-direction velocity fluctuation to a prescribed maximum value, to assess whether improved solutions can be obtained relative to those reported in this study. Furthermore, because symmetric coupler paths exhibited the most constant X-direction velocity, future desired trajectories may be redesigned to explicitly promote vertical-axis symmetry.

Although the results provide clear guidance for future optimization studies, additional investigation is required to resolve two outstanding objectives. Specifically, Y-direction velocity fluctuation must be characterized empirically to determine a maximum permissible value suitable for use as a constraint. At the same time, the role of link length norm in influencing dynamic performance must be evaluated through dynamic simulation and optimization to determine how reductions in link length affect overall system behaviour.

In this thesis, it was assumed that large Y-direction velocity fluctuations during extraction are associated with increased membrane tension and an elevated risk of tearing. This assumption motivated the inclusion of Y-direction velocity fluctuation as an optimization objective. However, this relationship was not validated experimentally. An essential next step is therefore to empirically evaluate the influence of different velocity profiles on membrane extraction, either to validate the assumptions made in this work or by identifying a maximum permissible Y-direction velocity fluctuation. A testing apparatus designed for this purpose is shown in Figure 51. It consists of a

stepper-motor-driven vertical winch with linear bushings that guide a rod connected to the toolhead.

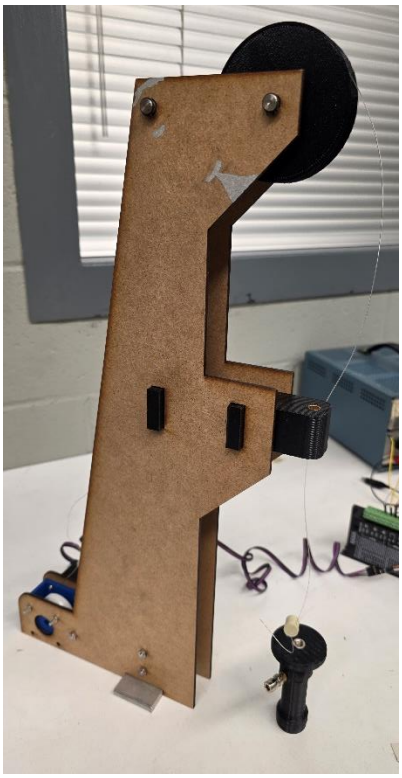

Figure 51. Velocity profile testing apparatus

Furthermore, it was assumed that minimizing link lengths would positively affect the dynamics of the linkage. As a next step, these assumptions should be thoroughly assessed by incorporating minimization of crank-arm input torque fluctuations as an optimization objective. The relationship between link length norm and input torque can then be examined. Using maximum and average input torque as optimization objectives would also reduce the required input motor size and eliminate alternating torque imposed on the motor by the mechanism, a phenomenon observed in the prototype and mitigated post-design with counterweights. This optimization would also improve the design's real-world feasibility, as the dynamics are critical to the machine's operation. The machine's input torque, shaking force, and shaking moment could also be minimized by incorporating counterweights into the design and optimizing them further to improve its dynamics and performance.

Throughout this research, crankarm velocity is constant and an open-loop control scheme is used to match the operational characteristics of the current egg-braking machines on the market.

However, since this machine has well-defined kinematic and dynamic models and the X-direction velocity fluctuation objective is imperative to reliability, the feasibility and effectiveness of a closed loop control system for improving egg speed matching and velocity fluctuation should be evaluated in future research. The dynamic and vibrational effects of varying crankarm angular velocity should also be evaluated to prevent any unwanted end-effector displacement inside the egg.

Furthermore, since the functions used to construct the velocity fluctuation and velocity ratio objectives may introduce discontinuities or singularities, alternative formulations warrant consideration. One such alternative is replacing the absolute value function by the quadratic function, given its smoothness and full differentiability.

Additional multi-objective optimization structures and algorithms should be explored and compared to the MOGA used in this study. Specifically, a hierarchical structure could be implemented since the X-direction velocity fluctuation objective is prioritized for its impact on system reliability. This structure would allow for a Pareto front to be built, as done in the MOGA, whilst also controlling the priority of the objectives, as done in a weighted sum global optimization. Furthermore, multi-objective versions of GS and CMA-ES would be good candidates for further research. The addition of cognate solutions and the open and crossed assembly modes should also be incorporated into future optimization algorithms to expand further the range of four-bar linkages considered for motion-based operations on objects moving at constant velocity.

Furthermore, PS should be investigated with a smaller mesh tolerance for the path-generation optimization, as the algorithm may have terminated due to the mesh tolerance criterion before the objective function converged. Reducing the mesh tolerance may allow PS to reach the same optimal value as GS or potentially achieve improved solutions.

It was mentioned that the metaheuristic and gradient based single-objective global optimizers can only produced results that are considered a strong candidate for the global optimum rather than a certified one. Therefore, evaluating deterministic global optimization algorithms that are proven to produce a global optimum such as Branch and Bound [131] algorithm tailored to the non-linear constrained formulation would be a good next step to validate the results shown in this thesis.

Finally, following a revised formulation of the kinematic optimization, once a maximum permissible Y-direction velocity fluctuation value is determined and an optimization study of the system's dynamics is performed, it is essential to build a physical prototype of the new optimal linkage to validate the reliability and efficiency improvements. It would also be beneficial to design a custom pinion-and-gear system to mechanically synchronize the linkage with the conveyor, mimicking the systems used in industrial egg-breaking machines. Furthermore, increasing the efficiency of the linkage prototype by adding multiple toolheads to the same end-effector could enable simultaneous operation on multiple eggs, allow extraction of more than one membrane per cycle, and significantly increase the extraction rate.

References

[1] Norton RL. Design of machinery: an introduction to the synthesis and analysis of mechanisms and machines. Choice Reviews Online 1992;30:76–140. https://doi.org/10.5860/CHOICE.30-1544.

[2] Kelly VRP, Lanteigne E, Hincke M. Recent Patents and Commercial Products Reflecting the Biomedical Applications of Eggshell Membrane. Eggshell for Biomedical Applications, Elsevier; n.d.

[3] Lanteigne E, Hincke MTC, Ahmed TAE, Insull PJ, Kelly VRP. System and Method for Extracting Eggshell Membrane (ESM). WO2024250114 (A1), 2024.

[4] Ahmed TAE, Suso H-P, Hincke MT. In-Depth Comparative Analysis of the Chicken Eggshell Membrane Proteome. Journal of Proteomics 2017;155:49–62. https://doi.org/10.1016/j.jprot.2017.01.002.

[5] Premiada una investigación de la UGR y el CSIC sobre las propiedades beneficiosas de la membrana de la cáscara de huevo como material para la regeneración ósea. Canal UGR n.d. https://canal.ugr.es/noticia/premiada-investigacion-propiedades-beneficiosas-de-la-membrana-de-la-cascara-de-huevo-como-material-para-la-regeneracion-osea/ (accessed November 19, 2025).

[6] Production | Gateway to poultry production and products. Food and Agriculture Organization of the United Nations n.d. https://www.fao.org/poultry-production-products/production/en/ (accessed December 2, 2024).

[7] Global Egg Market Data and Industry Growth Analysis 2021. https://www.thebusinessresearchcompany.com/press-release/global-egg-market-2021 (accessed December 2, 2024).

[8] Kulshreshtha G, Diep T, Hudson H-A, Hincke MT. High Value Applications and Current Commercial Market for Eggshell Membranes and Derived Bioactives. Food Chemistry 2022;382:132270. https://doi.org/10.1016/j.foodchem.2022.132270.

[9] Vlad V. Eggshell Membrane Separation Method. WO2006078824A2, 2006.

[10] Utgård B, Amundsen S, Suso H-P. Method of Processing Eggshell Residues. WO2015058790 (A1), 2015.

[11] Adams RG, Franklin MR. Vacuum Treatment of an Input Stream Without Ruining Delicate Output Fractions. US7017277 (B1), 2006.

[12] Monferrer BA, Cubero EN. Method to Separate the Internal Membrane of the Egg Shell. (machine-translation by Google Translate, Not Legally Binding). ES2327087 (A1), 2009.

[13] Hasebe Y. Manufacturing Method of Shell Membrane Lot and Manufacturing Method of Egg Related Product Group. JP2020111543 (A), 2020.

[14] Fujimura H. Taking-out of Egg Shell Film and Device Therefor. JPH0345264 (A), 1991.

[15] Miyazaki N, Mitani T, Torisaki A, Inoue M. Method and Device for Separating Membrane from Egg Shell. JPH08173838 (A), 1996.

[16] Kato M. Method of Treating Egg. JP2002263629 (A), 2002.

[17] Angeles J, Bai S. MECH541 Kinematic Synthesis 2016.

[18] Rotzoll M. Algebraic Input-Output Equations of Four-Bar Kinematic Chains: Planar; Spherical; Spatial. Doctor of Philosophy. Carleton University, 2023. https://doi.org/10.22215/etd/2023-15534.
[19] Freudenstein F. Design of Four-Link Mechanisms. Ph.D. dissertation. Columbia University, 1954.
[20] Freudenstein F. Approximate Synthesis of Four-Bar Linkages 2010.
[21] Copeland ZA. Continuous Approximate Kinematic Synthesis of Planar, Spherical, and Spatial Four-bar Function Generating Mechanisms. Doctor of Philosophy. Carleton University, 2024. https://doi.org/10.22215/etd/2024-15994.
[22] Sutherland GH, Siddall JN. Dimensional synthesis of linkages by multifactor optimization. Mechanism and Machine Theory 1974;9:81–95. https://doi.org/10.1016/0094-114X(74)90009-3.
[23] Liu Z, Angeles J. Least-Square Optimization of Planar and Spherical Four-Bar Function Generator Under Mobility Constraints. Journal of Mechanical Design 1992;114:569–73. https://doi.org/10.1115/1.2917045.
[24] Gupta KC. Design of Four-Bar Function Generators with Mini-Max Transmission Angle. Journal of Engineering for Industry 1977;99:360–5. https://doi.org/10.1115/1.3439231.
[25] Barker CR, Shu G-H. Three-Position Function Generation of Planar Four-Bar Mechanisms with Equal Deviation Transmission Angle Control. Journal of Mechanisms, Transmissions, and Automation in Design 1988;110:435–9. https://doi.org/10.1115/1.3258941.
[26] Chen F-C, Huang H-H. Application of Taguchi Method on the Tolerance Design of a Four-Bar Function Generation Mechanism. Volume 7: 29th Mechanisms and Robotics Conference, Parts A and B, Long Beach, California, USA: ASMEDC; 2005, p. 727–33. https://doi.org/10.1115/DETC2005-84199.
[27] Guj G, Dong ZY, Di Giacinto M. Dimensional synthesis of four bar linkage for function generation with velocity and acceleration constraints. Meccanica 1981;16:210–9. https://doi.org/10.1007/BF02128323.
[28] Russell K, Shen Q. Expanded spatial four-link motion and path generation with order and branch defect elimination. Inverse Problems in Science and Engineering 2013;21:129–40. https://doi.org/10.1080/17415977.2012.672418.
[29] Stoilov Todorov T. Synthesis of Four Bar Mechanisms as Function Generators by Freudenstein - Chebyshev. J Robot Mech Eng Resr 2015;1:1–6. https://doi.org/10.24218/jrmer.2015.01.
[30] Mehar K, Singh S, Mehar R. Optimal synthesis of four-bar mechanism for function generation with five accuracy points. Inverse Problems in Science and Engineering 2015;23:1222–36. https://doi.org/10.1080/17415977.2014.993982.
[31] Rose RS, Sandor GN. Direct Analytic Synthesis of Four-Bar Function Generators with Optimal Structural Error. Journal of Engineering for Industry 1973;95:563–71. https://doi.org/10.1115/1.3438191.
[32] Jaiswal A, Jawale HP. Comparative study of four-bar hyperbolic function generation mechanism with four and five accuracy points. Arch Appl Mech 2017;87:2037–54. https://doi.org/10.1007/s00419-017-1310-5.
[33] Input-Output Equation for Planar Four-Bar Linkages. Springer Proceedings in Advanced Robotics, Cham: Springer International Publishing; 2019, p. 12–9. https://doi.org/10.1007/978-3-319-93188-3_2.

[34] Rotzoll M, Hayes MJD, Husty ML. An algebraic input–output equation for planar RRRP and PRRP linkages. Trans Can Soc Mech Eng 2020;44:520–9. https://doi.org/10.1139/tcsme-2019-0156.
[35] Hayes MJD, Rotzoll M, Bucciol Q, Copeland ZA. Planar and spherical four-bar linkage vi−vj algebraic input–output equations. Mechanism and Machine Theory 2023;182:105222. https://doi.org/10.1016/j.mechmachtheory.2022.105222.
[36] Burmester LEH. Lehrbuch der Kinematik: Für Studirende der Maschinen-Technik, Mathematik und Physik Geometrisch Dargestellt. Leipzig, Germany: A. Felix; 1888.
[37] Chen C, Bai S, Angeles J. A Comprehensive Solution of the Classic Burmester Problem. Transactions of the Canadian Society for Mechanical Engineering 2008;32:137–54. https://doi.org/10.1139/tcsme-2008-0010.
[38] Advanced mechanism design: Analysis and Synthesis n.d.
[39] Sandor GN, Erdman AG. Advanced Mechanism Design: Analysis and Synthesis. vol. 2. Englewood Cliffs, NJ: Prentice-Hall; 1984.
[40] Angeles J, Bai S. Some Special Cases of the Burmester Problem for Four and Five Poses. Volume 7: 29th Mechanisms and Robotics Conference, Parts A and B, Long Beach, California, USA: ASMEDC; 2005, p. 307–14. https://doi.org/10.1115/DETC2005-84871.
[41] Hayes MJD, Zsombor-Murray PJ. Solving the Burmester Problem Using Kinematic Mapping. Volume 5: 27th Biennial Mechanisms and Robotics Conference, Montreal, Quebec, Canada: American Society of Mechanical Engineers; 2002, p. 1439–46. https://doi.org/10.1115/DETC2002/MECH-34378.
[42] Schröcker H-P, Husty ML, McCarthy JM. Kinematic Mapping Based Assembly Mode Evaluation of Planar Four-Bar Mechanisms. Journal of Mechanical Design 2007;129:924–9. https://doi.org/10.1115/1.2747635.
[43] Copeland ZA, Hayes MJD. Coupler Pose Curves for Planar 4R Mechanisms 2020.
[44] Deshpande S, Purwar A. A Task-Driven Approach to Optimal Synthesis of Planar Four-Bar Linkages for Extended Burmester Problem. Journal of Mechanisms and Robotics 2017;9:061005. https://doi.org/10.1115/1.4037801.
[45] McCarthy JM, Soh GS. Analysis of Planar Linkages. Geometric Design of Linkages, vol. 11, New York, NY: Springer New York; 2011, p. 15–53. https://doi.org/10.1007/978-1-4419-7892-9_2.
[46] Kempe AB. On a General Method of describing Plane Curves of the n^{th} degree by Linkwork. Proceedings of the London Mathematical Society 1875;s1-7:213–6. https://doi.org/10.1112/plms/s1-7.1.213.
[47] Verstraten E. Cognate Linkages the Roberts – Chebyshev Theorem n.d.
[48] Wu R, Li R, Bai S. A fully analytical method for coupler-curve synthesis of planar four-bar linkages. Mechanism and Machine Theory 2021;155:104070. https://doi.org/10.1016/j.mechmachtheory.2020.104070.
[49] Sandor GM. A General Complex-Number Method for Plane Kinematic Synthesis with Applications n.d.
[50] Morgan AP, Wampler CW. Solving a Planar Four-Bar Design Problem Using Continuation. Journal of Mechanical Design 1990;112:544–50. https://doi.org/10.1115/1.2912644.
[51] Subbian T, Flugrad DR. Four-Bar Path Generation Synthesis by a Continuation Method. Journal of Mechanical Design 1991;113:63–9. https://doi.org/10.1115/1.2912752.

[52] Li X, Wei S, Liao Q, Zhang Y. A novel analytical method for four-bar path generation synthesis based on Fourier series. Mechanism and Machine Theory 2020;144:103671. https://doi.org/10.1016/j.mechmachtheory.2019.103671.
[53] Qian Y, Zhong H, Wang T, Wang L. An Improved Fourier-Based Method for Path Generation of Planar Four-Bar Linkages without Prescribed Timing. Machines 2024;12:299. https://doi.org/10.3390/machines12050299.
[54] Bai S, Angeles J. Coupler-curve synthesis of four-bar linkages via a novel formulation. Mechanism and Machine Theory 2015;94:177–87. https://doi.org/10.1016/j.mechmachtheory.2015.08.010.
[55] Hayes MJD, Rotzoll M. The Algebraic Parametric Coupler Point Curve Equation n.d.
[56] Constrained Nonlinear Optimization Algorithms - MATLAB & Simulink n.d. https://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html (accessed November 7, 2025).
[57] Bakthavachalam N, Kimbrell JT. Optimum Synthesis of Path-Generating Four-Bar Mechanisms. Journal of Engineering for Industry 1975;97:314–21. https://doi.org/10.1115/1.3438556.
[58] Angeles J, Alivizatoss A, Akhras R. An unconstrained nonlinear least-square method of optimization of RRRR planar path generators. Mechanism and Machine Theory 1988;23:343–53. https://doi.org/10.1016/0094-114X(88)90048-1.
[59] Mariappan J, Krishnamurty S. A generalized exact gradient method for mechanism synthesis. Mechanism and Machine Theory 1996;31:413–21. https://doi.org/10.1016/0094-114X(95)00077-C.
[60] Sancibrian R, Viadero F, García P, Fernández A. Gradient-based optimization of path synthesis problems in planar mechanisms. Mechanism and Machine Theory 2004;39:839–56. https://doi.org/10.1016/j.mechmachtheory.2004.02.012.
[61] Sancibrian R, García P, Viadero F, Fernández A. A general procedure based on exact gradient determination in dimensional synthesis of planar mechanisms. Mechanism and Machine Theory 2006;41:212–29. https://doi.org/10.1016/j.mechmachtheory.2005.04.006.
[62] Kang Y-H, Lin J-W, You W-C. Comparative Study on the Synthesis of Path-Generating Four-Bar Linkages Using Metaheuristic Optimization Algorithms. Applied Sciences 2022;12:7368. https://doi.org/10.3390/app12157368.
[63] How GlobalSearch and MultiStart Work - MATLAB & Simulink n.d. https://www.mathworks.com/help/gads/how-globalsearch-and-multistart-work.html (accessed November 7, 2025).
[64] Conor AM, Douglas SS, Gilmartin MJ. The kinematic synthesis of path generating mechanisms using genetic algorithms 1995;8.
[65] Roston GP, Sturges RH. Genetic algorithm synthesis of four-bar mechanisms. AIEDAM 1996;10:371–90. https://doi.org/10.1017/S0890060400001700.
[66] Laribi MA, Mlika A, Romdhane L, Zeghloul S. A combined genetic algorithm–fuzzy logic method (GA–FL) in mechanisms synthesis. Mechanism and Machine Theory 2004;39:717–35. https://doi.org/10.1016/j.mechmachtheory.2004.02.004.
[67] Lin W-Y. A GA–DE hybrid evolutionary algorithm for path synthesis of four-bar linkage. Mechanism and Machine Theory 2010;45:1096–107. https://doi.org/10.1016/j.mechmachtheory.2010.03.011.
[68] Ahmadi B, Ahmadi B. Optimal synthesis of crank-rocker mechanisms with optimum transmission angle for desired stroke and time-ratio using genetic programming. Advances

in Mechanical Engineering 2022;14:168781322211312. https://doi.org/10.1177/16878132221131291.
[69] Storn R. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces n.d.
[70] Price KV, Lampinen JA, Storn RM. Differential evolution: a practical approach to global optimization. Berlin [u.a.]: Springer; 2005.
[71] Cabrera JA, Simon A, Prado M. Optimal synthesis of mechanisms with genetic algorithms. Mechanism and Machine Theory 2002;37:1165–77. https://doi.org/10.1016/S0094-114X(02)00051-4.
[72] Shiakolas PS, Koladiya D, Kebrle J. On the Optimum Synthesis of Four-bar Linkages Using Differential Evolution and the Geometric Centroid of Precision Positions. Inverse Problems in Engineering 2002;10:485–502. https://doi.org/10.1080/1068276021000054594.
[73] Cabrera JA, Ortiz A, Nadal F, Castillo JJ. An evolutionary algorithm for path synthesis of mechanisms. Mechanism and Machine Theory 2011;46:127–41. https://doi.org/10.1016/j.mechmachtheory.2010.10.003.
[74] Lin W, Hsiao K. A new differential evolution algorithm with a combined mutation strategy for optimum synthesis of path-generating four-bar mechanisms. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 2017;231:2690–705. https://doi.org/10.1177/0954406216638887.
[75] Sleesongsom S, Bureerat S. Optimal Synthesis of Four-Bar Linkage Path Generation through Evolutionary Computation with a Novel Constraint Handling Technique. Computational Intelligence and Neuroscience 2018;2018:1–16. https://doi.org/10.1155/2018/5462563.
[76] Pickard JK, Carretero JA, Merlet J-P. Appropriate synthesis of the four-bar linkage. Mechanism and Machine Theory 2020;153:103965. https://doi.org/10.1016/j.mechmachtheory.2020.103965.
[77] Kang Y-H, Lin J-W, You W-C. Comparative Study on the Synthesis of Path-Generating Four-Bar Linkages Using Metaheuristic Optimization Algorithms. Applied Sciences 2022;12:7368. https://doi.org/10.3390/app12157368.
[78] Acharyya SK, Mandal M. Performance of EAs for four-bar linkage synthesis. Mechanism and Machine Theory 2009;44:1784–94. https://doi.org/10.1016/j.mechmachtheory.2009.03.003.
[79] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan: IEEE; 1995, p. 39–43. https://doi.org/10.1109/MHS.1995.494215.
[80] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, Perth, WA, Australia: IEEE; 1995, p. 1942–8. https://doi.org/10.1109/ICNN.1995.488968.
[81] Kennedy J, Eberhart RC, Shi Y. Swarm intelligence. 8. [print.]. San Francisco: Morgan Kaufmann; 2009.
[82] Lee CK, Cheng YC. Significant Factors Identification for Particle Swarm Optimization Algorithm to Solving the Design Optimization Problem of a Four-Bar Linkage for Path Generation. AMM 2012;249–250:1180–7. https://doi.org/10.4028/www.scientific.net/AMM.249-250.1180.
[83] Lee C-T, Lee C-C. On a hybrid particle swarm optimization method and its application in mechanism design. Proceedings of the Institution of Mechanical Engineers, Part C: Journal

of Mechanical Engineering Science 2014;228:2844–57. https://doi.org/10.1177/0954406214522206.
[84] Singh R, Chaudhary H, Singh AK. A new hybrid teaching–learning particle swarm optimization algorithm for synthesis of linkages to generate path. Sādhanā 2017;42:1851–70. https://doi.org/10.1007/s12046-017-0737-2.
[85] Optimization by Simulated Annealing n.d.
[86] Burkard RE, Rendl F. A thermodynamically motivated simulation procedure for combinatorial optimization problems. European Journal of Operational Research 1984;17:169–74. https://doi.org/10.1016/0377-2217(84)90231-5.
[87] Smaili AA, Diab NA, Atallah NA. Optimum Synthesis of Mechanisms Using Tabu-Gradient Search Algorithm. Journal of Mechanical Design 2005;127:917–23. https://doi.org/10.1115/1.1904640.
[88] Martínez-Alfaro H. Four-bar Mechanism Synthesis for n Desired Path Points Using Simulated Annealing. In: Siarry P, Michalewicz Z, editors. Advances in Metaheuristics for Hard Optimization, Berlin, Heidelberg: Springer Berlin Heidelberg; 2007, p. 23–37. https://doi.org/10.1007/978-3-540-72960-0_2.
[89] Ebrahimi S, Payvandy P. Efficient constrained synthesis of path generating four-bar mechanisms based on the heuristic optimization algorithms. Mechanism and Machine Theory 2015;85:189–204. https://doi.org/10.1016/j.mechmachtheory.2014.11.021.
[90] Wu J, Ge QJ, Gao F, Guo WZ. On the Extension of a Fourier Descriptor Based Method for Planar Four-Bar Linkage Synthesis for Generation of Open and Closed Paths. Journal of Mechanisms and Robotics 2011;3:031002. https://doi.org/10.1115/1.4004227.
[91] Ullah I, Kota S. Optimal Synthesis of Mechanisms for Path Generation Using Fourier Descriptors and Global Search Methods. Journal of Mechanical Design 1997;119:504–10. https://doi.org/10.1115/1.2826396.
[92] McGarva J, Mullineux G. Harmonic representation of closed curves. Applied Mathematical Modelling 1993;17:213–8. https://doi.org/10.1016/0307-904X(93)90109-T.
[93] Liu Y, Xiao R. Optimal Synthesis of Mechanisms for Path Generation Using Refined Numerical Representation Based Model and AIS Based Searching Method. Journal of Mechanical Design 2005;127:688–91. https://doi.org/10.1115/1.1898342.
[94] Starosta R. Application of Genetic Algorithm and Fourier Coefficients (ga-Fc) in Mechanism Synthesis. Journal of Theoretical and Applied Mechanics 2008;46:395–411.
[95] Buśkiewicz J, Starosta R, Walczak T. On the application of the curve curvature in path synthesis. Mechanism and Machine Theory 2009;44:1223–39. https://doi.org/10.1016/j.mechmachtheory.2008.08.001.
[96] Sharma S, Purwar A, Jeffrey Ge Q. An Optimal Parametrization Scheme for Path Generation Using Fourier Descriptors for Four-Bar Mechanism Synthesis. Journal of Computing and Information Science in Engineering 2019;19:014501. https://doi.org/10.1115/1.4041566.
[97] Kim J-W, Seo T, Kim J. A new design methodology for four-bar linkage mechanisms based on derivations of coupler curve. Mechanism and Machine Theory 2016;100:138–54. https://doi.org/10.1016/j.mechmachtheory.2016.02.006.
[98] Kim J-W, Jeong S, Kim J, Seo T. Numerical hybrid Taguchi-random coordinate search algorithm for path synthesis. Mechanism and Machine Theory 2016;102:203–16. https://doi.org/10.1016/j.mechmachtheory.2016.04.001.
[99] Goulet V, Li W, Cheong H, Iorio F, Quimper C-G. Four-Bar Linkage Synthesis Using Non-Convex Optimization. In: Rueher M, editor. Principles and Practice of Constraint

Programming, vol. 9892, Cham: Springer International Publishing; 2016, p. 618–35. https://doi.org/10.1007/978-3-319-44953-1_39.
[100] Vasiliu A, Yannou B. Dimensional synthesis of planar mechanisms using neural networks: application to path generator linkages. Mechanism and Machine Theory 2001;36:299–310. https://doi.org/10.1016/s0094-114x(00)00037-9.
[101] Khan N, Ullah I, Al-Grafi M. Dimensional synthesis of mechanical linkages using artificial neural networks and Fourier descriptors. Mech Sci 2015;6:29–34. https://doi.org/10.5194/ms-6-29-2015.
[102] Li X, Chen P. A Parametrization-Invariant Fourier Approach to Planar Linkage Synthesis for Path Generation. Mathematical Problems in Engineering 2017;2017. https://doi.org/10.1155/2017/8458149.
[103] Hoskins JC, Kramer GA. Synthesis of mechanical linkages using artificial neural networks and optimization. IEEE International Conference on Neural Networks, San Francisco, CA, USA: IEEE; n.d., p. 822-J-822-N. https://doi.org/10.1109/icnn.1993.298663.
[104] Galán-Marín G, Alonso FJ, Del Castillo JM. Shape optimization for path synthesis of crank-rocker mechanisms using a wavelet-based neural network. Mechanism and Machine Theory 2009;44:1132–43. https://doi.org/10.1016/j.mechmachtheory.2008.09.006.
[105] Chen C-H. Application of Multiple Deep Neural Networks to Multi-Solution Synthesis of Linkage Mechanisms. Machines 2023;11:1018. https://doi.org/10.3390/machines11111018.
[106] Wang J, Chen B, Wang Y, Pu D, Jia X. Research on four-bar linkage trajectory synthesis using extreme gradient boosting and genetic algorithm. Journal of Computational Design and Engineering 2024;11:1–21. https://doi.org/10.1093/jcde/qwae015.
[107] Nobari AH, Srivastava A, Gutfreund D, Ahmed F. LINKS: A dataset of a hundred million planar linkage mechanisms for data-driven kinematic design 2022. https://doi.org/10.48550/arXiv.2208.14567.
[108] Nobari AH, Srivastava A, Gutfreund D, Xu K, Ahmed F. LInK: Learning Joint Representations of Design and Performance Spaces through Contrastive Learning for Mechanism Synthesis 2024. https://doi.org/10.48550/arXiv.2405.20592.
[109] Pan Z, Liu M, Gao X, Manocha D. Joint search of optimal topology and trajectory for planar linkages. The International Journal of Robotics Research 2023;42:176–95. https://doi.org/10.1177/02783649211069156.
[110] Nurizada A, Dhaipule R, Lyu Z, Purwar A. A Dataset of 3M Single-DOF Planar 4-, 6-, and 8-Bar Linkage Mechanisms with Open and Closed Coupler Curves for Machine Learning-Driven Path Synthesis. Journal of Mechanical Design 2025;147. https://doi.org/10.1115/1.4067014.
[111] Nurizada A, Lyu Z, Purwar A. Path Generative Model Based on Conditional β-Variational Auto Encoder for Four-Bar Mechanism Design. Journal of Mechanisms and Robotics 2025;17. https://doi.org/10.1115/1.4067169.
[112] Sutton RS, Barto AG. Reinforcement learning: an introduction. Second edition. Cambridge, Massachusetts: The MIT Press; 2018.
[113] Viquerat J, Rabault J, Kuhnle A, Ghraieb H, Larcher A, Hachem E. Direct shape optimization through deep reinforcement learning. Journal of Computational Physics 2021;428:110080. https://doi.org/10.1016/j.jcp.2020.110080.
[114] Gallego JA, Muñoz JM, Viquerat J, Aguirre ME. Design of Fourbar Linkages Using a Reinforcement Learning Optimization Method. Volume 7: 46th Mechanisms and Robotics

Conference (MR), St. Louis, Missouri, USA: American Society of Mechanical Engineers; 2022. https://doi.org/10.1115/detc2022-89953.
[115] Lipson H. Evolutionary synthesis of kinematic mechanisms. AIEDAM 2008;22:195–205. https://doi.org/10.1017/s0890060408000139.
[116] Vermeer K, Kuppens R, Herder J. Kinematic Synthesis Using Reinforcement Learning. Volume 2A: 44th Design Automation Conference, Quebec City, Quebec, Canada: American Society of Mechanical Engineers; 2018. https://doi.org/10.1115/detc2018-85529.
[117] Fogelson MB, Tucker C, Cagan J. GCP-HOLO: Generating High-Order Linkage Graphs for Path Synthesis. Journal of Mechanical Design 2023;145. https://doi.org/10.1115/1.4062147.
[118] Miettinen K. Nonlinear Multiobjective Optimization. New York, NY, United States: Springer; 1998.
[119] Nariman-Zadeh N, Felezi M, Jamali A, Ganji M. Pareto optimal synthesis of four-bar mechanisms for path generation. Mechanism and Machine Theory 2009;44:180–91. https://doi.org/10.1016/j.mechmachtheory.2008.02.006.
[120] Khorshidi M, Soheilypour M, Peyro M, Atai A, Shariat Panahi M. Optimal design of four-bar mechanisms using a hybrid multi-objective GA with adaptive local search. Mechanism and Machine Theory 2011;46:1453–65. https://doi.org/10.1016/j.mechmachtheory.2011.05.006.
[121] Farmani MR, Jaamialahmadi A, Babaie M. Multiobjective optimization for force and moment balance of a four-bar linkage using evolutionary algorithms. J Mech Sci Technol 2011;25:2971–7. https://doi.org/10.1007/s12206-011-0924-8.
[122] Etesami G, Felezi ME, Nariman-zadeh N. Pareto Optimal Balancing of Four-bar Mechanisms Using Multi-Objective Differential Evolution Algorithm. J Appl Comput Appl Mech 2020. https://doi.org/10.22059/jcamech.2020.290187.435.
[123] Farmani MR, Jaamiolahmadi A. Optimization of Force and Moment Balance of a Four-Bar Linkage Using Multi-Objective Genetic Algorithm. Volume 4: Design and Manufacturing, Lake Buena Vista, Florida, USA: ASMEDC; 2009, p. 259–68. https://doi.org/10.1115/imece2009-12740.
[124] Chaudhary K, Chaudhary H. Shape Optimization of Dynamically Balanced Planar Four-bar Mechanism. Procedia Computer Science 2015;57:519–26. https://doi.org/10.1016/j.procs.2015.07.378.
[125] Miler D, Birt D, Hoić M. Multi-Objective Optimization of the Chebyshev Lambda Mechanism. Sv-Jme 2023;68:725–34. https://doi.org/10.5545/sv-jme.2022.349.
[126] Uicker JJ, Pennock GR, Shigley JE. Theory of machines and mechanisms. Fifth edition. New York Oxford: Oxford University Press; 2017.
[127] BreakerPro 18: Ultimate Egg Breaking Machine | Highest Yield & Eco-Friendly Design n.d. https://www.sanovogroup.com/en/egg/solutions/egg-processing/breakerpro-18/ (accessed March 22, 2026).
[128] Yang X-S, He X-S, Fan Q-W. Mathematical framework for algorithm analysis. Nature-Inspired Computation and Swarm Intelligence, Elsevier; 2020, p. 89–108. https://doi.org/10.1016/B978-0-12-819714-1.00017-8.
[129] Hinkle DE, Wiersma W, Jurs SG. Applied statistics for the behavioral sciences. 5. Aufl. Boston: Houghton Mifflin Company; 2003.
[130] Carlton MA, Devore JL. Probability with Applications in Engineering, Science, and Technology. Cham: Springer International Publishing; 2017. https://doi.org/10.1007/978-3-319-52401-6.

[131] Land AH, Doig AG. An Automatic Method of Solving Discrete Programming Problems. Econometrica 1960;28:497. https://doi.org/10.2307/1910129.

Appendix

A. In-Depth Mathematical Derivation of the Fourier Harmonic Decomposition

This study's trajectory-matching objective function corresponds to the norm of the difference between the Fourier coefficients of the current ($\hat{\boldsymbol{c}}_{\boldsymbol{x}}$) and desired ($\hat{\boldsymbol{c}}_{desired}$) paths, as shown in (60). In essence, the smaller the norm of the difference between the coefficients, the greater the similarity between the two path shapes. The number of coefficients is a function of the number of harmonics j selected to best match the set of points (61). In this case, five harmonics best represent the desired path; therefore, there are 11 coefficients.

$$f_1 = \sqrt{\sum (\hat{\boldsymbol{c}}_{\boldsymbol{x}} - \hat{\boldsymbol{c}}_{desired})^2} \tag{60}$$

$$\hat{\boldsymbol{c}} = \left[\hat{c}_{-j}, \hat{c}_{-j-1}, \dots, \hat{c}_0, \dots, \hat{c}_{j-1}, \hat{c}_j\right]^T, \hat{\boldsymbol{c}} \in \mathbb{C}^{2j+1} \tag{61}$$

The Fourier coefficients are calculated using (62), where $\boldsymbol{z}_{\boldsymbol{k}}$ is the representation of the path points in the complex plane, N is the number of path points, and m is the coefficient index.

$$c_m = \frac{1}{N}\sum_{k=0}^{N-1} z_k e^{-\frac{2\pi i m k}{N}},\ m = [-j, \dots, 0, \dots, j] \tag{62}$$

$$\boldsymbol{z}_{\boldsymbol{k}} = \boldsymbol{x}_{\boldsymbol{O5},\boldsymbol{k}} + \boldsymbol{i}\boldsymbol{y}_{\boldsymbol{O5},\boldsymbol{k}} = \boldsymbol{x}_{\boldsymbol{desired},\boldsymbol{k}} + \boldsymbol{i}\boldsymbol{y}_{\boldsymbol{desired},\boldsymbol{k}} \tag{63}$$

This equation is derived by assuming that if z(t) satisfies Dirichlet's condition, t goes from 0 to 1, and m from $-\infty$ to ∞, the closed path z(t) can be represented by the doubly infinite series of Fourier coefficients in (64).

$$z(t) = \sum c_m\, e^{2\pi i m t}, m = [-\infty, \dots, 0, \dots, \infty] \tag{64}$$

Fourier coefficients themselves, which are complex numbers, are calculated using (65).

$$c_m = \int_{t=0}^{t=1} z(t)\, e^{-2\pi i m t} dt\,, m = [-\infty, \dots, 0, \dots, \infty] \tag{65}$$

Since the points are along a curve, one can compute the equation numerically with N points given as in (66).

$$z_0, z_1, z_2, \dots z_{N-1} \tag{66}$$

Since the curve is circular, z_0 is the same as z_{N-1}. It is then assumed that all the points are equally spaced regarding t. Thus, the integration step length is 1/N, and the trapezoidal rule can be used to derive the approximations for the Fourier coefficients used in this study, as shown in (62).

To normalize the Fourier coefficients such that they become independent of their pose and scale, two shifting factors α and β and the inverse of the magnitude of the first harmonic $|c_1|^{-1}$ are used to form (67). This gives the normalized coefficients $\hat{c}_m$ which are then used in the objective function. The fundamental coefficient, c_0, which represents the centroid of the set of points, is also set to 0 to center the normalized path about the origin.

$$\hat{c}_m = c_m\, e^{-i\alpha} e^{-im\beta} \frac{1}{|c_1|}, m = [-j, \dots, 0, \dots, j] \tag{67}$$

Now, to solve for the shifting factors α and β. First, the first coefficient is considered, also known as the first harmonics, which form the ellipse h_1 using (68).

$$h_1(t) = c_1\, e^{2\pi i t} + c_{-1}\, e^{2\pi i t} \tag{68}$$

Which can also be represented using (69).

$$h_1(t) = e^{i\alpha}[(r_1 + r_{-1})\cos(2\pi t + \beta) + \mathrm{i}(r_1 + r_{-1})\sin(2\pi t + \beta)] \tag{69}$$

Where r_1 and r_{-1} are the magnitudes of c_1 and c_{-1}, respectively, and thus the following equations can represent α and β:

$$\alpha = [\arg(c_1) + \arg(c_{-1})]/2 \tag{70}$$

$$\beta = [\arg(c_1) - \arg(c_{-1})]/2 \tag{71}$$

The coefficients can also be used to graph the closed Fourier harmonic curve that represents the set of points. This can be done by using (72) where p represents the number of harmonics that are part of the partial sum from the first to the p^{th} harmonic.

$$z_t = \sum_{k=0}^{N-1} c_m \, e^{2\pi i m t} \, , m = [-p, \dots, 0, \dots, p] \tag{72}$$

This can be seen in Figure 52 where the partial sums of the first to the fifth harmonics are graphed. One can see that the harmonics better approximate the set of points as the number of harmonics increases.

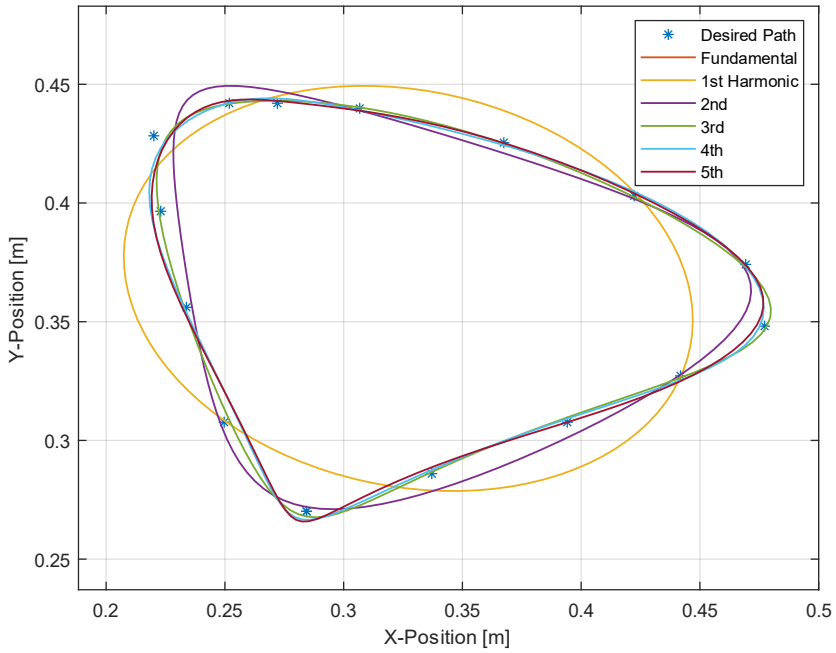

Figure 52. Plot of the desired path points and the closed curves formed using the path points' fundamental harmonic and first through fifth Fourier harmonics

B. Extra Tables

In Table 9 below, one can see the design parameters corresponding to this linkage, starting with the six primary path-shape design parameters from the optimization, followed by the rest of the full set of design parameters produced following the three-step linear transformation described above.

Table 9. Design parameters of the Baseline linkage and the trajectory-optimized linkage, all in meters [m]

	r_{12}	r_{23}	r_{34}	$d_{35x'}$	$d_{35y'}$	x_1	y_1	x_4	y_4
Baseline	0.121	0.239	0.239	0.254	1×10^{-4}	0	0	0.254	0
Trajectory	0.098	0.489	0.417	0.184	0.354	0.713	-0.275	0.991	0.472

The dominant result from the GS trajectory-matching optimization and the *Outlier*'s design parameters are presented in Table 10, and their resultant linkages are presented in Figure 27 and Figure 28, respectively. The norm of their link lengths is also displayed in the respective figure legends.

Table 10. Design parameters of the trajectory-optimized linkage and the algorithm's Outlier result, all in meters [m]

	r_{12}	r_{23}	r_{34}	$d_{35x'}$	$d_{35y'}$	x_1	y_1	x_4	y_4
Trajectory	0.098	0.489	0.417	0.184	0.354	0.713	-0.275	0.991	0.472
Outlier	0.114	0.277	0.505	0.101	0.215	-0.082	0.326	0.158	-0.279

All nine linkage design parameters of both the *Baseline* and Case 2, the overall optimal configuration, are shown in Table 11 below.

Table 11. Design parameters of the Baseline and the Case 2 optimal configuration, all in meters [m]

	r_{12}	r_{23}	r_{34}	$d_{35x'}$	$d_{35y'}$	x_1	y_1	x_4	y_4
Baseline	0.121	0.239	0.236	0.254	1×10^{-4}	0	0	0.254	0
Case 1.A	0.101	0.276	0.270	0.188	0.123	0.590	-0.011	0.677	0.422
Case 1.B	0.111	0.272	0.298	0.146	0.187	-0.102	0.287	0.174	-0.060
Case 2 XOpp	0.096	0.351	0.337	0.276	0.244	-0.215	-0.011	0.290	-0.213
Case 3 XOpp	0.096	0.227	0.293	0.227	0.108	-0.021	0.087	0.310	-0.072
Case 4 XOpp	0.093	0.393	0.2417	0.418	0.057	-0.260	-0.182	0.186	-0.121

C. Path Efficiency Improvement Calculation

The efficiency improvement calculation is shown in (73) to (79) below. In these equations L_{path} and $L_{path}{}'$ represent the *Baseline* and Case 2 path lengths, and their ratio is calculated in (73).

$$\frac{L_{path}{}'}{L_{path}} = \frac{0.6162\ m}{0.8647\ m} = 0.7125\ \therefore L_{path}{}' = 0.7125\ L_{path} \tag{73}$$

Δ_{eggs} and $\Delta_{eggs}{}'$ represent the ratio of eggs per meter of conveyor for the *Baseline* and Case 2 and is the inverse of the path length to ensure the continuous and synchronized operation of the machine. The increase in eggs per meter of conveyor from the *Baseline* to Case 2 is shown in (76).

$$\Delta_{eggs} = \frac{1}{L_{path}} = 1.1565\ \left[\frac{eggs}{m}\right] \tag{74}$$

$$\Delta_{eggs}{}' = \frac{1}{L_{path}{}'} = 1.6228\ \left[\frac{eggs}{m}\right] \tag{75}$$

$$\therefore \Delta_{eggs}{}' = 1.4033\Delta_{eggs} \tag{76}$$

v_{egg} and $v_{egg}{}'$ represent the conveyor velocity, which is set to $\bar{v}_{path}$ and $\bar{v}_{path}{}'$, the average horizontal velocity of the end effector within the egg, as shown in (77). This is done to ensure the tool is aligned within the egg.

$$\bar{v}_{path} = \bar{v}_{path}{}' = v_{egg} = v_{egg}{}' \left[\frac{m}{s}\right] \tag{77}$$

Finally, R_{egg} and $R_{egg}{}'$ are the rates of eggs processed per second. (78) and (79) show that, for a constant conveyor velocity across both configurations, the Case 2 configuration processes 40% more eggs per second because the eggs can be spaced closer together.

$$R_{egg}\left[\frac{eggs}{s}\right] = \Delta_{eggs}\ v_{egg} \tag{78}$$

$$\therefore R_{egg}{}' = \Delta_{eggs'} v_{egg} = 1.4035\ \Delta_{eggs} v_{egg}\ \left[\frac{eggs}{s}\right] \tag{79}$$