

University of Cincinnati

Date: 11/10/2014

I, Jeannette Geisler, hereby submit this original work as part of the requirements for the degree of Master of Science in Mechanical Engineering.

It is entitled:

The Localization of Free-Form Surfaces by the Maximization of the Shared Extent

Student's name: Jeannette Geisler

This work and its defense approved by:

Committee chair: Sundararaman Anand, Ph.D.

Committee member: Thomas Richard Huston, Ph.D.

Committee member: David Thompson, Ph.D.



12812

The Localization of Free-Form Surfaces by the Maximization of the Shared Extent

A thesis submitted to the
Graduate School
of the University of Cincinnati
in partial fulfillment of the
requirements for the degree of

Master of Science

November 2014

in the Department of Mechanical & Material Engineering
of the College of Engineering and Applied Sciences

by

Jeannette Geisler

BS Rose-Hulman Institute of Technology

Committee Chair: Dr. Sam Anand

UMI Number: 1585293

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 1585293

Published by ProQuest LLC (2015). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ABSTRACT

Feedback, such as an inspection of a part, is a key step in the design and manufacture of complex products. It determines where a product or manufacturing process should be re-evaluated to conform to design specifications. The inspection of a part is characteristically accomplished by comparing the CAD model to the measurements of a manufactured part. This is simple for parts that contain a commonality: central axis, plane on a flat side, center of a sphere, etc. When a part does not share a commonality—like free-form surfaces—the comparison analysis becomes complex.

This complexity occurs when determining the process for correspondence of every point on a manufactured part to every point on a design model. Whenever one coordinate system is shifted, the comparison can be lost and, then, has to be reevaluated, creating an iteration. The demand for substantial iterations protracts the process and thwarts optimization. It is, also, challenging to mathematically determine which points should be compared to another. Is the selected point optimal for comparison? Is a higher resolution of points needed? This problem of how the coordinate systems of the CAD model and the measured part can be aligned is termed as localization and is extensively researched [1]. Currently, most algorithms use a line or surface fitting technique that minimizes the sum of the square of the errors, drawing upon Gunnarsson and Prinz's original idea [2]. Such nonlinear approaches may result in local minima when minimized, resulting in false solutions. Additionally, a solution achieved may not be optimal due to averaging of outliers in the data.

This thesis proposes a methodology that automatically aligns the coordinate systems of free-form CAD models to collected manufactured measurements, with resiliency to outliers of the fit and false solutions given by local minima, by maximizing the shared extent depending on dimension. To perform this, data from the manufactured surface and design surface are polygonized and compared until geometrically similar. Then, the overlapping or intersecting extent is calculated depending on the dimension and maximized using a heuristic approach, particle swarm optimization. At the maximum

shared extent, two coordinate systems should be aligned in the optimal position. In this work, only two dimensional free-form curves are used to determine if the maximization of the shared extent results in an optimal solution, reducing complexity from three dimensions. Results obtained validated the approach and that the manufactured curve was aligned to the design, as measured by the sum of the squared errors. Also, the method was discovered to resist outliers, demonstrated by the tight alignment of consistent sloped areas while not necessarily aligned to peaks and valley features. Error observed is mainly due to inaccurate polygon geometry between the curves rather than the maximization of shared area process.

ACKNOWLEDGEMENTS

The deepest appreciation and gratitude go to the following. Without any of you, this thesis would have not been possible.

First of all, I would like to sincerely recognize my advisor, Dr. Sam Anand. Without your guidance and expertise, this paper would still be just an idea or an eventual goal. Thank you for the consistent push and high expectations when it was needed and leniency when it was not. I apologize for the occasional late evening, bouts of stubbornness, and the long winded run-ons that needed correcting.

To Dr. Tom Huston and Dr. David Thompson, thank you for taking time out of your schedule to be a part of my defense committee. Both of you, in and out of the classroom, genuinely care for your students, which is rare and appreciated.

My gratitude goes out to all the people who I have crossed along the way at the University of Cincinnati, including Dr. Bernie Rudd, Dr. Kelly Anderson, Dr. Allyn Phillips, Dr. Mark Schulz, Dr. Edzel Lapria, Scott Phillips, and all the graduate students under Dr. Anand.

Mr. Steve Brickner, Dr. Andrew Mech, Dr. Richard Stamper, and Dr. Patrick Ferro, thank you for creating and continuing my passion in mechanical engineering. Each of you has driven me to pursue this degree, to strive to better understand the world around me, and to impart this zeal to the next generation.

To Anthony Pankala, you are my support, my anchor, and every other cliché. The small things like fresh coffee, doing laundry, walking the dog, and trying to cook have not gone unnoticed. I hope to one day return the favor.

Sherlock, thank you for keeping me awake and company during those late nights and early mornings. As much as I think you would enjoy this paper, you are not allowed to eat it!

For the constant nagging...I mean support, blank glazed stares when describing this thesis, and the random distractions, I would like to thank my parents, brother, and the rest of my extended family and friends. *It is finally done!*

TABLE OF CONTENTS

	Page
Abstract	ii
Acknowledgements.....	5
Table of Contents.....	7
List of Tables	9
List of Figures.....	9
Chapter 1 Introduction	13
1.1 Introduction.....	13
1.2 Free-Form Surfaces.....	13
1.3 Localization.....	14
1.4 Need for Research.....	16
1.5 Thesis Contribution.....	17
1.6 Layout of Thesis.....	17
Chapter 2 Problem Statement	19
2.1 Problem Statement.....	19
2.2 Initial Consideration.....	19
2.3 Goal of the Proposed Shared Area Maximization Algorithm.....	21
Chapter 3 Literature Review	22
3.1 Localization for Plane Surface Parts	22
3.2 Fundamental Localization Techniques for Free-Form Parts.....	22
3.3 Current Research in Localization.....	27
Chapter 4 Shared Area Maximization Algorithm.....	28
4.1 Overview of Algorithm Logic.....	28

4.2	Curve Creation	29
4.3	Randomization	31
4.4	Polygonization	32
4.4.1	Liaison Sequencing	33
4.4.2	Shape Characteristics	34
4.4.3	Shape Characteristic Error Calculation	39
4.5	Particle Swarm Optimization	39
Chapter 5	Analysis of Results	44
5.1	Methods of Analysis	44
5.2	Observations of the Results	46
5.3	The Sum of Squared Errors Comparison	49
5.4	Shared Area Analysis	52
5.5	Error Area Analysis	55
5.6	Comparison of Swarm Transformation Results to Randomization Values	57
Chapter 6	Conclusion	61
6.1	Overview of Thesis	61
6.2	Deductions from Results	61
6.3	Possible Limitations of the Algorithm	62
6.4	Future Research	66
References	68
Appendix	73
	Table of results from algorithm sequences	74

LIST OF TABLES

Table	Page
Table 1: Steps within the liaison sequencing algorithm.....	34
Table 2: Gradient increase of the convergence factor.....	42
Table 3: Transformation parameters from the twenty five trials	58

LIST OF FIGURES

Figure	Page
Figure 1: An example of freeform curve creation via NURBS [5].....	14
Figure 2: A summarization of the localization problem [8].....	15
Figure 3: Comparison of a design surface, tolerance boundaries, manufactured surface, and variation of the particular manufactured surface [9].....	15
Figure 4: Tolerance vs. dimensions for typical free-form part industries [10]	16
Figure 5: Area between two curves [12].....	20
Figure 6: Geometrical entities in two and three dimensions [13]	21
Figure 7: Example of 3-2-1 method [4]	22
Figure 8: Two-dimensional example of Gunnarson and Prinz's algorithm [2]	23
Figure 9: An example of Zhang's Iterative Point Matching technique [21]	24
Figure 10: Symmetric features used in Li, Gou and Chu various algorithms [26]	25
Figure 11: Segmented normals by Orazi and Tani's method [33]	26
Figure 12: Delaunay pole spheres and the medial axis transform [37].....	27
Figure 13: The shared extend localization algorithm overview	28
Figure 14: The random placement of control points.....	29

Figure 15: The cubic spline interpolation of control points	30
Figure 16: The result of curve creation section.....	31
Figure 17: The random translation and rotation of MFGC	31
Figure 18: Flowchart of polygonization.....	32
Figure 19: Example of perimeter of a polygon	35
Figure 20: Example of area of a polygon.....	35
Figure 21: How to determine the area of a polygon.....	36
Figure 22: Example of compactness of a polygon	37
Figure 23: Example of convexity of a polygon.....	38
Figure 24: Example of solidity of a polygon	38
Figure 25: The shared area between the MFGC and CADC polygons	39
Figure 26: Flowchart of Particle Swarm Optimization [49]	40
Figure 27: Dispersion of swarm particles before optimization	41
Figure 28: Swarm particle results and maximization over iterations.....	43
Figure 29: Example of a localized MFGC	43
Figure 30: An example of shared area before and after localization	45
Figure 31: A shared area comparison with different polygonizations	45
Figure 32: An example of error area before and after localization	46
Figure 33: Result of localization via shared area maximization	47
Figure 34: Final localization conclusion based on shared area maximization	47
Figure 35: Shared area localization that aligns with initial MFGC.....	48
Figure 36: Steps of the shared area localization that aligns with initial MFGC	49
Figure 37: Sum of squared errors between the MFGC and CADC in initial position, before and after shared area maximization.....	50
Figure 38: Percentage decrease of the sum of squared errors	50

Figure 39: Comparison of percentage decrease between the initial position and after shared area maximization	51
Figure 40: Difference between the percentage decreases needed to obtain initial position and to perform the shared area maximization	52
Figure 41: Shared area initially and after shared area maximization.....	53
Figure 42: Percentage difference of the shared area initially and after shared area maximization.....	53
Figure 43: Steps of the shared area result for Run One	54
Figure 44: Error area before and after localization	55
Figure 45: Percentage difference of error area before and after localization.....	56
Figure 46: Steps of the shared area result for Run Three.....	56
Figure 47: The comparison of the actual and calculated theta values.....	58
Figure 48: The comparison of the actual and calculated delta x values.....	59
Figure 49: The comparison of the actual and calculated delta y values.....	59
Figure 50: Example of failure due to extreme shifts in slope	63
Figure 51: Steps of a failure example due to extreme shifts in slope.....	63
Figure 52: Example of a success concerning extreme shifts in slope	64
Figure 53: Steps of a successful example when extreme shifts in slope occur	64
Figure 54: Example of failure due to mirrored curve	65
Figure 55: Steps of a failure example due to extreme shifts in slope.....	65

CHAPTER 1 INTRODUCTION

1.1 Introduction

The perpetual goal of manufacturing is the replication of a perfect product from raw materials. However, perfection is infeasible, which has created the acceptance of errors by specifying tolerances on part models produced by any manufacturing process. Yet, with computer-aided design (CAD) models, the ideal part is now virtually captured and can be manipulated. Naturally, this generates an ambition to compare the ideal part, the CAD model, to any actual part to gauge the quality of the manufacturing process.

This initiates the question: how are two objects compared to each other? It may seem simple. Determine shared features—shapes, axis, and planes— and then align or nest these features inside each other. If performed correctly, differences between the two shapes become visible. And yet, this relies on the assumption that the two objects share obvious features or characteristics. What if the two objects do not share any recognizable features? How would the objects be compared? Would the results be accurate and would those results be intuitive? This is the case for free-form curves, and these questions are not simply answered.

1.2 Free-Form Surfaces

What are free-form surfaces and why can they not be compared through conventional means? Free-form surfaces are three-dimensional surfaces that do not have a constant geometrical shape or set pattern, are classified as either constrained or unconstrained, and are typically produced using sophisticated CAD programs [3]. They are widely used in the design and manufacturing of dies, molds, and products ranging from the plastics, biomedical, entertainment applications to automotive and aerospace industries [1]. To create a free-form surface, the CAD models are defined by piecewise parametric functions to create the shape of a part, using one of the three methods: Bezier, Spline or NURBS [4]. An example of a NURBS curve and surface are displayed in Figure 1.

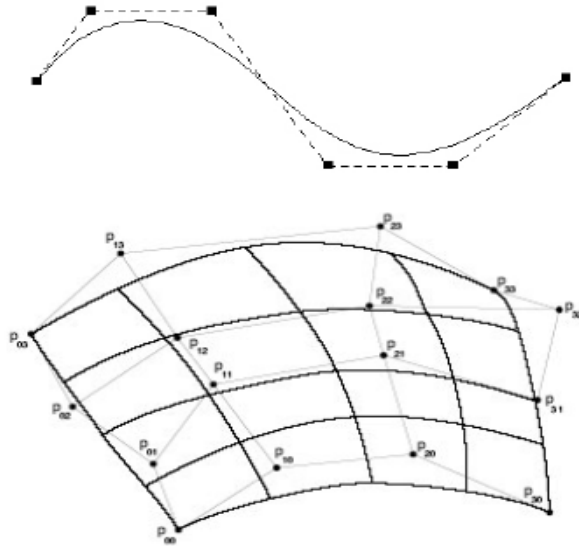


Figure 1: An example of freeform curve creation via NURBS [5]

From Figure 1, it is apparent that a CAD model surface data can be extracted via several methods: points along the surface, collecting control point positions, surface equation estimation, etc. Regardless of the method used eventually, for comparison of the CAD surface to the manufacturing surface, specific positions or direct points will be needed along the CAD surface. A similar dataset will need to be collected from the manufactured part. Contact or non-contact measurement sensors along with computer-controlled coordinate measurement machines (CMMs) are commonly used to achieve high-precision dimensional measurements of complex parts, such as a free-form surface [4]. Once both datasets are collected, how can the coordinate systems be aligned to compare the manufactured and CAD surfaces?

1.3 Localization

Localization is the process in which two objects are compared and aligned when they do not share any recognizable features, shapes, axis, and planes, such as free-form curves [6]. In the scope of manufacturing, the alignment is concerned with the determination of the relationship between two coordinate reference frames: measurements from a part and surface points generated by a CAD model [7]. Simply phrased, it is the process that provides the user with the appropriate translation and rotation values

to align CAD model coordinate system and the manufactured part coordinate system, which is shown in Figure 2.

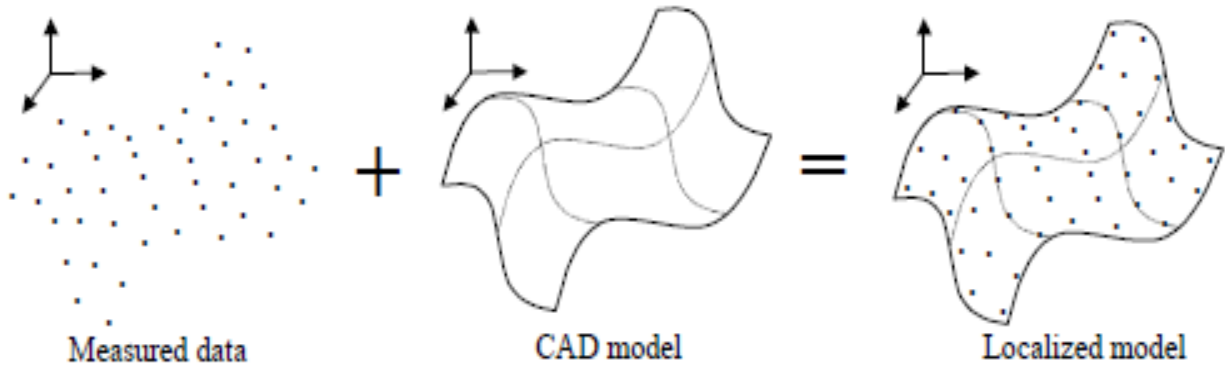


Figure 2: A summarization of the localization problem [8]

Although a CAD surface is depicted in Figure 2, for this research, a collected set of data points from the surface is utilized.

Again, the goal of localization is to align the manufacturing points onto the CAD surface [7].

Once aligned properly, tolerance analysis on the surface can begin, which is demonstrated in Figure 3 [9].

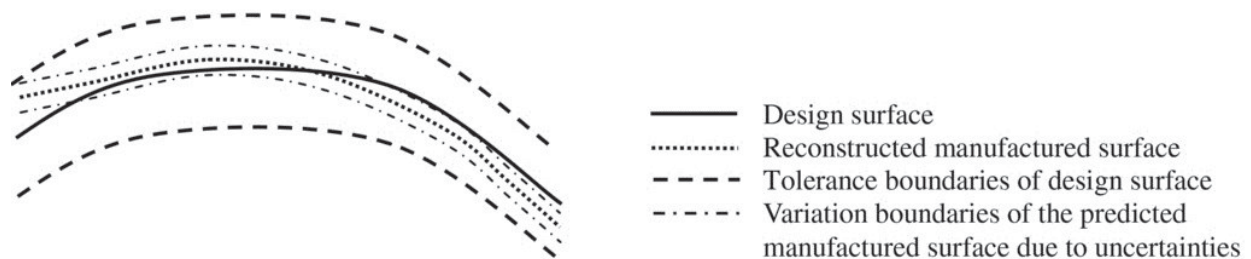


Figure 3: Comparison of a design surface, tolerance boundaries, manufactured surface, and variation of the particular manufactured surface [9]

In Figure 3, the design surface has a tolerance boundary that a manufactured surface can vary within, due to manufacturing defects [9]. This manufactured surface is depicted by the closely dashed line, as well as

the variation of the part [9]. From Figure 3, there is a possibility for arbitrary adjustments, meaning alignment or localization is not always intuitive.

1.4 Need for Research

Currently, computing technology is approaching the ability to solve difficult optimizations and manipulate enormous sets of data. This allows for new methods to localization that were deemed previously unattainable. The potential solution to localization is key to industries such as consumer products (mobile phones, cameras, etc.), household appliances (water pumps, fans, etc.), aerospace (turbine blades, impellers, aerodynamic parts, etc.), and automotive (car bodies, 3D-cams, seals, gears, etc.) [10]. Each of these have a desired tolerance range, shown in Figure 4, which has to be met for safe consumer usage or other benefit to the manufacturer.

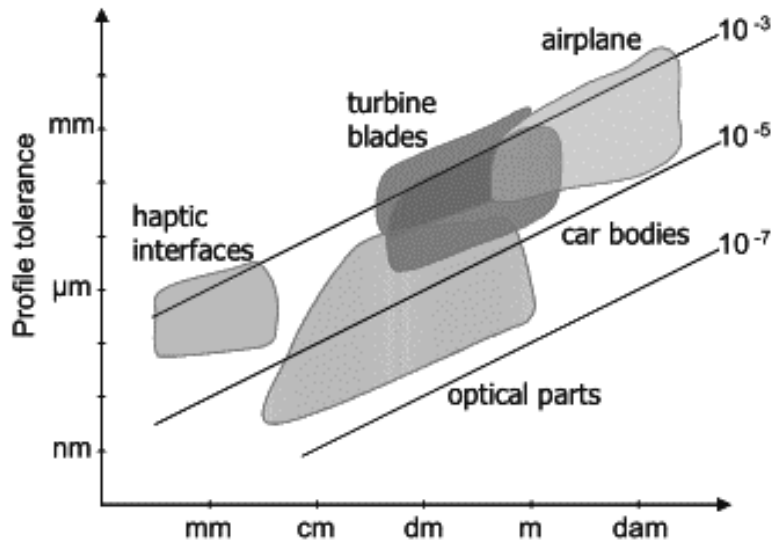


Figure 4: Tolerance vs. dimensions for typical free-form part industries [10]

Without the proper alignment discussed in Figure 3, the tolerances, shown in Figure 4, may be unmet and not intuitively measured. The feedback cycle in design phase is deficient. Optimization of material use and feature placement is impracticable, creating more expensive, over-engineered parts. Failure analysis using real world comparisons is deficient as well, meaning the failure of a part may occur

repetitively. If localization could be performed rapidly, it would open the possibility for automated quality control on a manufacturing line, guaranteeing that a product met the given specification range, such as the requirements listed in Figure 4.

1.5 Thesis Contribution

This thesis proposes the methodology that could automatically compare and align measurements from a manufactured part to points generated from a CAD surface. The approach is resilient to local minima and will not average outliers for the fit. To achieve this, the algorithm focuses on creating a digital mesh of the design surface and the manufactured part. Once a geometrically similar mesh is created for each, the overlapping volume or area of the meshes could be calculated. Then by maximizing the shared volume or area, the CAD model and the manufactured part coordinates would be aligned in the optimal position.

As the foundation of this work, only two dimensional free-form curves are used instead of surfaces to reduce complexity and to test the hypothesis that maximization of the shared extent, area in this case, results in an optimal solution. Similar control polygons are generated for both the manufactured part curve and the CAD model curve. The overlapping area is maximized creating a localization solution. Obtained results demonstrated an independent reduction of the sum of the squared errors between the two curves, validating this hypothesis. Also, the method was resistant to outliers, demonstrated by the tight alignment of consistent sloped areas while not necessarily aligned to peak and valley areas. Error generated by this method is mainly due to inaccurate creation of polygon geometry rather than the maximization of shared area process.

1.6 Layout of Thesis

The remainder of this thesis will be arranged as follows. First, the Problem Statement in Chapter 2 will clearly state the scope, initial considerations, and the overall goal of this research. In Chapter 3, the Literature Review section will cover the traditional approaches of localization, additional techniques, and

current research. Subsequently, Chapter 4, the Shared Area Maximization Algorithm section, will explain step-by-step the proposed algorithm. The analysis of the results from the algorithm will follow in Chapter 5. Chapter 6 will conclude inferences formed from the results and discuss the future direction for this research.

CHAPTER 2 PROBLEM STATEMENT

2.1 Problem Statement

The difficulties in developing a solution to localization originates from how points are compared to each other. Generally, every point on one surface is compared to every on the other surface via distance rays, as shown in Figure 8 and Figure 9 [2]. The sum of least squares is an often used direct approach, which is prone to local minima during optimization [11]. Another difficulty is trying to mathematically determining where or which point to compare a data point along a curve. It may not always be the closest point. Any curve estimation to determine where the point fits may introduce its own error into a solution through the averaging of points, giving outliers significant weight. Without knowing which point on a surface to compare with another, the number of iterations are substantial, time to solutions are large, and significant computing power is needed.

Thus, the direct approach of least squares or curve approximation is not always an optimal approach for localization. Are there alternative possibilities to compare and align datasets? Are there ways to be less influenced by outliers in the data and not add error of its own? Would this direction create less iterations and be less likely to fall into local minima?

2.2 Initial Consideration

Instead of the sum of least squares approach, perhaps, localization could be considered akin to a basic calculus problem. The area or volume between the two curves or surfaces could be found by the integral between the curves or surfaces, shown in Figure 5.

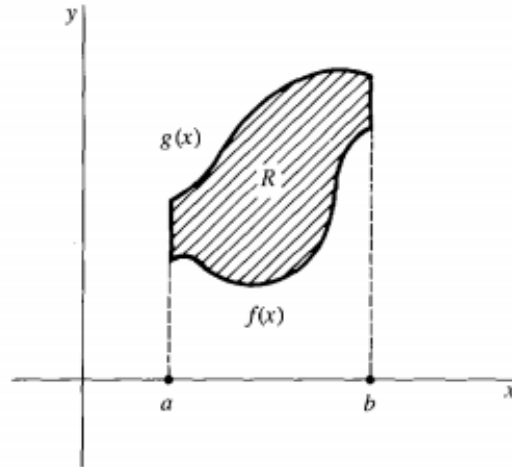


Figure 5: Area between two curves [12]

Once the area or volume was determined by Equation 1 [12], it could be minimized by an optimization technique.

$$A = \int_a^b (f(x) - g(x)) dx \quad [1]$$

Since freeform curves do not have a constant geometrical shape or set pattern [3], a piecewise function would be used in Equation 1. To implement this, an equation approximation of a curve or line is needed for each section of the CAD curve and between each point in the dataset for the manufactured part. These equations would be algorithmically intensive if determined automatically, especially when the data points are potentially unorganized. Also this method, by using approximations, would create its own error and be skewed by outliers. The end connection and any crossing of the curves would need to be determined and evaluated to guarantee the area between the functions is consistently positive. If part of the area was negative, an optimization technique would not minimize the area between the curves properly. This requirement is potentially problematic to automate within an algorithm, requiring significant iterations and sweeping of datasets searching for intersections. Solution time and computing requirements would significantly increase due to this.

2.3 Goal of the Proposed Shared Area Maximization Algorithm

The goal is to develop a method that does not add error by approximation of a curve or averaging outliers, reduces the number of iterations, and will be more resistant to falling in a local minima. To determine if this is plausible, the scope of the analysis will focus only on two-dimensional free-form curves with the intention that the algorithm could be expanded to three dimensions. This expansion from two-dimensional to three-dimensional is shown in Figure 6.

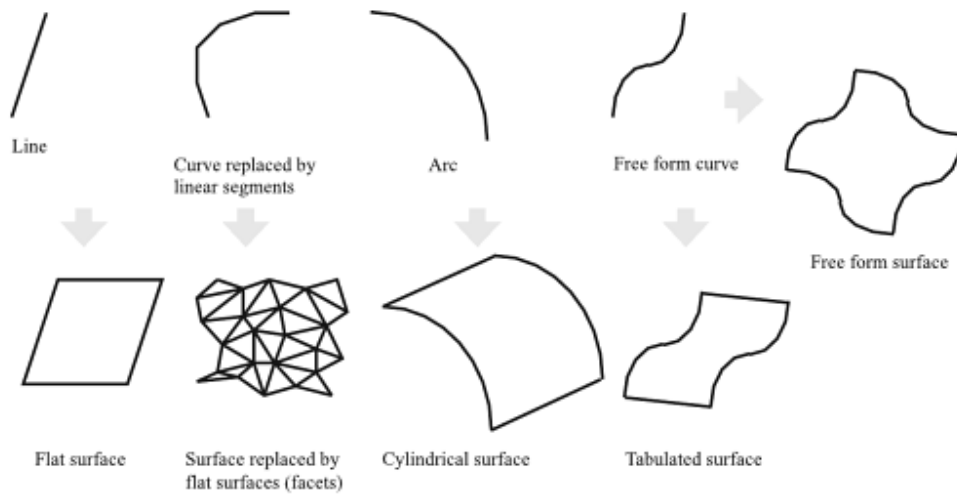


Figure 6: Geometrical entities in two and three dimensions [13]

To achieve this goal, the shared area maximization algorithm will compare data points of a manufactured part and CAD model by bounding each curve in a polygon, then confirm similarity of the polygons based on shape characteristics, and maximize the shared area of the two polygons by using particle swarm optimization (PSO). This is the equivalent of creating a digital outline of the CAD curve and then searching for the most geometrically similar outline from the manufactured part's data points by comparing shape morphology. By maximizing the shared area between the two outlines, the two datasets should align the coordinates systems in the optimal position. This method will reduce the number of iterations since a data point sweep is not necessary to determine relation between each point during optimization. By using the area of the polygons, the method will be more resistant to outliers, and by not approximating, additional uncertainty is not added.

CHAPTER 3 LITERATURE REVIEW

3.1 Localization for Plane Surface Parts

Localization can be considered as a two-step process: first, determining the relationship between measurement and design surfaces and, second, solving the 3D rigid transformation between the two surfaces to align the coordinate systems [1]. Typically once the relationship between the two coordinate frames is created, the three dimensional transform is implied. Creating the correct correlation with free-form curves varies significantly throughout existing research, unlike a part with a datum or a flat surface. In a basic part, the 3-2-1 method is used [1]. Three points are measured from the first datum to establish a plane; two more are measured from the second datum to establish another plane perpendicular to the first; and the last point is measured from the datum perpendicular to the first two datum [1] [14] [15]. This is displayed in Figure 7.

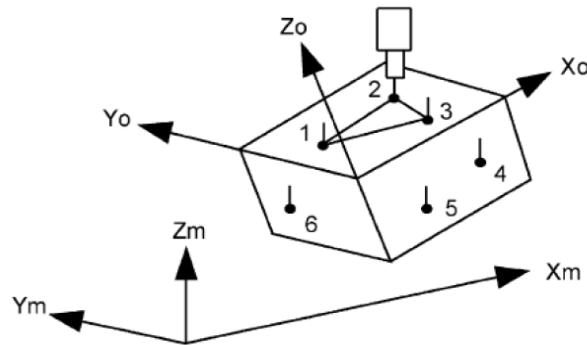


Figure 7: Example of 3-2-1 method [4]

Unfortunately, this method does contain vulnerabilities to measurement and dimensional errors due to the selection of a datum from surface data, similar to the free-form case [1].

3.2 Fundamental Localization Techniques for Free-Form Parts

Starting in 1987, Gunnarsson and Prinz created a foundational solution to attempt matching CAD models to manufacturing data [2]. They created an algorithm that localized the data by minimizing the sum of the squared errors between the two surfaces; this process is visible in Figure 8 [2].

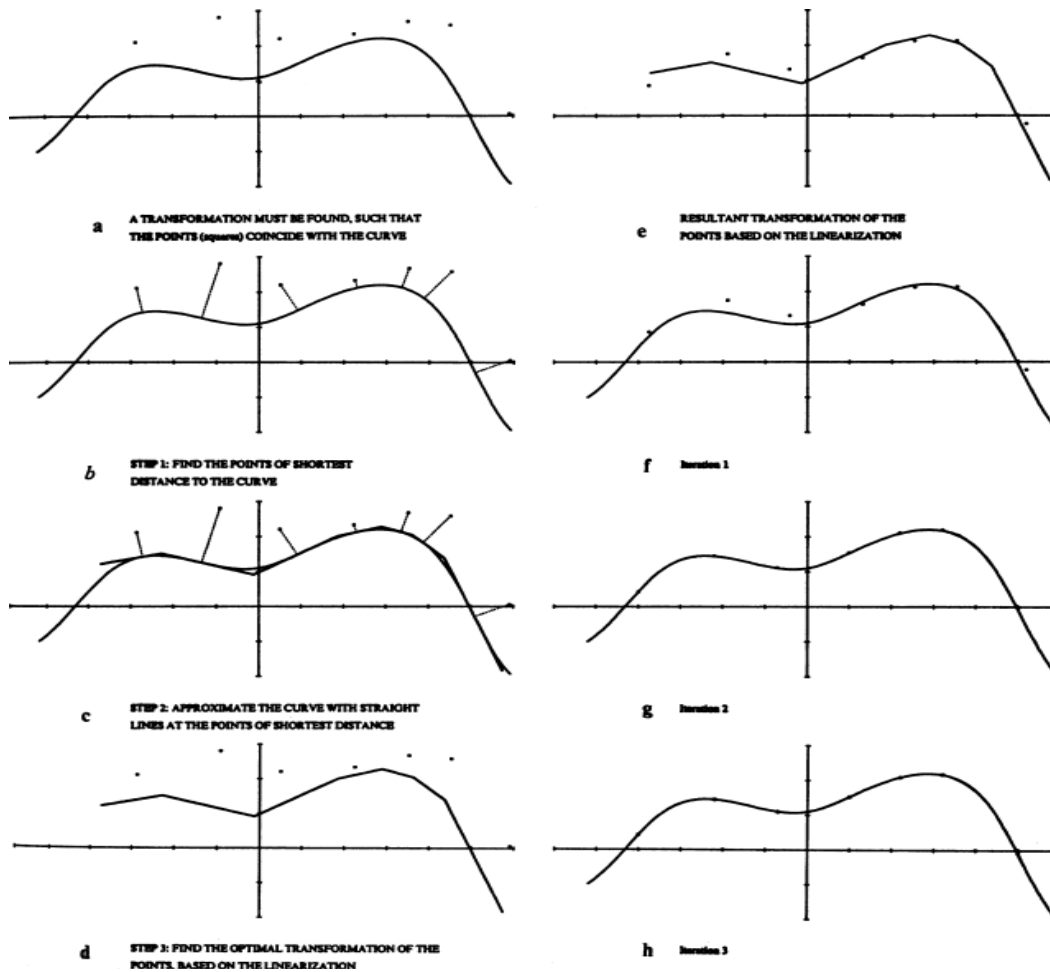


Figure 8: Two-dimensional example of Gunnarson and Prinz's algorithm [2]

From Gunnarson and Prinz's work, Arun and Horn expanded on the mathematics of the algorithm to create a closed form solution of the localization problem [16] [17]. Horn, specifically, organized a rotation matrix using quaternions and solved for the best rotation using eigenvectors and eigenvalues [17]. To expand upon this idea, Patrikalakis and Bardis formed a method to correlate the edges or vertices of the two surfaces before rotational optimization occurred [18]. Sahoo and Menq combined both of these ideas and generated a single objective function to minimize by using a transformation matrix, able to rotate and translate the coordinate systems at the same time [19]. A year later, in 1992, Menq, Yau and Lai were able to increase the efficiency of the previous method by approximately tenfold [4].

To increase the rate of convergence between the two surfaces, Besl and McKay used the Iterative Closest Point (ICP) scheme to minimize the smallest distance between the two surfaces within each iteration [20]. This process decrease the amount of iterations significantly, however each iteration was more computationally intense. Instead of using the closest point singularly, Zhang developed Iterative Point Matching that found the closest point for the entire dataset and then minimized the sum of the square of the errors [21]. This method can be seen in Figure 9.

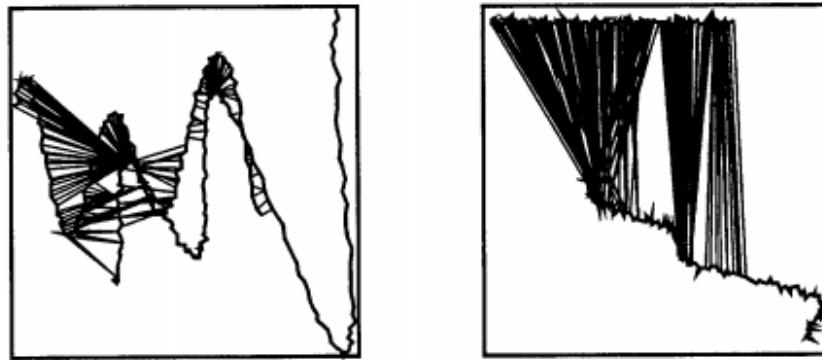


Figure 9: An example of Zhang's Iterative Point Matching technique [21]

Huang also utilized a point matching algorithm throughout the dataset that was inside another algorithm driving the determination of a transformation matrix [6] [22]. Another extension of Besl and McKay's ICP algorithm is Ahmed, Farag, and Yamany's grid closest point (GCP) technique, which remembers the grid association of the data sets and manipulates one data set each iteration, instead of searching the dataset each iteration, decreasing the overall runtime while maintaining accuracy [23]. Gudla constructs from Ahmed's grid technique by using generalized Hopfield networks on the both data sets before using the closest point minimization [24].

Eleven years after Gunnarson and Prinz's work, research gravitated towards increasing reliability of the localization fit. Che and Ni studied the effect of uncertainty when propagated through a transformation matrix [25]. This same year, Li, Gou and Chu explore various geometric algorithms for localization, breaking complicated parts into manageable pieces, shown in Figure 10 [26].




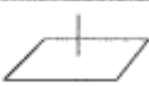





Features	Symbols	Isotropy Subgroup G_1	Configuration Space Q	Description of Isotropy Subgroup	Dimension
Circle in E^2		SO(2)	$SE(2)/G_1$ $=T(2)$	Rotation around the center of circle	2
Straight line in E^2		T(1)	$SE(2)/G_1$	Translation along the line	2
Sphere		SO(3)	$SE(3)/G_1$ $=T(3)$	Rotation about the center of sphere	3
Plane		SE(2)	$SE(3)/G_1$	Rotation about the normal of the plane & translation in the plane	3
Straight line in E^3		$SO(2) \times T(1)$	$SE(3)/G_1$	Rotation about and translation along the line	4
Cylinder		$SO(2) \times T(1)$	$SE(3)/G_1$	Rotation about and translation along the axis	4
Cone (resolved surface)		SO(2)	$SE(3)/G_1$	Rotation about the axis of cone	5
Tabular Surface		T(1)	$SE(3)/G_1$	Translation along the sweeping direction	5
Sculptured Surface without symmetry		1	SE(3)	Identity element	6

Figure 10: Symmetric features used in Li, Gou and Chu various algorithms [26]

A year later, the six equations used in the transformation matrix is reduced by Qu and Sarma to four equations using eigenvalues [27]. Next alteration, explored by Li, Xu, Cen, and Ding, was changing the minimization of the least squares of residual errors to the least median, trying to reduce the effect of outliers on the fit [28]. Sharp integrated invariant features, such as curvature, second order moments, spherical harmonics, and global density, to Besl and McKay's ICP algorithm, allowing for more efficient corresponding point matching in images [29].

By exploring invariant features or features of the surface, the trend of research shifts to evaluating surface qualities. In 2002, Yamany and Farag constructed a method to reduce the complexity of 3D surfaces by capturing the surface curvature information from key points to a surface signature [30]. To

find the closest point for ICP, Aware also maps the surface features, by determining the surface signature of a triangulated Bezier surface by using k-d trees [31]. Instead of creating a surface skeleton, Ko, Maekawa, and Patrikalakis use the constant Gaussian and mean curvature lines to find corresponding surface features to optimize a transformation [32]. The Gaussian curvature was also employed by Orazi and Tani, who compared the normals of the segmented surfaces to find corresponding points, shown in Figure 11 [33].

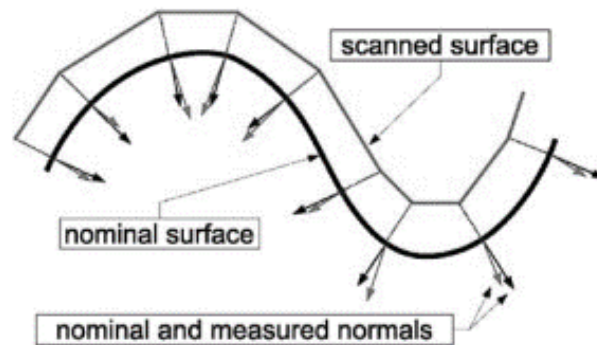


Figure 11: Segmented normals by Orazi and Tani's method [33]

Along with the Gaussian curvature, Li and Gu continued with a series of internal and external properties of the surface to help facilitate ICP for rough alignment and GCP for smooth alignment [34]. This two-step process, rough and smooth alignment, was also used by Sije, who exploited Genetic Algorithms (GA) to determine the surface matching [35]. To aid in a possible GA, Narayanaswamy created the idea of a search window to find the discrete design point set, which is closest to any particular measurement point and suggests using the u and v values of that point as a starting point for Newton's method [36]. Also using Newton's Method, Joshi designs a heuristic approach to determine to obtain an initial guess for minimizing distances and was able reduce the total number of iterations required to calculate the transformation parameters when compared to the traditional ICP approach [7].

3.3 Current Research in Localization

This trend of examining surface features for localization has continued into current research. OuYang localized the design and measurement surfaces by comparing medial axis transforms, shown in Figure 12, formed by using Delaunay pole spheres (DPS) [37].

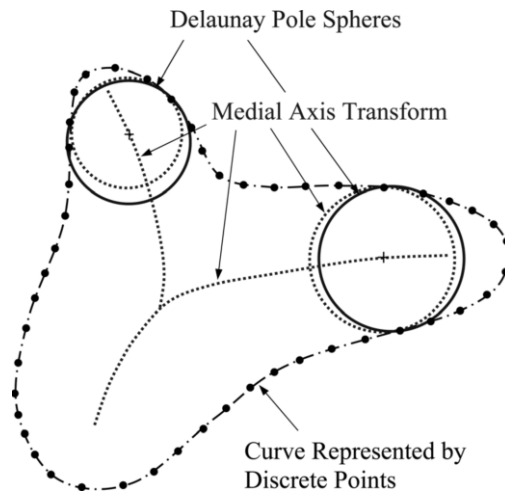


Figure 12: Delaunay pole spheres and the medial axis transform [37]

Mehrad, Xue, and Gu increased the robustness of rough localization by using the similarities of distances between the surfaces and the similarities of major and minor curvatures [38]. For fine localization, Mehrad optimized the coordinate system based on the surface features [38]. Also based on the surface features, Mehrad attempted to predict variance in data points and determine uncertainty ranges [38] [9]. Lastly, Wen et al. utilized quasi particle swarm optimization to search the transformation parameters and surface subdivide method, reevaluating the closest point via the NURB data, to evaluate the surface profile error of free-form surfaces [39].

This research draws upon Wen's idea of using a heuristic optimization, like particle swarm optimization, to drive discovery of transformation parameters and combining it with Li's thoughts of utilizing the internal and external properties of a surface for localization [34] [39].

CHAPTER 4 SHARED AREA MAXIMIZATION ALGORITHM

The shared area maximization algorithm will compare data points of a manufactured part and CAD model by bounding each curve in a polygon, then confirm similarity of the polygons based on shape characteristics, and maximize the shared area of the two polygons by using particle swarm optimization (PSO). This chapter will explain this process in further detail.

4.1 Overview of Algorithm Logic

The shared area maximization algorithm is divided into four parts: curve creation, randomization, polygonization, and particle swarm optimization as shown in Figure 13.

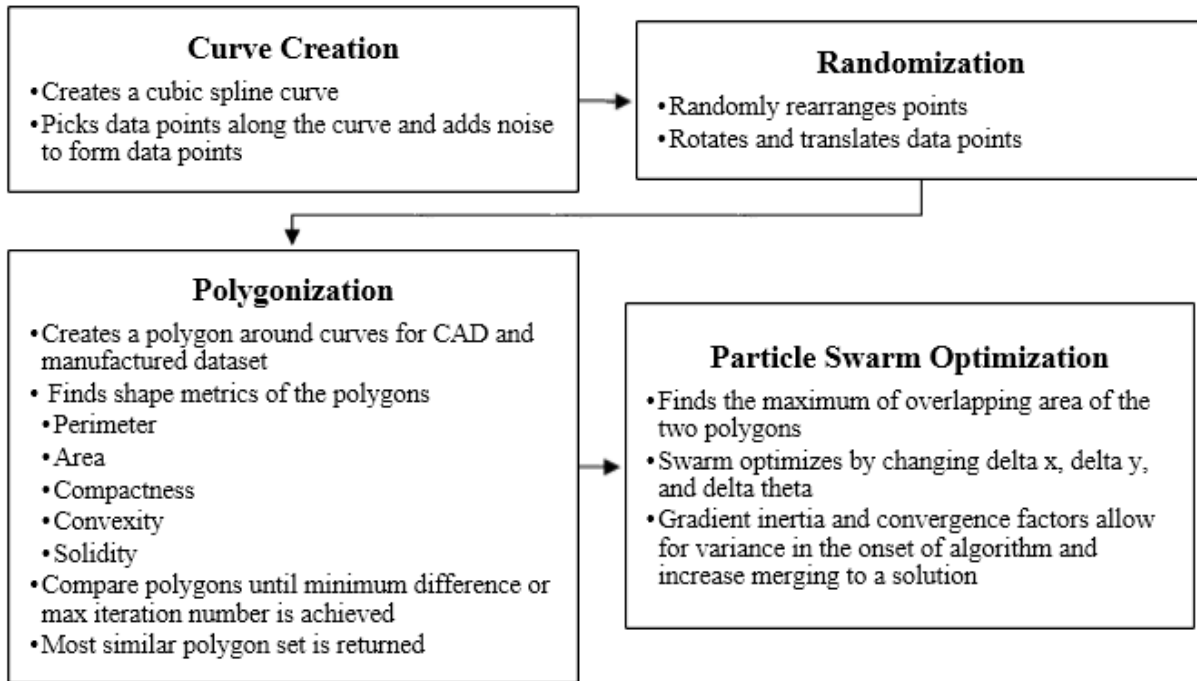


Figure 13: The shared extend localization algorithm overview

The first two sections of Figure 13 are the initialization of the localization problem. The curve creation section creates a smooth cubic spline curve, simulating the CAD curve (CADC), and a rough curve imitating the data points measured from a manufactured part (MFGC). Randomization translates, rotates, and reorders the MFGC, thus setting the problem of localization.

The concentration of this thesis is contained within the final two sections of Figure 13. The polygonization section creates two polygons out of each curve data structure. The first polygon consisting of the points from the CADC, and the second polygon using the MFGC points. Then, the MFGC data polygon is rearranged until the most similar polygon to the CADC data is generated using shape morphology. The shared area between the two polygons is determined, and particle swarm optimization maximizes the shared area by changing the rotation and translation variables of the MFGC polygon. By maximizing the shared area, the algorithm is able to determine how the MFGC should be translated or rotated to achieve the best fit to the CADC, testing the hypothesis that the maximization of shared area results in localization. Each of these processes will be explained in further detail in the remainder of this chapter.

4.2 Curve Creation

To start, a smooth cubic spline curve simulating a CAD curve is created. The control points are generated by evenly spacing an arbitrary amount of points on the x axis between -1 and 1. For this thesis, five control points are selected to make a smooth surface, with some oscillation. Large amount of control points will result in extreme variations in the curve, which should be avoided. On the y axis, the control points are allowed to randomly locate between -1 and 1 as well. This random placement can be seen in Figure 14.

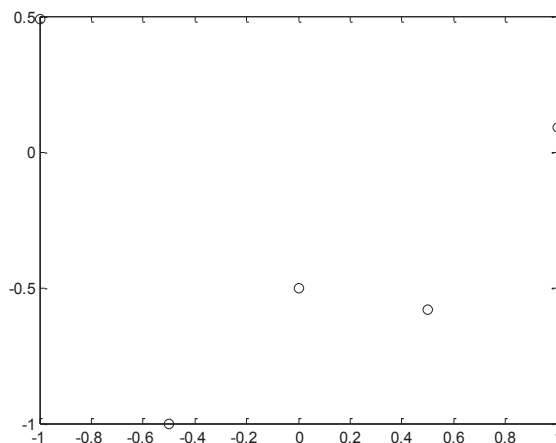


Figure 14: The random placement of control points

These control points are then used in a cubic spline interpolation, using the following Equation 2 [40]:

$$Y_i(t) = a_i + b_it + c_it^2 + d_it^3 \quad [2]$$

where $t \in [0,1], i = 0, \dots, n - 1$, and $n = \#$ of control points

The resolutions of the CADC is created using a hundredth deviation for t , resulting in a hundred points. Whereas, the MFGC is created by using a step size of five hundredths, giving a resolution of twenty data points. This gives the data points shown in Figure 15. The open circles are the control points shown in Figure 14. The x marks are the MFGC interpolation; the solid dots are equal to the CADC.

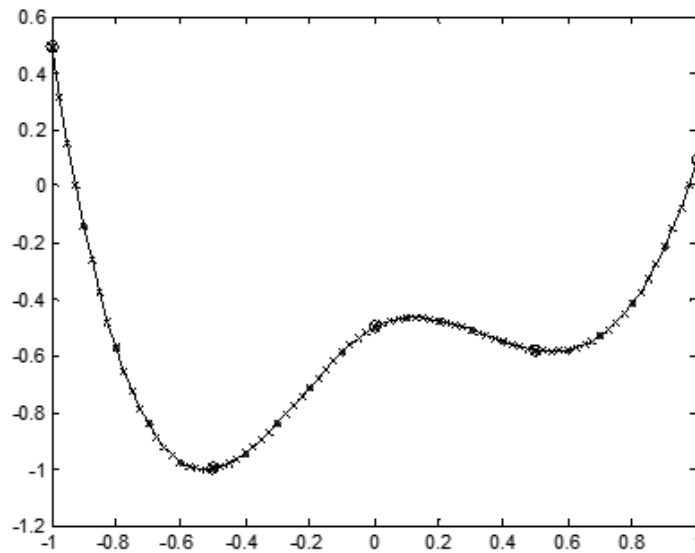


Figure 15: The cubic spline interpolation of control points

In Figure 15, the MFGC contains no error and does not simulate a real part. Each of these points are independently given a random nudge in the x and y direction within a reasonable range. This simulates how a manufactured part will have a flawed curve. Then the MFGC points are randomly ordered as well. This might seem unnecessary, since the data points would be collected in some order. Nonetheless, it was performed to prevent the algorithm from containing a bias towards the order of the points. The result of these calculations is shown in Figure 16.

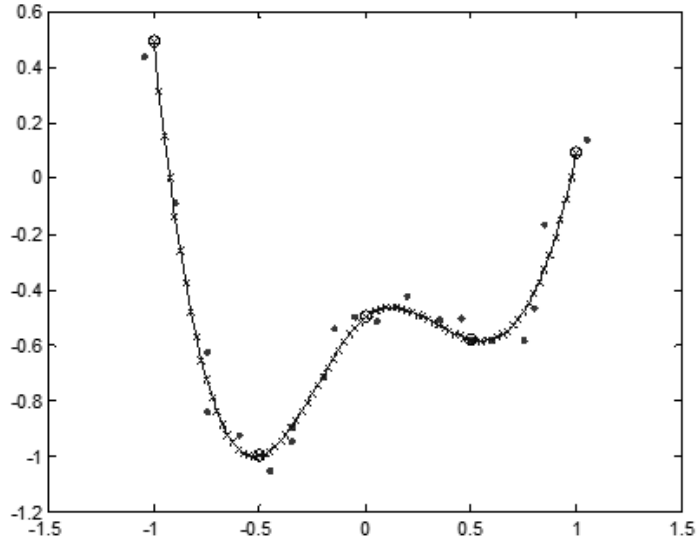


Figure 16: The result of curve creation section

4.3 Randomization

After the data points are achieved, as shown in Figure 16, the MFGC is still aligned to the original CADC. In order to simulate the difficulty of localization, the data points from the MFGC are randomly rotated and then randomly translated in the x and y direction. Clearly visible in Figure 17, the MFGC, solid dots, is no longer aligned to the CADC data.

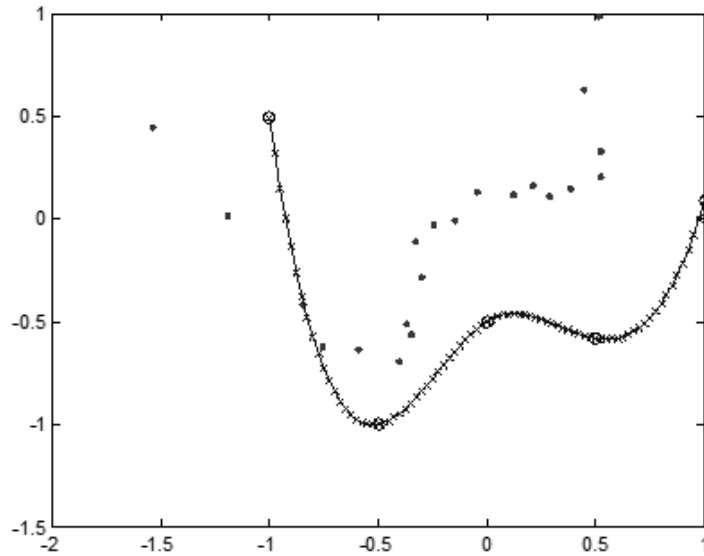


Figure 17: The random translation and rotation of MFGC

Thus the MFGC is properly simulating measurement from a part, setting the problem of localization for the rest of the algorithm. Also, the random values for delta theta, delta x, and delta y are recorded for comparison to any results in later chapters.

4.4 Polygonization

The goal of this section is to create a unique polygon for each the MFGC and CADC. This process is shown in the flowchart within Figure 18.

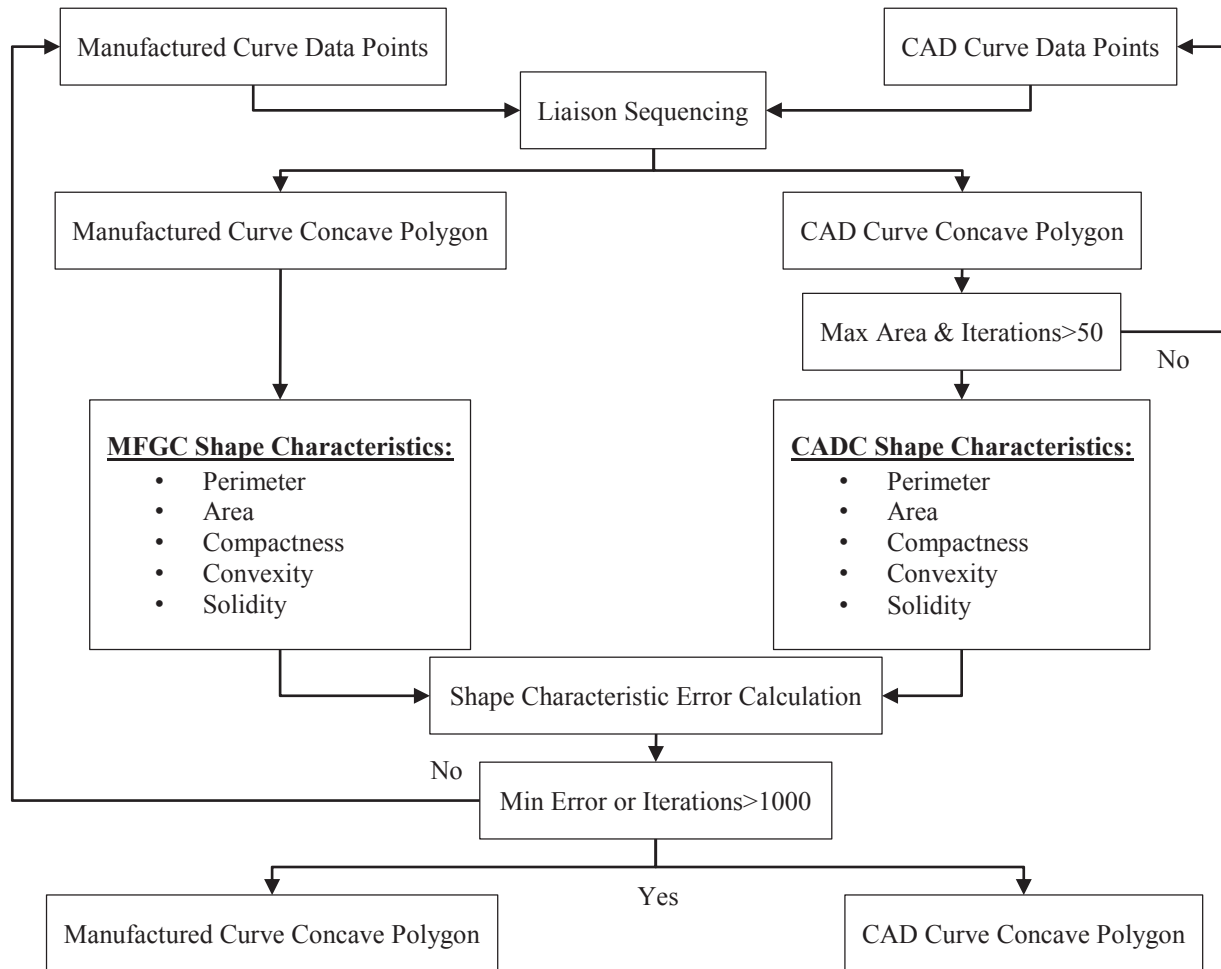


Figure 18: Flowchart of polygonization

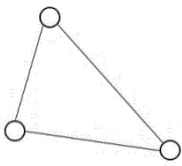
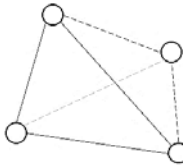
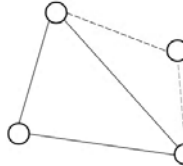
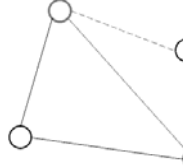
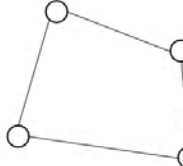
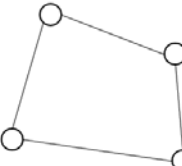
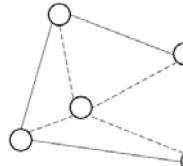
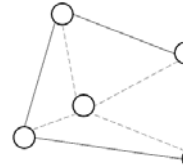
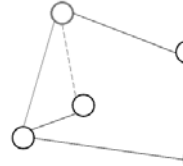
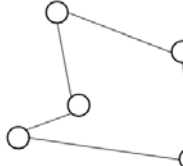
First step creates a concave hull around the data points of each curve; this logic is called liaison sequencing. Liaison sequencing gives an ordered connection that forms a non-intersecting concave polygon. The CADC liaison sequencing is maximized within a set amount of iterations for the maximum area, resulting in the best comparison to the MFGC polygon during optimization. Several hundred iterations of liaison sequencing is applied to the MFGC, utilizing shape characteristics to find the most geometrically similar MFGC polygon to the CADC polygon. The two geometrically similar polygons move to the particle swarm optimization section of the algorithm. The liaison sequencing algorithm, the shape characteristic used, and the shape characteristic error calculation will be explained in further detail within the rest of this chapter.

4.4.1 Liaison Sequencing

This part of the algorithm is able to create a closed non-intersecting concave polygon around a dataset. It is sensitive to the order that the dataset, meaning that a new polygon could exist for each unique reorder of the data. Though this adds complexity, it was a way to compare different size curves or datasets, since using vectors or matrices is not an option.

First, liaison sequencing selects the first three points and connects these together using a liaison. Once the next point is introduced, it searches for the nearest neighbor within the polygon without intersecting any liaisons and within an angle allowance. This angle allowance eliminates most jagged connections that would be uncharacteristic for a spline curve. The closest neighbor is selected, and the rest of the known points in the polygon are cycled to determine a shared connection to the selected neighbor. This shortest, shared connection between the selected neighbor and a known point, without intersections, is selected and erased. A new connection is made instead to the fourth point and the known point. This process is repeated until the polygon is completed, and all points are connected by two liaisons. This is demonstrated in Table 1 for an internal and external example.

Table 1: Steps within the liaison sequencing algorithm

Step 1	Step 2	Step 3	Step 4	Step 5
Connect first 3 points together or Use data from Step 5	Find neighbors without intersections	Choose nearest neighbor within angle allowance	Determine the closest shared point that both neighbor and the new point connect on	Complete the polygon
				
				

4.4.2 Shape Characteristics

Once a set of MFGC and CADC polygons has been created by using liaison sequencing, the two polygons are compared to gauge geometric similarity. To determine this, the shape of each has to be evaluated. Shape can roughly be defined as the description of an object without consideration to its position, orientation, and size; therefore, shape features or characteristics should be invariant to translation, rotation, and scale [41]. This means any characteristic used in the comparison of the MFGC and the CADC polygons cannot be skewed by rotation or translation. Five shape characteristics conform to these restrictions with unique outcomes: perimeter, area, compactness, convexity, and solidity.

4.4.2.1 Perimeter

To find the perimeter of the MFGC and the CADC polygons for comparison, the sum of the lengths of the boundaries has to be determined. An example of this is displayed in Figure 19.

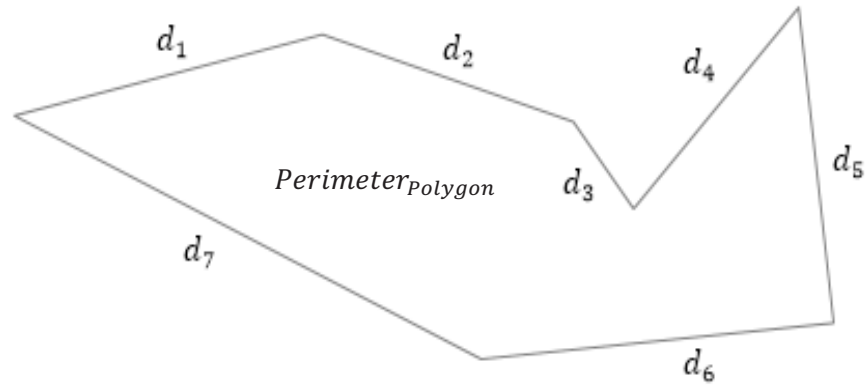


Figure 19: Example of perimeter of a polygon

This is achieved by finding the sum of the distance between the connections of each polygon, using Equation 3.

$$perimeter = d_{total} = \sum_{i=1}^n \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad [3]$$

where $i = 0, \dots, n - 1$, and $n = \# \text{ of points}$

Equation 3 also assumes that the data array contains the first and last data point information to complete the shape distance information.

4.4.2.2 Area

The next characteristic used for comparison is the area of a polygon, shown in Figure 20.

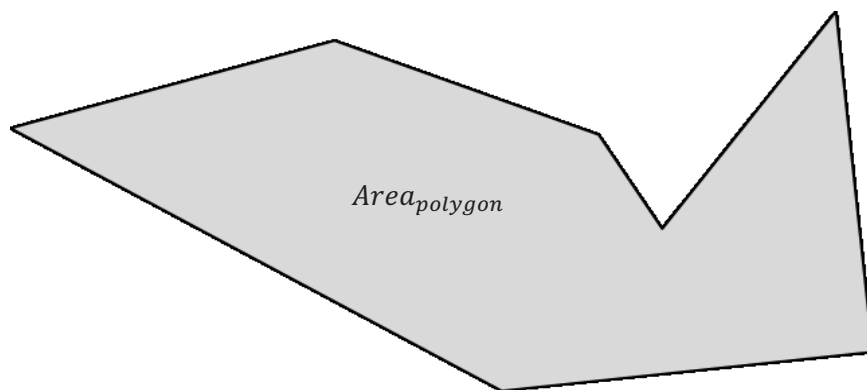


Figure 20: Example of area of a polygon

The area of this polygon is determined by creating vectors out of the points in a set direction along polygon [42]. Half of the cross product is used to find the area between the vectors [42]. These are aggregated until the polygon has been completed [42]. This logic is shown in Figure 21 and Equation 4 [42].

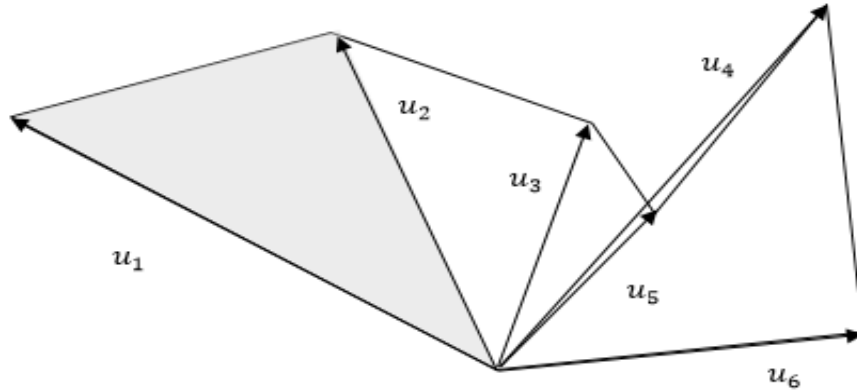


Figure 21: How to determine the area of a polygon

$$Area = 1/2 \left(\begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + \begin{vmatrix} x_2 & x_3 \\ y_2 & y_3 \end{vmatrix} + \dots + \begin{vmatrix} x_n & x_1 \\ y_n & y_1 \end{vmatrix} \right) \quad [4]$$

In the algorithm, this process was expedited by using the *polyarea* function within MATLAB. *Polybool* is a similar function in MATLAB that finds the overlapping area of two polygons, by determining where the polygons intersect and then perform the same logic as above for only the overlapping polygon. This function is utilized in the particle swarm maximization.

4.4.2.3 Compactness

The third characteristic used to compare the MFGC and the CADC polygons is compactness. Compactness is the ratio of the area of the image object to the area of a circle with the same perimeter, shown in Equation 5 [43].

$$Compactness = 4\pi Area / Perimeter \quad [5]$$

In other words, this characteristic measures how close an object is to a solid circle, which would have the maximum value of 1. This can be seen in Figure 22.

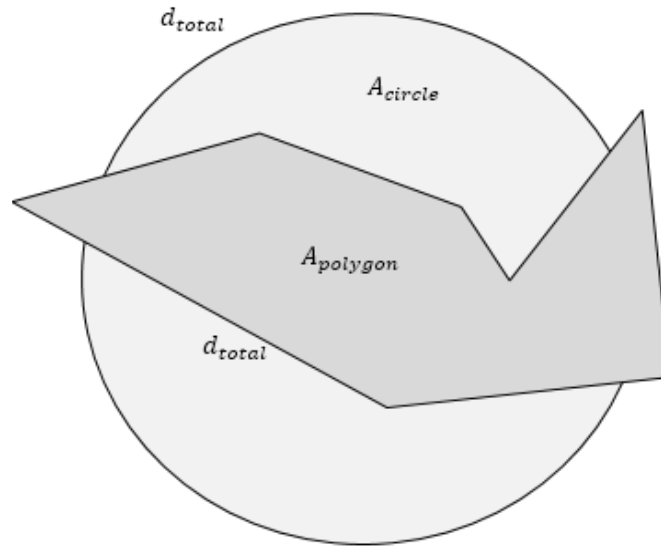


Figure 22: Example of compactness of a polygon

4.4.2.4 Convexity

The next shape characteristic used is convexity or comparison of the MFGC or CAD/C polygon to the convex hull of the same polygon. It is calculated from the ratio of the perimeter of the object's convex hull to the object's perimeter shown in Equation 6 [43].

$$Convexity = \frac{Perimeter_{Convex\ Hull}}{Perimeter} \quad [6]$$

This is to determine how concave an object is or how jagged the boundary of a shape is. This is seen at the top of the shape in Figure 23.

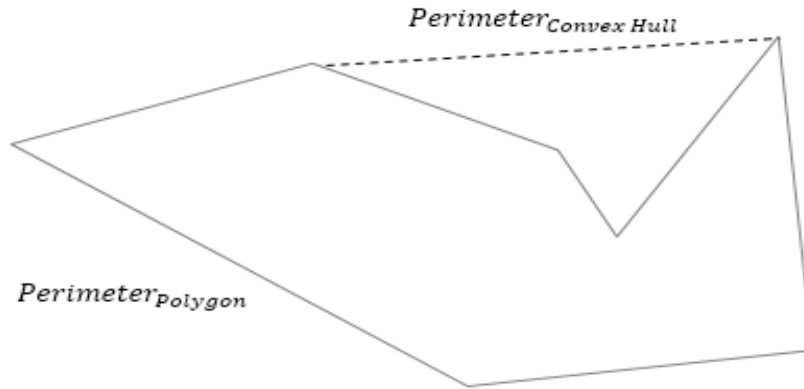


Figure 23: Example of convexity of a polygon

4.4.2.5 Solidity

The final characteristic used to compare the MFGC and the CADC polygons is solidity. It is obtained as the ratio of the area of an image object and the area of the object's convex hull, which is demonstrated in Equation 7 [44].

$$Solidity = \frac{Area}{Area_{Convex\ Hull}} \quad [7]$$

This conceptually is the density of the polygon or how solid is it. A polygon with several indentations or holes will have a small solidity value; this is shown in Figure 24.

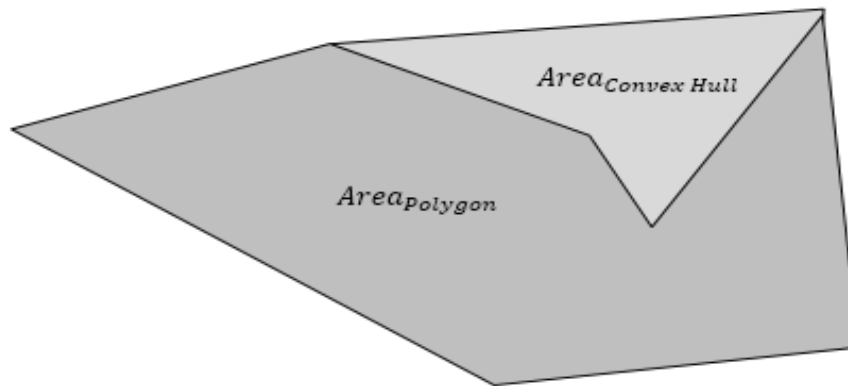


Figure 24: Example of solidity of a polygon

4.4.3 Shape Characteristic Error Calculation

To calculate the error between the ideal CADC polygon and the created MFG polygon from liaison sequencing algorithm, Equation 8 is utilized.

$$Error = \sqrt{(P_{MFGC} - P_{CADC})^2 + (A_{MFGC} - A_{CADC})^2 + (Cp_{MFGC} - Cp_{CADC})^2 + (Cv_{MFGC} - Cv_{CADC})^2 + (S_{MFGC} - S_{CADC})^2} \quad [8]$$

Equation 8 compares the properties, area, perimeter etc. of the MFGC and the CADC polygons, in five dimensional space. The origin of this five dimensional space, or goal, is the properties of the CADC polygon. The error calculation is the Euclidean distance from the origin to the MFGC point in the five dimensional space. Equation 8 is minimized by implementing different MFGC polygons via random reordering in liaison sequencing. The smallest error, calculated out of the possible liaison sequencing of MFGC, determines the most similar MFGC polygon to the CADC polygon. In this research, it was determined that weights given to certain characteristics had no benefit on the determination of the most similar MFGC polygon.

4.5 Particle Swarm Optimization

Once the liaisons of the MFGC and the CADC polygons have been determined, the last part of the algorithm rotates and translates the MFGC so as to maximize the shared area between the two polygons. An example of a step in the maximization of the shared area is shown in Figure 25.

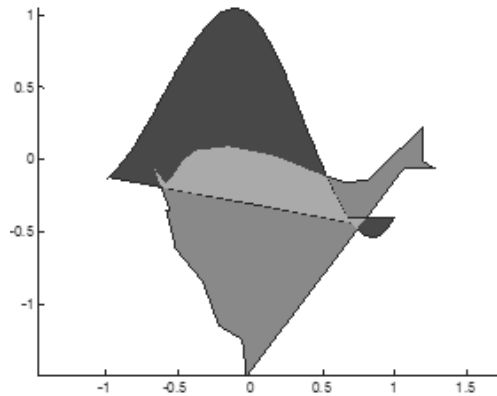


Figure 25: The shared area between the MFGC and CADC polygons

When the shared area is maximized, the curves should be in the optimal position for comparison. To calculate the shared area, light gray in Figure 25, the same logic is used that was discussed in Section 4.4.2.2.

For the maximization of the shared area method, a mathematical optimization technique will be needed. Due to factors, such as speed of algorithm, complexity of the objective function, and the sizable search of possible solutions, a heuristic technique is preferable, such as particle swarm optimization (PSO) [45]. PSO is a population-based stochastic algorithm for continuous optimization, inspired by the social interaction behavior of birds flocking and fish schooling [46] [47]. Developed by Kennedy and Eberhart in 1995, PSO creates a search space full of particles; each particle is determined to have a fitness value base on what is being evaluated; and the iterations continue, by giving particles a velocity based on the global best location, until a criteria is reached [46] [48]. This is explained further in the flowchart seen in Figure 26.

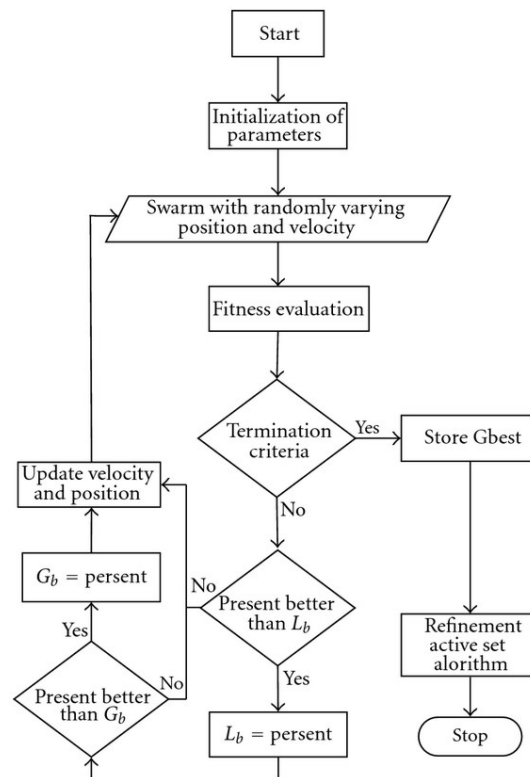


Figure 26: Flowchart of Particle Swarm Optimization [49]

To initialize the parameters, each particle is assigned a value of delta x, delta y, and theta used to translate the MFGC polygon onto the CAD/C polygon. The initial grid of particles are shown in Figure 27, ranging from -2 to 2 for both the delta x and delta y, $-\pi$ to π for theta.

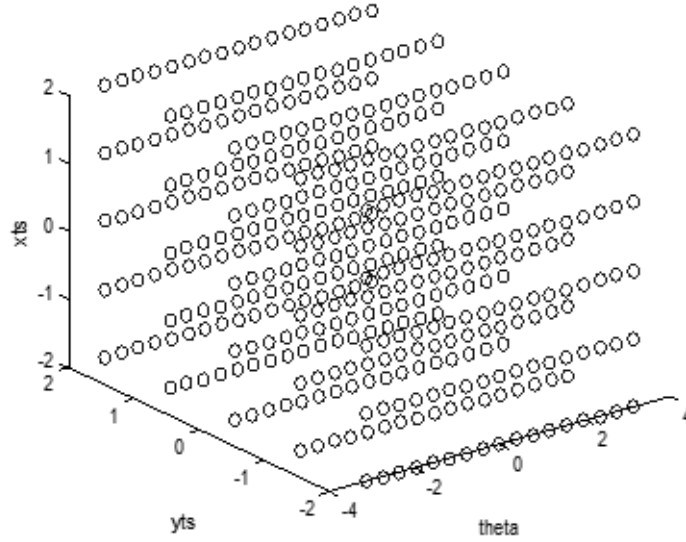


Figure 27: Dispersion of swarm particles before optimization

Each dot, in Figure 27, coordinates with a delta x, delta y, and theta value; each will calculate the shared area between the MFGC and the CAD/C, using the logic discussed in Figure 26. One dot will be determined to be the local and potentially global best value. This position will have the largest shared area between the MFGC and the CAD/C.

Based on the local and global best value, a new velocity and position is determined by using Equation 9 and Equation 10 [49].

$$v(t + 1) = \omega v(t) + C_1(p(t) - x(t)) + C_2(l(t) - x(t)) \quad [9]$$

$$x(t + 1) = x(t) + v(t + 1) \quad [10]$$

where v is velocity, x is position, p is global best position,
 l is local best position, ω is the inertia factor, C is the convergence factor

This velocity calculation draws other particles toward the local and global best by the convergence factor, while randomly misdirecting it by an inertia factor. Again, the inertia factor deter

the algorithm from producing a local maxima or minima. For the case of shared area maximization, the convergence factor is the same for the local and global convergence and is set to a gradient demonstrated in Table 2.

Table 2: Gradient increase of the convergence factor

Iteration Number	0	20	30	40	50	60	70	80	90	100	110	120
Convergence Factor, C	0.01	0.05	0.10	0.20	0.30	0.40	0.50	0.75	1.00	2.00	3.00	5.00
Inertia Factor, ω	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2

The gradient increase in the convergence factor is key due to the nature of the shared area maximization. The algorithm needs to vary significantly to explore the multitudes of possibilities in the theta direction and slowly increase towards a solution. Another solution to this is to increase the number of particles along the theta direction. This, however, leads to an increase in run time per iteration, slowing the algorithm.

For shared area maximization, the termination criteria was two-fold. The first criteria was a maximum iteration limit, preventing overwork on the computer hardware if a solution was not determined. The second criteria is that the median of the shared area over a user-defined amount of iterations, typically 30, was within a certain factor of the global best position. This requires the algorithm to level out as the convergence factor increases. A final result of the swarm convergence is displayed in part (a) of Figure 28. The largest generated shared area for each iteration is also recorded in a bar graph in part (b) of Figure 28, which demonstrates how this second criteria works.

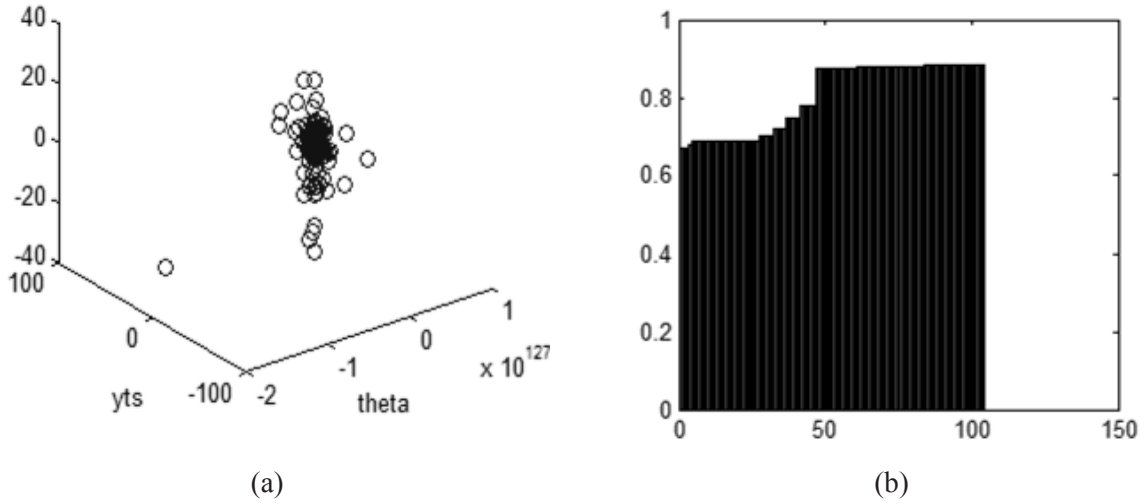


Figure 28: Swarm particle results and maximization over iterations

Overall, PSO searches a given window using particles to determine the location that maximizes the shared areas. It is able to manipulate the position of the MFGC in part (a) in Figure 29 and form the localization solution given in part (b) of Figure 29.

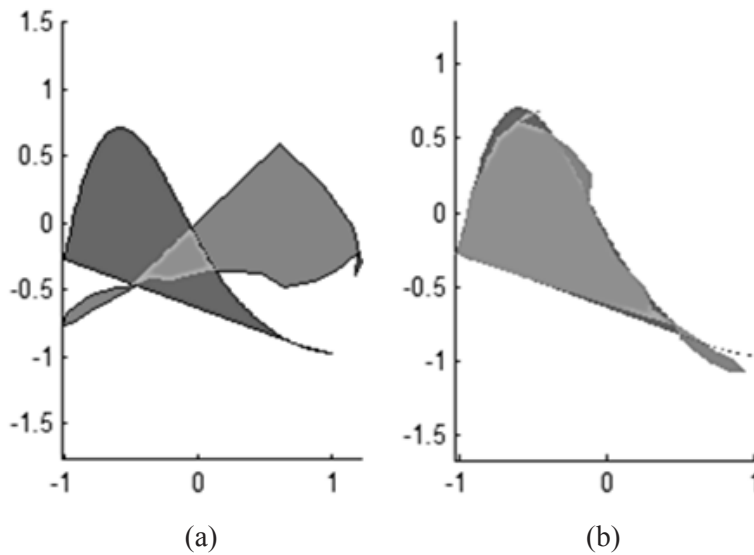


Figure 29: Example of a localized MFGC

Once in the final localized position of part (b) in Figure 29, the delta theta, delta x, and delta y can be recorded from the particle swarm cloud, part (a) of Figure 28, completing the needed information for a transformation matrix.

CHAPTER 5 ANALYSIS OF RESULTS

5.1 Methods of Analysis

This chapter will examine the collected data of twenty-five localizations of the shared area maximization method, found in Appendix A, by exploring five metrics: visual observations, the standard of the sum of squared errors, shared area, error area, and the transformation values. By examining each, it will be determined if shared area maximization is indeed a successful localization process.

Two of the metrics are self-explanatory, observations and the transformation values. The observation section is the visual inspection of the results and conclusions from this. The transformation values section will compare of the rotation and translation values generated in randomization and the values from the particle swarm optimization. However the rest of this section will describe the rest of the metrics.

The sum of squared errors is calculated from the Euclidean distance using each point on the MFGC to the closest point on the CADC as seen in part (a) of Figure 8 and is expressed in Equation 11.

$$Error_{SSE} = \sum_{m=1}^{C_1} \min_{1 \leq n \leq C_2} \left(\sqrt{(P_{mxMFGC} - P_{nxCADC})^2 + (P_{myMFGC} - P_{nyCADC})^2} \right) \quad [11]$$

where P is a point in a dataset, C_1 is the number of points in MFGC, C_2 is the number of points in CADC

Equation 11 is applied in three instances: the first instance is before randomization in the initial position; the second instance is after randomization of the MFGC; and the third instance is the result after shared area maximization method. For reference, the initial or original position is the result after curve creation, show in Figure 16. The MFGC has been created with error but not rotated or translated. By comparing the first and third instances, a determination can be made if the shared maximization method was able to meet or exceed what is needed to obtain the initial position.

The shared area between the MFGC and the CADC is demonstrated in Figure 30. Part (a) is the final result: the circles are MFGC before randomization, the dots are the MFGC result after the shared area method, and the xs are the CADC. Part (b) is the shared area before any randomization, the circles and the xs. Part (c) is after localization using the shared area method, the dots and the xs.

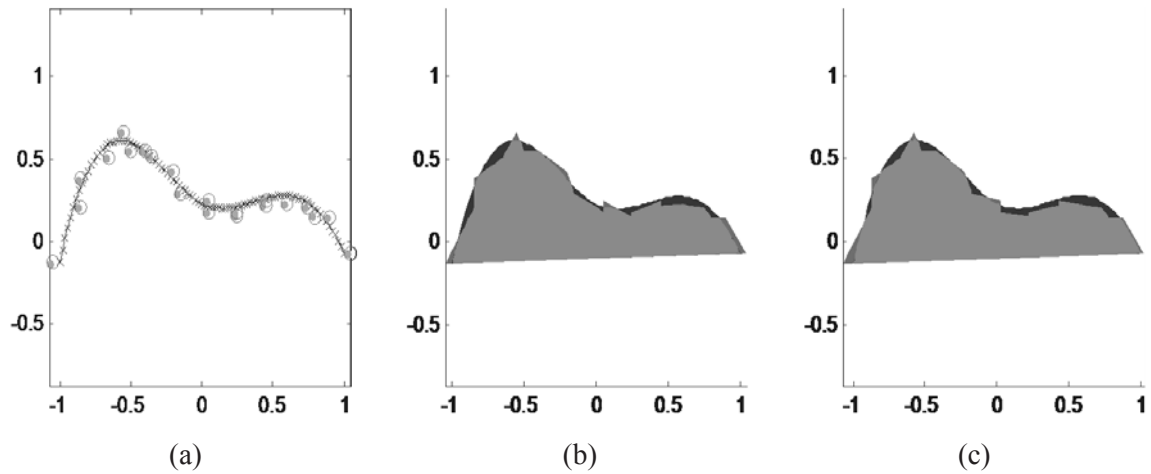


Figure 30: An example of shared area before and after localization

From Figure 30, again, it is visible that the polygonization around part (b) and (c) did completely capture the same shape as before. Therefore the shared area between the MFGC and the CADC will match the area in the initial position before randomization. Although, geometrically similar polygonization sometimes does not occur, as in Figure 31.

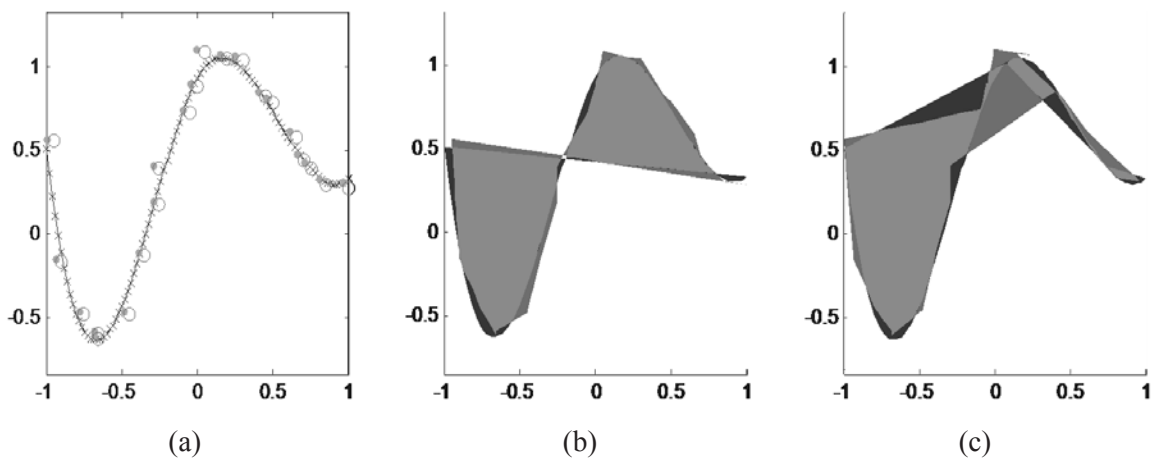


Figure 31: A shared area comparison with different polygonizations

As displayed in Figure 31, the polygonization of the two datasets are not similar and will have different shared area values. Therefore, the shared area becomes a significant indicator on how the algorithm performed and is used as one of the metrics. If the shared area after localization is greater than the area before randomization, then the algorithm produced a more optimal fit than the original error, meaning that it indeed performed localization.

The final metric, the error area, is shown in Figure 32. Again, part (a) is the final result; part (b) is the shared area before any randomization; part (c) is after localization using the shared area method.

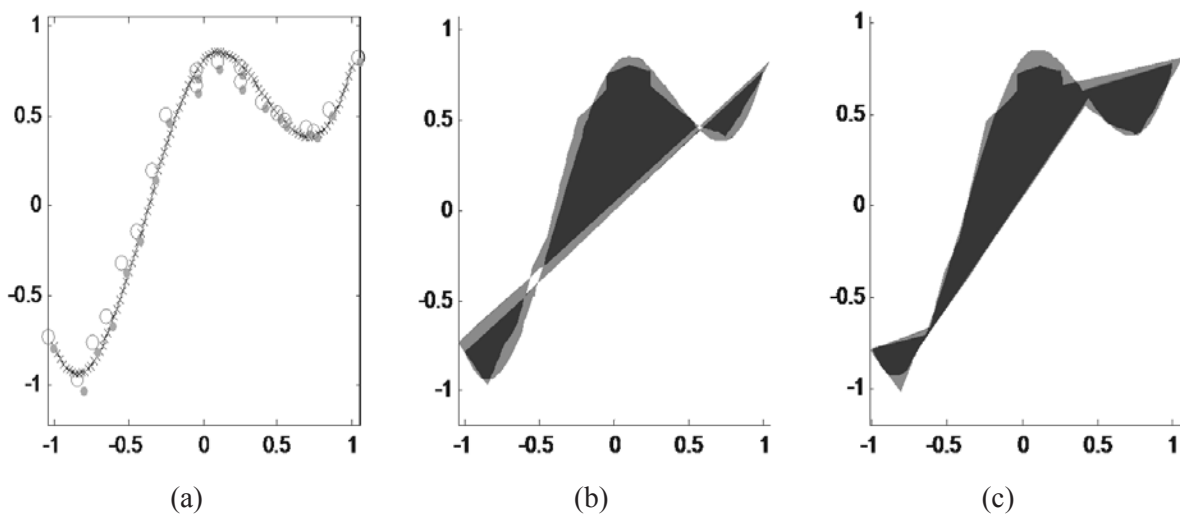


Figure 32: An example of error area before and after localization

In Figure 32, there is a decrease in error area from part (b) and (c). In part (c), the MFGC polygon aligns to the CADC closer during areas of constant slope, seen between -0.5 to 0.5 on the x axis. The significant decrease in error area after shared area maximization lends credibility to the method as a localization technique. Therefore, the analysis of the error area can be another indicator of how the algorithm performed localization.

5.2 Observations of the Results

An example of the results from the shared area maximization is shown in Figure 33.

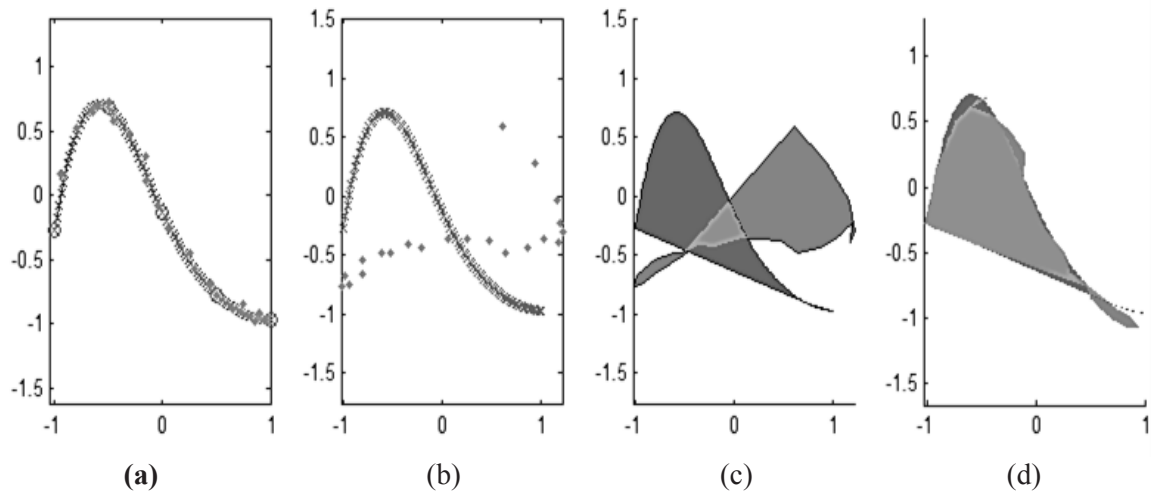


Figure 33: Result of localization via shared area maximization

In Figure 33, part (a) is the result of curve creation, modeling the CAD and simulation of MFGC. The outcome of randomization is shown in part (b), after being randomly translated and rotated. Part (c) demonstrates the polygonization of the data in part (b). The most similar MFGC polygon is generated from comparing characteristics from the CAD polygon. PSO is used to maximize the area shared between the two polygons, observed in Part (d), and complete the localization process. This completed product is shown in Figure 34, where the MFGC is aligned to the CAD.

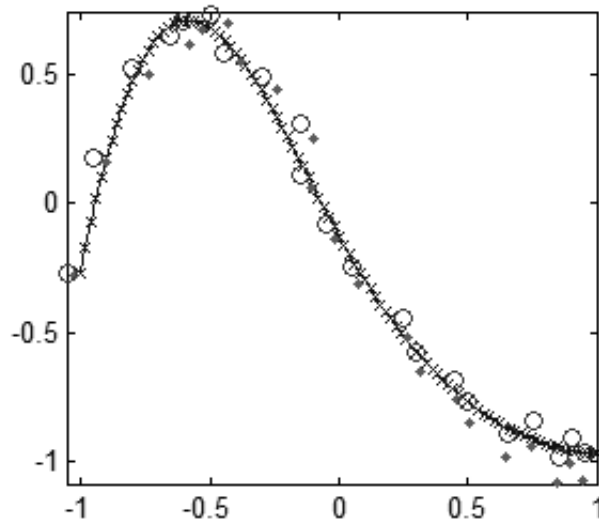


Figure 34: Final localization conclusion based on shared area maximization

Once more, the circle marks displayed in Figure 34 are generated data of the MFGC before randomization and the diamonds are the result of MFGC after localization process via shared area maximization. The circle marks in this manner are a target for the localization process. It is a gauge on how well the algorithm succeeded. However, it is not a perfect benchmark to access success or failure. Clearly from Figure 34, the diamond marks are not aligned perfectly with the circle marks. This method aligns the MFGC and the CADC based on global positioning and not individual points, meaning that outliers are less influential. Visually this can be ascertained by Figure 34, the lack of fit occurs at the peaks and valley features of the curve. Meanwhile, data with a more consistent slope is tightly aligned, more so than the original position of the MFGC.

However, this misalignment is not always the case, as shown in Figure 35.

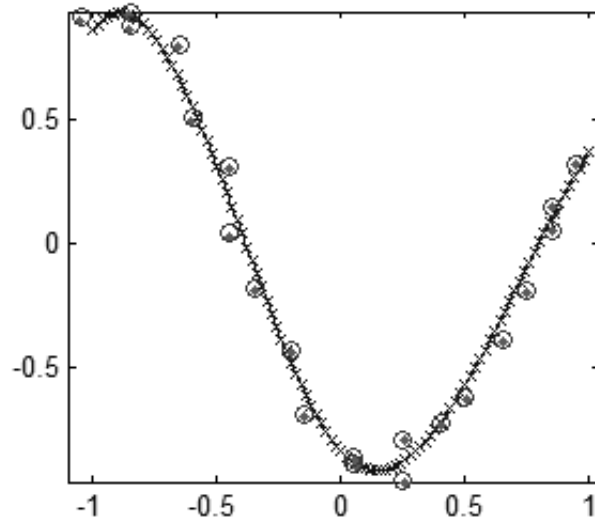


Figure 35: Shared area localization that aligns with initial MFGC

Figure 35 demonstrates a typical result from the shared area maximization, aligned directly with the originally generated MFGC. This result visually shows that constant slope areas, not the peaks and valleys features, that are aligned closer than the original MFGC generation. Why did this example align to the original MFGC than in Figure 34? To answer this, the results of each step in the shared extent algorithm needs to be examined in Figure 36.

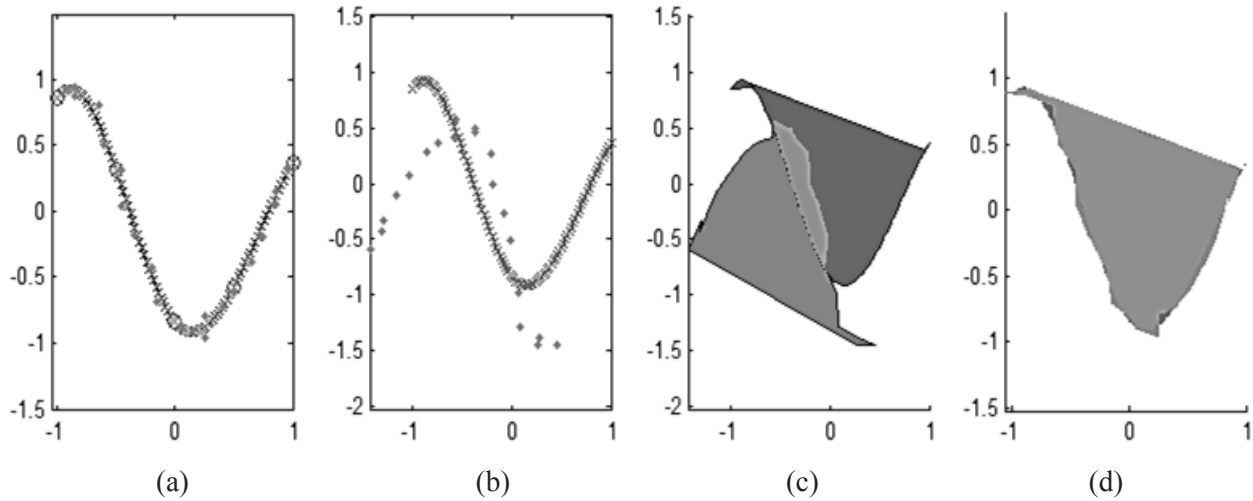


Figure 36: Steps of the shared area localization that aligns with initial MFGC

The difference between the two datasets alignments is visible in parts (c) and (d) in Figure 33 and Figure 36. The polygonization of the MFGC, in Figure 36, is a more accurate representation of the CAD/C than in Figure 33, mainly due to the top curve folding in on itself. Without an accurate polygon geometry, the area will not be able to be maximized fully and results will misalign. By observation alone, it is clear that polygonization is key to an accurate localization using the shared area method.

5.3 The Sum of Squared Errors Comparison

Again, the sum of squared errors is applied in three instances: first in the initial position before randomization, for example part (a) of Figure 36; second in the random position of the MFGC, shown in part (c) of Figure 36; and third to the MFGC after shared area maximization method, part (d) of Figure 36. This is significant to determine if the sum of squared errors is minimized and is presented in Figure 37.

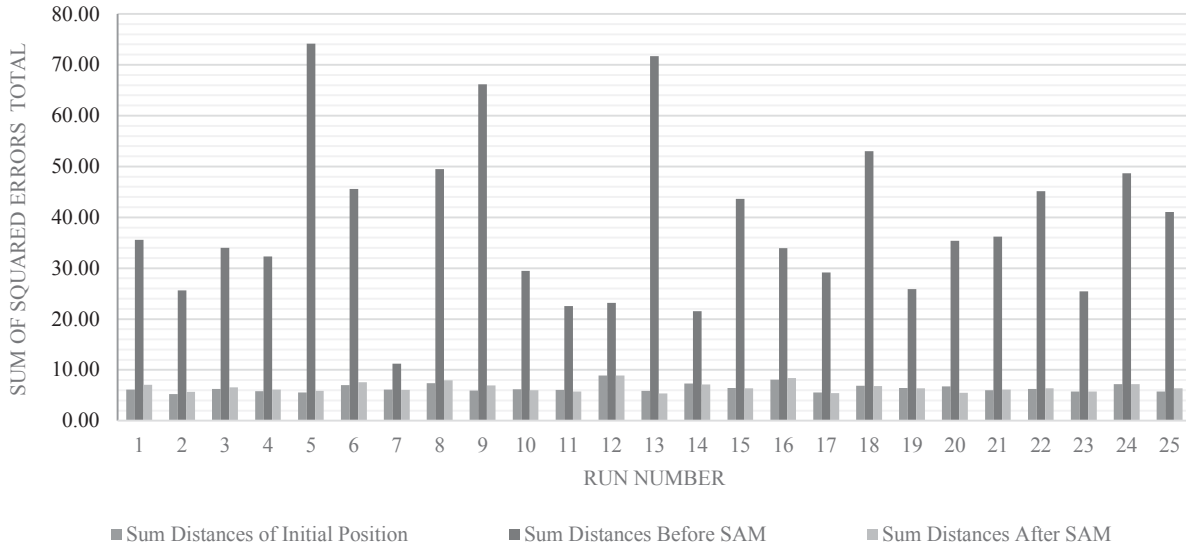


Figure 37: Sum of squared errors between the MFGC and CADC in initial position, before and after shared area maximization

From Figure 37, a consistent minimization of the sum of distances is exhibited. The percentage decrease of shared area maximization, as depicted in the second and third columns in Figure 37, was calculated to facilitate analysis and is displayed in Figure 38.

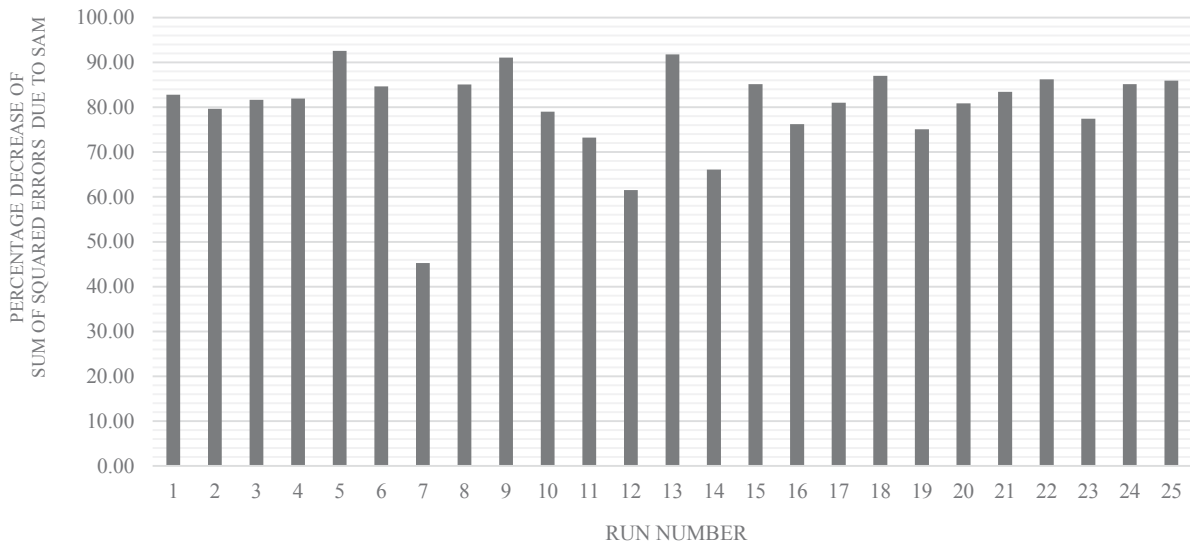


Figure 38: Percentage decrease of the sum of squared errors

Figure 38 demonstrates the reduction of the sum of distances after the shared area maximization, an average reduction of 79.8 percent. Thus, shared area maximization does localize, but how closely does it reduce to the original position? To continue, the percentage decrease from the randomized position to the initial position was calculated, to simulate if the curve localized from the random position to the initial position. This calculation, along with the data from Figure 38, is compared in Figure 39.

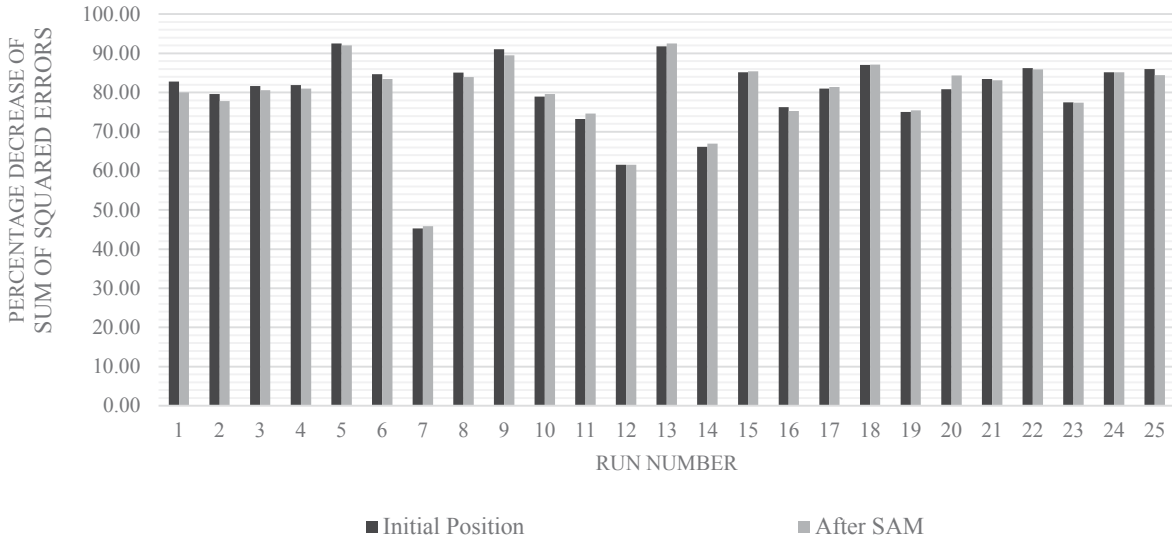


Figure 39: Comparison of percentage decrease between the initial position and after shared area maximization

Figure 39 exhibits that shared area maximization was able to obtain a similar percentage decrease of the sum of the squared error when compared to the percent decrease needed to position the randomized curve into the original position. To quantify this further, the difference of the percentage decreases, columns represented in Figure 39, is calculated and demonstrated in Figure 40.

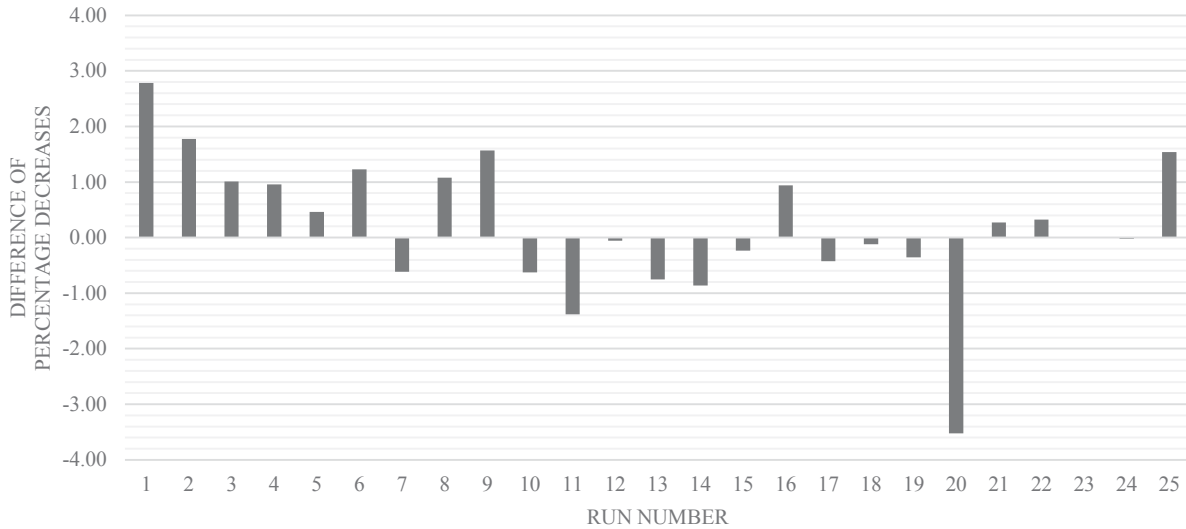


Figure 40: Difference between the percentage decreases needed to obtain initial position and to perform the shared area maximization

In Figure 40, the positive results show that the shared area maximization method was not able to decrease the sum of the squared errors to the same amount needed to achieve the initial position. The negative results occur when the shared area maximization was able to find a position with a smaller sum of the squared errors than the original position. The significance, in Figure 40, is that the shared area maximization method was able to perform the localization task within 3 percent of the desired target of the initial position. This verifies that shared area maximization does reduce the sum of squared errors without falling into local minima, proving it is a valid localization technique.

5.4 Shared Area Analysis

Again if the shared area after localization is greater than the area before randomization, then the algorithm produced a better fit than the original error, meaning that localization was indeed performed. The shared area before and after is visible in Figure 41.

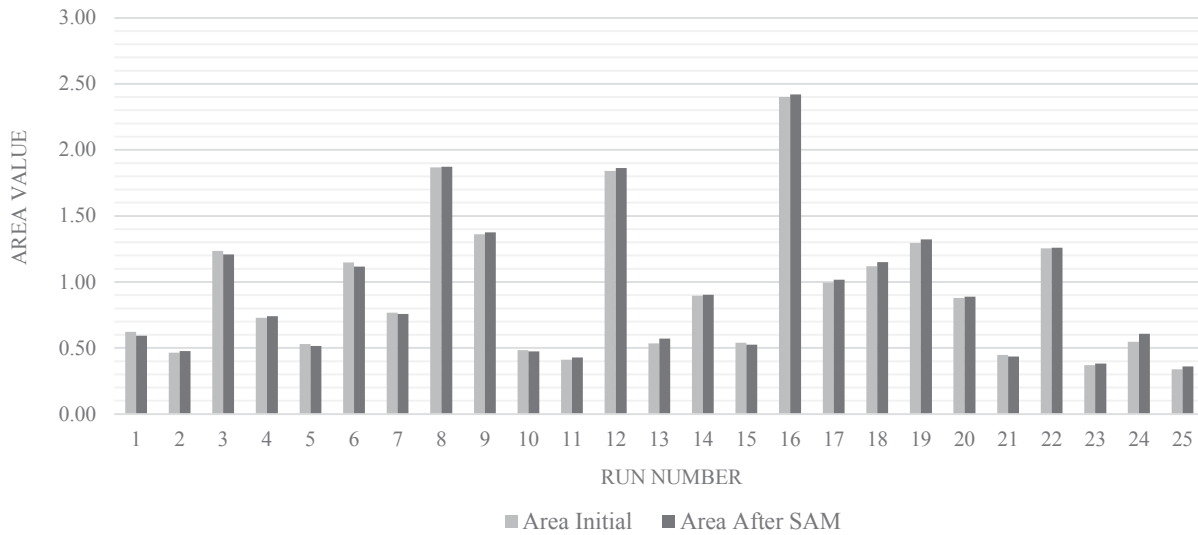


Figure 41: Shared area initially and after shared area maximization

To better demonstrate the change of the shared area after the algorithm, Figure 42 takes the difference between the two columns, shown in Figure 41, divided by the original shared area, creating a percentage increase or decrease. A positive result displays the improvement of shared area after the algorithm and negative indicates that the original area was not achievable.

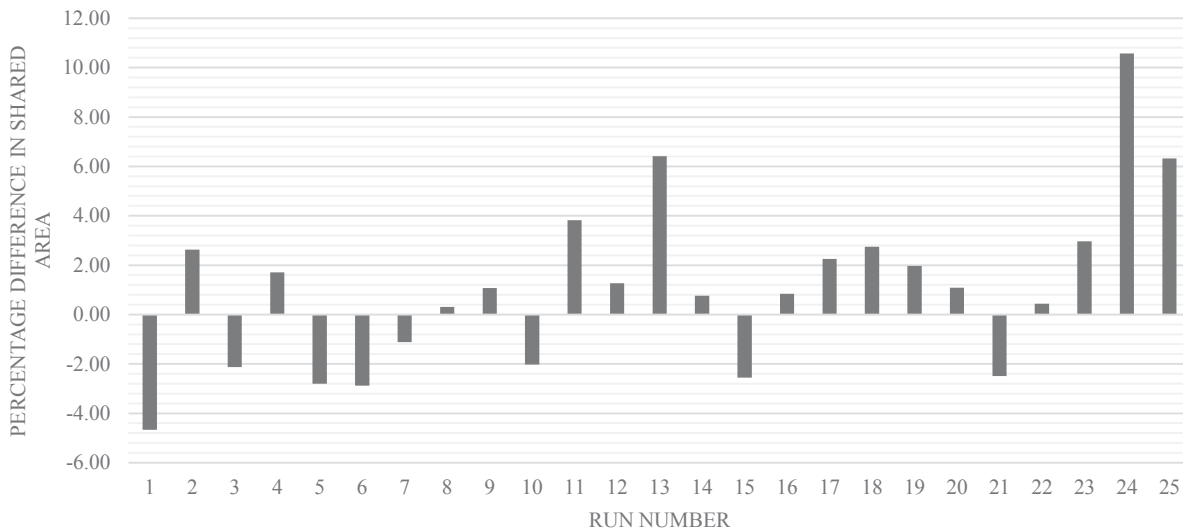


Figure 42: Percentage difference of the shared area initially and after shared area maximization

From Figure 42, one result significantly fails to achieve the same shared area, Run One. The rest were able to reach at least ninety-six percent of the original shared area or more. What occurred in this run to make it perform improperly? The results of Run One is presented in Figure 43. Again, part (a) is the final result; part (b) is the shared area before any randomization; part (c) is after localization using the shared extent method.

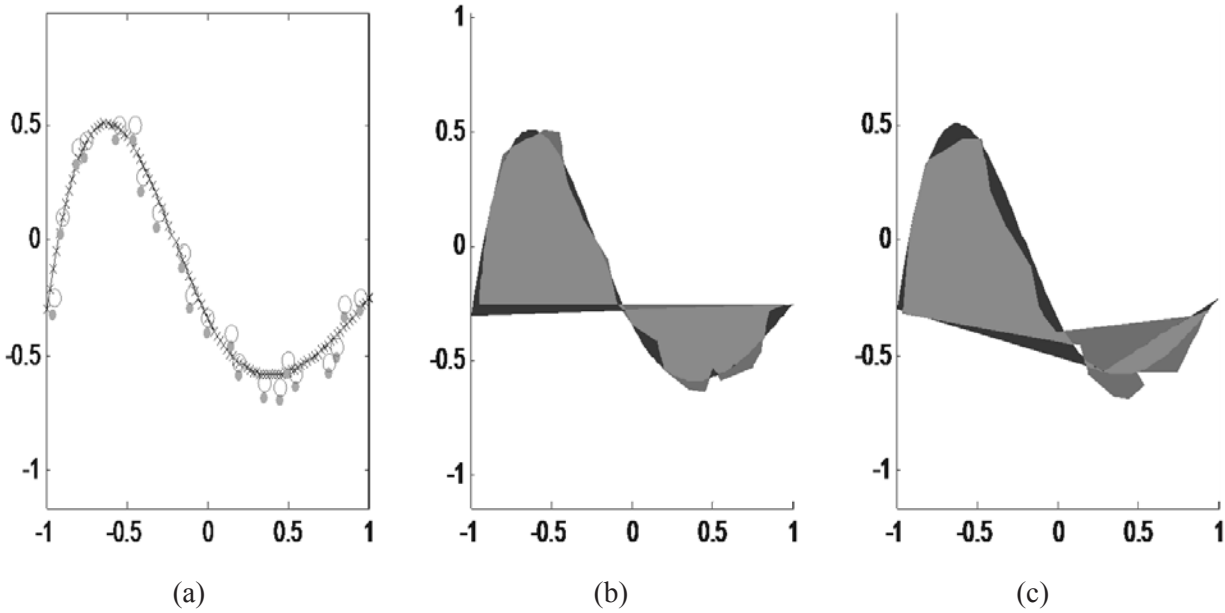


Figure 43: Steps of the shared area result for Run One

In parts (b) and (c) of Figure 43, the geometries of the MFGC and CADC polygons have some significant differences. In part (c), the CADC polygon changed to allow for more area to aid in comparison.

However the polygonization was not able to match to the new shape. This could be due to an insufficient amount of points to perform the narrow necking seen between 0 and 0.5 on the x axis. Or the polygonization may need more iterations to perform correctly, since the number of possible connections is in the range of 10^{18} power. The iterations was capped in order to expedite the algorithm, but possibly a heuristic could be added to guide the connection selected during polygonization.

Overall, the MFGC polygon, in part (c) of Figure 43, appears to match the CADC polygon of part (b) rather than its own. These inaccuracies in geometry between the MFGC and the CADC polygons

results in a shared area that is significantly smaller than the original shared area, explaining the error in Figure 42. Regardless of the geometric differences, Run One was still able to roughly localize, shown in part (a) of Figure 43, proving that the idea of shared area maximization is indeed valid. But without geometric similarity of the MFGC and CADC polygons, fine attunement is impossible. If the geometric properties were able to ascertain an enhanced polygonization, by either adding more data points in the MFGC or optimizing that section of the algorithm, the difference of shared area before and after localization could be an on-off control in future research to determine if localization occurred. This would be key in establishing reliability in results and the method before moving to unknown fitting parts.

5.5 Error Area Analysis

Section 5.4 discussed the shared area before and after localization; this section will discuss the inverse or the area that is not shared before and after localization, error area. The error area before and after localization for the twenty five runs is located in Figure 44.

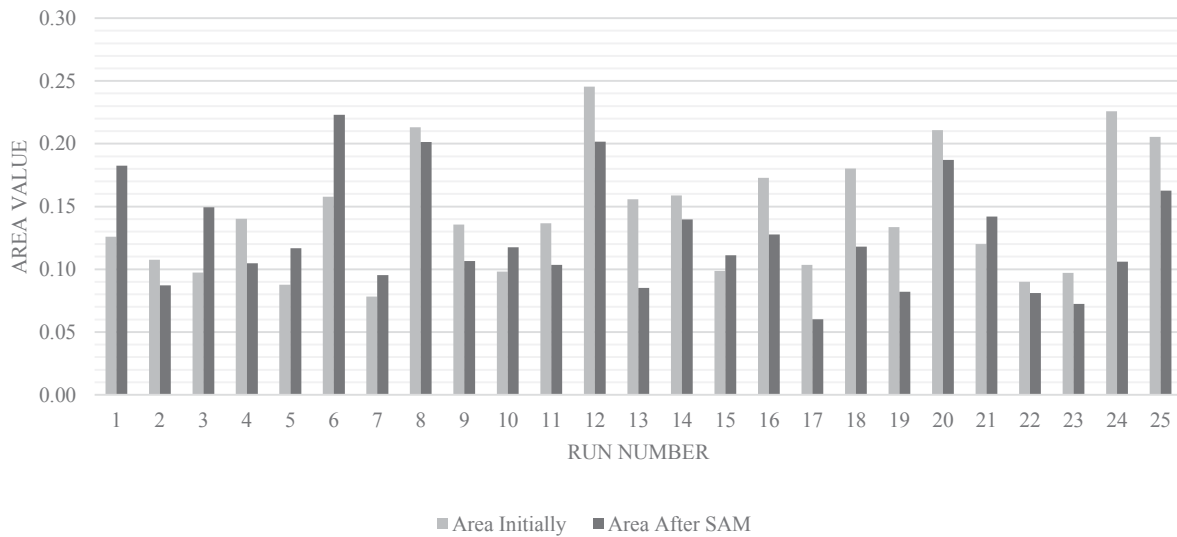


Figure 44: Error area before and after localization

Again, the percentage difference of Figure 44 is calculated and displayed in Figure 45.

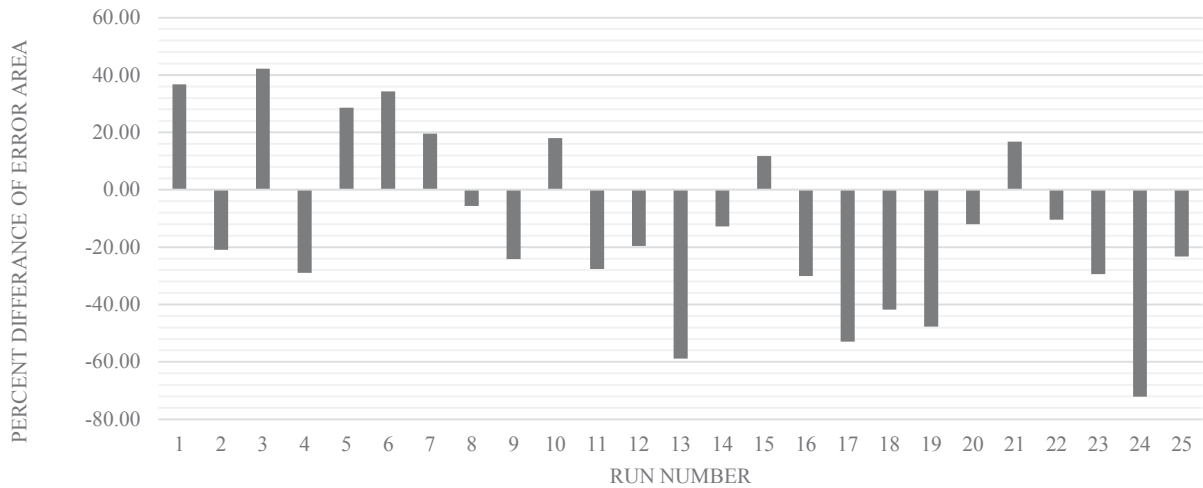


Figure 45: Percentage difference of error area before and after localization

From Figure 45, four of the twenty-five runs significantly increased the amount of error area after localization, Runs One, Three, Five, and Six. Otherwise, the error area from shared area maximization was able to achieve under a twenty percent increase of the error area before randomization. In fact, in Figure 45, most runs are negative, concluding that the error area was reduced from the original conditions. This is consistent with Run One in Figure 43; the lack of geometric fit creates an excess in error area. Is this the same for Run Three, displayed in Figure 46, with one of the highest error area?

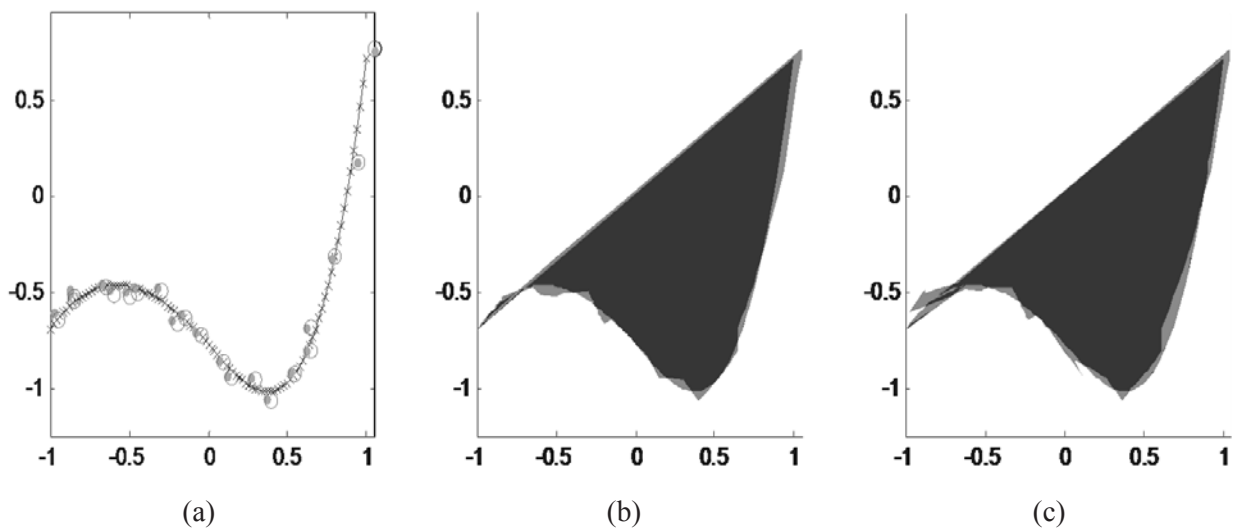


Figure 46: Steps of the shared area result for Run Three

In Figure 46, polygonization seems to have performed, and the shared area maximization localized. There is a small area that is not shared between -1 and -0.5 on the x axis in part (c). This however could have been sacrificed for the closer alignment of the top edge between -0.5 to 1 on the x axis. As part of the nature of shared area maximization, larger areas are going to supersede the alignment of small areas. However Run Three, Figure 46, does not seem to have the geometric issues discussed in Run One, Figure 43, yet Run Three was deemed more significant than Run One. This is due to the nature of percent difference. The amount of error area is small in Run Three, therefore any slight difference in error area will register as more extreme than a curve that contains an inherently large error area due to noisy data.

Stated before, the analysis of the shared area before and after localization could be an on-off control, in future research, to confirm if localization occurred. However, the error area analysis may be a more significant gauge on how the CAD/C and MFGC polygons fit together. However, this gauge is extremely sensitive and may be unreliable at inherently small error areas or smooth MFGC data. In future research and analysis, this error area could be minimized in a heuristic by changing the polygon geometries after initial localization to see if an improvement of the overall fit could occur, however this will create a longer runtime and increase complexity of the algorithm.

5.6 Comparison of Swarm Transformation Results to Randomization Values

The results of the transformation parameters—delta theta, delta x, and delta y values—from the swarm particles in Figure 28 were calculated and compared to the recorded random values used during randomization. The collection of these parameters are shown in Table 3.

Table 3: Transformation parameters from the twenty five trials

Run	θ Actual	θ SAM	θ Diff	X Actual	X SAM	X Diff	Y Actual	Y SAM	Y Diff
1	-40.000	-39.471	0.529	0.100	0.222	0.122	0.300	0.285	0.015
2	14.000	11.724	2.276	-0.500	-0.508	0.008	0.000	0.086	0.086
3	56.000	54.817	1.183	0.500	0.394	0.106	0.000	0.305	0.305
4	29.000	26.672	2.328	0.200	0.269	0.069	-0.200	-0.118	0.082
5	-91.000	-90.505	0.495	-0.350	-0.345	0.005	0.450	0.483	0.033
6	81.000	79.780	1.221	-0.200	-0.258	0.058	0.200	0.228	0.028
7	-10.000	-9.918	0.082	0.050	0.055	0.005	-0.100	-0.106	0.006
8	98.000	100.461	2.461	-0.150	-0.232	0.082	-0.200	-0.105	0.095
9	135.000	133.272	1.729	-0.300	-0.010	0.290	0.300	0.458	0.158
10	-38.000	-35.333	2.667	-0.250	-0.298	0.048	0.200	0.104	0.096
11	21.000	21.993	0.993	0.450	0.429	0.021	0.100	0.287	0.187
12	-27.000	-22.591	4.409	-0.300	-0.230	0.070	0.100	0.230	0.130
13	-95.000	-93.832	1.168	0.150	0.218	0.068	0.400	0.375	0.025
14	-29.000	-28.217	0.783	0.350	0.254	0.096	-0.200	-0.367	0.167
15	68.000	67.596	0.404	0.100	0.149	0.049	-0.100	-0.060	0.040
16	-25.000	-26.503	1.503	-0.500	-0.624	0.124	-0.400	-0.110	0.290
17	6.000	4.587	1.413	-0.300	-0.308	0.008	0.350	0.322	0.028
18	143.000	141.384	1.617	0.050	0.034	0.016	-0.200	-0.228	0.028
19	-37.000	-37.669	0.669	0.500	0.364	0.136	-0.100	-0.326	0.226
20	12.000	14.911	2.911	-0.350	-0.156	0.194	-0.500	-0.579	0.079
21	93.000	93.034	0.034	-0.350	-0.337	0.013	-0.300	-0.320	0.020
22	63.000	62.659	0.341	0.050	0.242	0.192	0.450	0.369	0.081
23	-30.000	-30.911	0.911	0.450	0.459	0.009	0.050	0.214	0.164
24	58.000	59.010	1.010	-0.150	-0.139	0.011	-0.100	-0.016	0.084
25	-76.000	-72.333	3.667	-0.250	-0.293	0.043	-0.250	-0.190	0.060

To facilitate understanding of Table 3, calculated and actual values of each parameter are compared in bar graphs, exhibited in Figure 47, Figure 48, and Figure 49.

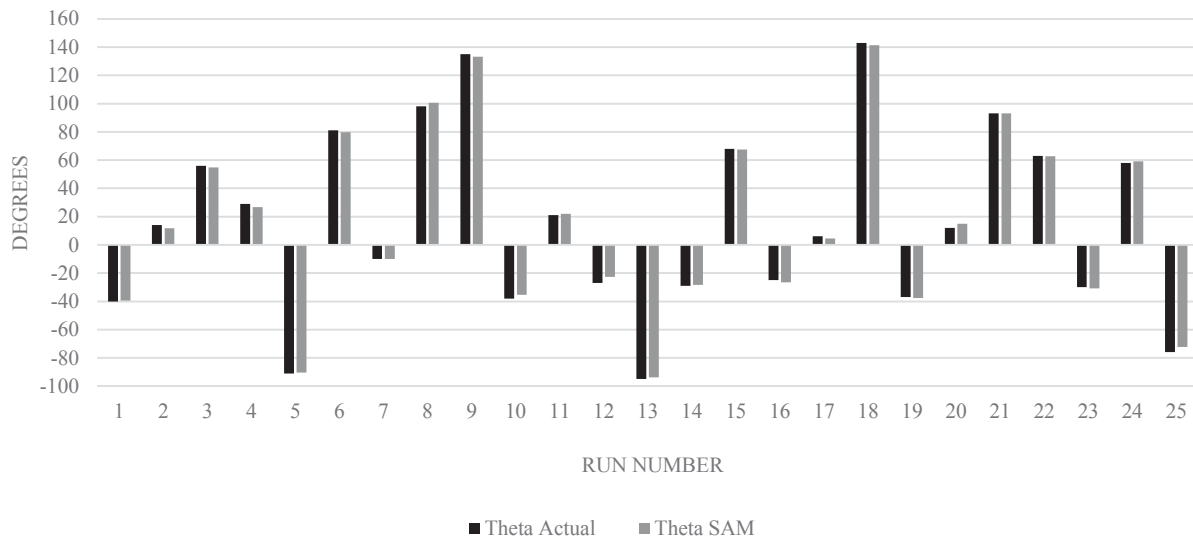


Figure 47: The comparison of the actual and calculated theta values

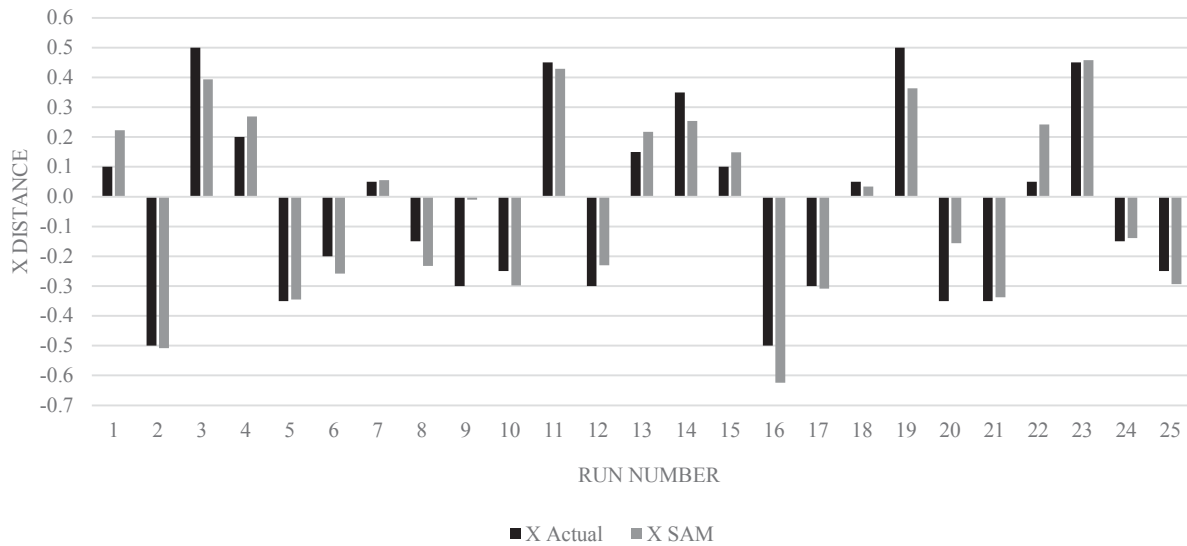


Figure 48: The comparison of the actual and calculated delta x values

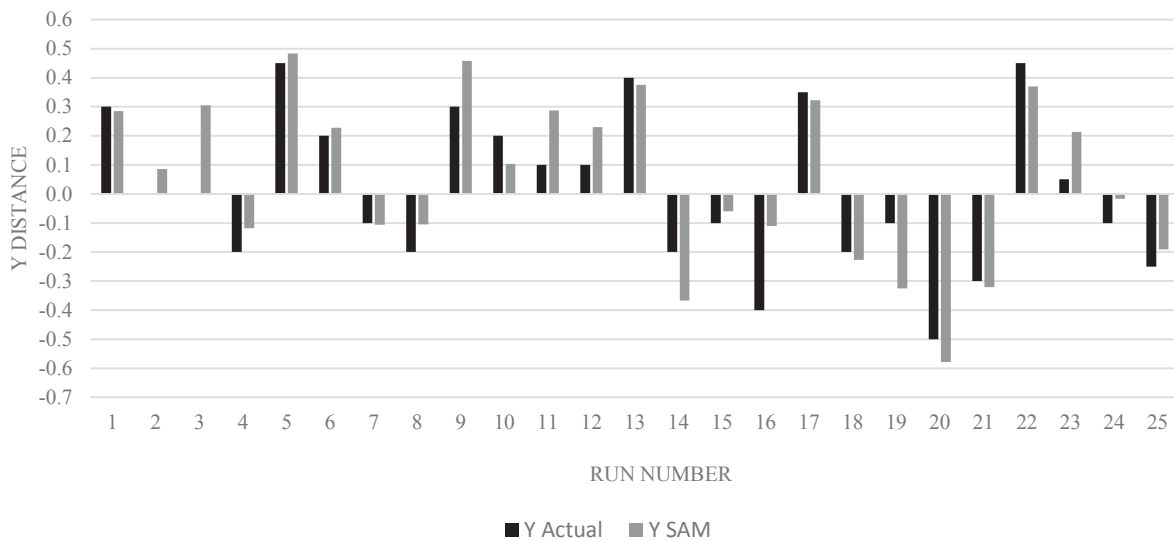


Figure 49: The comparison of the actual and calculated delta y values

In Figure 47, theta values determined by the shared area maximization method were able to match within at least five degrees and with an average error of 1.5 degrees. This is an error of 0.41 percent, proving again the shared area maximization localized.

However the delta x and y values, in Figure 48 and Figure 49, struggled to achieve the same type of accuracy. The delta x value has an average error value of 6.56 percent; delta y has 9.46 percent. This

change in error and lower performance may be due to less swarm particles in those directions. Theta requires a smaller deviation between particles to allow for rotation possibilities and ensure localization. The delta x and y deviations of the swarm are usually sacrificed for run time of an algorithm. Additionally, the shared area maximization parameters are being compared to the typically unknown transformation values from the noise data. The noise data position was not optimized to begin with, ergo there may be a more optimal position for the noise data than the original position. This slight difference would affect the average delta x and delta y values, more than the overall angle of the curve.

CHAPTER 6 CONCLUSION

6.1 Overview of Thesis

Overall, this thesis proposed a methodology that automatically aligns the coordinate systems of free-form CAD models to collected manufactured measurements by maximizing the shared area in this two dimensional case. Data from the manufactured surface and design surface were polygonized and compared until geometrically similar. The overlapping or intersecting area was calculated and maximized using a heuristic approach, particle swarm optimization. At the maximum shared area, the two coordinate systems were aligned. This section will explore the conclusions during the analysis of the result, the limitations to the method, and how the algorithm can be extrapolated to three dimensions.

6.2 Deductions from Results

From the results in Chapter 5, the following can be concluded. The sum of squared errors were reduced and avoided local minima, as demonstrated in Figure 37 through Figure 40. The shared area maximization method was able to reduce the sum of the errors within plus or minus three percent of the values given in the initial position, proving this method is a localization technique. The method is resistant to outliers in the dataset of MFGC curve. This is demonstrated by the tight alignment of consistent sloped areas while not necessarily to peaks and valley areas, displayed in Figure 34 and Figure 35. The theta transformation parameter determined by shared area maximization, when compared to the recorded value from randomization, was accurate, with an average error of 0.41. The delta x and y values achieved an average error value of 6.56 and 9.46 percent. While close to true delta x and y values, the error may be due to less swarm particles in the x and y directions, reduced to decrease run time of the algorithm. This should be remedied by expanding the amount of swarm particles in these directions. Additionally, these calculated transformation parameters from shared area maximization are being compared to the transformation parameters to achieve the initial position of the MFGC, which was not

localized or fit onto the CAD. This could affect the error in the x and y directions, more than the angle of the curve.

The majority of the results localized. However, four of the twenty-five runs significantly increased the amount of error area after localization. The analysis of the error area before and after localization was determined to be a gauge of how similar the polygon geometry is between the CAD and the MFGC. This could be utilized in future analysis to improve the overall localization. The difference between the shared area of before and after localization was also determined to be an indicator of whether localization occur. If the shared area is significantly smaller than before localization, it implies one of two malfunctions: the MFGC and the CAD polygons did not maximize correctly giving a false solution or that the MFGC and the CAD polygons are not geometrically similar. In the twenty-five test cases, the error was revealed to be due to the second reason. This is mainly due to limited amount of shape parameters that do not skew during rotation and challenges of the geometry of the curves. This demonstrates that polygonization is the key to an accurate localization using the shared area method and is visible in parts (b) and (c) in Figure 31, Figure 36, and Figure 43.

6.3 Possible Limitations of the Algorithm

The following were assumed or determined within this research which may lead to unforeseen difficulties in the future. One reference frame needs to undergo rotation and translation, the manufacturing data points onto the CAD data points. No one point can become an axis or origin. This origin point could contain error or be an outlier. Heuristics are favored, because a simple linear optimization is not possible. Local minima solutions can and will be given by heuristic methods.

During algorithm development to examine shared area maximization, a few geometric challenges of free-form curves were discovered. Extreme shifts in slope in a small time step, drastic wiggling, will not localize correctly. Mirror images or uniform sloped curves are difficult to orient as well. Both of these complications will be discussed further in the remainder of this section.

Typically for extreme shifts in slope, the algorithm will perform as displayed in Figure 50.

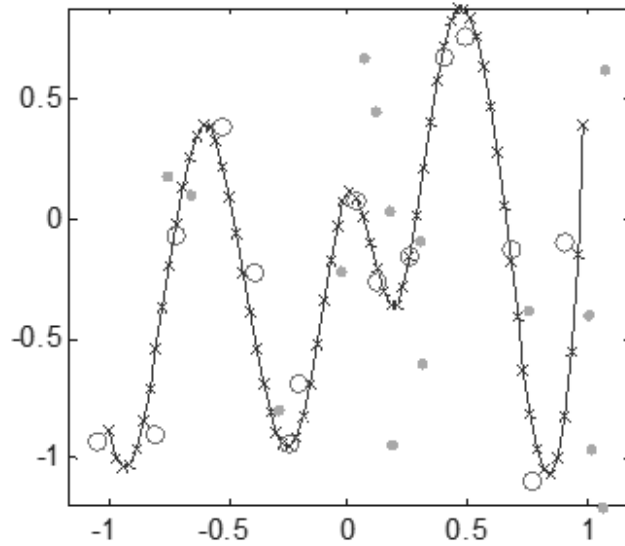


Figure 50: Example of failure due to extreme shifts in slope

As observed in Figure 50, localization does not occur. To determine how the algorithm fails, Figure 51 shows the steps performed. Like in Figure 33 and Figure 36, part (a) is the curve generation; part (b) is the randomization; part (c) is the polygonization; and part (d) is the shared area maximization.

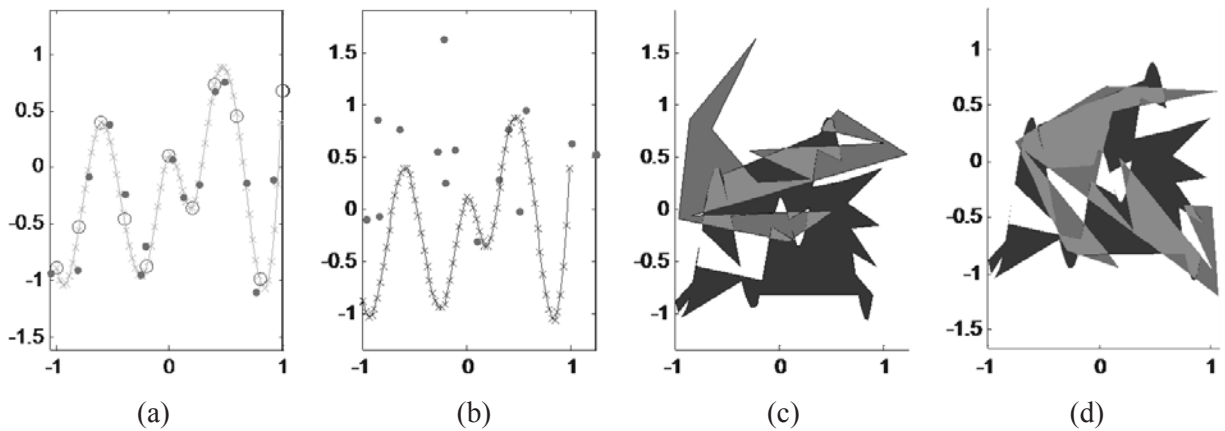


Figure 51: Steps of a failure example due to extreme shifts in slope

In part (c) of Figure 51, it is clear to see that the polygonization did not determine a geometrically similar match between the MFGC and the CADC. Due to this, the shared area maximization was not able to

localize the curves. With an inadequate number of shape characteristics that do not skew, additional accuracy to the polygonization technique is limited.

However, more data points can be added to help distinguish differences in the shape characteristics. This will lead to longer run times as a concession. With more data points, a successful localization is more prone to occur, as shown in Figure 52 and Figure 53.

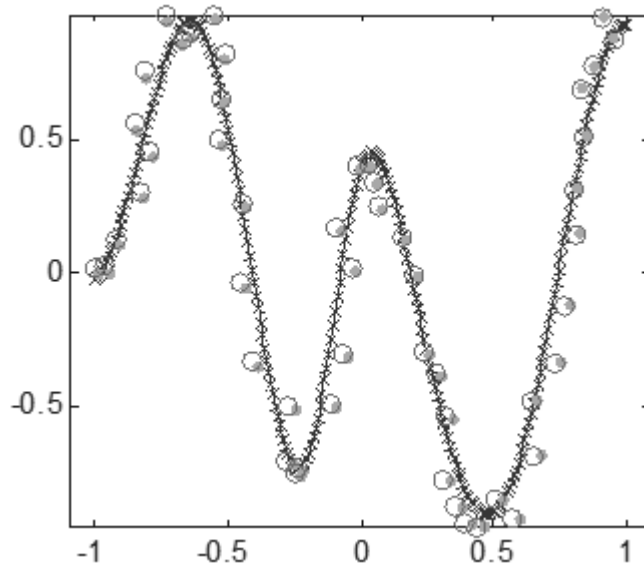


Figure 52: Example of a success concerning extreme shifts in slope

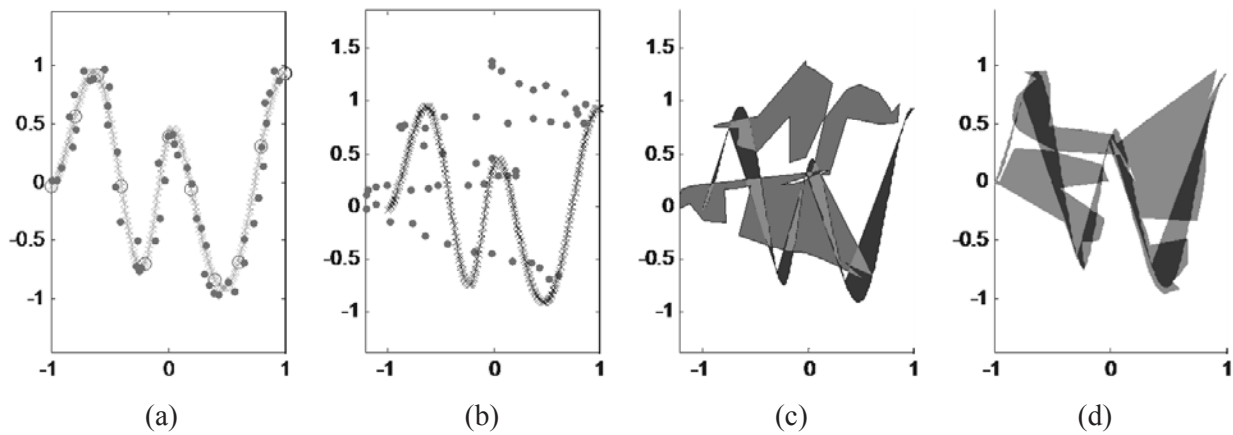


Figure 53: Steps of a successful example when extreme shifts in slope occur

Part (c), in Figure 53, correspondingly demonstrates the lack of polygonal fit on the two datasets. However, the fit was a close enough approximation to allow for shared area maximization. From this, it is clear more shape characteristics are needed that do not skew with rotation to develop the polygonization algorithm further.

The other conclusion discovered during programming of the algorithm is that mirrored curves may not localize. This is presented in Figure 54 and Figure 55.

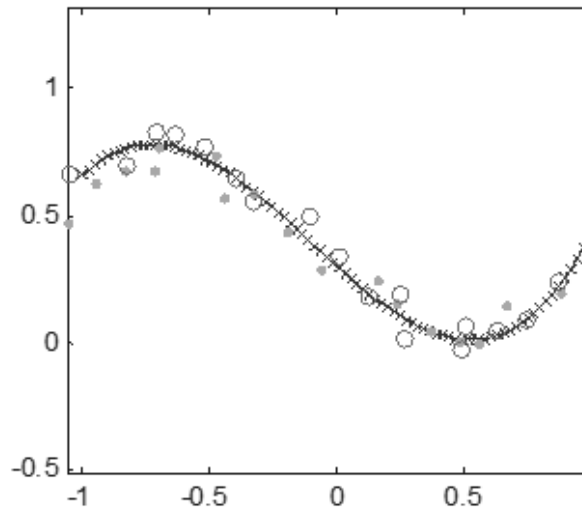


Figure 54: Example of failure due to mirrored curve

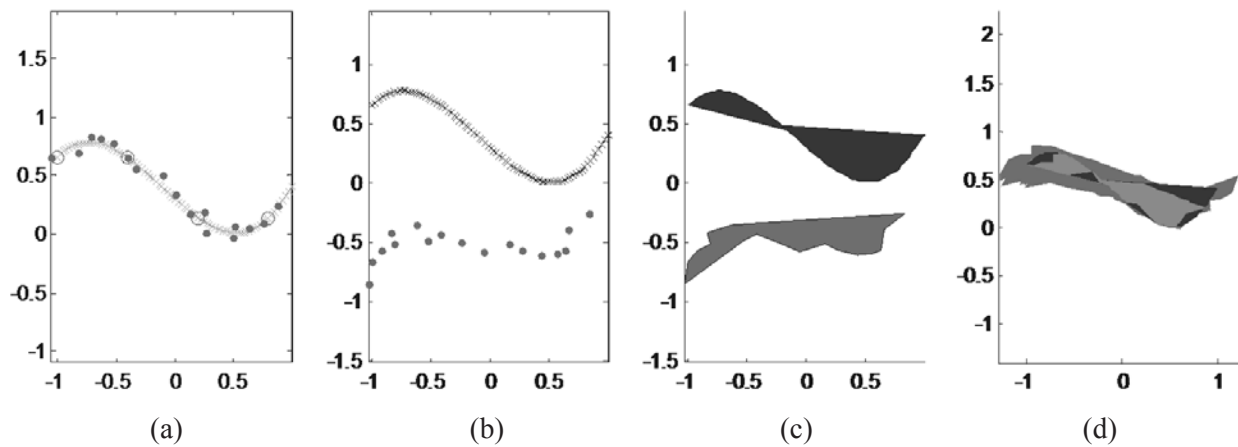


Figure 55: Steps of a failure example due to extreme shifts in slope

Figure 54 indicates that shared area maximization occurred; however the result needs to be rotated 180 degrees. To explain why this happened, part (c) of Figure 55 is significant. The most geometrically similar polygons occurs backwards on the MFGC due to the limited amount of points and error add to simulated manufacturing. For this case, more data points may increase likelihood of correct localization. Nevertheless, this type of curve is approaching the edge of the definition of freeform curves, and as such another method of localization may be more effective. Also, if three dimensions were utilized, this circumstance may no longer transpire.

6.4 Future Research

As the foundation of this work, only two dimensional free-form curves were examined in this thesis, reducing complexity and concluding that maximization of the shared area may perform localization. In future research, an additional third dimension will need to be accessed to study shared volume maximization and to render value to industry. The remainder of this section will discuss how this could be done and some of difficulties that may arise.

In order to add the third dimension, moving from a curve to a free-form surface, several aspects of the algorithm will need to be changed, however the most of the logic diagrams in Figure 13, Figure 18, and Figure 26 would remain similar. Curve creation and randomization, discussed in Figure 13, should be implemented with limited difficulty. The first challenge arises in the current polygonization section, logic in Figure 18. A triangular mesh of design surface and the manufactured part will need to be created and determined to be geometrically similar. For the CAD surface, this would be the simple utilization of the .stl format or other similar modified formats. However, for the manufactured part, the triangular mesh creation is a complex traveling salesman problem with three dimensional shape characteristics. Some of these three dimensional features can be extrapolated from the two dimensional versions, such as volume, surface area, sphericity, etc. Further research into particle analysis of materials and other fields dealing with shape morphology may be needed.

Additionally, the overlapping or intersecting volume of the meshes would be need to be calculated rapidly. This may be difficult to achieve with limited iterations or small run time. This is needed as the driving function for optimization. Particle swarm optimization could still be a valuable approach, displaying the delta x, y, and z values on one three dimensional graph and the degree rotations in another. If the overlapping volume can be calculated rapidly, the maximum shared volume should align the two coordinate systems of the CAD surface and the manufactured part with the minimal addition of errors during analysis.

REFERENCES

- [1] Y. Li and P. Gu, "Free-Form Surface Inspection Techniques State of the Art Review," *Computer-Aided Design*, vol. 36, pp. 1395-1417, 2004.
- [2] K. J. Gunnarsson and F. B. Prinz, "CAD Model-Based Localization of Parts in Manufacturing," *Computer*, vol. 20, no. 8, pp. 66-74, Aug. 1987.
- [3] G. R. Bertoline and E. Wiebe, *Fundamentals of Graphics Communication*, McGraw Hill, 2002.
- [4] C. H. Menq, H. Yau and G. Lai, "Automated Precision Measurement of Surface Profile in CAD-Directed Inspection," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 2, pp. 268-278, 1992.
- [5] S. Zlatanova, S. Pu and W. Bronsvoot, "Freeform Curves and Surfaces in DBMS: A Step Forward in Spatial Data Integration," *Archives of ISPRS*, vol. 34, no. 3A, pp. 407-412, 2006.
- [6] X. Huang, P. Gu and R. Zernike, "Localization and Comparison of Two Free-Form Surfaces," *Computer-Aided Design*, vol. 28, no. 12, pp. 1017-1022, December 1996.
- [7] M. Joshi, "A Heuristic Approach to Inspection of Sculptured Surfaces Using Data Localization," MS thesis, Dept. Mech. Eng., Univ. of Cincinnati, Cincinnati, OH, 2007.
- [8] K. Lee, S. Son and H. Park, "CAD-based Laser Scan Planning System for Complex Surfaces," *Euro-RP 2001*, 2001.
- [9] V. Mehrad, D. Xue and P. Gu, "Inspection of Freeform Surfaces Considering Uncertainties in Measurement, Localization and Surface Reconstruction," *Measurement Science and Technology*, vol. 24, 2013.
- [10] E. Savio, L. De Chiffre and R. Schmitt, "Metrology of Freeform Shaped Parts," *Annals of the CIRP*,

vol. 26, no. 2, pp. 810-814, 2007.

- [11] J. M. ten Burg, *Least Squares Optimization in Multivariate Analysis*, Leiden: DSWO Press, 1993.
- [12] J. H. Keisler, *Elementary Calculus: An Infinitesimal Approach*, Prindle, Weber & Schmidt, 2000.
- [13] L. Horváth and I. J. Rudas, *Modeling and Problem Solving Techniques for Engineers*, Elsevier, 2004, p. 330.
- [14] S. N. Spitz, "Dimensional Inspection Planning for Coordinate Measuring," PhD dissertation, Dept. of Computer Science, Univ. of Southern California, Los Angeles, CA, 1999.
- [15] P. Gu and K. Chan, "Generative Inspection Process and Probe Path Planning," *J Manufact Syst*, vol. 15, no. 4, pp. 240-255, 1996.
- [16] K. S. Arun, T. S. Huang and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions of Pattern Anal. and Machine Intelli*, vol. 9, no. 5, pp. 698-700, 1987.
- [17] B. Horn, "Closed-Form Solution of Absolute Orientation Using Unit Quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629-642, 1987.
- [18] N. M. Patrikalakis and L. Bardis, "Localization of Rational B-Spline Surfaces," *Engineering with Computers*, vol. 7, no. 4, pp. 237-252, 1991.
- [19] C. H. Menq and K. C. Sahoo, "Localization of 3-D Objects Having Complex Sculptured Surfaces Using Tactile Sensing and Surface Description," *Journal of Engineering for Industry*, vol. 113, no. 1, pp. 85-92, 1991.
- [20] P. J. Besl and N. D. McKay, "A Method for Registration of 3-D Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992.
- [21] Z. Zhang, "Iterative Point Matching for Registration of Free-Form Curves and Surfaces," *International Journal of Computer Vision*, vol. 13, no. 2, pp. 119-152, 1994.
- [22] X. Huang and P. Gu, "CAD-Model Based Inspection of Sculptured Surfaces with Datums," *Int. J.*

- Prod. Res.*, vol. 36, no. 5, pp. 1351-1367, 1998.
- [23] M. Ahmed, S. Yamany and F. A.A., "Fast Algorithm for Registration of Free-Form Curves and Surfaces," in *IEEE International Conference on Image Processing*, Santa Barbara, CA, 1997.
- [24] R. Gudla, "Sculptured Surface Localization using Generalized Hopfield Networks," MS thesis, Dept. Mech. Eng., Univ. of Cincinnati, Cincinnati, OH, 2000.
- [25] C. Che and J. Ni, "A Generic Coordinate Transformation Uncertainty Assessment Approach and Its Application in Machine Vision Metrology," *International Journal of Machine Tools and Manufacture*, vol. 38, no. 10-11, p. 1241–1256, 1998.
- [26] Z. Li, J. Gou and Y. Chu, "Geometric Algorithms for Workpiece Localization," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp. 864-878, 1998.
- [27] J. Qu and R. Sarma, "Least-Square Curve and Surface Localization for Shape Conformance Checking," *Journal of Manufacturing Systems*, vol. 19, no. 5, pp. 297-304, 2000.
- [28] Z. Li, Z. Xu, M. Cen and X. Ding, "Robust Surface Matching for Automated Detection of Local Deformations Using Least-Median-of-Squares Estimator," *Photogrammetric Engineering and Remote Sensing*, vol. 67, no. 11, pp. 1283-1292, 2001.
- [29] G. C. Sharp, S. W. Lee and D. K. Wehe, "ICP Registration Using Invariant Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 90-102, 2002.
- [30] S. M. Yamany and A. A. Farag, "Surface Signatures: An Orientation Independent Free-Form Surface Representation Scheme for the Purpose of Objects Registration and Matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1105-1120, 2002.
- [31] K. Aware, "Closest Point Correspondence in Localization of Parametric Surfaces," Technical Report,

- Dept. Mech. Eng., Univ. of Cincinnati, Cincinnati, OH, 2003.
- [32] K. Ko, T. Maekawa and N. Patrikalakis, "An Algorithm for Optimal Free-Form Object Matching," *Computer-Aided Design*, vol. 35, p. 913–923, 2003.
- [33] L. Orazi and G. Tani, "Geometrical Inspection of Designed and Acquired Surfaces," *Int J Adv Manuf Technol*, vol. 34, p. 149–155, 2007.
- [34] Y. Li and P. Gu, "Automatic Localization and Comparison for Free-Form Surface Inspection," *Journal of Manufacturing Systems*, vol. 25, no. 4, pp. 251-268, 2003.
- [35] Y. Sijie, P. Fangyu and L. Xide, "Research on the Localisation of the Workpieces with Large Sculptured Surfaces in NC Machining," *Int J Adv Manuf Technol*, vol. 23, p. 429–435, 2004.
- [36] B. Narayanaswamy, "A Novel Approach for Arriving at Starting Solution for Least Squared Deviation Algorithms in Data Localization Problem for Free Form Surfaces," Technical Report, Dept. Mech. Eng., Univ. of Cincinnati, Cincinnati, OH, 2006.
- [37] D. OuYang, H. Feng, N. Jahangir and S. H., "Robust Initial Matching of Free-Form Objects Represented by Point Clouds," *Journal of Manufacturing Science and Engineering*, vol. 134, no. 2, 2012.
- [38] V. Mehrad, D. Xue and P. Gu, "Robust Localization to Align Measured Points on the Manufactured Surface with Design Surface for Freeform Surface Inspection," *Computer-Aided Design*, vol. 53, pp. 90-103, 2014.
- [39] X. Wen, Y. Zhao, D. Wang, X. Zhu and X. Xue, "Accurate Evaluation of Free-Form Surface Profile Error based on Quasi Particle Swarm Optimization Algorithm and Surface Subdivision," *Chinese Journal of Mechanical Engineering*, vol. 26, no. 2, pp. 406-413, 2013.
- [40] R. H. Bartels, J. C. Beatty and B. A. Barsky, "Hermite and Cubic Spline Interpolation," in *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*, San

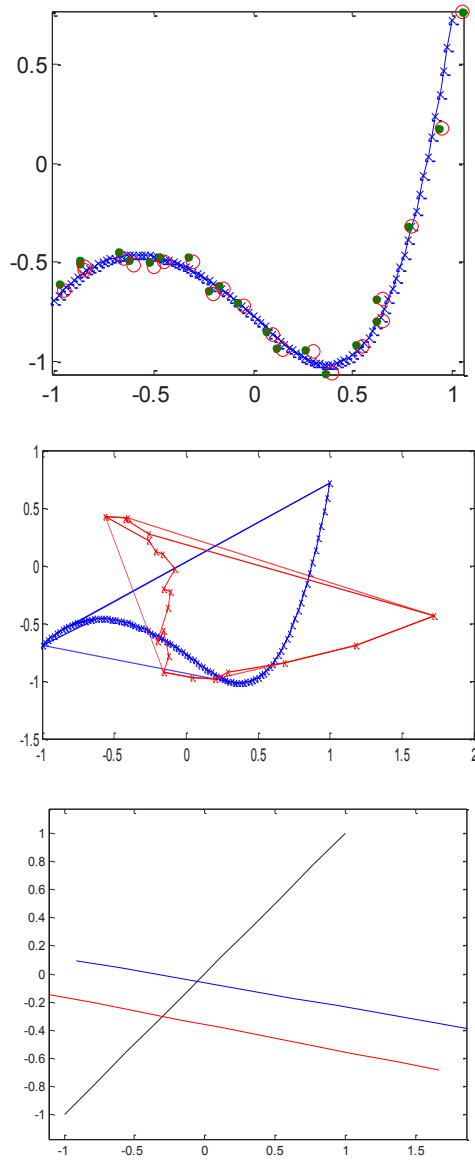
- Francisco, CA: Morgan Kaufmann, 1998, pp. 9-17.
- [41] T. Acharya and A. K. Ray, *Image Processing Principles and Applications*, Hoboken, New Jersey: Wiley Interscience, 2005.
- [42] W. H. Beyer, *CRC Standard Mathematical Tables*, 28th ed., Boca Raton, FL: CRC Press, 1987.
- [43] N. Jamil, Z. Abu Bakar and T. Sembok, "Image," *Proceedings of the Geometric Modeling - New Trends*, pp. 171-176, 2006.
- [44] M. Sarfraz and A. Ridha, "Content-Based Image Retrieval Using Multiple Shape Descriptors," *Computer Systems and Applications*, pp. 730-737, 2007.
- [45] D. Poole and A. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*, Cambridge: University Press, 2010.
- [46] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *IEEE International Conference on Neural Networks*, Perth, WA, 1995.
- [47] W. N. Z. J. Chen and H. S. H. Z. W. Chung, "A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 20, pp. 278-300, 2010.
- [48] M. Clerc, "Standard Particle Swarm Optimisation," 13 Dec 2012. [Online]. Available: http://hal.archives-ouvertes.fr/docs/00/76/49/96/PDF/SPSO_descriptions.pdf. [Accessed 16 Sept 2014].
- [49] M. A. Z. Raja, J. A. Khan, S. I. Ahmad and I. M. Qureshi, "A New Stochastic Technique for Painlevé Equation-I Using Neural Network Optimized with Swarm Intelligence," *Computational Intelligence and Neuroscience*, vol. 2012, p. 10, 2012.

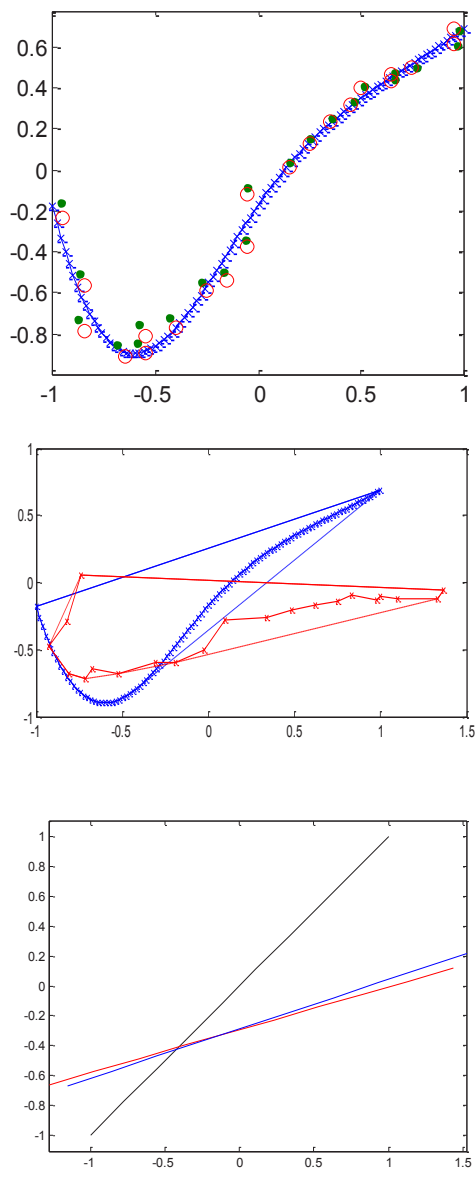
APPENDIX

Table of results from algorithm sequences

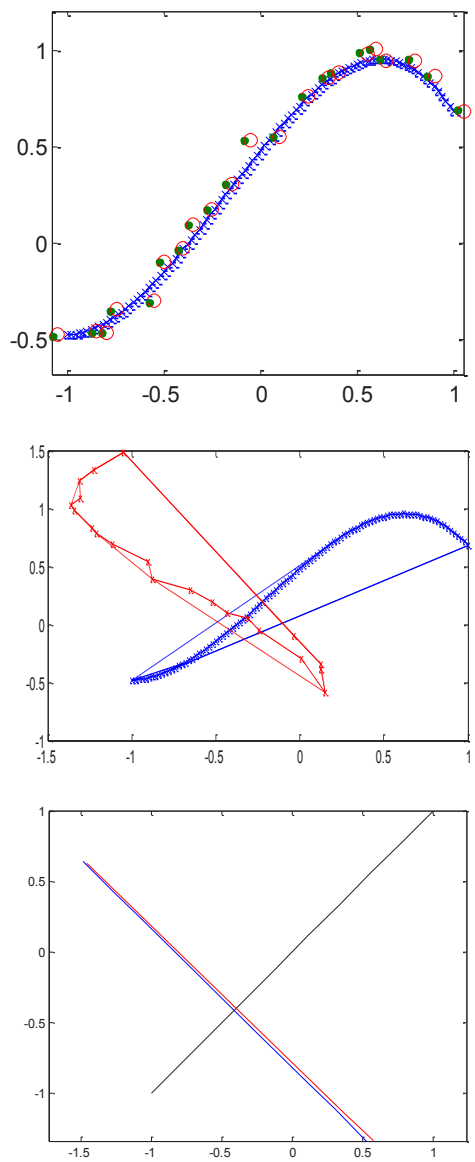
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
1	0.6212	0.5929	0.1685	0.2353	35.5512	7.0974	

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
2	0.4637	0.4761	0.1882	0.1547	25.6361	5.6791	

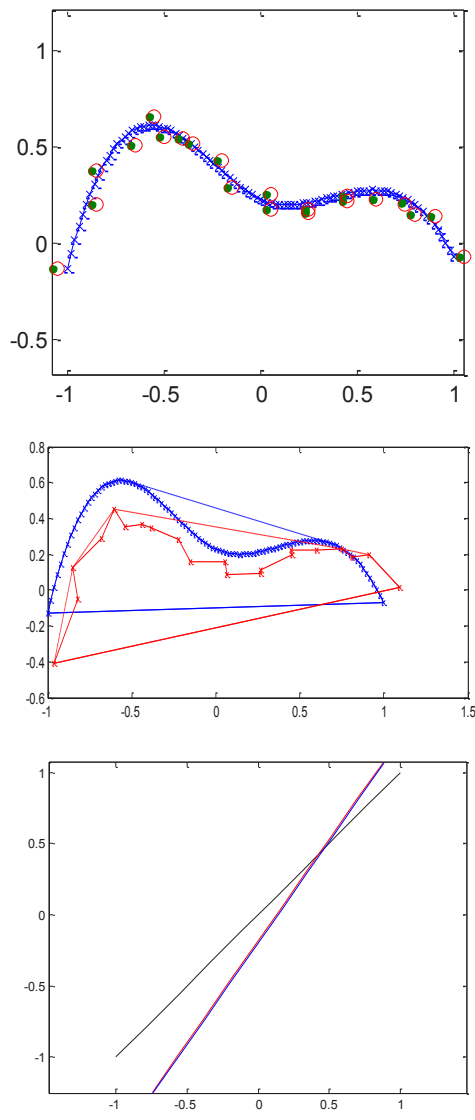
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
3	1.2340	1.2080	0.0731	0.1100	33.9796	6.5898	

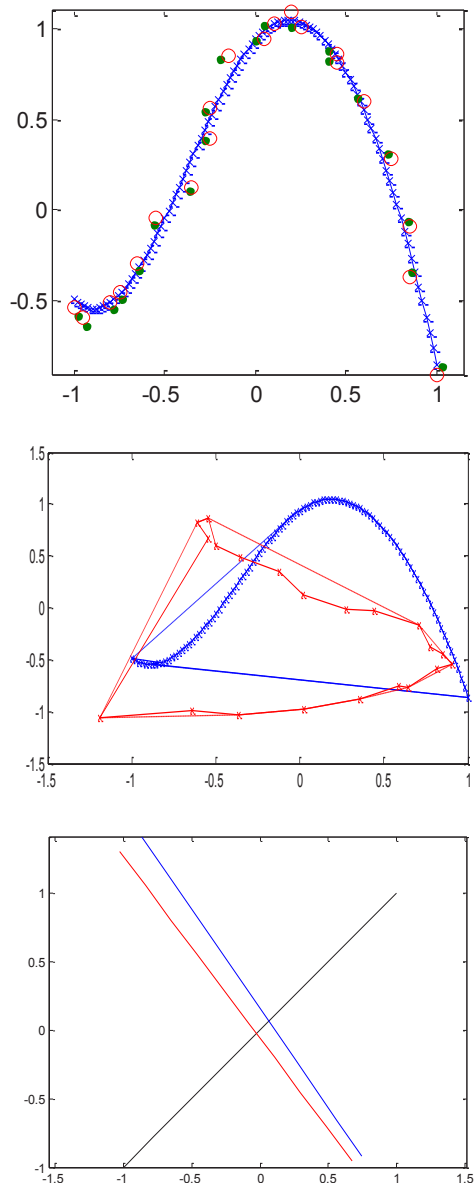
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
4	0.7277	0.7403	0.1615	0.1239	32.2905	6.1410	 <p>The figure consists of three vertically stacked plots. The top plot shows data points (green and red circles) and a blue fit curve. The middle plot shows the same data points and a red fit curve. The bottom plot shows the same data points and a black fit curve. The x-axis for all plots ranges from -1 to 1.5, and the y-axis ranges from -1 to 1.</p>

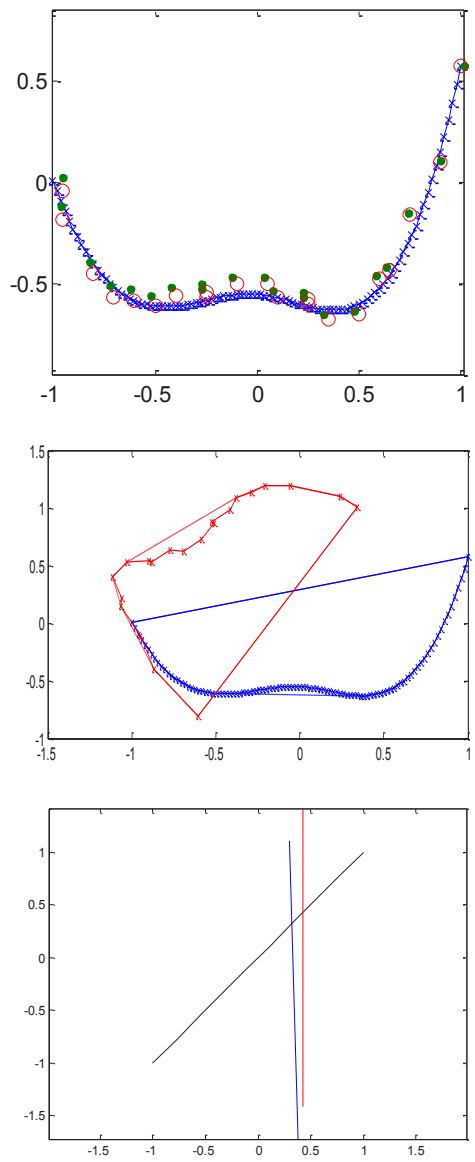
0

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
5	0.5295	0.5149	0.1420	0.1850	74.1850	5.8830	 <p>The figure consists of three vertically stacked plots. The top plot shows a set of data points (green and red circles) and a blue curve fit. The middle plot shows the same data points and a red curve fit. The bottom plot shows the same data points and a black curve fit. The x-axis for all plots ranges from -1 to 1, and the y-axis ranges from -0.5 to 1.0.</p>

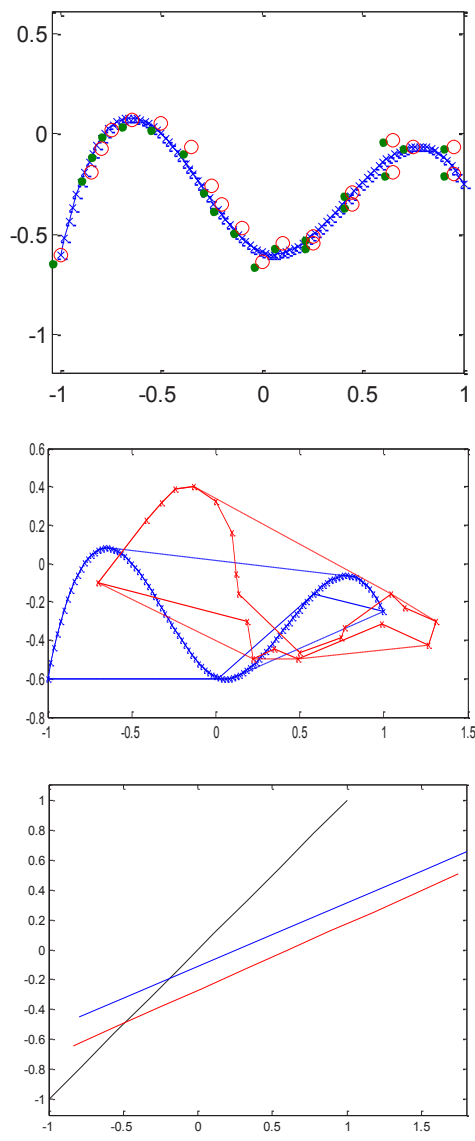
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
6	1.1486	1.1159	0.1209	0.1667	45.5705	7.5562	<p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows a set of data points (green circles, red circles, blue crosses) and a blue curve that fits the data. The middle subplot shows the same data points and fit curve, but with red lines connecting the points, forming a polygonal shape. The bottom subplot shows a red line and a blue line, both decreasing linearly, with a black line representing the identity function $y=x$.</p>

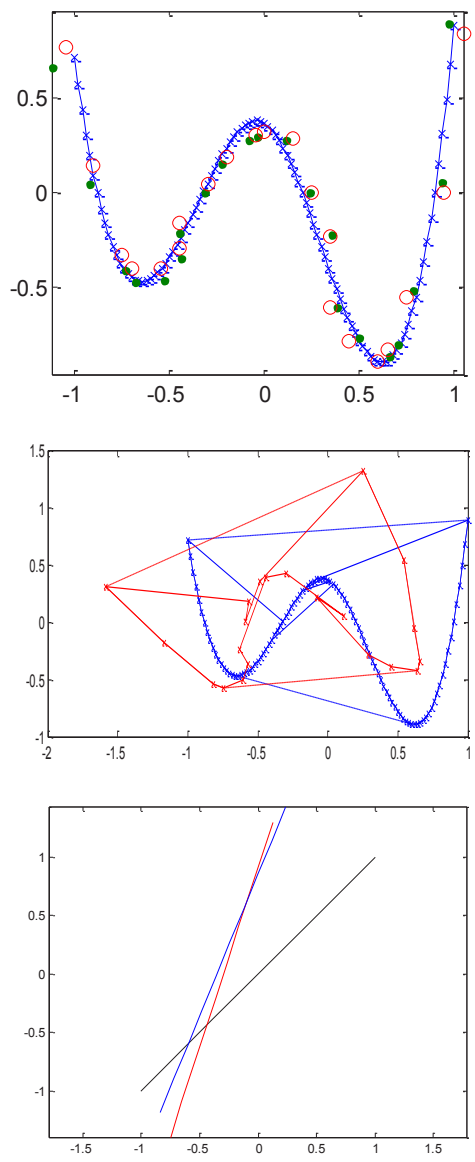
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
7	0.7672	0.7587	0.0927	0.1117	11.2238	6.0768	 <p>The 'Plot' column contains three vertically stacked plots. The top plot shows data points (red circles) and a blue curve fit over the range [-1, 1] on the x-axis and [-0.5, 1] on the y-axis. The middle plot shows the error regions (red lines) and the blue curve fit over the range [-1, 1.5] on the x-axis and [-0.6, 0.8] on the y-axis. The bottom plot shows a linear fit (purple line) and a reference line (black line) over the range [-1, 1] on both axes.</p>

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
8	1.8667	1.8726	0.1025	0.0971	49.4920	7.9337	 <p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows a set of data points (red and green circles) and a blue curve fit. The middle subplot shows the same data points with red lines connecting them, and a blue curve fit. The bottom subplot shows three lines: a red line, a blue line, and a black line, all intersecting at the origin.</p>

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
9	1.3617	1.3763	0.0906	0.0718	66.1657	6.9655	 <p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows a set of data points (green circles) and a blue parabolic curve fit. The x-axis ranges from -1 to 1, and the y-axis from -0.5 to 0.5. The middle subplot shows the same data points and a red piecewise linear fit consisting of several connected line segments. The x-axis ranges from -1.5 to 1.5, and the y-axis from -1 to 1.5. The bottom subplot shows the same data points and a single blue straight line fit. The x-axis ranges from -1.5 to 1.5, and the y-axis from -1.5 to 1.</p>

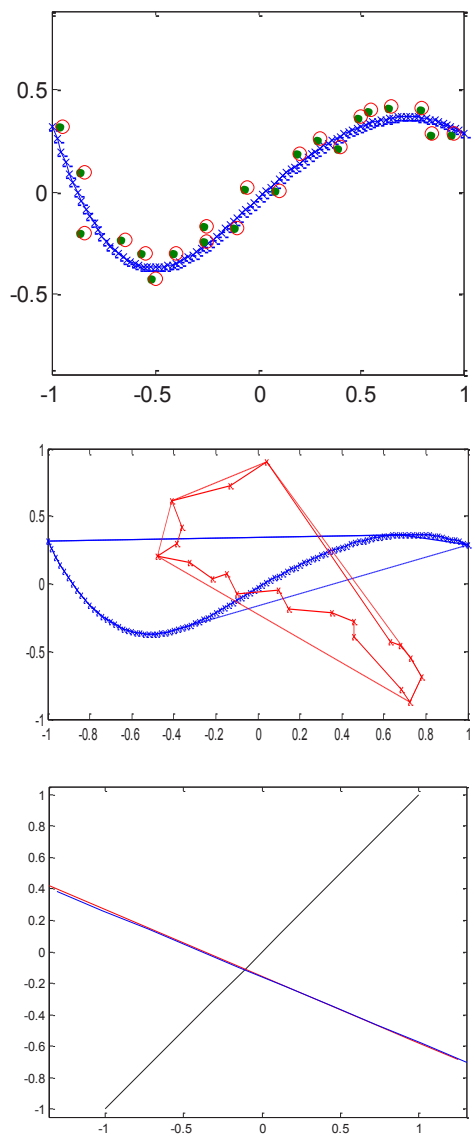
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
10	0.4851	0.4754	0.1683	0.1982	29.4830	6.0082	

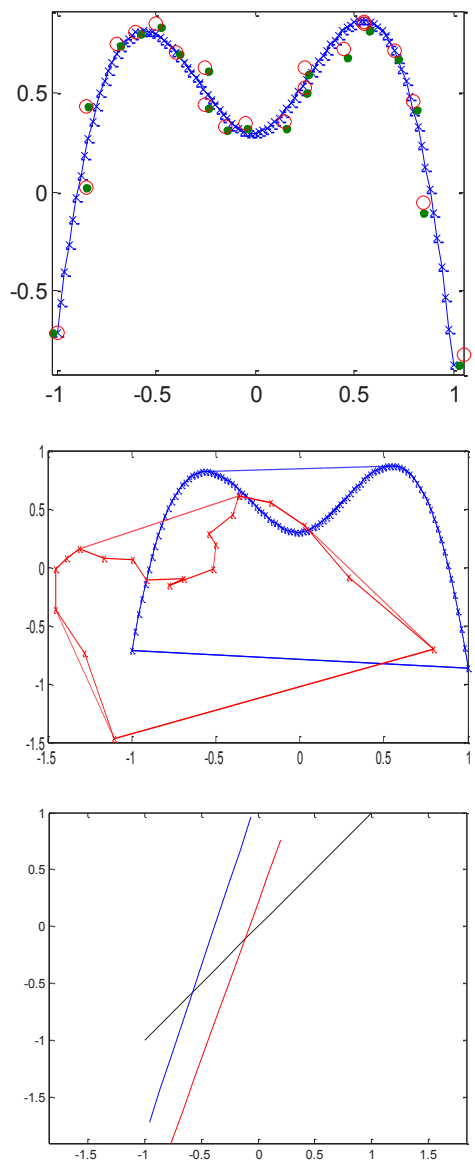
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
11	0.4120	0.4280	0.2492	0.1948	22.5645	5.7304	 <p>The 'Plot' column contains three vertically stacked plots. The top plot shows data points (red circles, green circles, blue crosses) and a blue fitted curve over the range x = [-1, 1] and y = [-1, 0.5]. The middle plot shows the same data points and blue curve, but with red lines connecting the points, forming a polygonal shape, over the range x = [-1, 1.5] and y = [-0.8, 0.6]. The bottom plot shows three linear regression lines: a black line with the steepest slope, a blue line with a moderate slope, and a red line with the shallowest slope, over the range x = [-1, 1.5] and y = [-1, 1].</p>

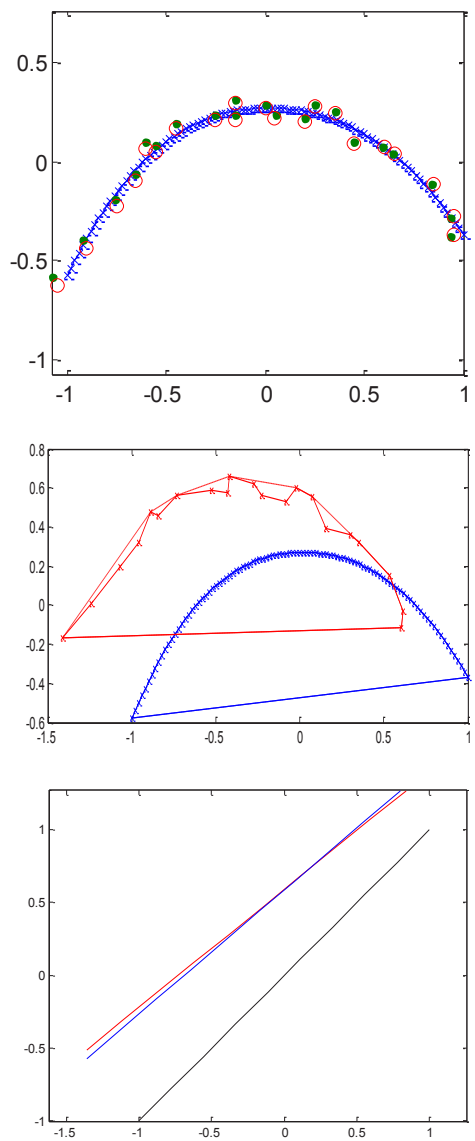
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
12	1.8401	1.8636	0.1177	0.0976	23.1539	8.8981	 <p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows data points (red circles, green circles, blue crosses) and a blue curve fit. The middle subplot shows the same data points and a red curve fit. The bottom subplot shows the same data points and a black curve fit.</p>

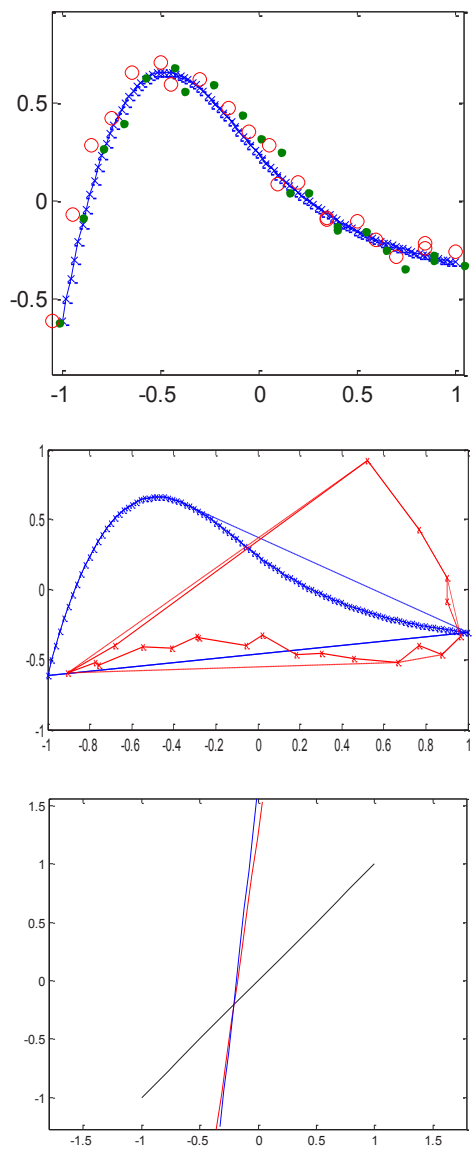
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
13	0.5347	0.5701	0.2257	0.1298	71.7097	5.3687	<p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows a set of data points (red and green circles) and a blue fitted curve. The middle subplot shows a red polygonal path and a blue curve. The bottom subplot shows a red line, a blue line, and a black diagonal line.</p>

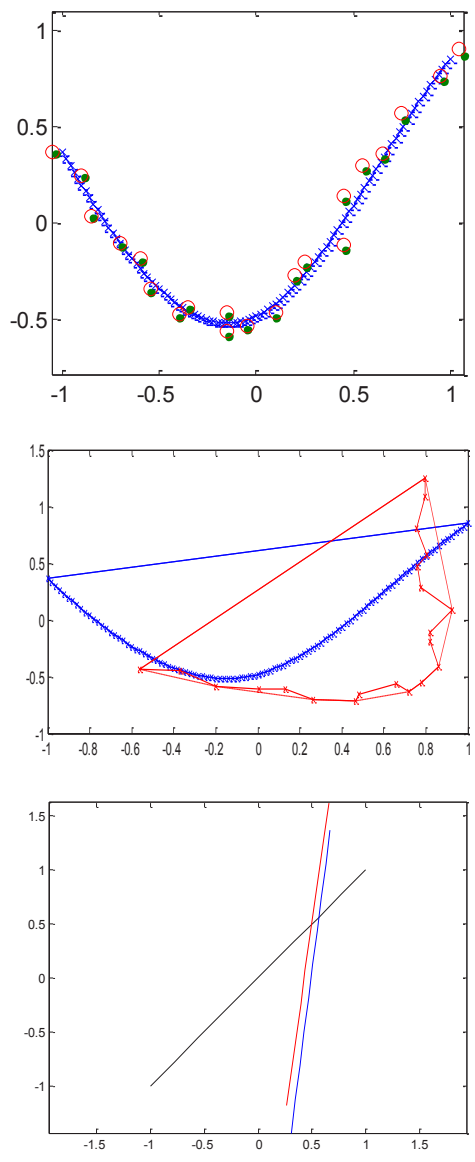
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
14	0.8960	0.9028	0.1505	0.1339	21.5483	7.1203	<p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows data points (green circles, red circles, blue crosses) and a blue curve fit. The middle subplot shows the same data points and a red curve fit. The bottom subplot shows a black line and a red line fit.</p>

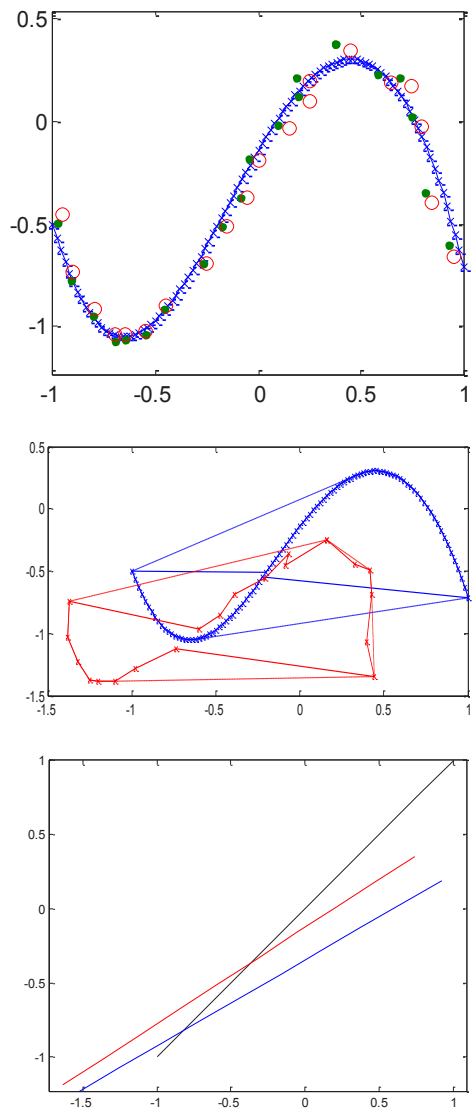
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
15	0.5396	0.5260	0.1546	0.1744	43.6432	6.3697	 <p>The 'Plot' column contains three vertically stacked plots. The top plot shows data points (green circles) and a blue curve fit over the range x = [-1, 1] and y = [-0.5, 0.5]. The middle plot shows the same data points and fit, with red lines connecting the points, likely representing the error or residuals. The bottom plot shows a coordinate system with x and y axes ranging from -1 to 1, with a black diagonal line and a purple line representing a linear fit or error component.</p>

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
16	2.3993	2.4196	0.0672	0.0501	33.9295	8.3808	 <p>The figure consists of three vertically stacked plots. The top plot shows a set of data points (red circles, green circles, blue crosses) and a blue curve fit. The middle plot shows a red line connecting the data points and a blue curve fit. The bottom plot shows a red line, a blue line, and a black line representing different model fits.</p>

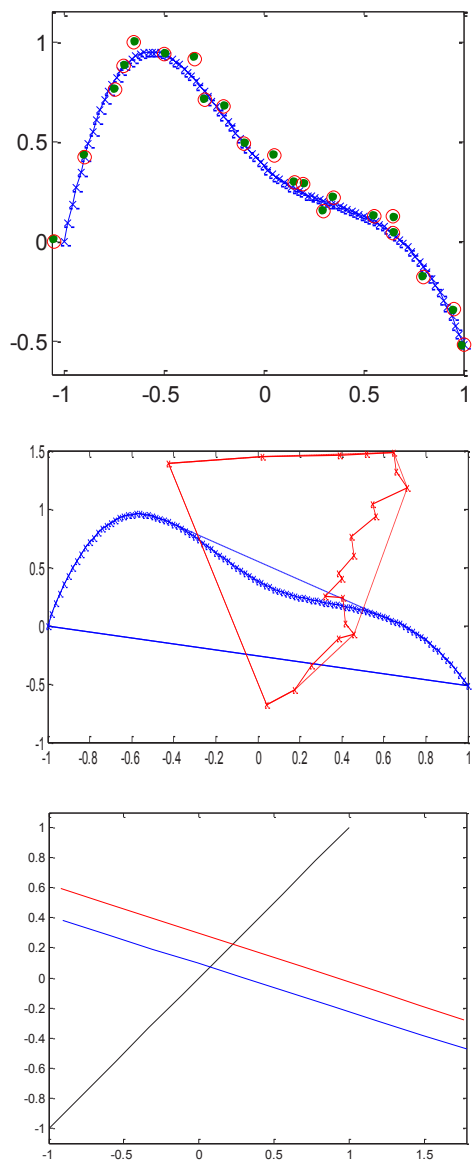
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
17	0.9954	1.0181	0.0942	0.0558	29.1661	5.4166	 <p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows a set of data points (red and green circles) and a blue curve fit, with the x-axis ranging from -1 to 1 and the y-axis from -1 to 0.5. The middle subplot shows the same data points and a red piecewise linear fit, with the x-axis ranging from -1.5 to 1 and the y-axis from -0.6 to 0.8. The bottom subplot shows the data points and a black linear fit, with the x-axis ranging from -1.5 to 1 and the y-axis from -1 to 1.</p>

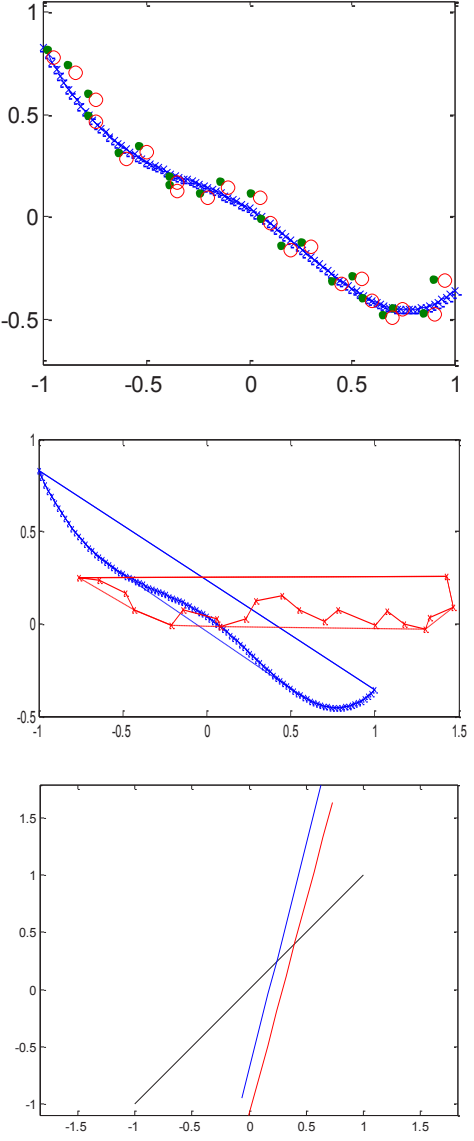
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
18	1.1198	1.1510	0.1386	0.0930	53.0306	6.8193	

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
19	1.2958	1.3215	0.0934	0.0585	25.9003	6.3708	

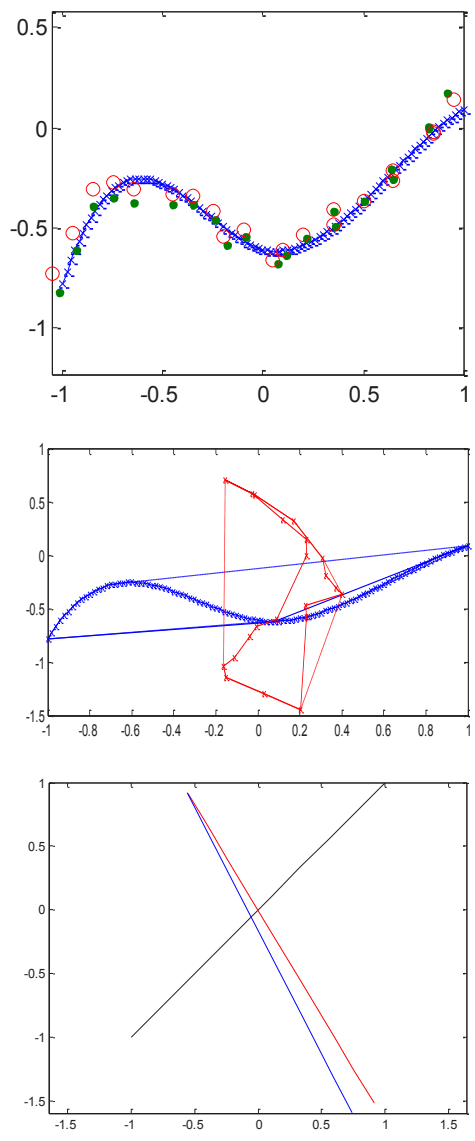
Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
20	0.8786	0.8882	0.1936	0.1740	35.4084	5.5314	 <p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows a set of data points (red circles and green dots) and a blue curve fit. The middle subplot shows the same data points and a red curve fit. The bottom subplot shows the same data points and a black curve fit. The x-axis for all plots ranges from -1 to 1, and the y-axis ranges from -1 to 0.5.</p>

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
21	0.4473	0.4363	0.2116	0.2456	36.2292	6.0999	<p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows a set of data points (red and green circles) and a blue curve fit. The middle subplot shows a red polygonal path and a blue curve fit. The bottom subplot shows two intersecting lines (red and blue) and a black line.</p>

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
22	1.2540	1.2594	0.0670	0.0605	45.1272	6.3785	 <p>The 'Plot' column contains three vertically stacked subplots. The top plot shows data points (green circles with red outlines) and a blue curve fit over the x-axis range [-1, 1] and y-axis range [-0.5, 1]. The middle plot shows the same data points and blue curve, with a red line connecting the points, likely representing the error or residual. The bottom plot shows three linear functions: a black line with a positive slope, a red line with a negative slope, and a blue line with a negative slope, plotted over the x-axis range [-1, 1.5] and y-axis range [-1, 1].</p>

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
23	0.3700	0.3811	0.2080	0.1594	25.4202	5.7347	 <p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows data points (red and green circles) and a blue curve fit. The middle subplot shows the same data points and a red curve fit. The bottom subplot shows the same data points and a black curve fit. The x-axis for all plots ranges from -1 to 1, and the y-axis ranges from -0.5 to 1.</p>

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
24	0.5469	0.6080	0.2923	0.1486	48.6582	7.2060	<p>The 'Plot' column contains three vertically stacked subplots. The top plot shows data points (red circles and green squares) and a blue curve fit over the x-axis range [-1, 1] and y-axis range [-1, 0.5]. The middle plot shows the same data points with red lines connecting them, forming a complex shape, and a blue curve fit, with axes ranging from -1.5 to 1.5. The bottom plot shows a linear fit (black line) and two other lines (red and blue) over the same x-axis range [-1, 1] and y-axis range [-1, 1].</p>

Run	Overlap Area Before	Overlap Area After	Error Area Before	Error Area After	Sum Least Squares Before	Sum Least Squares After	Plot
25	0.3383	0.3604	0.3780	0.3110	41.0524	6.3961	 <p>The 'Plot' column contains three vertically stacked subplots. The top subplot shows data points (red circles and green dots) and a blue curve fit. The middle subplot shows the same data points and a red curve fit. The bottom subplot shows the same data points and a black curve fit.</p>