



# Proof-of-concept for the Optimised Control of the English Wheel using a Non-conventional Parametric Approach

## FINAL YEAR MEng PROJECT REPORT

*Submitted by*

ADAM KIRCHEL

Candidate no. 13252

Supervisor: Evripides Loukaides  
Assessor: Debbie Janson

Department of Mechanical Engineering  
Faculty of Engineering and Design  
University of Bath

Word count: 12552

*“I certify that I have read and understood the entry in the Student Handbook for the Department of Mechanical Engineering on Cheating and Plagiarism and that all material in this assignment is my own work, except where I have indicated with appropriate references.”*

*Author’s signature:*

A handwritten signature in black ink, appearing to read 'ADAM KIRCHEL'.

**Submitted: 04 May 2022**

## **Acknowledgements**

I would like to thank my supervisor, Dr Evros Loukaides, and colleague, Dan Bowen, for their continual support in providing guidance and feedback throughout this project.

## **Links**

Github repository: [https://github.com/adamkirchel/FYP\\_submission.git](https://github.com/adamkirchel/FYP_submission.git)

## Summary

There has not yet been any significant development in the automation of the English Wheel (EW) due to limited understanding of the process mechanics. Other flexible forming processes have been adopting the use of machine learning (ML) for automation due to its applicability to non-linear problems. This paper takes inspiration from these processes, by presenting a novel neural network control architecture that can be applied to the EW.

The architecture requires a parametric representation of both the sheet blank geometry and tool path. A method is presented to categorise the tool path using techniques derived from the traditional smith known as ‘tracking strategies’, which produce standard geometries. From this heuristic knowledge, a synthetic dataset of sheet blank geometries is created, which is used to train and tune the architecture using an iterative framework. Within this framework, a variation of geometrical representation techniques and model hyperparameters are considered. The final iteration of the model gave an unbiased accuracy that served as a partial proof-of-concept to justify its application with experimental data.

To deliver a full proof-of-concept, this paper outlines an experimental proposal to be implemented in future work. This proposal includes a user interface (UI) application that enables efficient generation of ‘tracking strategies’ for use on the EW machine. Methodologies are presented for gathering data using a handheld scanning tool, and post-processing by transforming the generated point cloud. Given the results from the synthetic data, amendments to the experimental procedure regarding the sample population size and ML architecture are suggested.

## Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
1.1	Background . . . . .	7
1.2	Aims and objectives . . . . .	8
1.3	Structure . . . . .	9
<b>2</b>	<b>Literature Review</b>	<b>10</b>
2.1	English Wheel mechanics . . . . .	10
2.2	Parameterisation of flexible forming processes . . . . .	12
2.2.1	Tool path . . . . .	12
2.2.2	Workpiece . . . . .	13
2.3	Overview of machine learning techniques . . . . .	15
2.4	Non-conventional control applied to flexible metal forming . . . . .	17
<b>3</b>	<b>Parameterisation of the EW process</b>	<b>19</b>
3.1	Sheet parameters . . . . .	19
3.2	Machine parameters . . . . .	21
3.3	Tracking pattern parameters . . . . .	21
3.3.1	Parametric approach . . . . .	21
3.3.2	Non-parametric approach . . . . .	24
<b>4</b>	<b>Design Convergence Study</b>	<b>25</b>
4.1	Experimental design specification . . . . .	25
4.2	Parameter Study . . . . .	26
4.3	Experimental concepts . . . . .	26
4.4	Evaluation of experimental concepts . . . . .	28
4.5	Tracking strategies . . . . .	28
4.6	Evaluation of tracking strategies . . . . .	31
4.7	Sample space investigation . . . . .	34
4.7.1	Selection of samples . . . . .	34
4.7.2	Investigating sampling parameters . . . . .	35
4.8	Proposed test cases . . . . .	37
<b>5</b>	<b>Experimental procedure</b>	<b>41</b>
5.1	G-code generation . . . . .	41
5.2	Collecting data . . . . .	45
5.3	Data post-processing . . . . .	46

<b>6 Model testing framework design</b>	<b>48</b>
6.1 Overview of model and testing framework . . . . .	48
6.2 Architecture . . . . .	50
6.3 Performance metrics . . . . .	52
6.4 Data tuning . . . . .	52
6.4.1 Geometry representation . . . . .	52
6.4.2 Data augmentation . . . . .	54
6.5 Model tuning . . . . .	55
6.5.1 Machine learning method . . . . .	55
6.5.2 Hyperparameters . . . . .	57
<b>7 Results</b>	<b>58</b>
7.1 Overview of input data . . . . .	58
7.2 Unit testing . . . . .	59
7.2.1 First layer . . . . .	59
7.2.2 Second layer . . . . .	62
7.3 Integration testing . . . . .	63
<b>8 Discussion and future work</b>	<b>66</b>
8.1 Analysis of the final architecture . . . . .	66
8.2 Limitations of data . . . . .	66
8.3 Limitations of model . . . . .	67
8.4 Future work . . . . .	67
<b>9 Conclusion</b>	<b>67</b>
<b>A Design of experiments documentation</b>	<b>71</b>
A.1 Design specification . . . . .	71
A.2 Experimental concepts . . . . .	72
A.3 Tracking strategies . . . . .	76
<b>B User-interface documentation</b>	<b>85</b>
B.1 Process parameter documentation . . . . .	85
B.2 Tracking strategy structure documentation . . . . .	88
B.3 User-interface Quickstart Guide . . . . .	90
B.3.1 Defining the process parameters . . . . .	90
B.3.2 Generate the tracking pattern . . . . .	91
B.3.3 Simulate the machine process . . . . .	94
B.3.4 Generate samples for experimentation . . . . .	95

<b>C Model documentation</b>	<b>101</b>
C.1 Hyperparameters . . . . .	101
C.2 Description of preliminary layer . . . . .	102
<b>D Model results</b>	<b>103</b>
D.1 Overview of input data . . . . .	103
D.2 Unit testing . . . . .	105
D.2.1 First layer . . . . .	105
D.2.2 Second layer . . . . .	107

## Nomenclature

The following symbols and abbreviations are used throughout the body of the document and are provided here as reference.

$H$	Mean curvature
$K$	Gaussian curvature
$P_1$	Maximum principle curvature
$P_2$	Minimum principle curvature
ANN	Artificial neural network
CNN	Convolutional neural network
EW	English Wheel
FEA	Finite element analysis
FEM	Finite element model
MAE	Mean average error
ML	Machine learning
NN	Neural network
PDF	Probability density function
RMSE	Root-mean-square error
UI	User interface

## 1 Introduction

### 1.1 Background

Following recent technological developments, manufacturing is considered to have entered the so-called fourth industrial age, or Industry 4.0. [Groover \(2015\)](#) suggest one of the drivers of this new industrial age is changing customer demand towards mass customisation of parts and process flexibility. This introduces a problem, with [Yang et al. \(2018\)](#) concluding that current die-based manufacturing processes are too inflexible to meet this increase in demand for personalised production. In response, there has been increased research into die-less flexible forming technologies that can manufacture a large variety of part shapes and sizes.

Adopting flexible techniques have additional benefits, with [Daehn \(2021\)](#) describing the advantageous material microstructural properties that can be generated compared to conventional methods, such as work hardening, textural changes, and closing porosity. There is great environmental benefit to adopting flexible forming, with [Cooper et al. \(2017\)](#) concluding that considerable energy savings are possible compared to drawing operations for production numbers over the die lifespan of less than 100 parts. Also, there is limited waste with comparatively non-existent tooling cost and lead-times compared to conventional techniques.

[Schuh et al. \(2019\)](#) present several flexible technologies such as resin casting, arc spraying, hydroforming and multi-point forming. Despite the ambition, these technologies do not show the capability to produce low quantities of products reliably. Instead, [Bowen et al. \(2022\)](#) suggest new metal forming processes need to take inspiration from incremental deformation techniques used by the traditional metal smith. These techniques are proven to produce parts to a high precision at low quantities, and automation can enable this process to be more efficient. Shown in Figure 1, traditional metal smithing techniques include but are not limited to hammering (a), wheeling (b), driving (c), and spinning (d). These processes are all conducted manually, where the smith must monitor several material properties including geometry and surface finish.

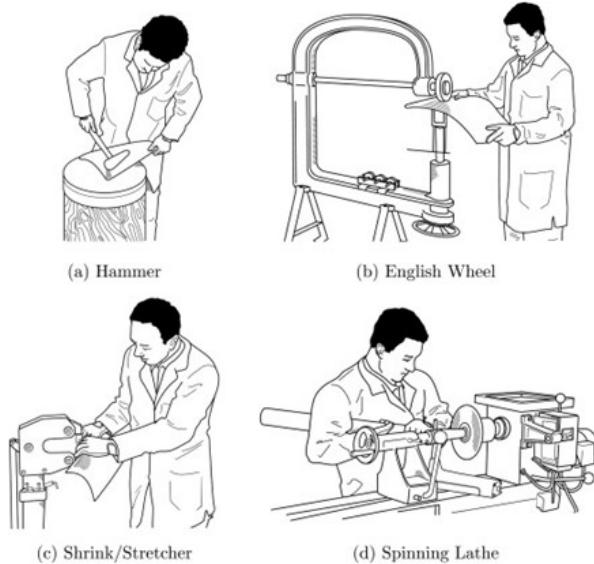


Figure 1: Overview of traditional forming processes used by the smith ([Bowen et al., 2022](#))

This study will focus on the automation of the wheeling process. The setup of the English Wheel (EW) which used to perform wheeling is illustrated in Figure 2. It is composed of upper and lower wheels that are separated by a small gap, through which the sheet is forced by the operator to induce incremental deformation. A novel machine has been developed at the University of Bath to automate this process, and will be available for use in this project.

There has been no success producing a feasibly marketable automated solution to flexible metal forming processes within the academic community. This is because of uncertainties arising from non-linearities in metal properties, such as the stochastic variation in grain size and orientation in metal cast parts. The

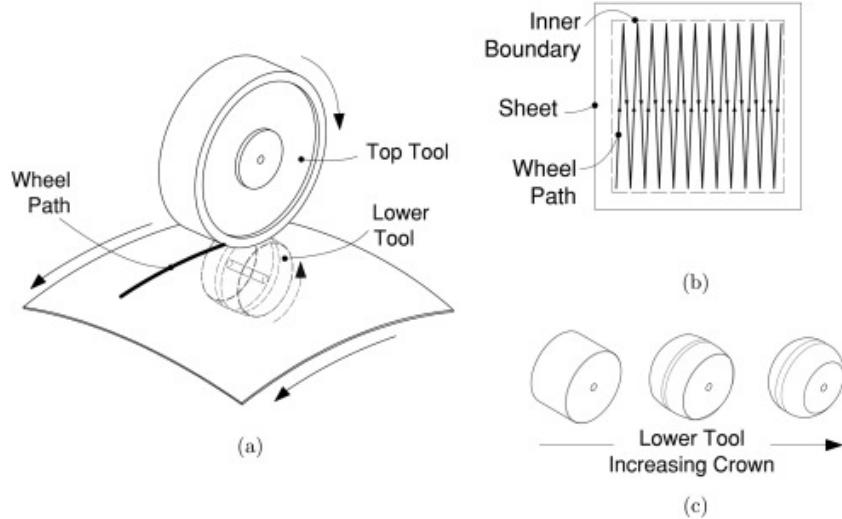


Figure 2: EW technical overview (Bowen et al., 2022)

traditional control methodologies implemented in state-of-the-art processes lack the precision of the traditional smith (Bowen et al., 2022), with Allwood et al. (2016) concluding that these systems do not have the high-tolerance prediction capabilities to minimise these uncertainties.

Instead, non-conventional methods such as machine learning (ML) look promising. ML relies on high computational power, and with Lundstrom (2003) suggesting Moore's Law, that the computational power of microprocessors is roughly doubling every two years, has the capability to continue for the foreseeable future, the use of these techniques will become more accessible. ML techniques also rely on a large pool of data, with Zhong et al. (2017) claiming that data is becoming more accessible in manufacturing due to the advent of novel technologies such as the internet-of-things (IoT), resulting in new 'big data' environments.

## 1.2 Aims and objectives

Given this technological capability, this study aims to develop and evaluate the suitability of ML techniques for automation of the EW using experimental data. This aim will be achieved through the following objectives:

1. Parameterise the EW process, including the sheet geometry, machine parameters, and tracking characteristics.
2. Design experiments that collect data that is relevant to the aim, have minimal bias, and are representative of the sample space.
3. Develop an automated method of converting derived parameters to G-code.
4. Build a large database of sheet geometries and process parameters through repeatable and reproducible experiments.
5. Develop and compare ML techniques for automating the process and optimise through iterative tuning.

### 1.3 Structure

The methodology and timescales for achieving these objectives are presented in Figure 3. The report until Section 6 details work completed with the intention of collecting experimental results. This work includes the parameterisation of the EW process, the experimental design, and the data pre and post-processing framework. A pivot in strategy was made at this point due to the impracticality of collecting experimental data. From Section 6 onwards, the work documented refers to synthetic data.

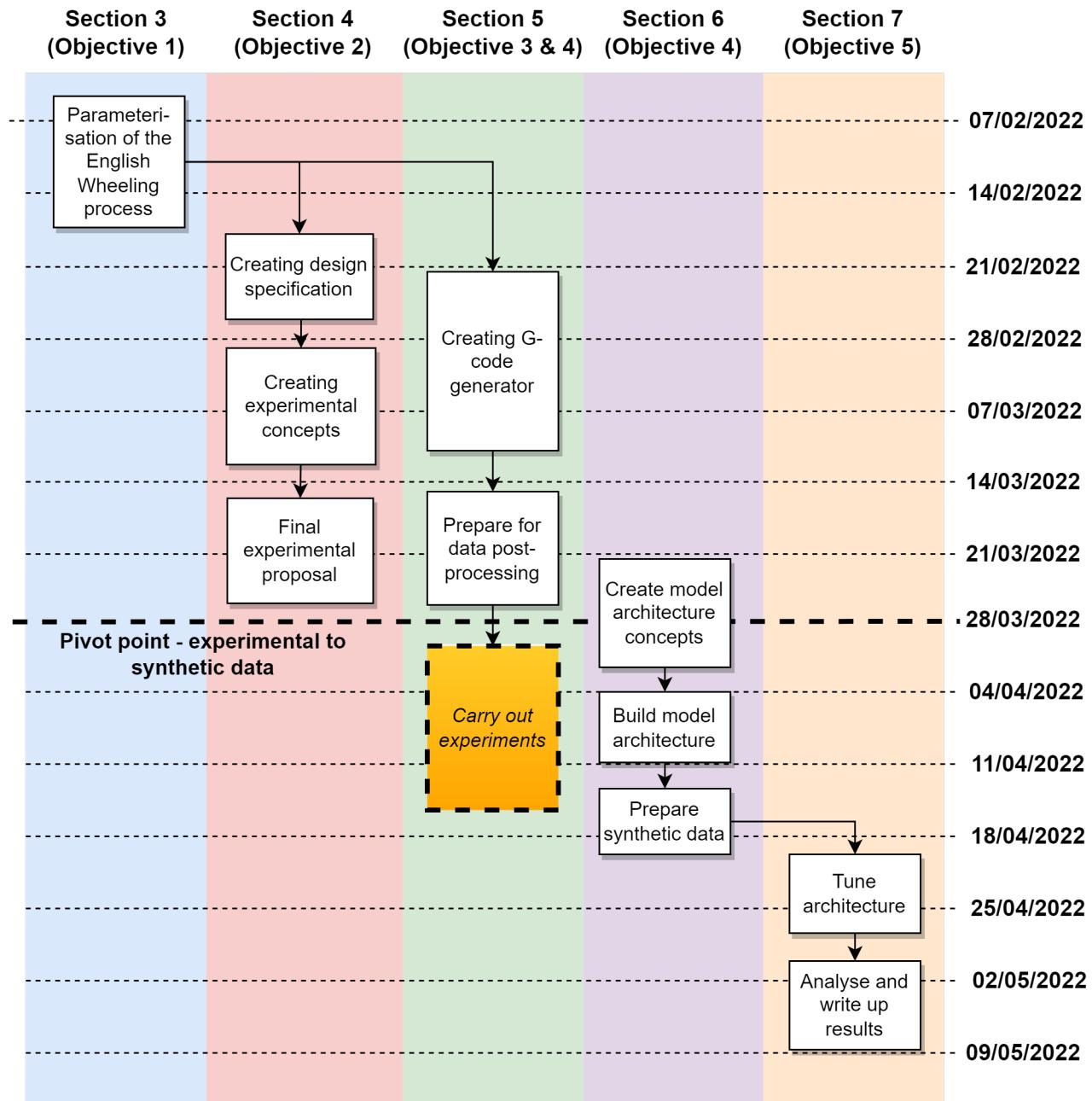


Figure 3: Methodology of project with references to sections in report with timescales

## 2 Literature Review

This review informs the methodologies and procedures proposed in the following study to enable the aims and objectives to be met.

### 2.1 English Wheel mechanics

The state-of-the-art in the understanding of EW mechanics can inform the design of experiments and ML model. [Bowen et al. \(2021\)](#) have categorised the mechanics of the EW to be a stretching process. They use a validated numerical model to examine the plastic deformation induced through wheeling. Figure 4 shows that there is through-thickness thinning at the point of contact and thickening beyond the surrounding region in the transverse direction. The greatest amount of thickening occurs in front of the tool in the direction of the wheel path. This thickening is a result of compressive stresses that stop the material from stretching further. Due to the difference between the strain in the top and bottom surfaces in Figure 4, curvature is induced in the transverse direction. This is the dominant direction of curvature. Despite the understanding gained from this finite element model, it is constrained to studying local phenomena due to the high computational power required.

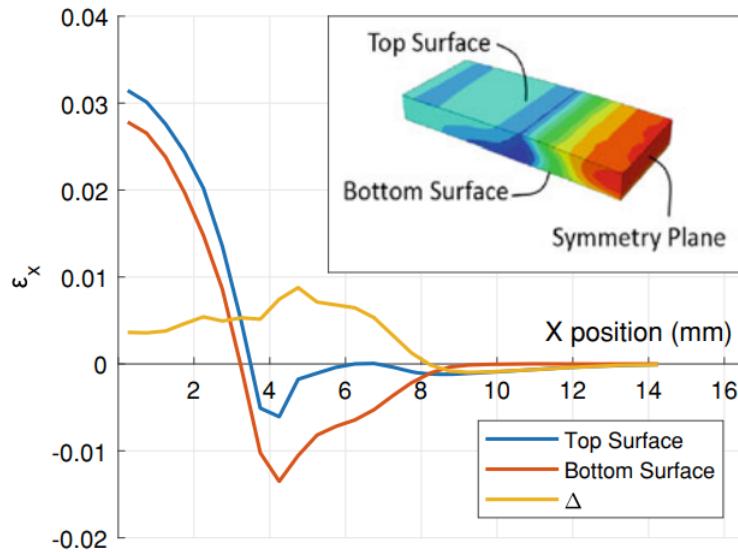


Figure 4: Strains in x-direction for wheeled section of sheet ([Bowen et al., 2021](#))

[Music and Allwood \(2012\)](#) generated data numerically and experimentally of the sheet response to a ‘spatial impulse’ from the EW. This is derived from the impulse response used in time domain control engineering, and is defined in this case as the geometrical response of the workpiece to an infinitesimal actuation of the tool. It was found that the EW had a global effect on the geometry of the workpiece. Figures 5a and 5d shows that impulse response of sheet was not significantly affected by changes to the thickness and prior deformation respectively. Figures 5b and 5c shows that the location had a significant effect with an impulse closer to the edge being more localised.

Currently, the most widely used methods to describe the EW process are heuristic in nature. [Eastwood \(2013\)](#) describe the effect of applying slight downward pressure while wheeling to induce incremental curvature in the sheet. Skilled artisans use different patterns to create varying curvature in the sheet. These ‘tracking patterns’ are produced from a repetitive procedure of wheeling in a straight line, pivoting the sheet by a fixed angle, and then linear tracking in the reverse direction. An example tracking pattern is displayed in Figure 6a. [Anderson \(1990\)](#) cites that tracking patterns that are closer together generate tighter curves. [Longyard \(2014\)](#) present techniques such as using three levels of pressure for different forming processes

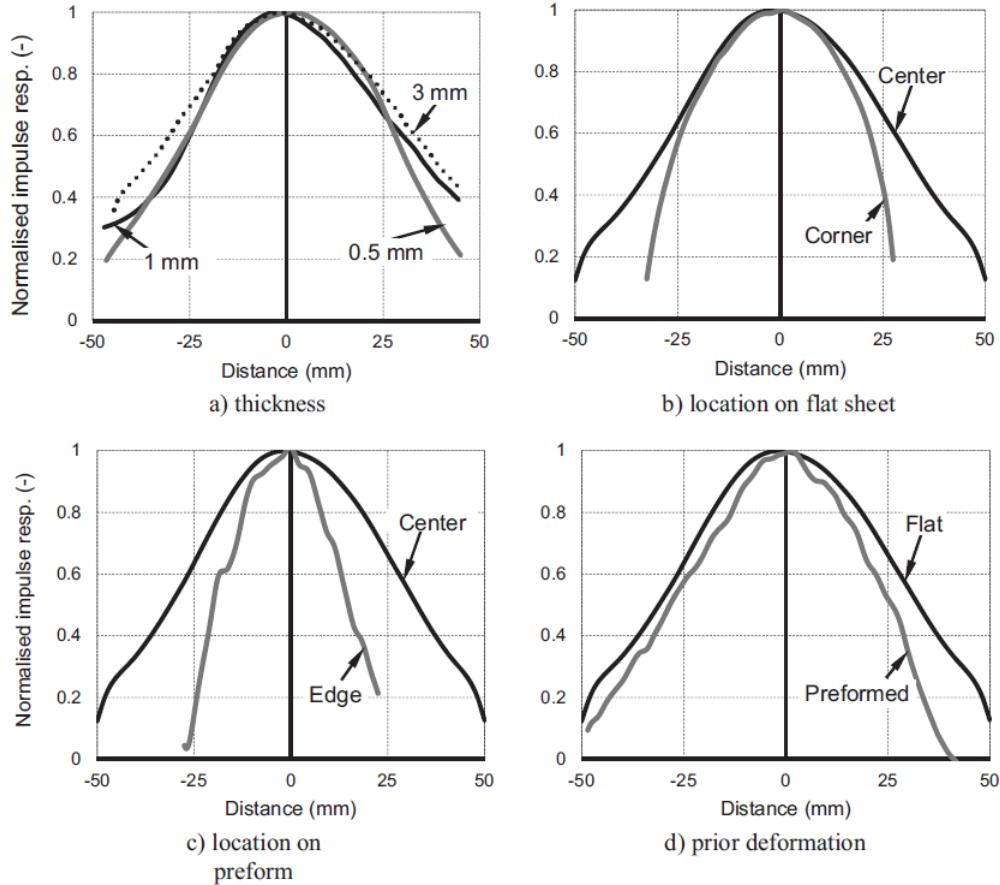


Figure 5: Effect of (a) thickness, (b) and (c) location, and (d) prior deformation on spatial impulse response of the EW ([Music and Allwood, 2012](#))

with the sheet. For example, light pressure, or ‘wash over’, is used to relieve the tensions of a panel that has previously been worked, whilst high pressure is used for flanging or embossing.

[Longyard \(2014\)](#) refers to basic tracking patterns. Figure 6a displays the simplest and most common tracking pattern in industry known as ‘N’ shaped tracks. The next common tracking pattern is referred to as ‘overlapping’, and involves wheeling each ‘leg’ of the ‘N’ shape three times. Each ‘leg’ will get a short, medium and long pass to raise the centre of the crown more. Figure 6c displays cross-tracking, which is when two ‘N’ tracking patterns are orientated 90 degrees to one another. This creates compound curves and the ‘bowl-like’ effect shown in Figure 6d. More advanced techniques include blending, which is used to smooth the curvature of a surface. [Longyard \(2014\)](#) also refers to a ‘no blow zone’ where the wheel should not pass beyond to prevent wrinkles at the edge of the sheet.

Currently, there is no accepted analytical theory describing the wheeling process, however given that it is a stretching process similarities can be made with theory presented in literature. [Marciniak et al. \(2002\)](#) describes mass production stretching processes with defined boundary conditions such as cylindrical deep drawing and hemispherical punch stretching. Despite the knowledge in this field, these operations cannot be made analogous to the EW process as the stretching process occurs locally on the sheet without simply defined boundary conditions. [Minton \(2017\)](#) presents a more rigorous approach where it is assumed the process can be described as a two-dimensional rolling model, and represents the mechanics of the process through a constrained optimisation equation to minimise energy. However, this solution is limited as the equation thus far has no solutions as initial guesses are non-trivial.

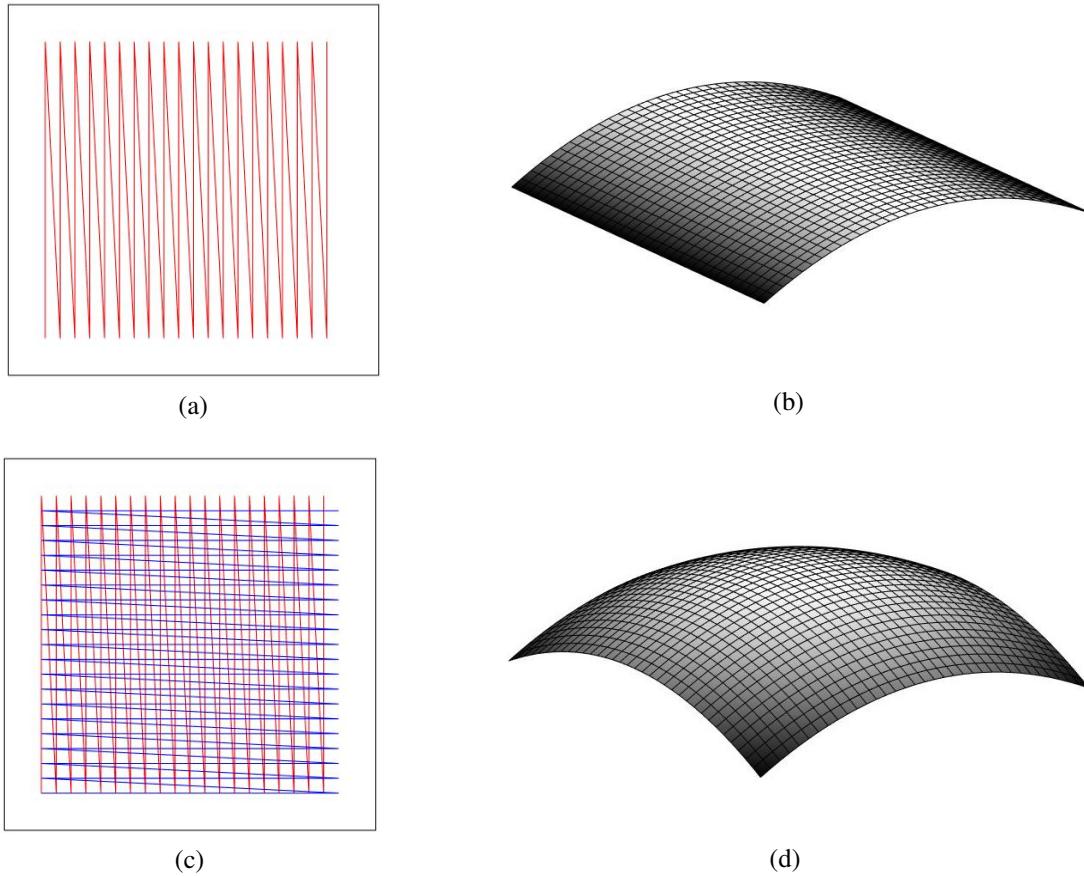


Figure 6: (a) ‘N’ shape tracking technique as described by [Longyard \(2014\)](#), (b) Shape of curve created from ‘N’ shape tracking technique, (c) Cross-wheeling tracking technique as described by [Longyard \(2014\)](#), and (d) Shape of curve created from cross-wheeling tracking technique

## 2.2 Parameterisation of flexible forming processes

The first objective is to form a parametric representation of the EW process that can be used in the design of experiments and ML model. [Hartmann et al. \(2016\)](#) present a method to parameterise the inputs and outputs of a neural network model shown in Figure 7 for automation of incremental free-forming (IFF). The desired digital part geometry and sheet metal blank are transformed using standardised discretisation. These are combined with material process parameters to form the input vectors to the neural network, which then predicts the parameterised form of the tool path strategy. Despite being applied to IFF instead of the EW, the concept is still applicable. This section will evaluate state-of-the-art techniques at representing the inputs and outputs of the system.

### 2.2.1 Tool path

[Opritescu et al. \(2012\)](#) presents a method to automate the driving process by defining a set of ‘tracking strategies’ that are applicable to standard sheet geometries. These strategies are defined by the force of the stroke combined with the 6 positional coordinates of the tool centre. It is proposed that these strategies can be combined and scaled to define geometrical variations of standard sheet geometries. From this, [Opritescu and Volk \(2016\)](#) analytically parameterises the tracking strategies so they can be interpolated to generate variations in the standard sheet geometries. This approach does not consider the infinite number of strategies that can create a specific geometry, instead relying on known strategies with predictable geometries.

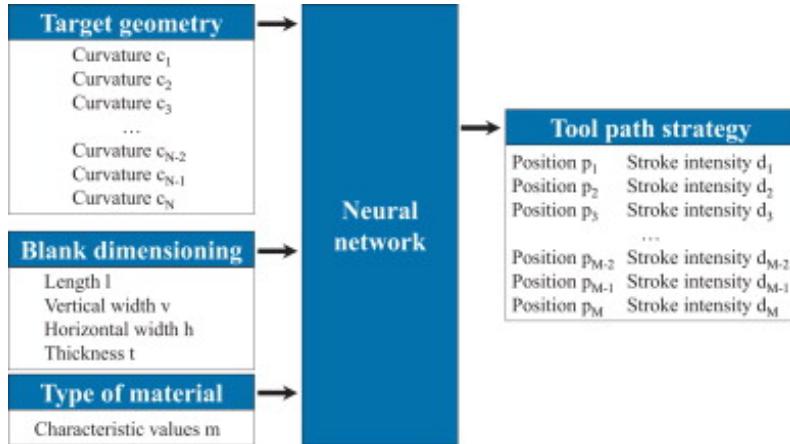


Figure 7: Input and output parameters of the neural network for automation of IFF (Opritescu and Volk, 2015)

Building upon this work, Hartmann and Volk (2019) proposes a non-parametric method to enable greater scalability of manufacturing strategies to encompass a wider variety of sheet geometries. In this study a stochastic approach is used to derive the tool paths, using a combination of probabilistic stroke density functions (PDF) and Voronoi partitioning. Despite minimising the deviation between the actual tool path and predicted tool path, there is no experimental example to show the accuracy and flexibility of the forming process using this technique.

## 2.2.2 Workpiece

The deformation of the workpiece can be represented locally using measurements of strain, or globally using geometrical characteristics. Bell et al. (2016) use a method to observe strain by drawing microscopic circles on the sheet using electrochemical etching. The variation in dimensions during deformation indicate shear and principle strain characteristics. This project does not have the resources to carry out this type of analysis, and will instead focus on the geometrical measurements.

In Figure 8, Ahmed et al. (2020) categorises geometry representation techniques as either Euclidean or non-Euclidean. Euclidean geometries are globally parameterised with respect to the same coordinate axes, whilst non-euclidean geometries are not globally parameterised and lacks vector space structure (Bronstein et al., 2017). The scope of this study encompasses volumetric, descriptor, projection, and point cloud techniques.

3D point clouds are an unstructured set of points that approximate a 3D geometry (Ahmed et al., 2020). This representation is most useful for capturing data using scanning techniques such as structured light metrology. Identifying geometrical features from these point clouds poses a challenge due to the lack of structure in the data (Ahmed et al., 2020). Instead, Zhang (2020) describes how this point cloud can be transformed extrinsically to give a volumetric representation. Volumetric data is a Euclidean-representation using a regular grid in 3D space. The geometry can be described as a surface given by equation 1, where  $(x, y, z)$  define a set of points.

$$z = f(x, y) \quad (1)$$

Using this form, the surface can be described via a volumetric polynomial patch. This can be represented as either a power basis that defines all points in space, or bases that define a restricted subdomain of the defining space such as the Bernstein-Bezier or B-spline basis (Bajaj, 1994). These algorithms compute planar triangular approximations by representing the geometry using intermediate control points as in Figure 9. The final explicit form considered is a projection from 3D to a 2D space. During this process there is information loss due to dimensionality reduction, however it enables well-researched learning models to be used that are applicable in 2D space (Sinha et al., 2016).

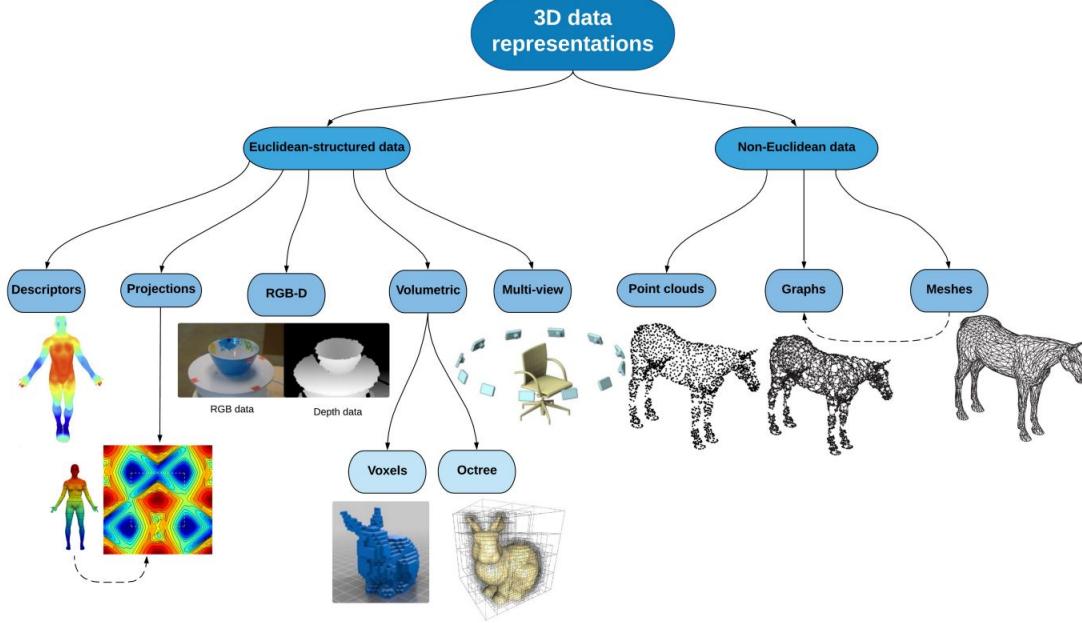


Figure 8: Categorisation of geometrical representation methods ([Ahmed et al., 2020](#))

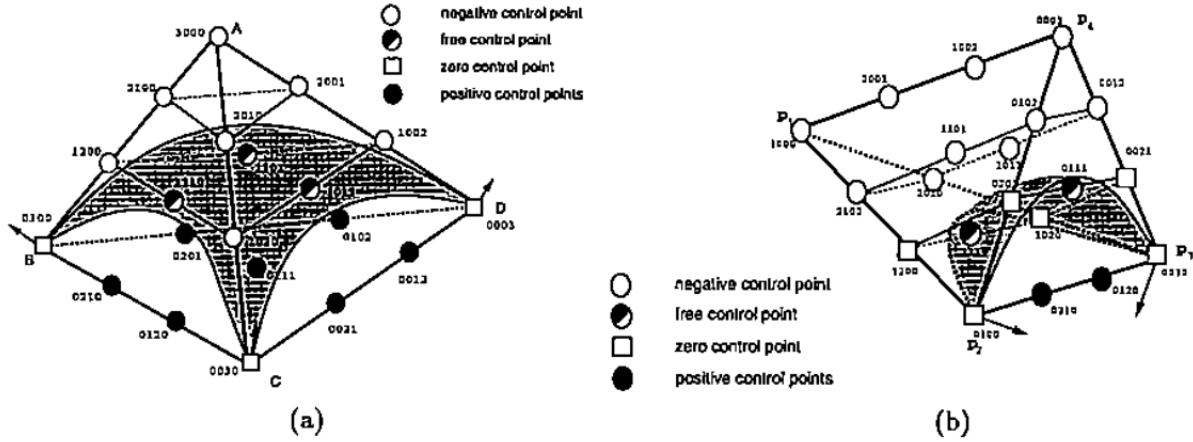


Figure 9: (a) A three sided patch tangent at E, C, D (b) A degenerate four sided patch tangent ([Bajaj, 1994](#))

Shape descriptors are simplified method to represent geometric characteristics. [Kazmi et al. \(2013\)](#) present multiple state-of-the-art shape descriptors such as view-based and transform-based descriptors that are used in 3D search engines and modelling systems. These techniques are advanced topics beyond the scope of this project. Instead, analytical shape descriptors such as curvature are considered. These descriptors can either be extrinsic, which is in reference to Euclidean space, or intrinsic, which is independent of Euclidean space.

Explicit measures of curvature include the maximum and minimum curvature measured at a point on the surface, termed the principle curvatures  $P_1$  and  $P_2$ . The mean of these two measurements,  $H$ , is a more general explicit representation. The product of the principle curvatures is known as the Gaussian curvature  $K$ , which is an intrinsic measure of curvature. When  $K = 0$ , the surface is described as developable and can be represented in Euclidean space ([Jia, 2020](#)). [Modes et al. \(2011\)](#) describe how stretching causes

non-developable surfaces when curvature is in more than one direction at a point, as is the case for the EW process mechanics. Therefore, doubly-curved surfaces can be described intrinsically using  $K$ .

### 2.3 Overview of machine learning techniques

A review of ML techniques is conducted to provide better understanding before a ML method is chosen. To provide the high-tolerance prediction capabilities required to minimise uncertainties in the flexible metal forming process, [Golle et al. \(2007\)](#) proposes a control architecture in Figure 10 that is composed of non-conventional open-loop and traditional closed-loop control methodologies. These non-conventional methods, such as ML, deploy open-loop control by using data collected before the operation to automate the manufacturing process. This hybrid architecture is used in a variety of mass production manufacturing processes such as injection moulding and machining ([Cho and Leu, 1998](#)), however it is has not yet been successfully implemented in a flexible forming context. This study will focus specifically upon non-conventional ML techniques due to the limitations in time and budget to implement a closed-loop control system.

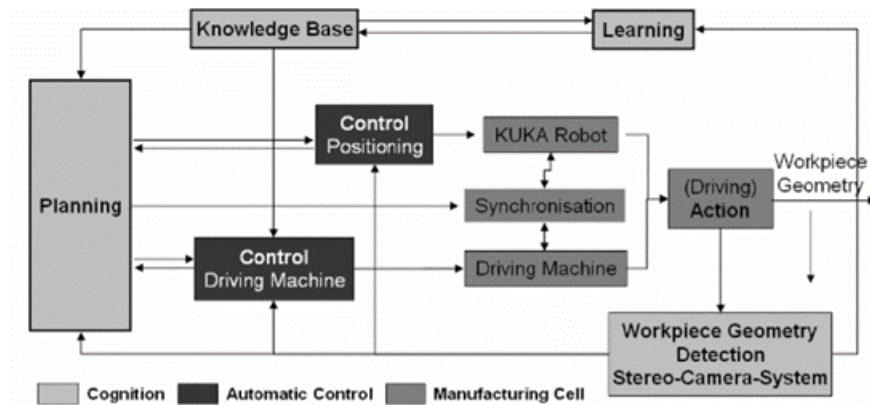


Figure 10: Ideal control architecture for flexible metal forming processes ([Golle et al., 2007](#))

[Alzubi et al. \(2018\)](#) categorise ML methods to be of ten classes. Of these, seven are relevant to this study; supervised, unsupervised, reinforcement, evolutionary, ensemble, Artificial Neural Networks (ANNs), and hybrid. Definitions and example algorithms of these techniques are detailed in Table 1.

These ML techniques can be categorised as either classification or regression. Classification and regression both use probabilistic techniques to predict an outcome, however the former defines a discrete output and the latter defines a continuous output ([Dasgupta and Nath, 2016](#)). Within classification, a problem can be defined as multi-class when there are more than one class that the output can be assigned to. If these classes are not defined as mutually exclusive, the problem can be defined as multi-label. In the case that each class is defined as a vector rather than a scalar, the problem can be defined as multi-output ([Xu et al., 2020](#)).

All ML techniques are applicable to 3D data, however have to be applied differently depending on the geometrical representations defined in Figure 8. [Shi et al. \(2015\)](#) introduces a technique called DeepPano, that uses a 2D CNN architecture to train a model on projected data. The input data is a cylindrical projection around an objects principle axis. This is more applicable for geometries with complex shapes rather than sheet blanks.

ShapeNet is a 3D ANN architecture for processing volumetric representations ([Wu et al., 2014](#)). It takes an input of a depth map and converts it to a representation in volumetric space using voxels, which it applies to a CNN for classification of images. This approach is found to be highly computationally expensive due to the large number of parameters within the convolutional layer ([Sinha et al., 2016](#)). Instead, a reduced architecture LightNet is proposed by [Zhi et al. \(2018\)](#), that achieves a faster convergence with less parameters.

Table 1: ML techniques and their applications ([Alzubi et al., 2018](#))

Class	Definition	Algorithms
Supervised	A dataset of features with corresponding labels are provided to the model for training. Based on this, the model detects a pattern between the features and labels, which enables predictions of labels to be made of a new set of features ( <a href="#">Praveena and Jainaesh, 2017</a> ).	Decision-trees, Naive-Bayes, Support vector machines (SVMs), logistic regression
Unsupervised	An unlabelled dataset is provided to the model, from which is must derive a rule set based on underlying patterns. From this rule set, classification predictions can be made.	K-means clustering, Hierarchical clustering
Reinforcement	Regarded as learning by trial and error, the algorithm is rewarded if the output is correct and punished if not. With a greater amount of data, the algorithm becomes more optimally tuned ( <a href="#">Salvador et al., 2020</a> ).	Upper confidence bound (UCB), Thompson sampling
Evolutionary	Algorithms take inspiration from biological processes to adapt to situations based on a fitness function ( <a href="#">Witczak, 2007</a> ).	Genetic algorithms (GA), Genetic programming (GP), Evolutionary programming (EP)
Ensemble	Many individual models are combined to solve a common problem. This has the benefit of reducing bias and variance, and improving precision.	Random Forest
ANN	Based on the biological construct, ANNs are made of layers of nodes that have weighted interconnections. This network learns by adjusting these weightings to increase accuracy of predictions.	Artificial neural networks, Convolutional neural networks
Hybrid	In some instances, ensemble models result in a reduced accuracy when applied to multiple classifiers. Instead, a hybrid approach of heterogeneous strategies can be used (ie. combining supervised and unsupervised techniques) to improve accuracy.	Variable, eg. <a href="#">Hou et al. (2019)</a> propose a neural network-integrated by auto-regressive integrated moving average using a fuzzy method to estimate urban traffic flow

## 2.4 Non-conventional control applied to flexible metal forming

The state-of-the-art in automating traditional flexible metal forming processes is now evaluated to inform the choice of ML method. In literature, ANNs are most commonly applied to flexible forming processes. [Hamouche and Loukaides \(2018\)](#) use a deep convolutional neural network (CNN) architecture for classification of geometries based on the forming process that was used. This process is commonly used in computer recognition processes such as object recognition ([Aloysius and Geetha, 2018](#)), however in this context it proves useful for recognising domain specific features with an accuracy of 89%. These geometries are derived analytically, and the overlooks the intermediate process of digitisation.

[Hartmann et al. \(2016\)](#) use an ANN architecture to predict the tool path required to create a target geometry in incremental metal sheet free-forming applications. Figure 11 shows the workflow for this manufacturing system, which requires the geometries to be digitised before implementation in a 3D neural network model. They use theory presented by [Opritescu and Volk \(2015\)](#) to build the model, which involves the discretisation of the sheet geometry into individual values for curvature. The methods proposed are of limited practical applicability as the exact NN methodology is unclear and no metric of accuracy is given.

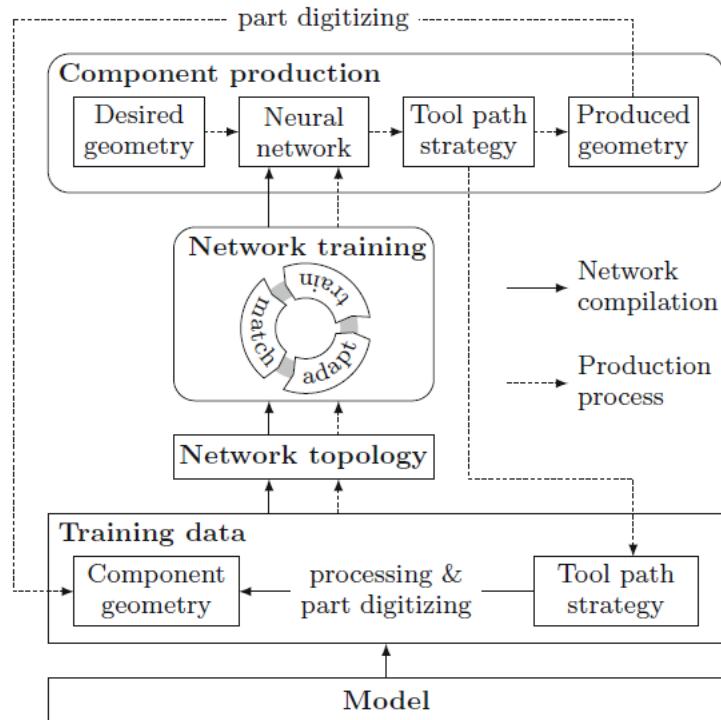


Figure 11: 3D neural network approach taken by [Hartmann et al. \(2016\)](#)

There are no examples of use of reinforcement-learning applied to flexible forming in literature, however there are many examples of use with mass production techniques. [Liu et al. \(2020\)](#) use a reinforcement learning algorithm in combination with a deep neural network to improve repeatability in the stamping of sheet metals. Successful training of the model was only possible with a large amount of data, which could only be provided via simulation.

[Rentsch \(2018\)](#) has created a method to control the autonomous operation of the spinning machine. This uses a finite element model (FEM) to calculate the deformation characteristics during the spinning operation to generate a target geometry. The shortest processing time was calculated to be 38.5 hours per geometry, which is not feasible for commercial use. Instead, a heuristic method is proposed that is more computationally efficient but less accurate.

[Ilangoan et al. \(2016\)](#) has created a automated hammering process named Mechatroforming® that uses a mechatronic system to monitor and control the process in real-time. The control architecture uses finite element analysis (FEA) to predict the incremental deformation and the next tool path. This approach is limited for use in flexible sheet forming of multi-point deformation as it is computationally inefficient. [Mori et al. \(1996\)](#) use an evolutionary algorithm to determine a hammering sequence depending on the difference between obtained and desired curvatures. It was found that the algorithm was computationally inefficient, therefore empirical rules were used to generate the shape during an initial ‘roughing’ procedure and the algorithm was used after during the ‘finishing’ procedure.

More relevant to this study, [Vazquez and Coleman \(2017\)](#) have developed a system that uses a Kuka 6-axis serial robot to mechanise the wheeling process. They develop an empirical procedure where ‘points of interest’ are identified as areas with significant Gaussian curvature. From this, the principle direction of curvature can be determined and a tool path can be created. This assumes that the mechanics of the wheeling process can be described entirely by the Gaussian curvature of the sheet, however it serves as a significant proof-of-concept. [Rossi and Nicholas \(2018\)](#) design a cyber-physical system that uses an ANN model to derive toolpath strategies whilst incorporating a Kinect visual sensor to scan the sheet and provide closed-loop feedback. The research was limited in its scope as it only had a sample size of 6 sheet blanks and only focused on varying one process parameter.

A similarity between all aforementioned examples is that the control and prediction techniques implemented do not result in comparative performance to the smith and do not explore the full capabilities of the flexible forming process. This will be investigated in the following study.

### 3 Parameterisation of the EW process

An overview of the parameters described in this section is displayed in Figure 12.

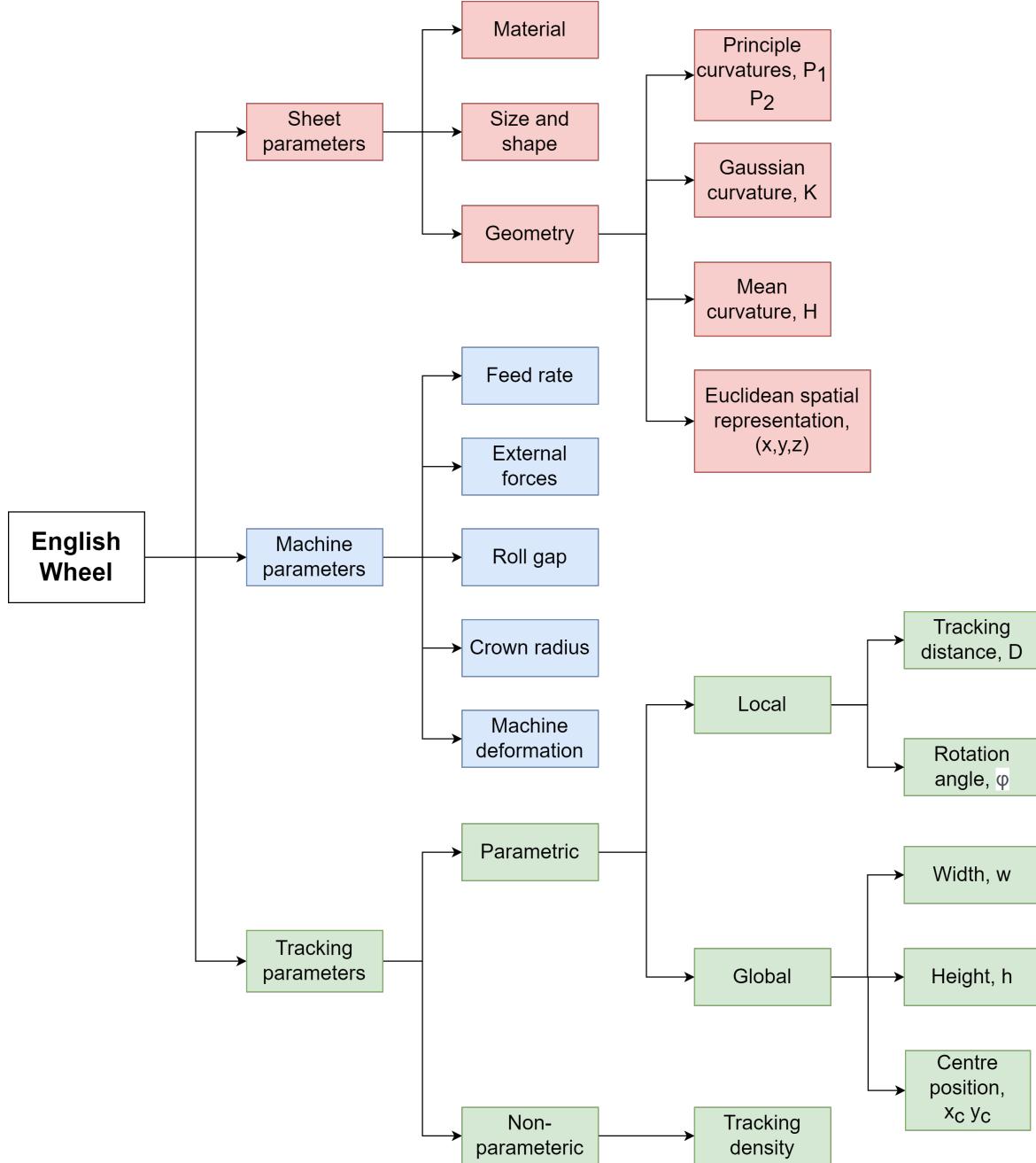


Figure 12: EW process parameters

#### 3.1 Sheet parameters

Parameterisation of the sheet blank is first considered. To simplify the process, only rectangular sheets are considered. For future transformational purposes, the sheet is described explicitly in relation to global axes as displayed in Figure 13. For comparison of sheet geometries, the position and orientation of the global axis must be the same across all measurements.

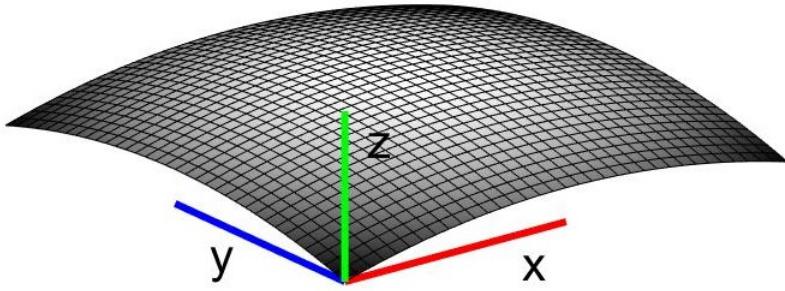


Figure 13: Global axes for explicit definition of sheet geometry

The x and y axes are normalised according to equations 2 and 3. This is to allow data to be scaled and compared more easily.

$$x_{in} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (2)$$

$$y_{in} = \frac{y_i - y_{min}}{y_{max} - y_{min}} \quad (3)$$

where  $x_{in}$  and  $y_{in}$  are the normalised values for x and y,  $x_i$  and  $y_i$  are the x and y positions,  $x_{min}$  and  $y_{min}$  are the minimum x and y values, and  $x_{max}$  and  $y_{max}$  are the maximum x and y values.

At a specific instance, the sheet can be defined via its material, dimensions, shape, initial geometry, and deformed geometry. The geometry is can be expressed explicitly, as defined in equation 1. This representation can be transformed to represent curvature at each  $(x, y)$  coordinate. The forms of curvature used in this study include Gaussian  $K$ , mean  $H$ , and the maximum and minimum principle curvatures,  $P_1$  and  $P_2$ . The first and second fundamental forms can be used to calculate these values analytically (Zhang, 2020). From the explicit definition, a parametric form of the surface,  $r$ , can be derived such that

$$r(u, v) = (x(u, v), y(u, v), z(u, v)) \quad (4)$$

The first fundamental form of a surface is given by  $v \cdot v$ , where  $v = r_u du + r_v dv$ .  $r_u$  and  $r_v$  are the first partial derivatives of  $r$  with respect to  $u$  and  $v$ . This can be rewritten as

$$Edu^2 + 2Fdudv + Gdv^2 \quad (5)$$

where  $E = r_u \cdot r_u$ ,  $F = r_u \cdot r_v$ , and  $G = r_v \cdot r_v$ .

The second fundamental form is with respect to a family of surfaces that vary in relation to a variable  $t$ . This is given by

$$R(u, v, t) = r(u, v) - tn(u, v) \quad (6)$$

The second fundamental form is given by

$$Ldu^2 + 2Mdudv + Ndv^2 \quad (7)$$

where  $L = r_{uu} \cdot n$ ,  $M = r_{uv} \cdot n$ , and  $N = r_{vv} \cdot n$ .  $r_{uu}$  is the second partial derivative of  $r$  with respect to  $u$ ,  $r_{vv}$  is respect to  $v$ , and  $r_{uv}$  is respect to  $u$  and  $v$ .

From this  $K$  is given by

$$K = \frac{LN - M^2}{EG - F^2} \quad (8)$$

and  $H$  is given by

$$H = \frac{EN + GL - 2FM}{2(EG - F^2)} \quad (9)$$

From this,  $P_1$  and  $P_2$  can be determined.

$$P_1 = H + \sqrt{H^2 - K} \quad (10)$$

$$P_2 = H - \sqrt{H^2 - K} \quad (11)$$

The instance during the tracking process at which these measurements are taken is important. The deformation mechanics of the wheeling process is continuous, but for the purposes for this study it will be described as discrete. The measurement of the geometry will only happen before and after a pass has been completed. This reduces the resolution of the process to the change in geometry  $\delta$  caused by one pass. This change of geometry applies to all geometrical descriptors measured at an arbitrary point on the sheet. Given the history of the geometry measured after each pass,  $\delta$  can be calculated for each pass. From this, overall changes in geometry  $\Delta$  from instance  $a$  to  $b$  can be calculated from the sum of the geometrical changes for each pass,

$$\Delta = \sum_{i=1}^n \delta_i \quad (12)$$

where  $n$  is the total number of passes between instance  $a$  to  $b$ , and  $i$  is the pass number.

### 3.2 Machine parameters

The machine being used for this study is presented in Figure 14. It has 8 degrees of freedom to enable rotation and translation operations of the sheet. This is so tracking patterns similar to those discussed in Section 2.1 can be wheeled. The top wheel is motorised and governs the feed rate of the process determined by the operator, represented as axes 1 in Figure 14. This wheel has a radius approximately double that as the lower wheel, and has a flat profile. The roll gap is the distance between the top and lower wheels. The lower wheel is actuated so that the roll gap can be altered, and is represented by axes 2 in Figure 14. Like the traditional process, The crown of the lower tool can be customised to control the curvature of part produced.

The movement of the sheet is governed by front and rear vacuum bellows that can each move in three degrees of freedom, indicated in Figure 14 as axes 3-8. These can move to create a tracking pattern on the sheet, which is parameterised in the subsequent section. Additional external forces can be applied to the sheet, as is the case in manual wheeling with slight pressure applied to the edges of the sheet to induce curvature. During wheeling, deformation occurs within the machine itself and can vary in different setups due to the material, geometry of parts, and the temperature of the environment.

### 3.3 Tracking pattern parameters

#### 3.3.1 Parametric approach

An example ‘N’ tracking style is presented in Figure 15a. This pattern can be parameterised locally by dividing it into a vector of nodal positions  $P$

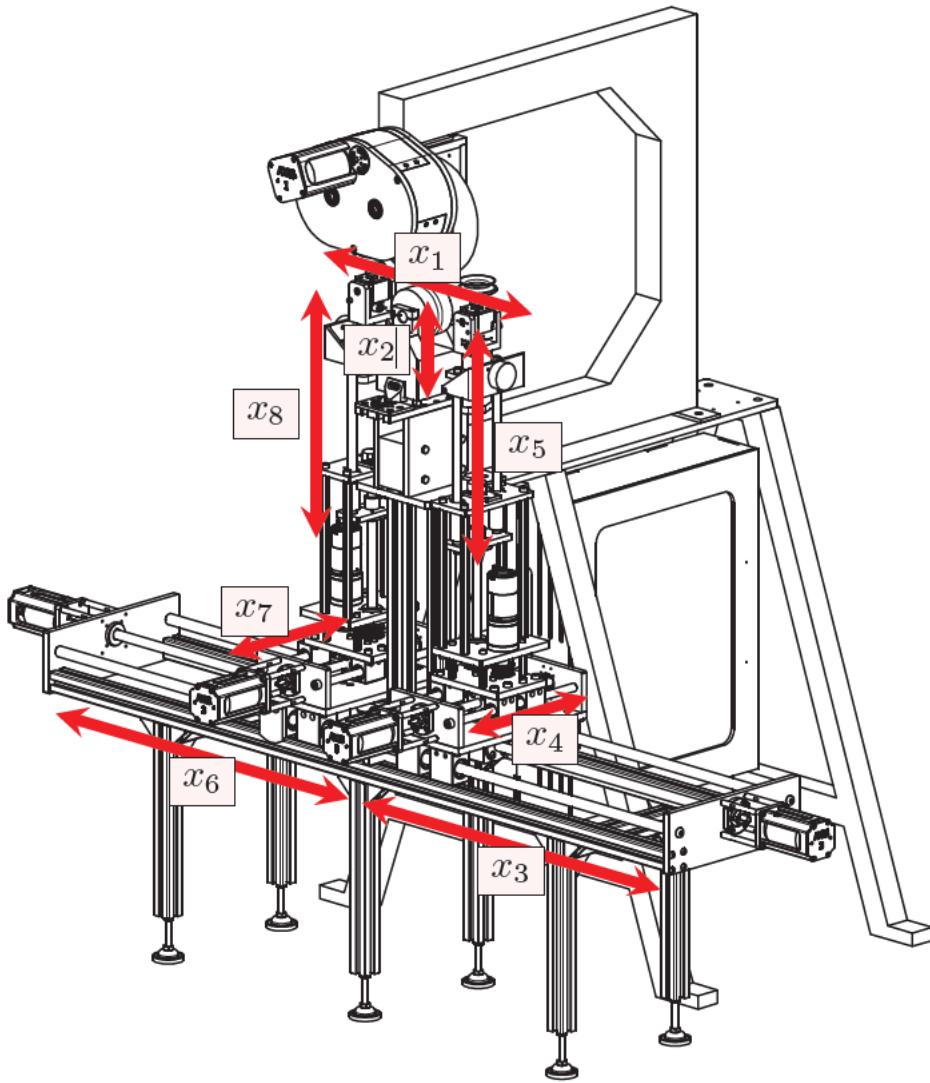


Figure 14: Machine with axes labelled

$$P = [p_0 \ p_1 \ p_2 \ p_3 \ \dots \ p_n]$$

where  $p_n$  is the final nodal position in the tracking pattern. These are shown as blue crosses. The tracking patterns used within this study will be regular, and can be described to have a constant tracking spacing  $t$  and tracking length  $d$  as shown in Figure 15a.

The tracking length will vary depending on the pattern used, for example when using an 'N' tracking style there is a repetitive sequence of a higher then lower value of  $d$ . This sequence is represented as a vector  $D$

$$D = [d_0 \ d_1 \ d_2 \ \dots \ d_n]$$

For the case of the 'N' tracking style,  $D = [d_0 \ d_1]$  as shown in Figure 15a, where

$$d_1 = \sqrt{d_0^2 + t^2}$$

The angle  $\phi$  between the previous and next track for each node in this case is constant, and is equal to

$$\phi = \text{atan}(t/d) \quad (13)$$

From this, a rotation matrix  $R_1^0$  can be calculated from  $\phi$  and a translation vector  $O_1^0$  from  $D$ , that describe  $p_1$  relative to  $p_0$ . This can be used to derive the homogeneous transformation matrix  $T_1^0$

$$T_1^0 = \begin{bmatrix} R_1^0 & O_1^0 \\ 0 & 1 \end{bmatrix} \quad (14)$$

The inverse of this matrix can be used to find  $p_1$

$$p_1 = (T_1^0)^{-1} p_0 \quad (15)$$

Given that the value of  $n$  is known, all values of  $P$  can be found.

The local definition is useful for practical application of the wheeling path, however a simplified form is required for describing the tracking patterns. Global characteristics such as the shape, size, and position can be used to define each pattern. In this study, each tracking pattern is confined to be rectangular in shape. Therefore, each pattern can be described explicitly as in Figure 15b using the central x and y position  $x_c$   $y_c$ , the width  $w$ , and height  $h$ . From this the local parameters can be determined, with  $p_0 = [x_c - \frac{w}{2}, y_c - \frac{h}{2}]$ , and  $D$  can be calculated given  $h$  and the specified tracking style.

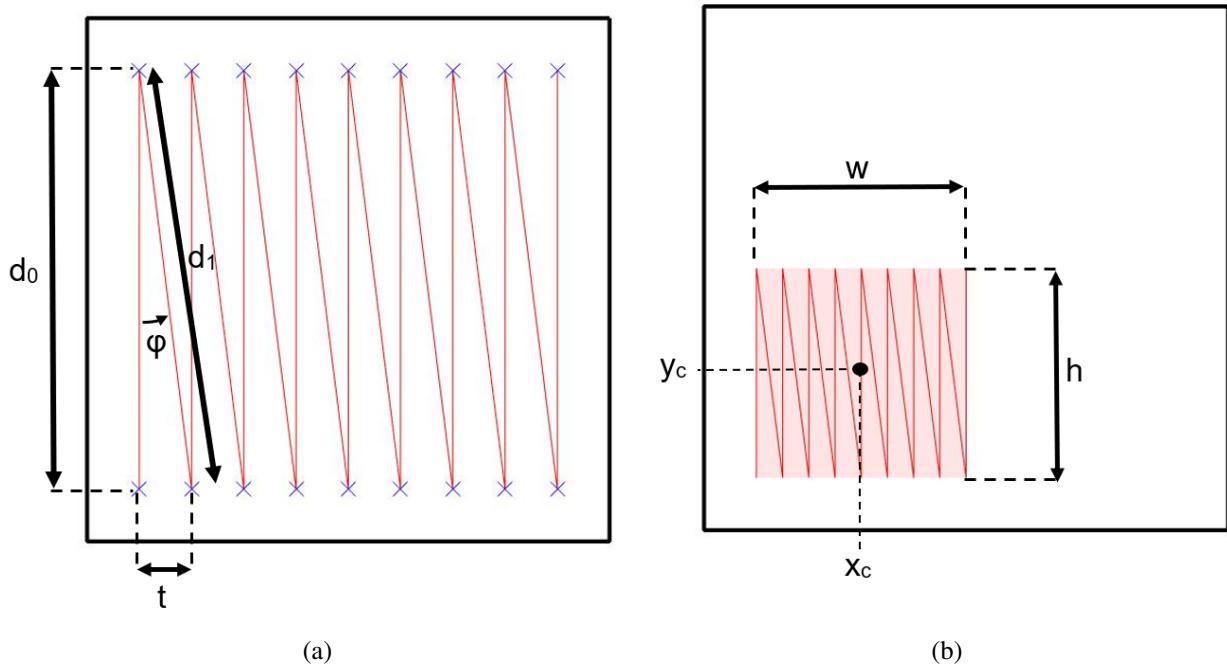


Figure 15: Parameterisation of tracking pattern using (a) local and (b) global characteristics

### 3.3.2 Non-parametric approach

A parametric approach is necessary for practical implementation of tracking patterns. However, when using this approach to describe multiple different tracking patterns that may overlap, the parameters becomes complex and specification bias arises for patterns that have varying characteristics (Botev et al., 2010). A non-parametric approach eliminates this problem and enables greater scalability between variations of patterns. It also allows for easier visualisation of the tracking patterns which will be useful later in the study.

The non-parametric approach is defined by the ‘tracking density’ shown in Figure 16b. This is a stochastic representation of the tracking path, with a higher density indicating it is more likely that area will be tracked. To compute this, the sheet is divided into an equally spaced grid of resolution equal to the tracking spacing. It is assumed that during one pass the tool makes contact with each cell within the tracking area twice, as in Figure 16a. A cumulative score of how often each cell within the grid is wheeled is recorded. This is input into a kernel density estimation function to determine the tracking density.

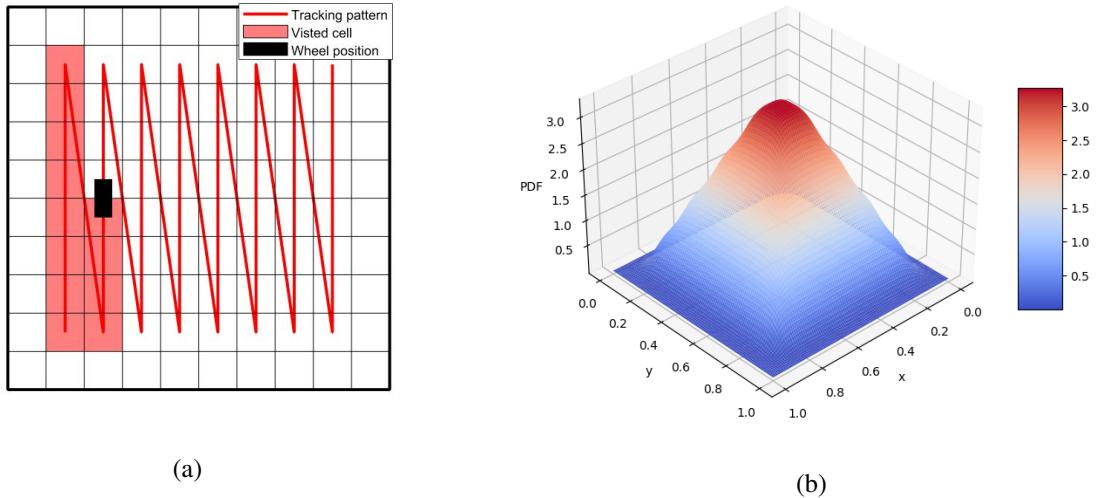


Figure 16: (a) Logging of discretised wheeled regions for processing in kernel estimation. (b) PDF representing the tracking density created using a kernel density function.

The assumptions made in this process do not consider that due to the overlapping tracking and angled path of the tool that some regions within the tracked area have more contact than others. Also, they do not consider the time the tool is at rest, such as a transfer and pivot points, and other process parameters such as feed rate and roll gap. However, the representation does not affect the practicalities of the experiment, and these factors can be considered during analysis.

## 4 Design Convergence Study

In this section a controlled convergence study is conducted to ensure the design of experiments is optimal given the relevant design specification. This will involve a phase of conceptualisation where the design process diverges, followed by a phase of elimination where the scope is narrowed. From this narrowed scope concepts are created and the process repeats until a final concept is decided upon.

### 4.1 Experimental design specification

The experiment aims to generate a data set that is representative of global deformation phenomena experienced by sheet metal during the EW process. This is for two primary objectives:

1. The data set will be applicable in the context of future research into EW sheet mechanics.
2. The data set will be used in this study to train and test the effectiveness of a ML model for making predictions.

These objectives will be fulfilled by following the design specification outlined in Table 2. ‘A’ listed criteria indicates musts, and ‘B’ listed criteria indicates wants. Reasons for how these apply to each objective are given in Appendix A.1.

Table 2: Specification for the design of experiments

Ref.	Criteria	Description
A1	Accuracy of measurement	Data needs to be defined to a specific accuracy
A2	Reproducibility of experiment	Method needs to be presented to allow experiment to be reproduced
A3	Repeatability of experiment	Procedure repeatable for use on the same machine in the same conditions
A4	Relevant results	Experiment relevant to industrial forming processes and traditional techniques used by artisans
A5	Amount of data	Strategy must be able to generate a large amount of data
A6	Diversity of data	Data must cover numerous strategies in order to broaden understanding of global deformation
A7	Scalability of data	The method in which the data is represented must be scalable to represent a spectrum of patterns for one manufacturing strategy
A8	Bias in data collected	Bias in the data must be accounted for and minimised
B1	Machine capabilities	The necessary resources and equipment are available to carry out the experiment
B2	Logistics of metrology	Equipment is easy to operate and is available for use during periods of testing
B3	Accuracy of metrology equipment	Equipment provides accurate and repeatable results
B4	Timespan of experiment	Experiment can be conducted within the timespan of the project

## 4.2 Parameter Study

There are many different methods used in the design of experiments in literature, such as one factor at a time (OFAT) design, factorial design, and Taguchi's methods (Yuangyai and Nembhard, 2010). Before these methods can be considered, the feasibility of each EW process parameter needs to be assessed to reduce the sample space. The process parameters are evaluated against feasibility criteria in Table 3. A parameter is only considered to be feasible if it complies with all criteria.

Table 3: Parameter elimination matrix

Reference	B1	B2	B3	B4
Criteria	Does the machine have the capability to vary this parameter?	Does the machine have the capability to measure this parameter?	Is the measurement equipment accurate enough?	Is it feasible in the timespan?
Sheet parameters	Size	Y	Y	Y
	Shape	Y	Y	Y
	Material	Y	Y	N
Machine parameters	Roll gap	Y	Y	Y
	Tool type	Y	Y	Y
	External forces	N	N	N
	Feed rate	N	Y	Y
	Machine deformation	Y	N	N
Tracking parameters	Area	Y	Y	Y
	Location	Y	Y	Y

The experimental setup did not have the sensory capacity to measure the external forces applied to the edge of the sheet and the deformation of the machine due to lack of resources. The sheet material was fixed as there was limited options and applications for testing. The feed rate of the machine is fixed so variation could not be measured. Process parameters that were deemed feasible include the distance of the tracking pattern from the edge, the roll gap, tool type, tracking pattern, area of tracking pattern, and size of sheet.

## 4.3 Experimental concepts

The budget of £250 only allowed 25 steel sheets to be ordered. It was decided 20 sheets would be used for training, tuning and testing the machine learning model virtually, while the remaining 5 would be used for testing the accuracy of the model on the machine. This provides a sufficient amount of data to build a model based upon the variation of one process parameter. This meant implementing an OFAT design methodology. OFAT experimental concepts were generated from the selected process parameters, which can be seen in Table 4. Visual explanations of these strategies can be viewed in Appendix A.2.

Table 4: Experimental concepts for the EW. For visual explanation of these concepts see Appendix A.2

Ref.	Name	Procedure	Interest
A.1	Tracking strategies from heuristic techniques	Parameterise tracking 'strategies' proposed in wheeling manuals such as <a href="#">Longyard (2014)</a> for classification of sheet geometries.	<a href="#">Longyard (2014)</a> and <a href="#">Barr (2013)</a> classify tracking patterns that produce predictable deformation in the sheet, which could be applied to a ML model.
A.2	Tracking spacing	For each sample the distance between each track is varied by incremental amounts whilst the tracking pattern remains constant.	This would identify a relationship between the tracking spacing and global curvature. However, it is widely accepted that the spacing should be invariant and small to keep the thickness of the sheet uniform.
B.1	Area wheeled with reference to a point	For each sample a different point is selected on the sheet and with each pass the area of wheel is increased or decreased incrementally.	Identifies relationship between the location wheeled on the sheet and global deformation.
B.2	Area wheeled with reference to a line	Similar to B.1 the area of the tracking pattern is varied between passes, however this area radiates from a line rather than a point.	Similar to B.1
C.1	Distance from the edge of the sheet	For each sample the distance of the centre of the tracking pattern from the outer edge is varied.	This experimental concept investigates the sheet mechanics if there is no material to restrict plastic deformation surrounding the wheeled region. Gain a greater understanding of wrinkling.
C.2	Discrete cellular wheeling	The sheet is divided into a grid of equally sized cells. Each pass involves wheeling a cell with a constant tracking pattern. For each sample a different sequence of cells are wheeled.	Assesses the relationship between the location of wheel and resulting deformation of the sheet.
C.3	Discrete wheeling striped	The sheet is divided into sections that are of equal width and extend down the entire length that can be wheeled. Each pass involves wheeling one section with a constant tracking pattern. For each sample a different sequence of sections are wheeled.	Similar to C.2.
D.1	Tool type	For each sample a different tool is used with variable crown and blow radii. As there are only five anvil pieces, another parameter must be varied for each tool tested.	Assess the effect of the tool piece used on the curvature of the sheet.

Table 4: Experimental concepts for the EW. For visual explanation of these concepts see Appendix A.2

Ref.	Name	Procedure	Interest
E.1	Reaction force with constant roll gap	For each sample the reactant force acting on the machine is measured with the roll gap remaining constant. This is measured for several passes. The roll gap is varied for each sample.	<a href="#">Longyard (2014)</a> explains how the roll gap needs to be varied during the wheeling process due to maintain constant pressure.
E.2	Effect of pressure on curvature	For each sample, a different pressure is applied to the sheet. The roll gap is varied to maintain constant pressure during each pass, and curvature is measured.	This assesses the effect of pressure on generated curvature.
F.1	Size of sheet	For a range of sheets with the same shape but scaled to different dimensions, a similar tracking pattern is used and the resultant global deformation of the sheet is measured.	Assesses the scalability of tracking patterns to variations of known geometries.
G.1	Shape of sheet	For a range of sheets with varying shapes, a similar tracking pattern is used and the resultant global deformation of the sheet is measured.	Similar to F.1.

#### 4.4 Evaluation of experimental concepts

The strategies were evaluated against relevant design criteria. Each strategy was compared using Pugh's controlled convergence matrix in Table 5, where strategy A.1 was selected as the datum. Strategies F.1 and G.1 were eliminated as there is no capability to machine all 25 sheets within the time span. Strategy E.2 was eliminated as the machine lacks the measurement and control capabilities to maintain the pressure of the tool acting on the sheet. Strategy E.1 was eliminated as there is no capability to measure the resultant force acting on the tool.

Of the remaining strategies, A.2 does not have results of significant relevance as the tracking spacing is held constant in industry to ensure uniformity in the thickness of the sheet. D.1 lacked scope in terms of the data collected as there are only 4 different anvil pieces that can be tested. The remaining strategies involve varying globally defined tracking pattern characteristics such as area and location. The advantage of A.1 compared to these concepts is that heuristic tracking strategies create known geometries that have industrial applications, therefore it is more reliable for producing a relevant dataset. Given these relative benefits, A.1 was selected as the final concept.

#### 4.5 Tracking strategies

Conceptual strategies with known geometries are described in Table 6 and illustrated in Figure 18. Each strategy is comprised of a set number of passes, and in this representation they each have 3. Figure 17 illustrates the centre-to-outside strategy tracking and the order of tracking patterns. The individual passes for the strategies shown in Figure 18 can be inferred from the descriptions in Table 6, and are given in Appendix A.3 for further clarification.

These strategies have been abstracted from EW manuals such as [Longyard \(2014\)](#) and [Barr \(2013\)](#). Each strategy has the condition that the tracking pattern of each pass cannot overlap itself. To derive each individual tracking pattern from the strategy, each strategy has accompanying characteristic equations. These are presented in Appendix A.3.

Table 5: Pugh's controlled convergence matrix for evaluation of experimental concepts

		Musts		Wants			
Reference		B1	B2	A4	A5	A6	Total
Criteria		Ability to vary independent parameter	Ability to measure dependent parameter	Relevance of results	Amount of data	Scope of data	
Concepts	A.1	Y	Y	Datum	Datum	Datum	0
	A.2	Y	Y	-	o	-	-2
	B.1	Y	Y	-	o	o	-1
	B.2	Y	Y	-	o	o	-1
	C.1	Y	Y	-	o	o	-1
	C.2	Y	Y	-	o	o	-1
	C.3	Y	Y	-	o	o	-1
	D.1	Y	Y	o	o	-	-1
	E.1	Y	N				-
	E.2	N					-
	F.1	N					-
	G.1	N					-

Table 6: Tracking strategy concepts

Name	Description	Ref.
Centre-to-outside	An initial area is tracked in the centre. In this case it will be a square. This dimensions of this area is then expanded linearly for each subsequent pass.	Figure 17a
Diagonal	Two points are selected at equal distances but from the opposite edges of the sheet. These are the centre points of the first and last rectangular tracking patterns in the strategy, with a defined number of pattern linearly spaced in between.	Figure 18a
Vertical expansion	A position is selected on the x axis that is the centre point of the initial tracking pattern. This has the smallest width, with the height defined as a constant. The consequent tracking pattern have the same central point but have a width determined by a scaling factor.	Figure 18b
Horizontal expansion	A position is selected on the y axis that is the centre point of the initial tracking pattern. This has the smallest height, with the width defined as a constant. The consequent tracking pattern have the same central point but have a height determined by a scaling factor.	Figure 18c
Overlaid	Each tracking pattern has the same central point, height and width.	Figure 18d

Table 6: Tracking strategy concepts

Name	Description	Ref.
Vertical tracking	A special case of horizontal expansion, but the tracking pattern only expands in one direction, not two.	Figure 18e
Horizontal tracking	A special case of vertical expansion, but the tracking pattern only expands in one direction, not two.	Figure 18f
Triangular	The tracking patterns accumulate to form a triangular shape. The first tracking pattern is as wide as the base with the smallest height, with the last equal to the height of the triangle with the smallest width.	Figure 18g

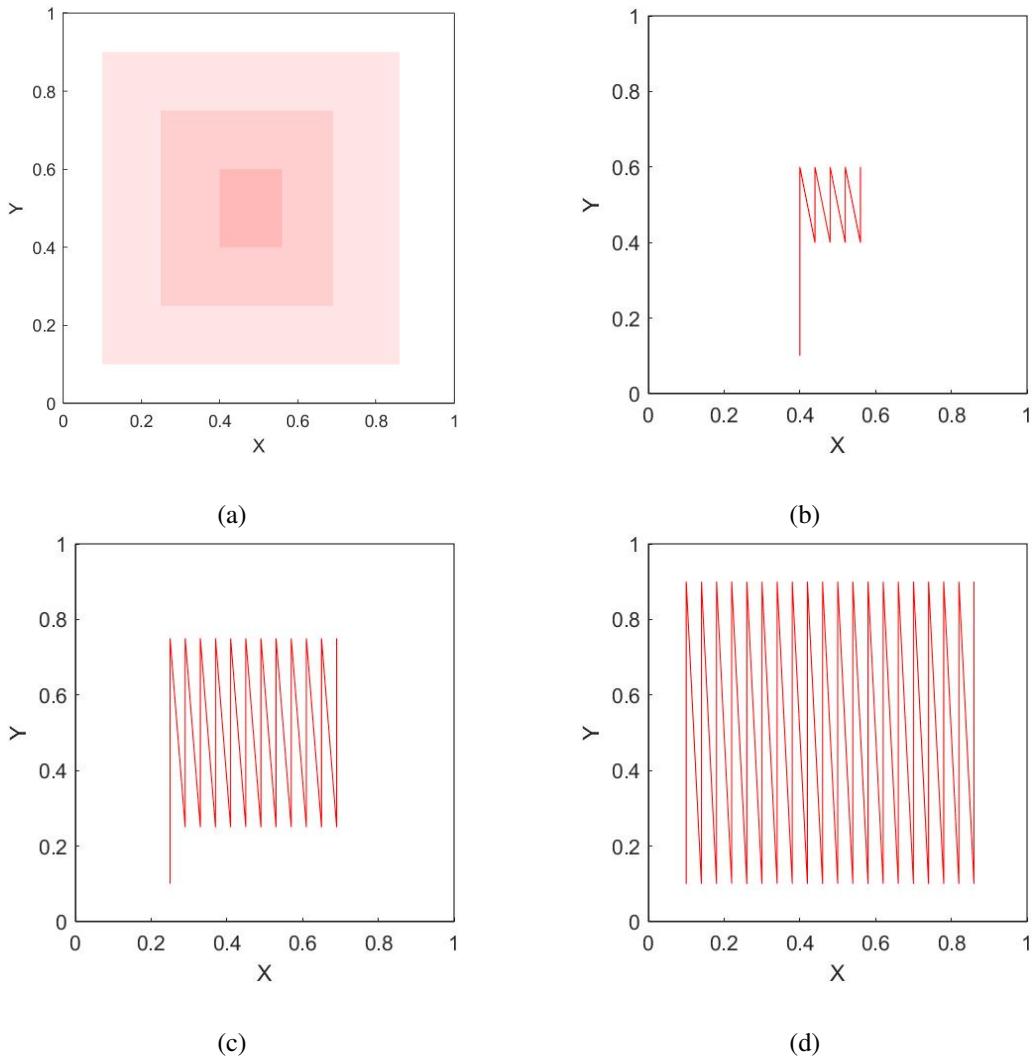


Figure 17: (a) Tracking density for the centre-to-outside tracking strategy, and (b), (c), and (d) showing the tracking pattern for each pass. Each plot is normalised between 0 and 1.

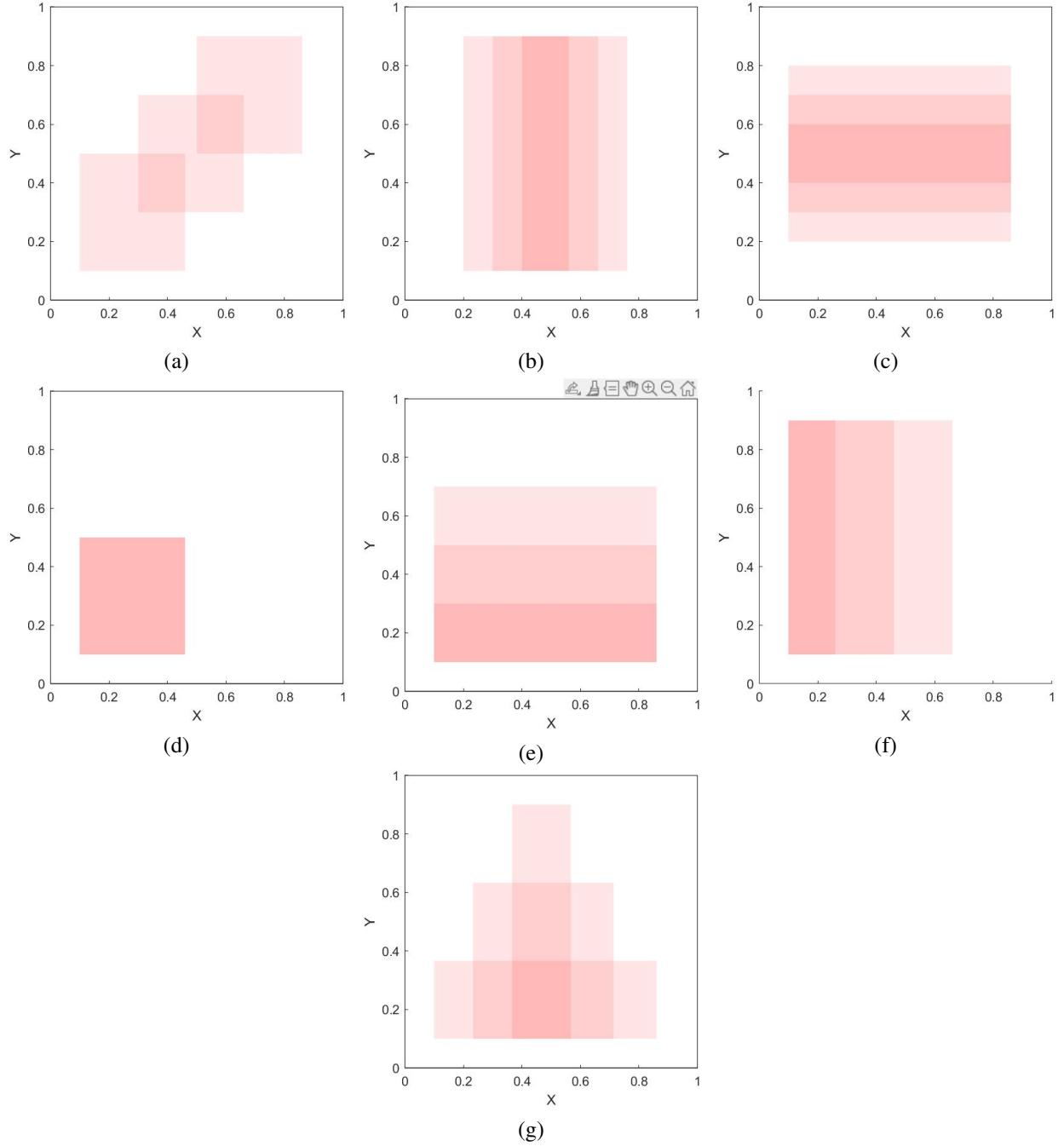


Figure 18: Tracking density for the (a) diagonal, (b) vertical expansion, (c) horizontal expansion, (d) overlayed, (e) vertical tracking, (f) horizontal tracking, and (g) triangular strategies. Each plot is normalised between 0 and 1.

#### 4.6 Evaluation of tracking strategies

There are 25 samples available to develop a dataset. This section only considers the experimental setup for the 20 sheets used for training the model, whilst the 5 comprising the test set will be randomly generated when required. Due to limitations in time for testing, the number of passes for each sample was limited to 4. It was decided to limit the number of strategies tested to five to ensure that the quantity and variation in data

for each strategy was substantial to give greater potential for the ML model to classify different strategies accurately. The choice for the selection of strategy was based on the diversity of geometry produced.

The horizontal and vertical tracking are a slight variation of the vertical and horizontal expansion tracking strategies, and thus have similar mechanics. Therefore, only the vertical and horizontal expansion tracking strategies were considered. The diagonal tracking strategy was found to be used less by traditional artisans, and was disregarded. This results in the five selected strategies: centre-to-outside, vertical expansion, horizontal expansion, overlayed, and triangular.

To assess the variation in geometries between strategies, geometries was derived heuristically based upon standard forms of these strategies with 4 passes. Figure 19a shows that the ‘centre-to-outside’ strategy produces a domed shape which is doubly-curved. This is also true for Figures 19b and 19e for the ‘overlayed’ and ‘triangular’ strategies. The ‘horizontal expansion’ strategy displays the greatest curvature in the y direction. Note this is not an exact representation but an approximation for visualisation purposes, and should not be used to analyse the process mechanics.

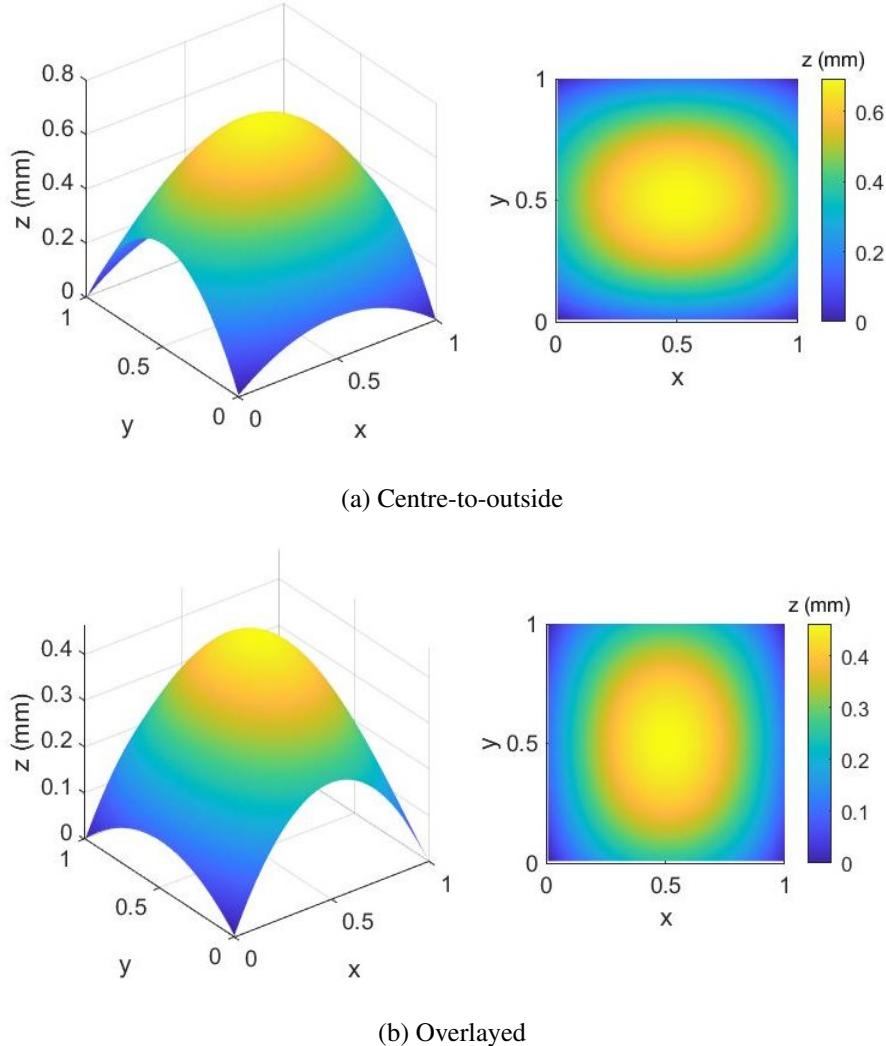
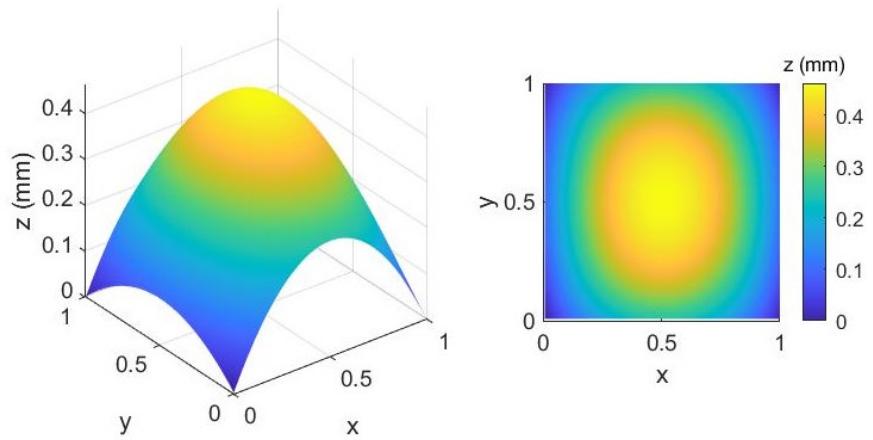
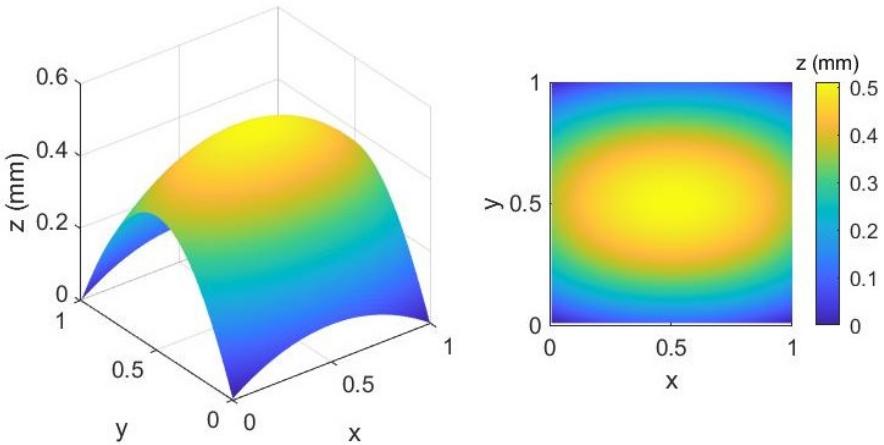


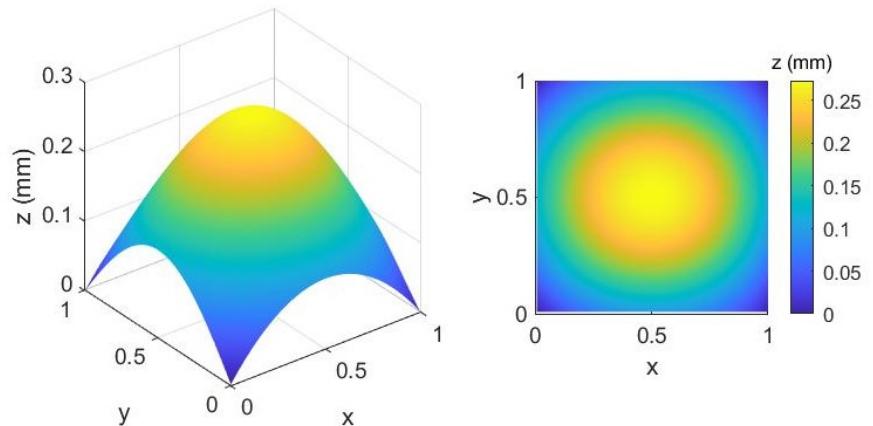
Figure 19: Synthetic data produced for the comparison of standard geometries produced by the tracking strategies suggested



(c) Vertical expansion



(d) Horizontal expansion



(e) Triangular

Figure 19: Synthetic data produced for the comparison of standard geometries produced by the tracking strategies suggested

## 4.7 Sample space investigation

### 4.7.1 Selection of samples

For each of the five selected tracking strategies, four individual test cases need to be derived. To accomplish this the sample space was explored. To define the sample space, boundaries were defined to mark the maximum and minimum value for each process parameters defining the strategy. The boundary values defined for each strategy are defined in Table 7.

Table 7: Boundary values for the sample space for each tracking strategy

Strategy	Parameter	Minimum	Maximum
Centre-to-outside	Minimum width (mm)	50	150
	Minimum height (mm)	50	150
	Centre X (mm)	50	250
	Centre Y (mm)	50	250
	Scaling factor	3	8
Overlaid	Width (mm)	50	400
	Height (mm)	50	400
	Centre X (mm)	50	250
	Centre Y (mm)	50	250
Vertical expansion	Minimum width (mm)	50	150
	Centre X (mm)	50	250
	Scaling factor	3	10
Horizontal expansion	Minimum height (mm)	50	150
	Centre Y (mm)	50	250
	Scaling factor	3	10
Triangular	Base width (mm)	100	400
	Height (mm)	100	400
	Base centre X (mm)	50	250
	Base centre Y (mm)	50	250

For each strategy the sheet displayed a differing number of planes of symmetry. For example, for the 'overlaid' strategy there are two planes of symmetry as shown in Figure 20. Therefore, the sample space is reduced to one quarter of the area of the sheet.

With the sample space defined, the hyperspace was discretised with an equal number of intervals  $n_i$  in each dimension according to equation 16. An equal number of intervals ensures bias is limited.

$$\text{interval} = \frac{p_{\max} - p_{\min}}{n_i} \quad (16)$$

The number of intervals  $n_i$  was tuned manually based on computational performance. The total number of data points  $n_d$  generated follows a power law related to the number of process parameters  $n_p$ , shown in

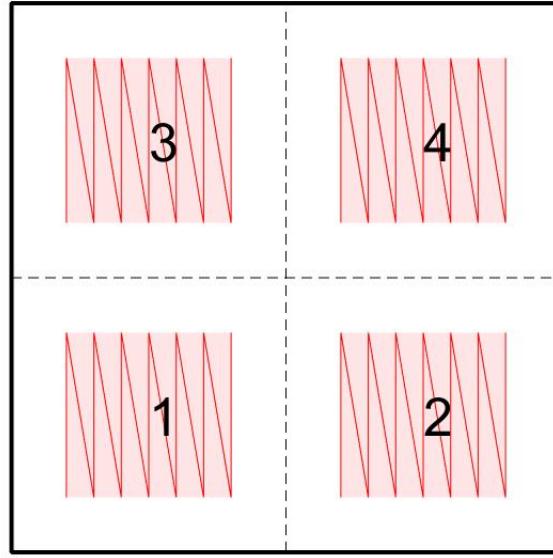


Figure 20: Symmetry within the 'overlaid' tracking strategy

equation 17. It is also affected by the number of desired samples  $n_s$ , as too small a sample set would result in the algorithm failing to find enough samples.

$$n_d = n_i^{n_p} \quad (17)$$

Each point in the generated hyperspace was normalised according to equation 18.

$$p_{inorm} = \frac{p_i - p_{min}}{p_{max} - p_{min}} \quad (18)$$

The tracking patterns were limited to within a specified distance from the edge. A proportion of the points in the hyperspace describe tracking paths that exceed these limits. These were filtered out by ensuring the maximal case for each strategy was within the defined limits. At this point, the sample space is multi-dimensional and non-linear, which means that traditional methods of selecting varied samples are limited.

An altered Latin Hypercube method is implemented for selecting a varied pool of samples. Initially, a base test case is selected from which can be used as a reference point for subsequent selections. For each strategy, this was created using the maximum values of process parameters within the sample space. The selection for each strategy can be viewed in Table 8.

Data points that have parameters within  $\pm 1/n_s$  of the data point selected are then eliminated to ensure that there is an evenly distributed spread for each parameter across all samples. The vector norm of the position of all remaining points in the sample space subtracted from the current data point is then calculated. The maximum norm indicates the set of parameters that differ the most from the current point. This is selected as the next point. Data points are again eliminated, and this process repeats until the sample set has been obtained.

#### 4.7.2 Investigating sampling parameters

A minimum area was defined for each tracking strategy to ensure that curvature induced in the sheet was measurable. Using an external parameter to define the reduction of the data set can introduce bias in the data.

Table 8: Base test cases for each tracking strategy

Strategy	Parameter	Value
Centre-to-outside	Minimum width (mm)	120
	Minimum height (mm)	120
	Centre X (mm)	250
	Centre Y (mm)	250
	Scaling factor	3.333
Overlaid	Width (mm)	400
	Height (mm)	400
	Centre X (mm)	250
	Centre Y (mm)	250
Vertical expansion	Minimum width (mm)	120
	Centre X (mm)	250
	Scaling factor	3.333
Horizontal expansion	Minimum height (mm)	120
	Centre Y (mm)	250
	Scaling factor	3.333
Triangular	Base width (mm)	400
	Height (mm)	400
	Base centre X (mm)	250
	Base centre Y (mm)	50

Figure 21 shows the relation between the area of tracking pattern and number of samples to vary per strategy. The minimum area chosen must maximise the variety of strategies, and therefore minimise the number of samples removed.

The selection of the minimum area is a compromise between the usefulness of the data collected based on the curvature generated, and the bias introduced into the data set in terms of the process parameters. For the centre-to-outside tracking strategy, Figure 22a shows the minimum height to be more biased towards larger values as the area increases. Beyond 70%, the size of the sample space decreases. This is more defined in Figure 22b, as there is a clear shift towards the centre of the sheet for larger tracking areas. For the triangular tracking strategy, Figure 23a shows the base width to be more biased towards larger figures as the area increases. The difference with this strategy is that is restricted by a maximum area of 50% of the total unrestricted area.

Given these distributions, a compromise was found with a minimum area of 20% for the centre-to-outside, vertical expansion, horizontal expansion, and overlaid strategies, whilst a minimum area of 10% for the triangular strategy. The dimensions of the final strategies are not within the defined resolution of the process. The values of the tracking patterns were rounded accordingly, with minimal impact to the sampling methodology.

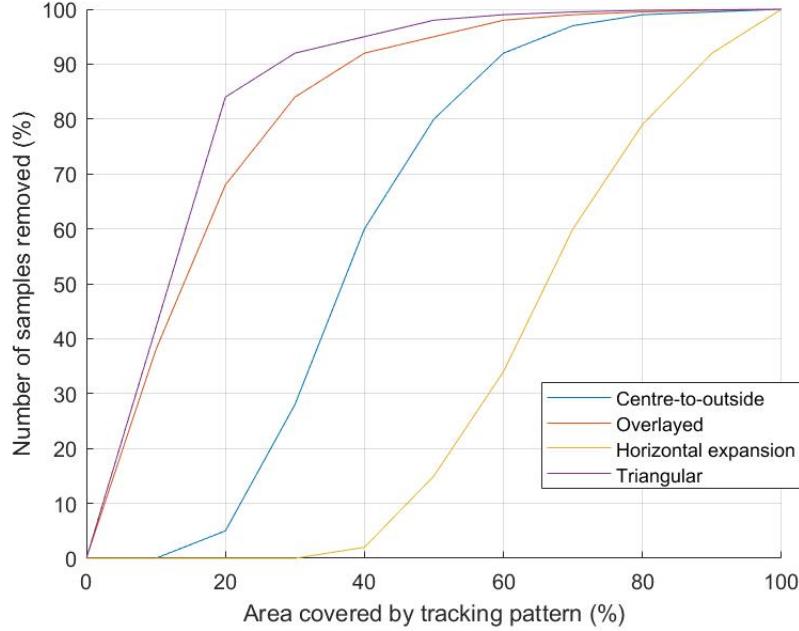


Figure 21: Variation in the overall number of samples with area covered by the tracking pattern for all strategies. Vertical expansion is not included as it is similar to Horizontal expansion.

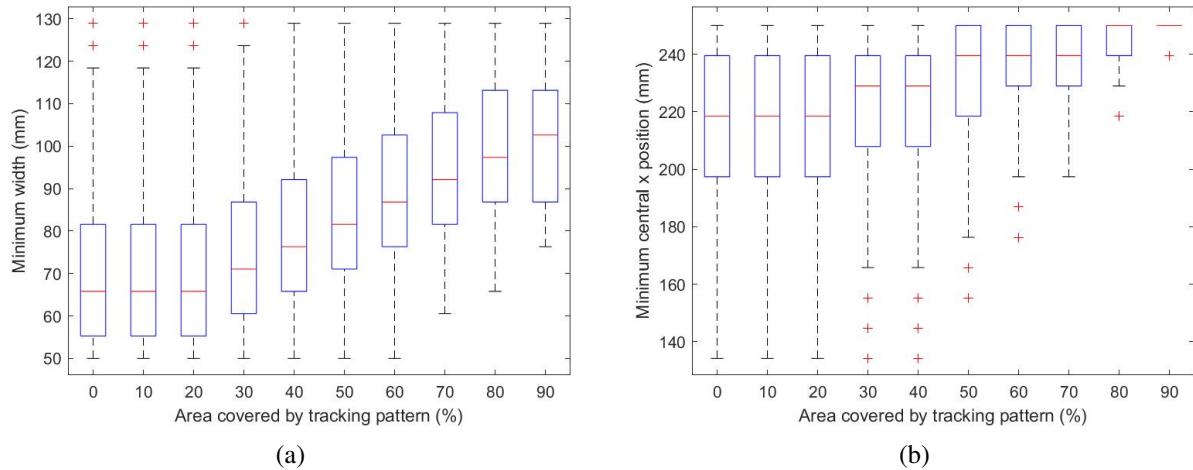


Figure 22: Variation of the (a) Minimum width and (b) Minimum central x position with the minimum area covered by the tracking pattern for the centre to outside tracking strategy.

#### 4.8 Proposed test cases

The number of samples for each strategy was varied as each strategy has a different size of sample space. Therefore, the 'Centre to outside' strategy, which has 5 defining parameters, should have more samples than the 'Vertical expansion' strategy that only has 3. Despite the clear variation in the sample parameters, there is a lack of standardisation in the samples for the data to be easily comparable for a small sample size. Therefore, these samples were adapted so that they are aligned in bottom left-hand corner. The exact test cases can be viewed in Figure 24.

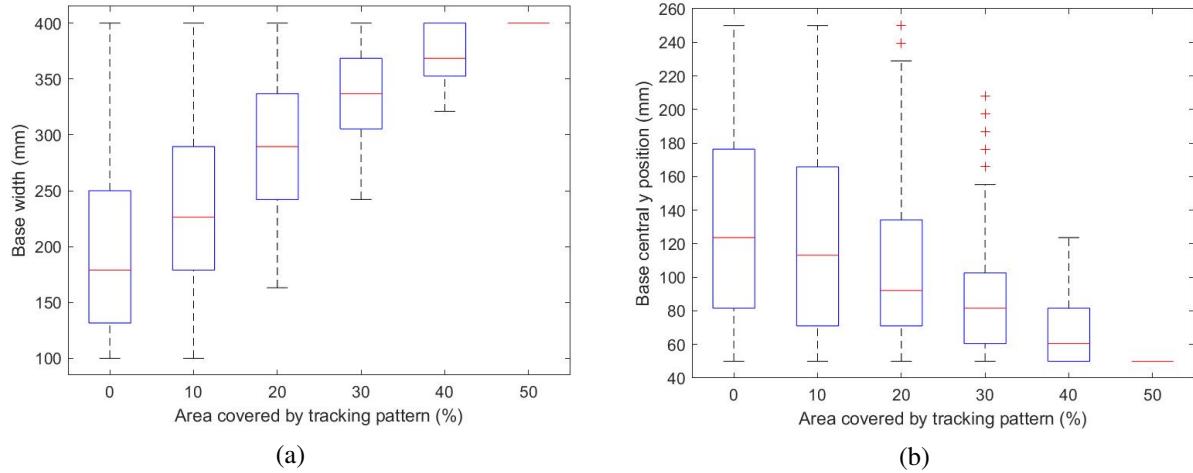


Figure 23: Variation of the (a) Base width and (b) Base central y position with the minimum area covered by the tracking pattern for the triangular tracking strategy.

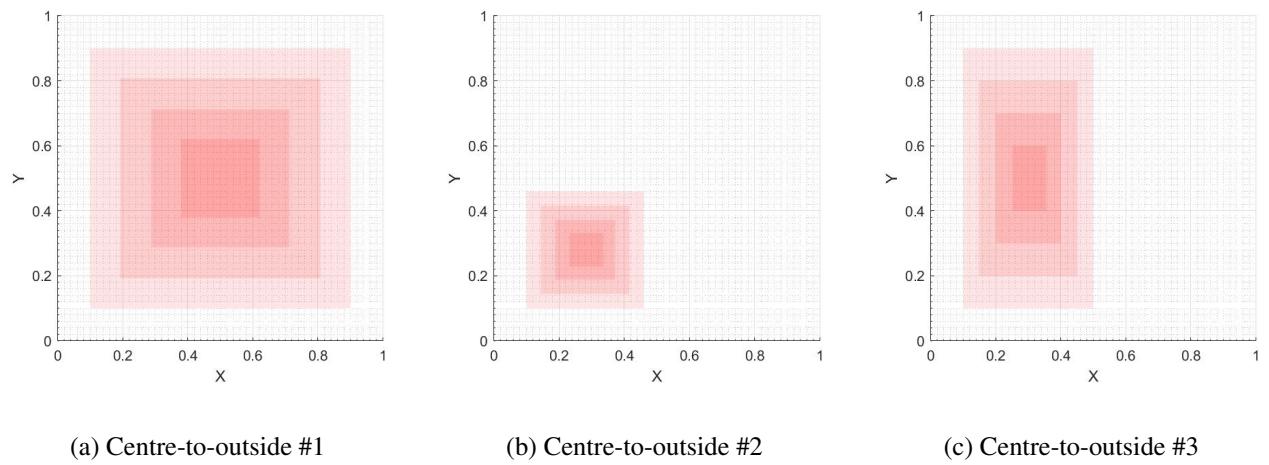


Figure 24: Final proposal of tracking patterns represented as tracking densities. x and y are normalised between 0 and 1

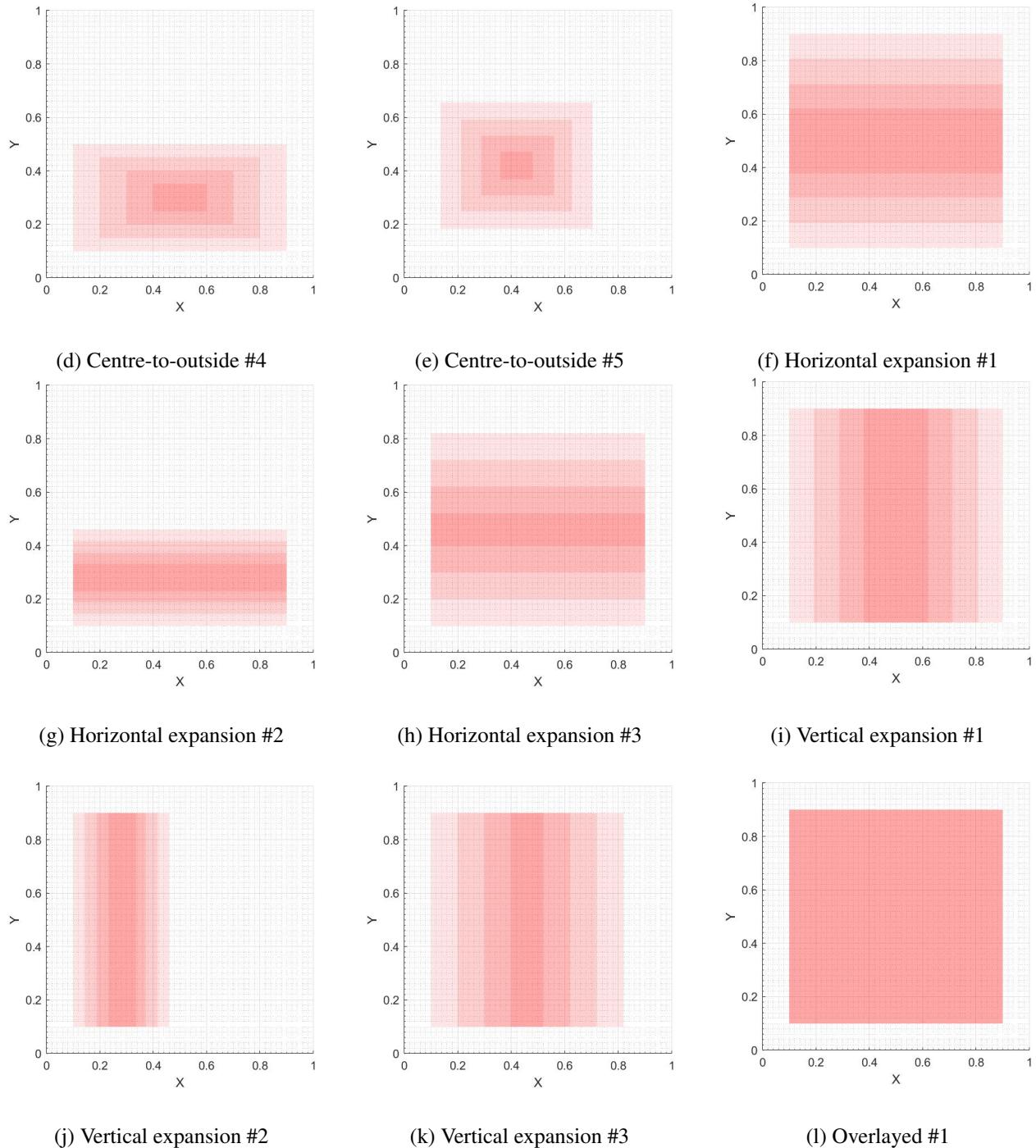


Figure 24: Final proposal of tracking patterns represented as tracking densities. x and y are normalised between 0 and 1

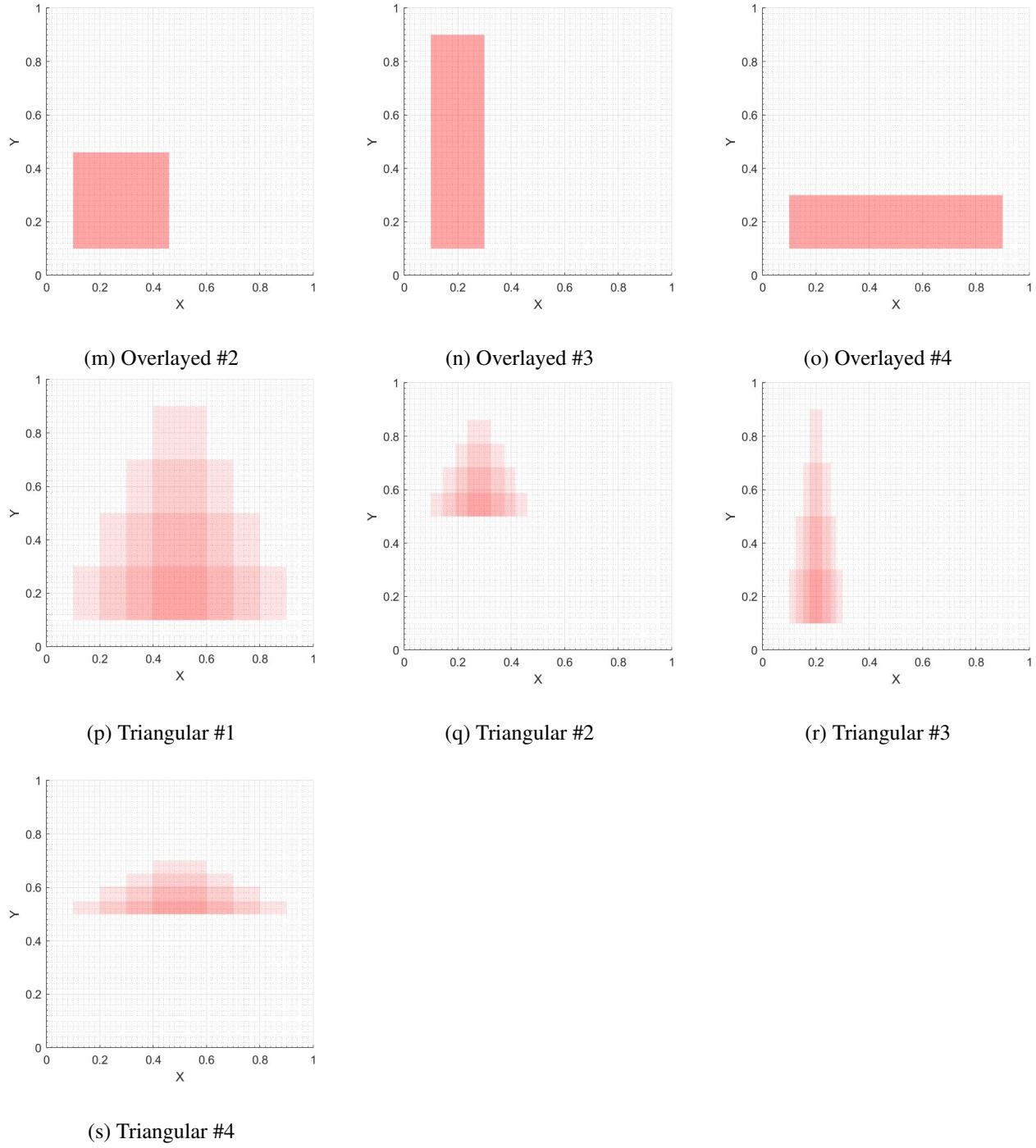


Figure 24: Final proposal of tracking patterns represented as tracking densities. x and y are normalised between 0 and 1

## 5 Experimental procedure

### 5.1 G-code generation

The strategies created in Section 4.8 must be converted to G-code commands for the machine to be able to perform the tracking paths. This process was made more accessible to the user by automating the generation of G-code through a user interface (UI). The quickstart guide in Appendix B.3 describes the steps to use this GUI to output G-code for a specific manufacturing strategy. The process of generating the G-code is described in Figure 25.

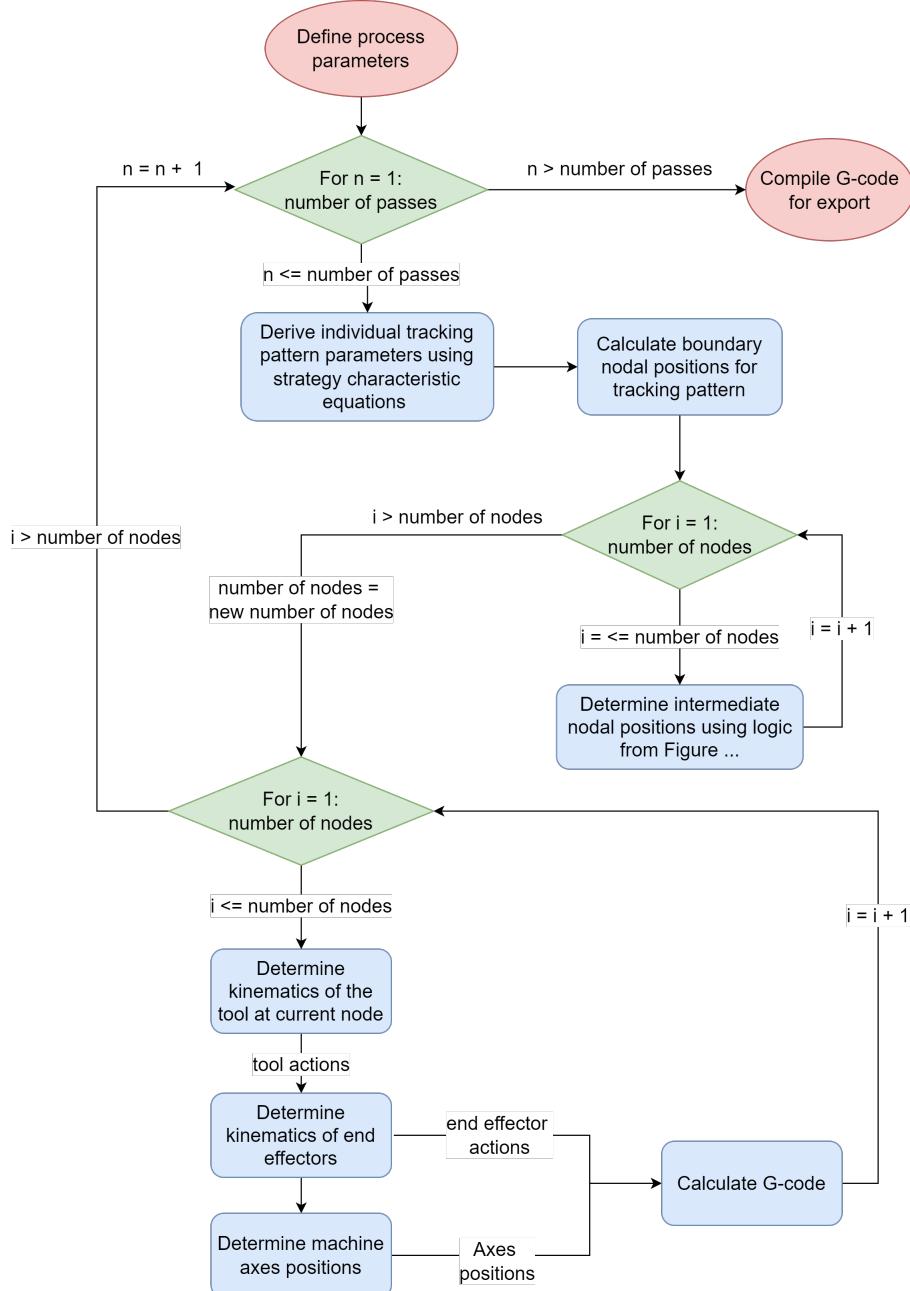


Figure 25: Flow chart describing the process of generating G-code from process parameters

Previous to this section, the process parameters have been defined without rigorous protocol. Table 16 in Appendix B.1 outlines the formal documentation for these process parameters that is implemented in the UI. The UI allows users to define the tracking strategy parameters, which are applied to the characteristic equations in Section 4.5 to determine the global tracking pattern parameters. From these parameters, the positions of each node within the tracking pattern can be calculated using the theory presented in Section 3.3, as shown in Figure 26.

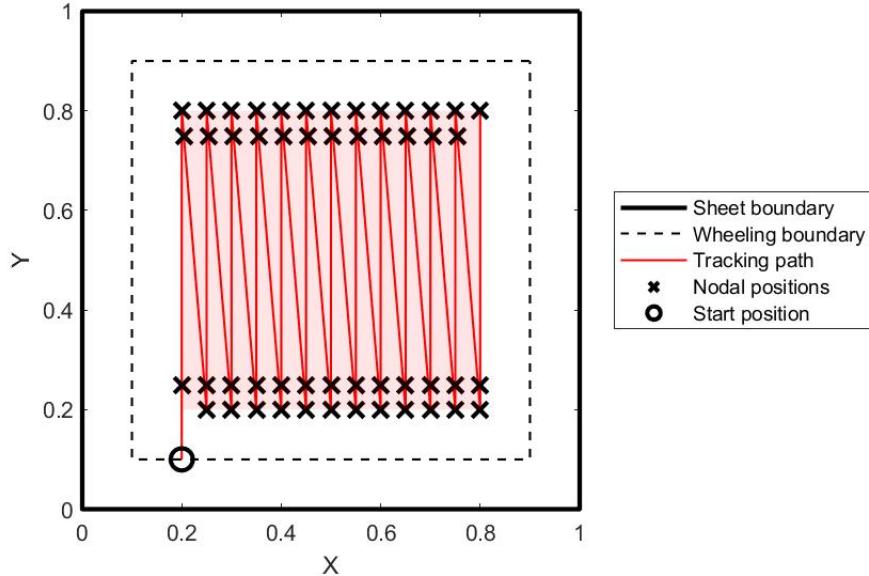


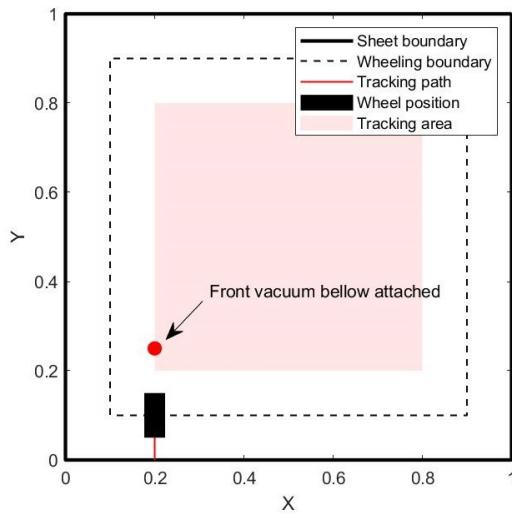
Figure 26: Example tracking pattern with nodes indicating machine action

Figure 26 displays intermediate nodes between the nodes indicating the boundary of the tracking area. These are points at which the vacuum bellows transfer. The logic determining the position of these nodes is outlined in Figure 27.

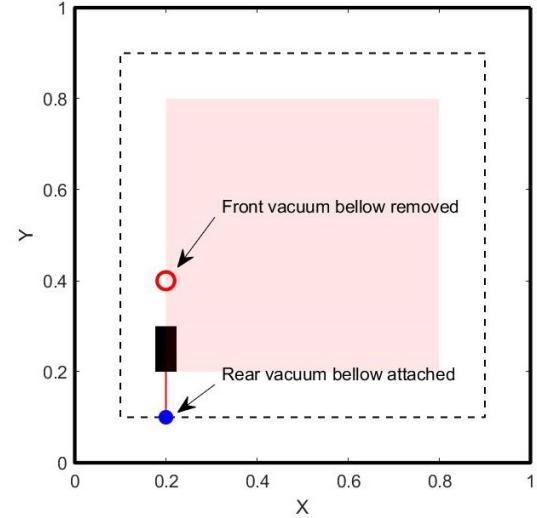
To enable the G-code generation software to be more easily transferable to the firmware, the wheeling process was characterised by discrete actions. Each node comprised of a series of these discrete actions, always starting with the termination of a translation action from the previous node, and ending with a translation action to the next node. The intermediate actions could include a pivot or transfer, as determined by the logic presented in Figure 27.

For each action, the kinematics of the tool relative to the position of the sheet was computed, using the axes defined in Section 3.1. The homogeneous matrices describing the transformation of the tool between nodes were computed as expressed in equation 14. From these transformations the kinematics of each individual end effector was calculated relative to the position of the tool.

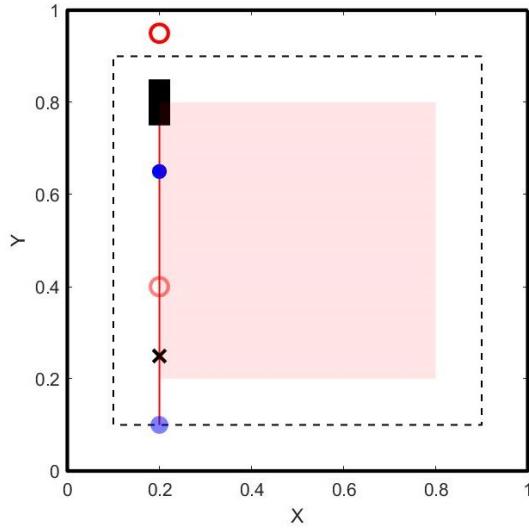
The end effector positions were then transformed to be representative of the machine axes. From the action name and axes positions at the start and end of each action, the G-code commands were generated. These follow the same documentation procedure as the firmware created for the machine. The documentation of the structure used to generate the G-code can be found in Appendix B.2.



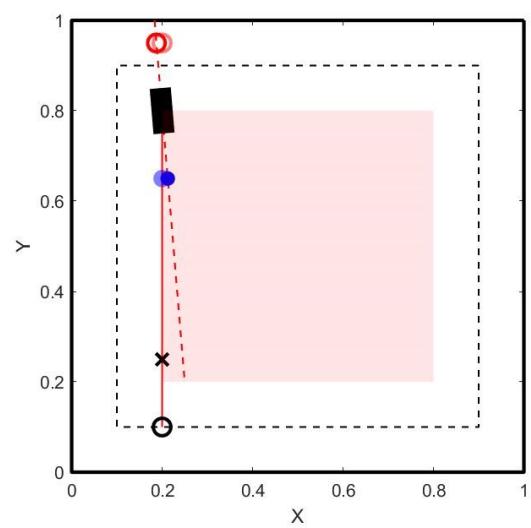
(a) Initial tracking of the sheet to push through the boundary to the first nodal position.



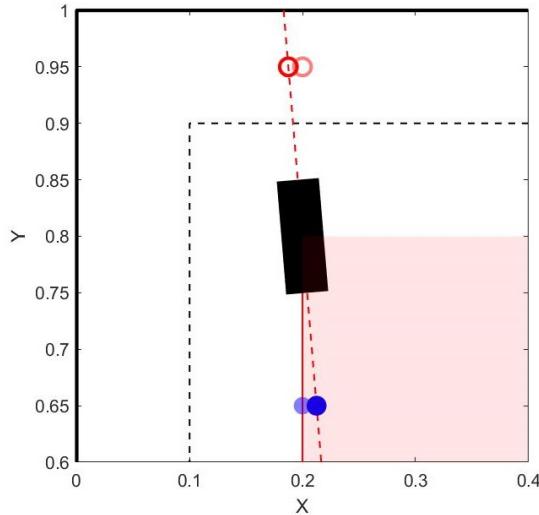
(b) Tracking of node 1 to node 2. This node is only for the transfer of end effectors. Its location is dependent on the limiting position of the end effector.



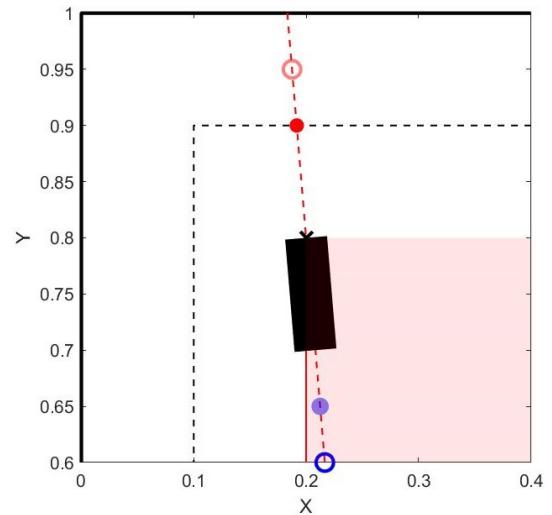
(c) Tracking of node 2 to node 3 with back end effector active.



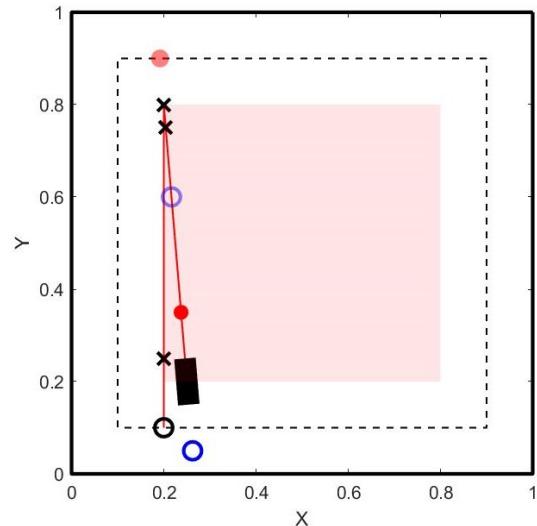
(d) A pivot occurs at node 3 at an angle calculated from the user-defined tracking spacing.



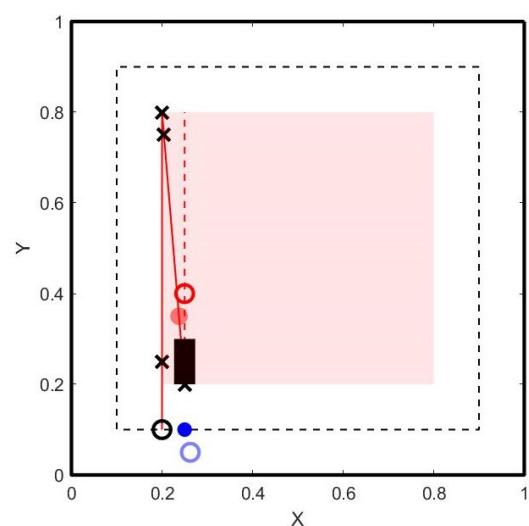
(e) Check if transfer is possible. In this case the front end effector is outside the permissible boundary for wheeling therefore transfer cannot occur.



(f) Calculate the additional translation required before transfer can occur. If it is not possible, continue with currently enabled end effector.

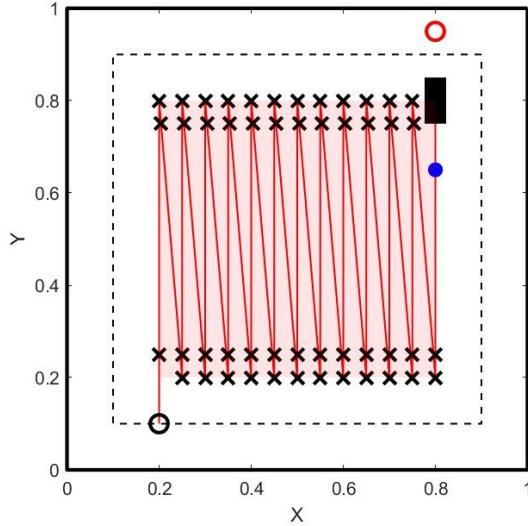


(g) Tracking from node 4 to 5 with front end effector active.



(h) Repeat steps d-g for all subsequent nodes.

Figure 27: Nodal positioning logic for machine action



- (i) This occurs for all nodes until the tracking pattern had been fully executed.

Figure 27: Nodal positioning logic for machine action

## 5.2 Collecting data

During the experiment, the geometry of the sheet is measured between passes. Various metrology equipment were evaluated based upon design specification B2 and B3, the accessibility and accuracy of the equipment. It was determined that the EinScan Pro handheld scanning device was the most appropriate tool as it can carry out measurements in situ, whilst also having a suitable accuracy of 0.05-0.1 mm in ‘hand-held mode’ ([Shining3D, 2022](#)).

The scanning required markers placed on the sheet to aid the identification of the surface. Figure 28 shows that these markers were placed at arbitrary positions for the scanning process. They did not remain on during the wheeling procedure as they would affect the mechanics of the process.

The EinScan Pro outputs a point cloud file in a .asc format and the marker file in a .p3 format. These both are a list of randomly ordered positional vectors. The point cloud is effectively non-euclidean, therefore a physical set of axes was created to re-orientate each scanned point cloud.

This axes was implemented using a jig built in-house. This jig, shown in Figure 28, ensures the sheet is always in the same position by constraining the same corner in the 90° elevated edge. Figure 28 displays three nodes placed around the sheet each composed of a separate number of markers. One node represents the position of the origin, while the other two intercept the x and y axes. These will be used to orientate the sheet virtually.

It is important that the markers indicating the axes positions are at a fixed spacing below 10 mm, and that the markers representing the sheet surface have at least 30 mm spacing. This enables the orientation algorithm to differentiate between the two types of markers when identifying the axes. Each node has a different number of markers so the algorithm can identify if the node represents the origin, x axis or y axis. These are indicated in Figure 28.

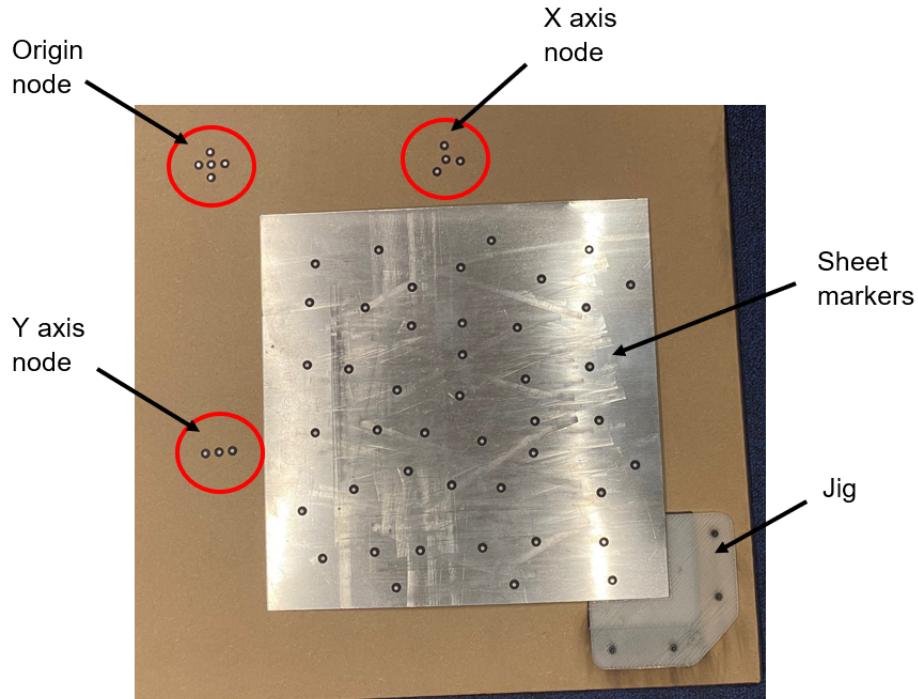


Figure 28: Sheet placed in jig for scanning with markers applied

### 5.3 Data post-processing

A test sheet with an arbitrary geometry was scanned in for verification of the subsequent computational processing procedures detailed in this section. It is made of 106682 data points.

A k-nearest neighbours algorithm was used to identify the markers that composed the nodes. If the nearest neighbour detected was within 10 mm, a node was detected. From this, the nodes could be differentiated by comparing the number of markers. To calculate the centre point of each node, the vector norm between each marker in the node was calculated. The marker that has the lowest summation of vector norms was considered to be central.

The point cloud was then transformed by first performing a translation equal to the negative of the origin positional vector. The rotational matrix was then calculated using ZYZ Euler angles to locate the two remaining nodal points along the x and y axes respectively. From this, a homogeneous transformation matrix can be created than was used to transform the point cloud, as shown in Figure 29.

The ground was then removed from the point cloud by defining a minimum z-value. Remaining unwanted points were removed manually by defining the sheet via its corners, and removing all external points. A point cloud was then translated by a pre-defined vector to position the corner on the origin. The final result can be seen in Figure 30.

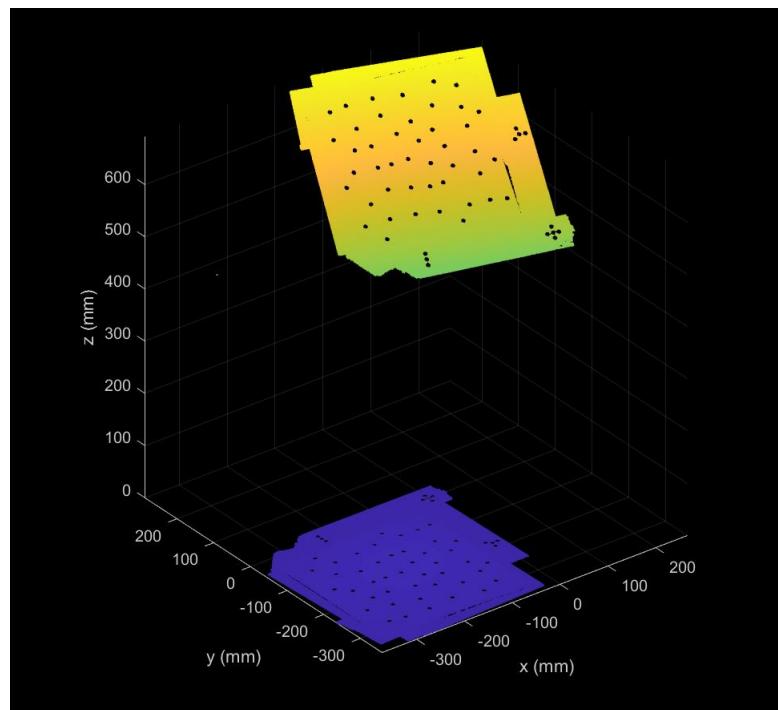


Figure 29: View of point cloud prior to (orange) and post (purple) orientation

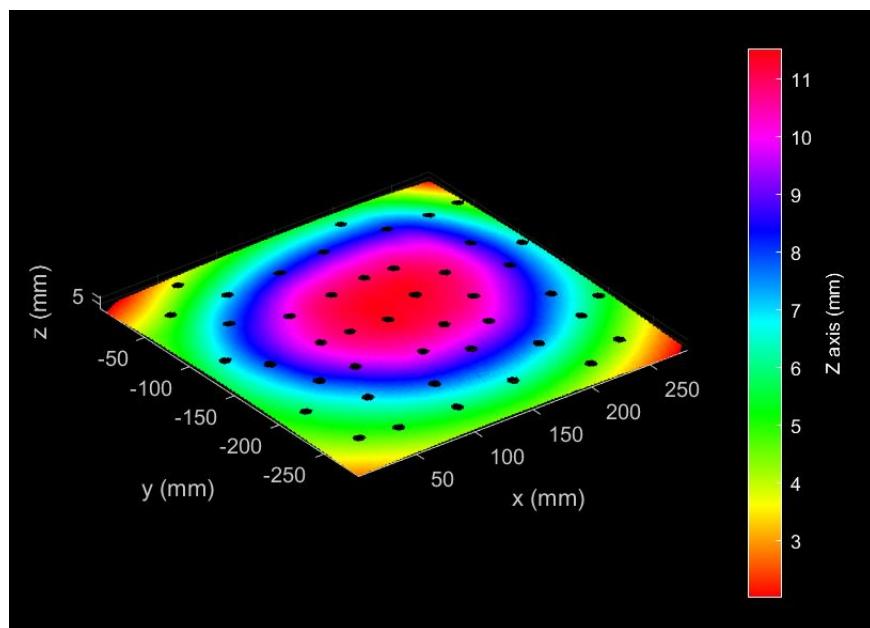


Figure 30: View of orientated and processed point cloud

## 6 Model testing framework design

### 6.1 Overview of model and testing framework

ML models require the data to be split into training, validation, and test sets. The training set is used to fit the model, however it introduces bias into the model by overfitting the data. The validation set provides an unbiased estimate of the model fit, and is useful for tuning model hyperparameters. However, over-tuning of hyperparameters can result in over-optimistic predictions. Therefore, a test set is required to assess the performance of a fully-specified classifier (Ripley, 1996). A 0.7:0.15:0.15 train-validation-test split is used for training, tuning and testing the model.

Each sample can be categorised by a set of features and labels. This model will use geometrical descriptors of the sheet as its dataset of features, parameters describing the tracking strategy as its labels. Using this trained model, the validation and test dataset will be used to tune and verify the accuracy of the model. An overview of the training and validation of the ML model is shown in Figure 31.

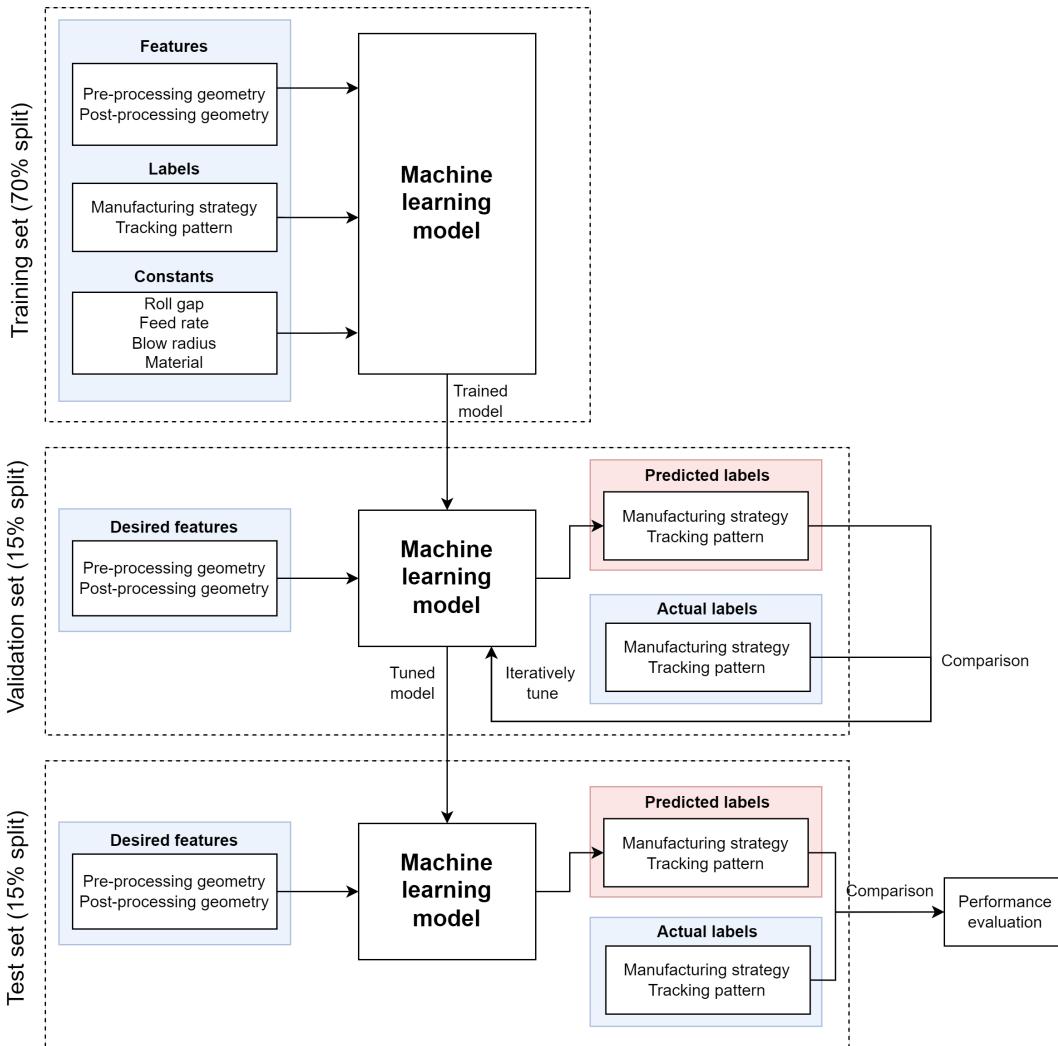
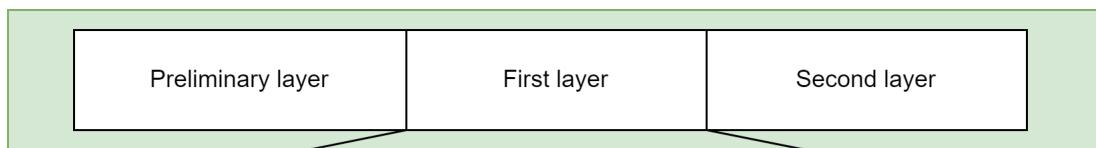


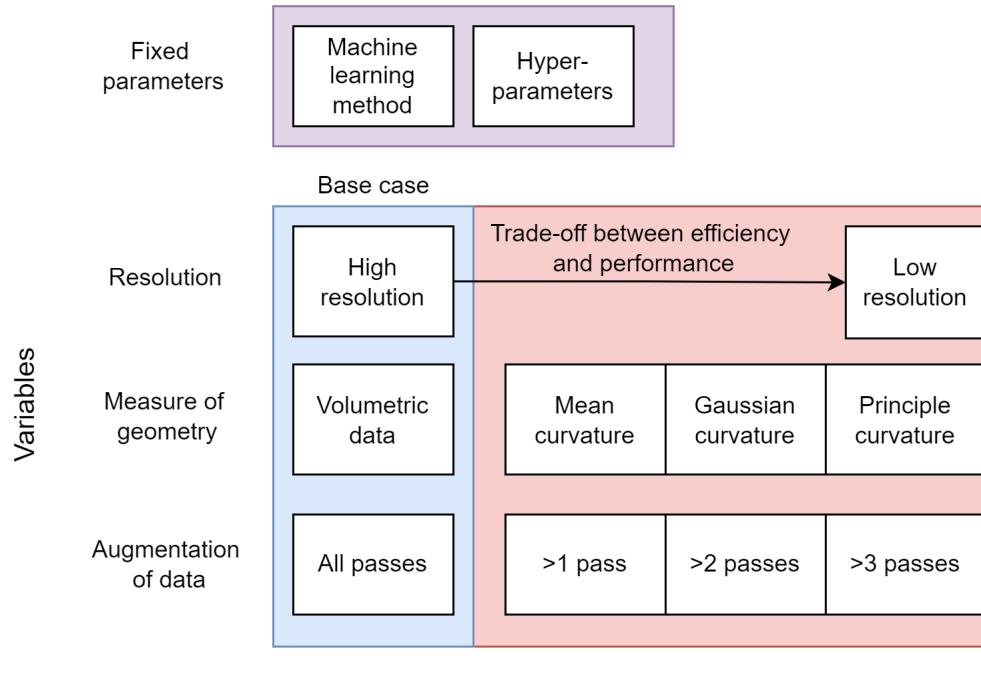
Figure 31: ML training, tuning, and testing overview

The tuning process will be performed according to the testing framework outlined in Figure 32. This is composed of three test stages; selection of architecture, data tuning, and model tuning. The tuning procedures will be discussed in the following sections.

### 1. Iterate through layers in architecture



### 2. Tune data based on selected model



### 3. Tune model using modified data

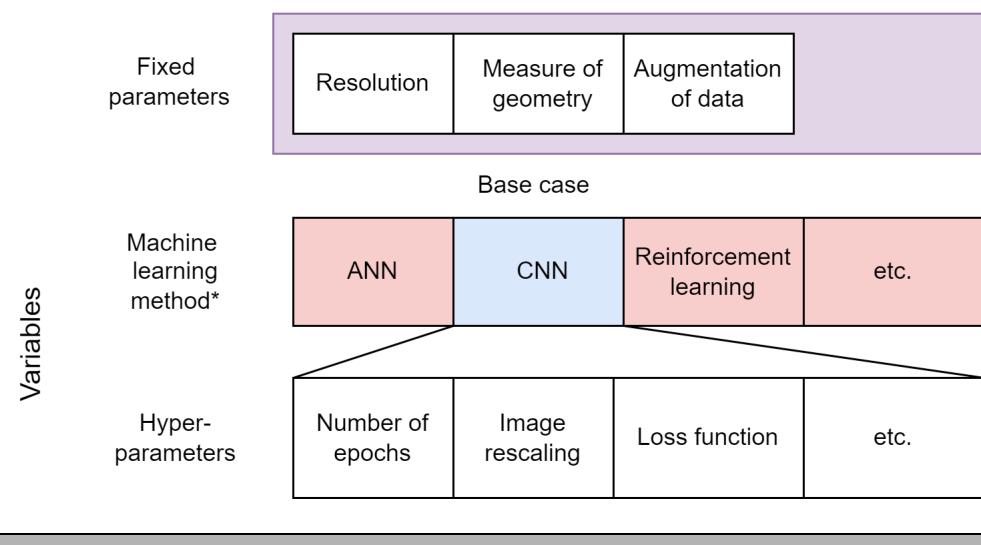


Figure 32: Testing framework for the ML architecture. \*This study is limited to CNN method, however this framework is still applicable for future study of other ML techniques.

## 6.2 Architecture

Each tracking strategy has a different set of defining parameters, therefore the labels for the ML model will vary per strategy. Therefore, to simplify the ML input and output characteristics the strategy needs to be defined prior to the prediction of the associated parameters. This study will consider two approaches to solving this that are both visually represented in Figure 33:

### 1. Geometric model

Classification of the strategy can be achieved using a data set of features describing the geometry, with corresponding labels indicating the name of strategy. Using this prediction, the data set is narrowed to include measurements only from that strategy. The process is repeated with a new model using this transformed data set. The features again describe the geometry, whilst the labels are the strategy parameters, with the final prediction being a set of strategy parameters.

### 2. Non-parametric model

The disadvantage of the geometrical feature-based prediction approach is that there is limited data for it to make accurate predictions per strategy class. Instead, it may be more effective to form a non-parametric method of describing the tracking strategy, which can be used to predict the strategy parameters. To achieve this a preliminary layer is required as shown in Figure 33 to create a data set of features representing the tracking density from the measured geometries. From this, it is possible to either predict the strategy, given labels detailing the strategy, or if the strategy is already known, the strategy parameters can be predicted using the same method implemented as the geometrical feature-based prediction method.

Due to time limitations, only the geometric model is implemented in this study. For future reference, a combination of both architectures can be used, with all configurations represented in the testing matrix given in Table 9.

Table 9: Testing matrix for different architecture configurations

Test case	Preliminary layer	First layer	Second layer
1	-	Geometric	Geometric
2	Non-parametric	Geometric	Non-parametric
3	Non-parametric	Non-parametric	Geometric
4	Non-parametric	Non-parametric	Non-parametric

Each layer within the architecture solves a different type of problem, as specified in Table 10.

Table 10: Classification of ML problem for each layer in architecture

Layer	Type	Multi-class?	Multi-output?
Preliminary	Regression		
First	Classification	✓	
Second	Regression		✓

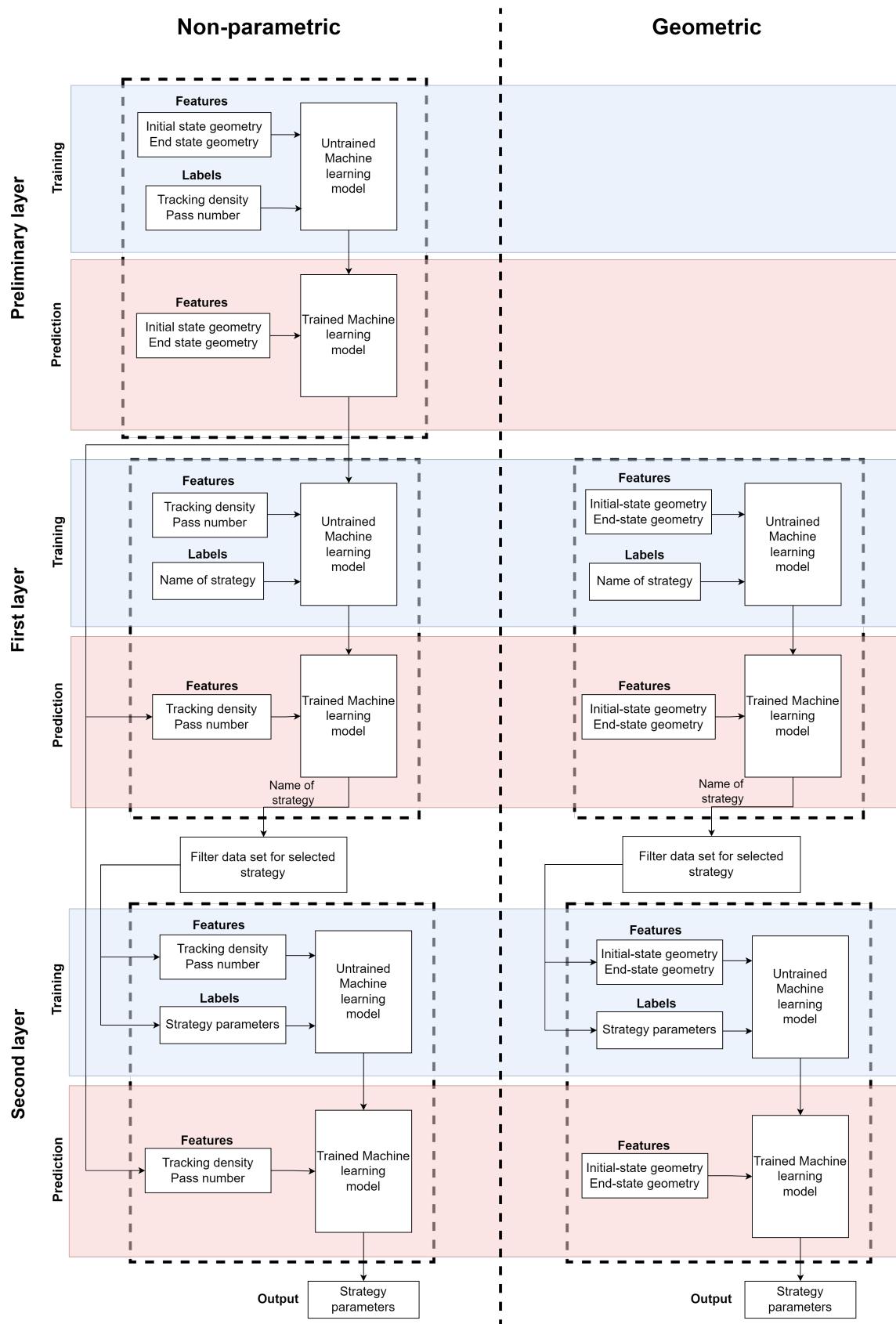


Figure 33: ML architecture concepts

### 6.3 Performance metrics

Metrics need to be identified to assess the performance of the model in the testing framework. These will vary depending on the model solving either a classification or regression problem. For the classification models the accuracy of the model is used to assess the performance. It is described as

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

A loss metric is required by the model to tune its performance after each iteration. The categorical cross-entropy was selected for the classification models. It can be calculated as

$$\text{Categorical crossentropy} = - \sum_{N=1}^{i=1} y_i \cdot \log \hat{y}_i \quad (19)$$

where  $\hat{y}_i$  is the  $i$ -th scalar value in the model output,  $y_i$  is the corresponding target value, and  $N$  is the number of scalar values in the model output.

For the regression model, the mean absolute error (MAE) is used as the performance indicator as well as the loss metric. It can be calculated as

$$\text{MAE} = \frac{\sum_{N=1}^{i=1} (\hat{y}_i - y_i)}{N} \quad (20)$$

For this performance metric to be comparative across all strategy parameters, the parameters are normalised before input into the model.

### 6.4 Data tuning

#### 6.4.1 Geometry representation

Initially, the geometry of the sheet is described via a point cloud in Euclidean space. As shown in Section 5.3, these point clouds typically consist of over 100000 points, and need to be reduced for computational efficiency. It was found that there are limited open-source ML methods for application to point clouds and methods available are computationally expensive. Instead, the data was transformed to a structured volumetric representation.

Surface representation methods such as B-spline patches had no ready-made package within MATLAB that could be used. Therefore, due to time constraints a power basis representation was used to describe a polynomial mesh. Figure 34a shows that there is small variation in the root-mean-square error (RMSE) for an order of polynomial surface beyond 8. Beyond an order of 14 the solution becomes unstable and singularities occur. This calculation is irrespective of resolution as RMSE is characteristic of the polynomial function rather than the mesh. It is required to have a high resolution to ensure accuracy, however there is a trade-off with computational cost. From Figure 34b, a resolution of 4mm was selected with order 10 as a compromise. Figure 35 shows that for this configuration the representation is similar.

The geometry can also be defined using descriptors such as measures of curvature  $K$ ,  $H$ , and  $P_1$  and  $P_2$ . These were calculated from the point cloud data using theory of fundamental forms. This theory could be easily applied to the polynomial approximation of the surface. Figure 36 shows the different forms of curvature have distinct and different characteristics. The effect of each form of the performance of the model will be analysed in this study.

The data is a 3D format, which is difficult to process unless state-of-the-art techniques such as those presented in Section 2.3 are understood. This study is using open-source ML packages that has limited scope in the 3D domain. Therefore, the 3D representation of the sheet, as either euclidean geometry or curvature, was projected onto a 2D surface in a grayscale image format.

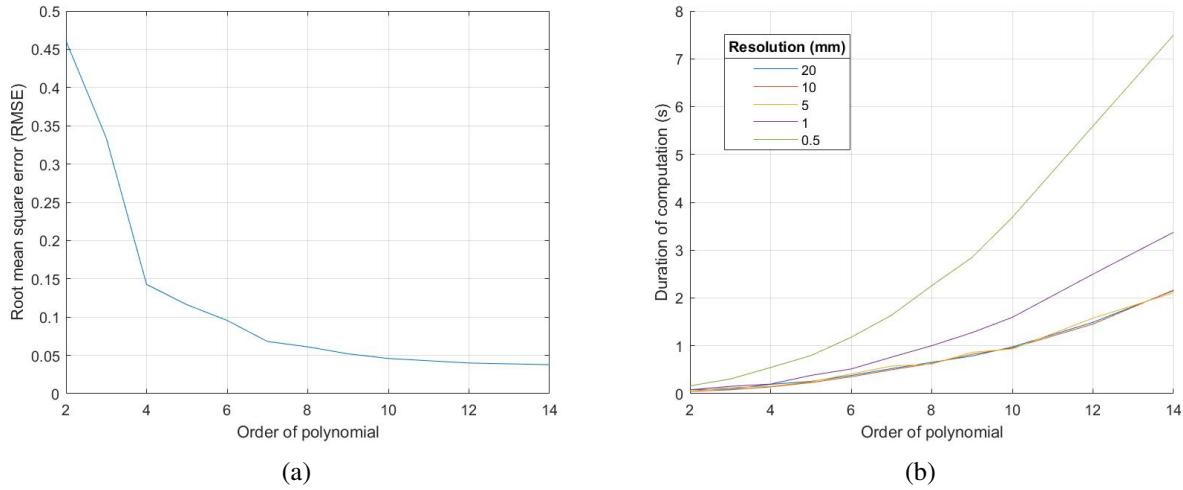


Figure 34: Evaluation for a singular polynomial surface of the (a) RMSE of varying orders of polynomial and (b) duration of computation with varying resolution and order of polynomial

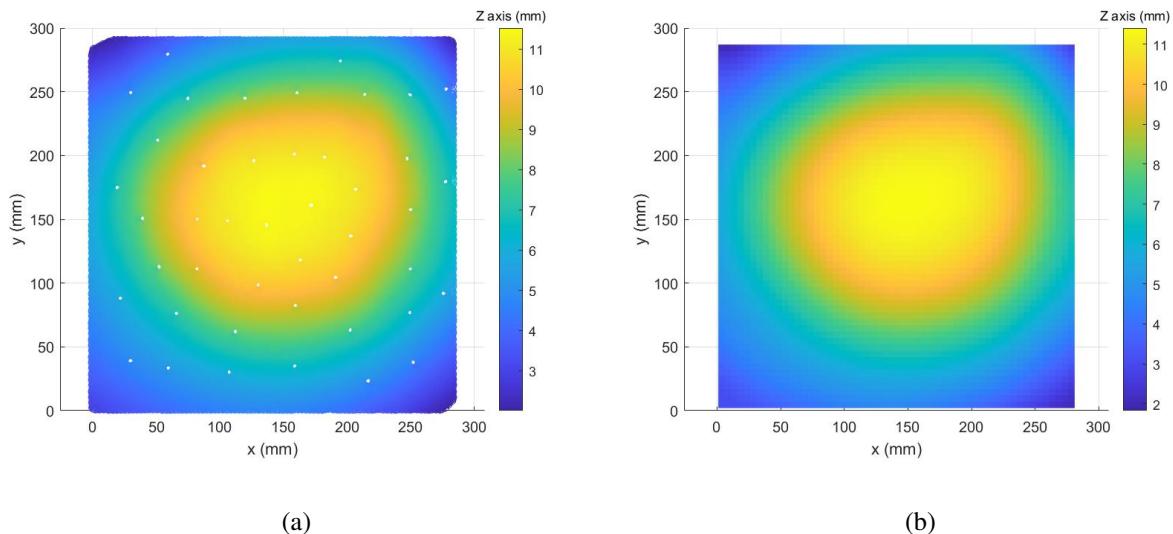


Figure 35: Comparison of the (a) point cloud representation of the surface with the (b) fitted polynomial surface of order 10 and resolution 4mm

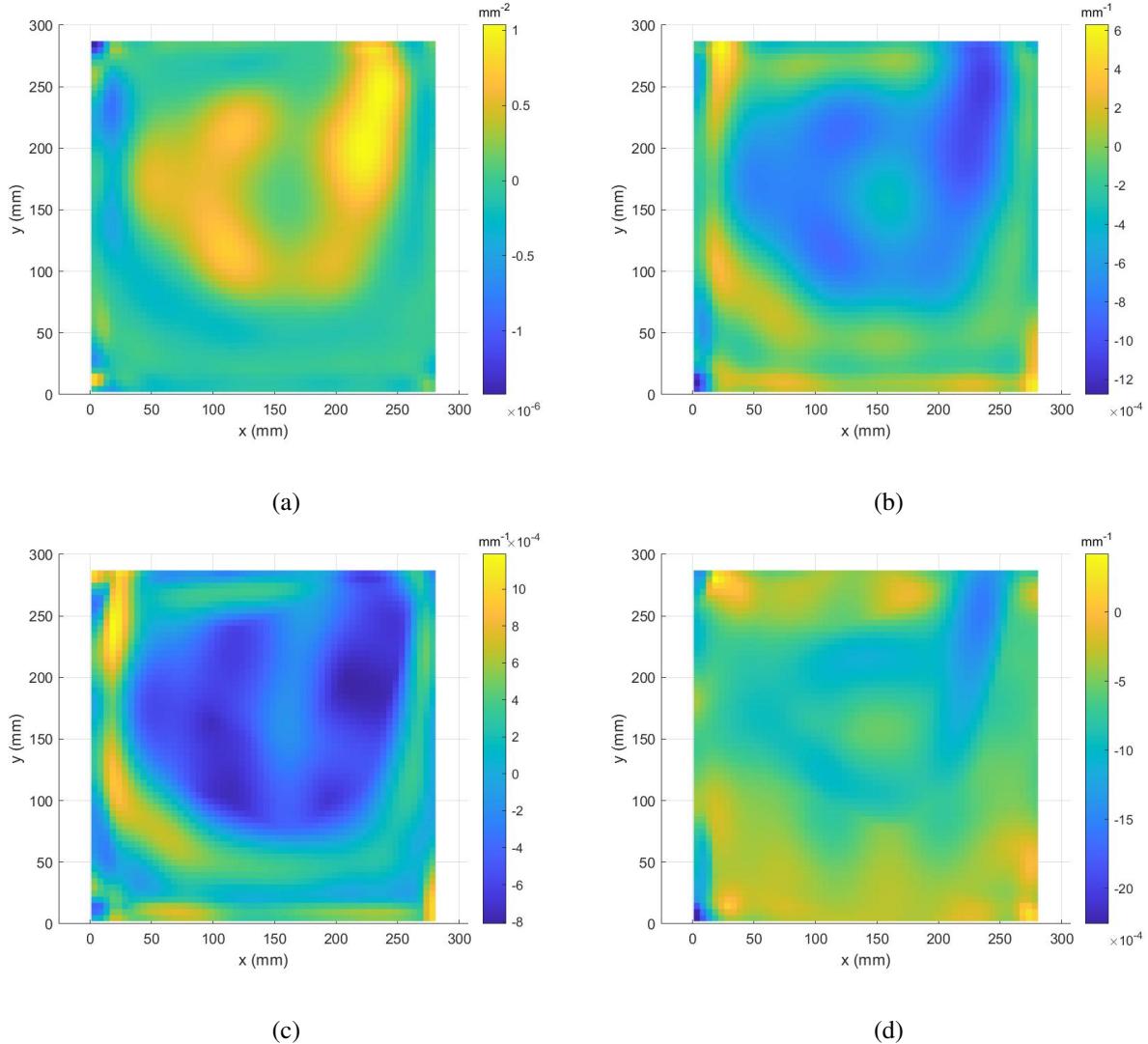


Figure 36: Comparison of (a) Gaussian curvature, (b) Mean curvature, (c) Maximum principle curvature, and (d) Minimum principle curvature calculated using a fitted polynomial surface of order 10 and resolution 4mm to the sample sheet

#### 6.4.2 Data augmentation

The change of geometry that occurs for a specific tracking density is of interest. The dataset can be expanded by considering all possible combinations of tracking patterns and calculating the corresponding changes in geometry using equation 12. An example is given in Figure 37. Currently, only the change of geometry is measured for passes 0-1, 0-2, 0-3, and 0-4. Figure 37 shows additional sets can be created, such as 1-2, 1-3, 1-4. Given that there are 4 passes for each sample, this increases the size of dataset by 250%.

Despite the advantage of greater data quantity, a larger dataset includes a higher number of measurements with a lower number of passes. This data is of lower quality as it is more difficult to differentiate between strategies. A compromise must be made between data quality and quantity. Tests will be conducted that vary the size of dataset based upon the number of passes, and evaluate its effect on the accuracy of prediction.

Additionally, the dataset will be expanded through augmentation based upon the orders of symmetry introduced in Section 4.7. Random transformations will be applied to the dataset during each ‘epoch’,

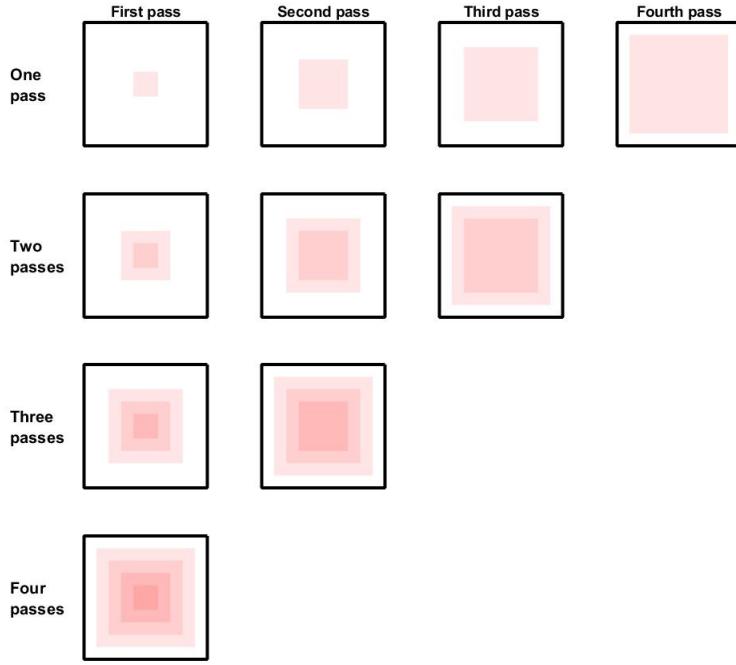


Figure 37: Example of the expansion of size of dataset using augmentation

with the greater data variety reducing overfitting and resulting in more accurate predictions ([Shorten and Khoshgoftaar, 2019](#)).

## 6.5 Model tuning

### 6.5.1 Machine learning method

The most applicable ML method to be used on the projected form of the geometry is a CNN. This method is effective at classifying images, and there is significant published documentation on building a CNN using open-source tool such as TensorFlow. Other ML techniques will not be evaluated as it is beyond the scope of this study.

The data is processed before input into the CNN model. Categorical labels are encoded using a ‘one-hot’ technique to be in a suitable format. The dataset is shuffled and divided into training, validation and test sets. The CNN model initially performs a ‘feature selection’ process on the input grayscale image. Figure 38 outlines the steps in feature selection. Initially, there is a ‘convolution layer’ within the network where each segment of the image is multiplied by an  $n \times n$  feature detector to create a feature map. This reduces the size of the image but retains the information. Multiple filters are used to create an array of feature maps.

Next is the ‘ReLU’ layer, where the rectifier function is applied to increase the non-linearity in the data. A ‘max pooling’ layer is then applied where the maximum value of  $n \times n$  sections of the feature map are taken to form a pooled feature map. These sections are defined by a stride length that is the distance moved between each section. This reduces the size of the dataset whilst retaining spatial position of features, reducing the overfitting of data. The ‘convolution’ and ‘max pooling’ layers can be repeated several times to improve performance. After this, a ‘flattening’ layer is applied where the pooled feature map is reduced to a column array. This is to be used in the ‘full connection’ layer, which is the input to the ANN.

Figure 39 shows that ANNs are constructed using input, hidden and output layers. Each node, represented by a circle, within a layer is analogous to a neuron in the brain. Connections between neurons, or synapses, are initially assigned weights. A weighted sum of inputs to each node is calculated, and subsequently an activation function is applied to the output. This occurs for all hidden layers until an output is generated. The error between the actual and predicted output of the network is calculated using a cost function. This error is then back-propagated through the network to adjust the weights to reduce this cost-function and increase performance (Zou et al., 2009).

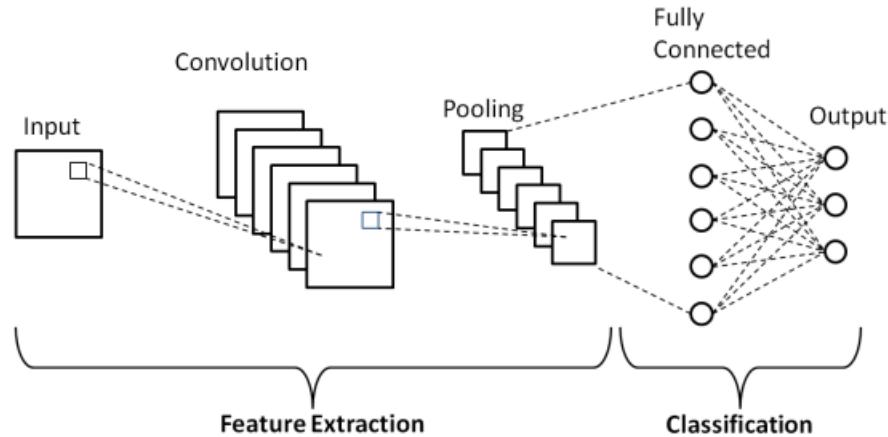


Figure 38: Overview of the CNN (Phung and Rhee, 2019)

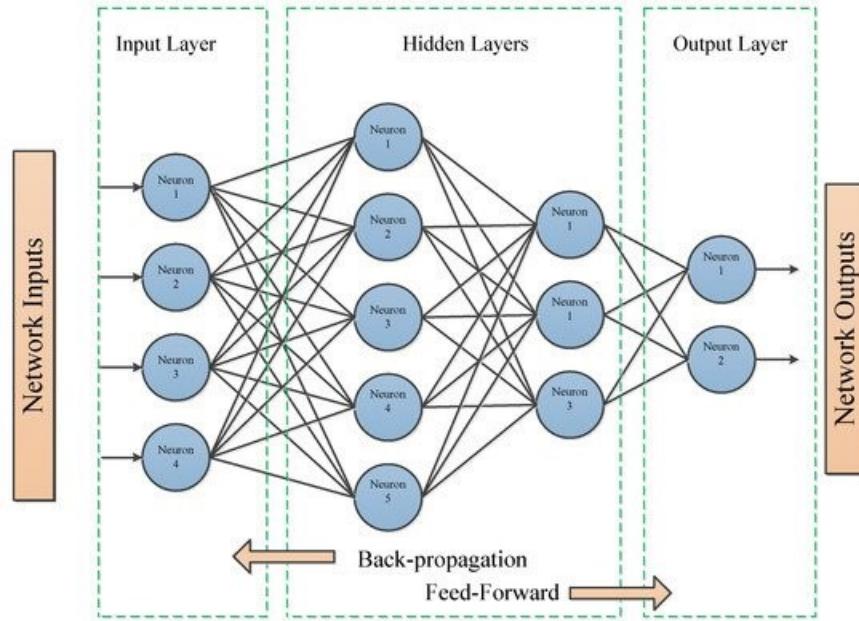


Figure 39: Artificial Neural Network architecture (Abdolrasol et al., 2021)

This theory is applied slightly differently to both the first and second layers within the architecture, as shown in Figure 40. Unlike the first layer, the second layer is multi-output and requires multiple models in series to make predictions simultaneously. Also, the second layer is a regression model, which requires a different measure of loss to the first layer classifier. These loss metrics are given in Section 6.3.

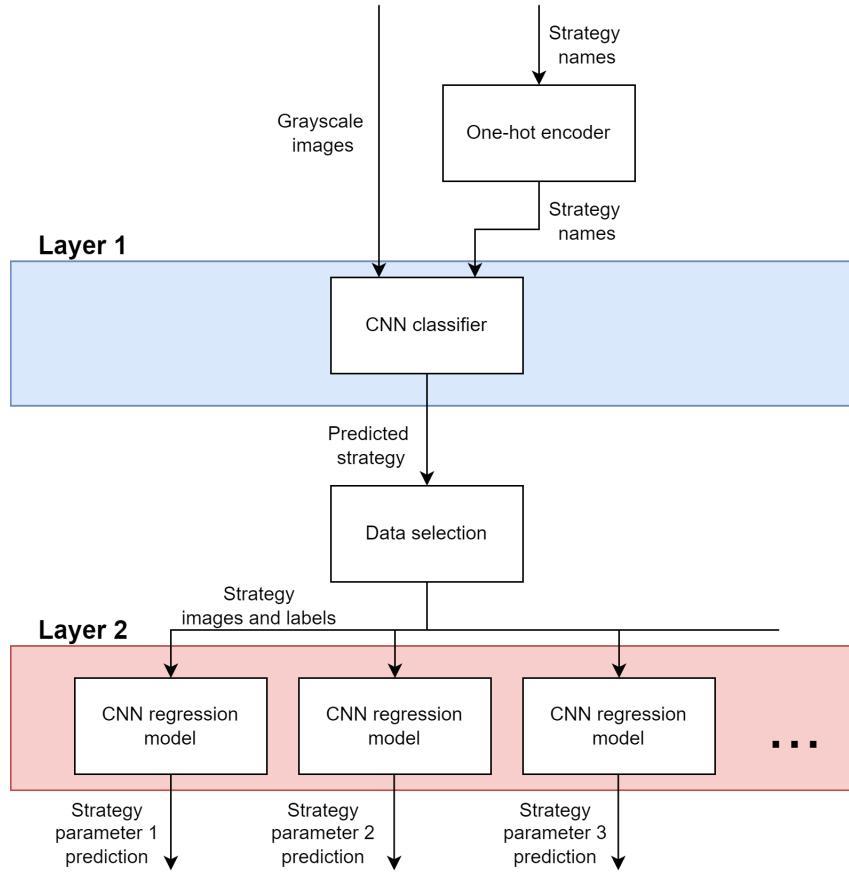


Figure 40: Methodology for implementing the CNN model for each layer in the architecture

### 6.5.2 Hyperparameters

Hyperparameters of each model can be tuned to increase model precision. These are listed in Table 18 in Appendix C.1. The hyperparameters that determine the size of network have the greatest effect on the performance of the neural network and will be tuned in this study. These include the number of hidden layers, neurons, and epochs. The other parameters will remain constant at the base values indicated.

The tuning of hyperparameters will be divided into four differently sized networks as indicated in Table 11. For each model, the size of network will be assessed against the relevant performance metric. This will be done across a number of epochs to ensure that the model is trained to maximum performance without overfitting the data.

Table 11: Size of networks to be tested for tuning of model

Name	Number of hidden layers	Number of neurons
Tiny	1	16
Small	2	16
Medium	3	64
Large	4	512

## 7 Results

### 7.1 Overview of input data

The ML architecture presented in the previous section was built with the intention of training using experimental data. However, the machine was not operational in time to carry out the experiments. Therefore, synthetic data was created instead that is based upon the pre-designed test cases that were presented in Section 4.8. A heuristic approach was taken to generate this data, using existing knowledge of wheeling mechanics from manuals and published sources of numerical data such as [Bowen et al. \(2021\)](#). This was mentioned previously in Section 4.6.

The data is projected to a 2D surface and converted to a grayscale image format. This is done for all geometrical forms of data and the tracking density. The values are normalised between 0 and 1 before formatting into an image. Normalisation has the negative effect of reducing the importance of features within the image due to outliers. The data was normalised using a z-score to eliminate these outliers, with images 3 standard deviations away from the mean held constant at either the maximum or minimum value. Through intuition, Figure 41 shows that normalisation produces an image that gives a better description of the surface. Normalisation is performed exclusively for each geometric form.

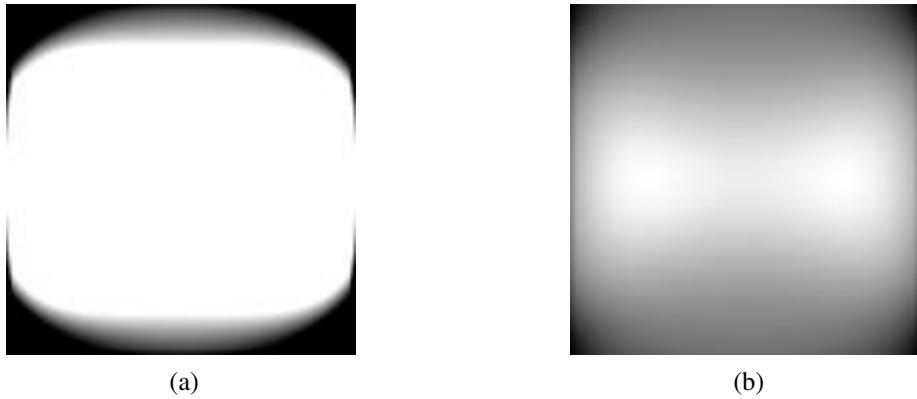


Figure 41: Comparison of the Gaussian curvature for a sample processed using the centre to outside tracking strategy in an image format using (a) unprocessed and (b) normalised data

Images across the dataset were visually verified to ensure that there are no errors. Figure 42 shows that the maximum principle curvature for this particular strategy displays more visible features than the Euclidean volumetric data despite the contrast being worse.

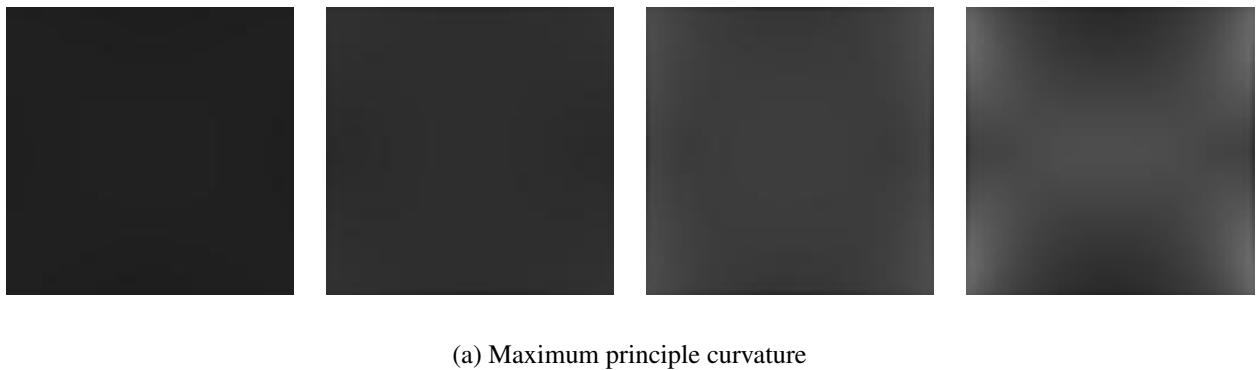


Figure 42: Comparison of image representations of different geometries for the 'centre-to-outside' #1 test case for all four passes. A comparison of all measures of geometry can be viewed in Appendix D.1

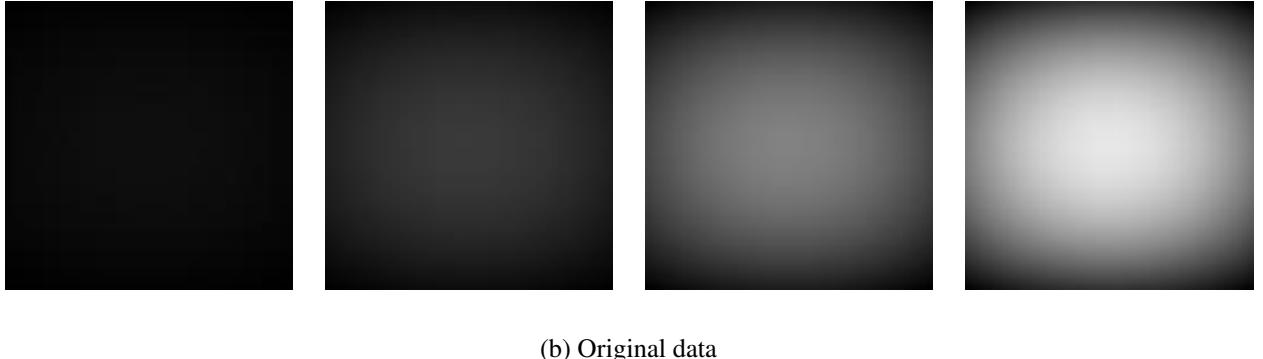


Figure 42: Comparison of image representations of different geometries for the 'centre-to-outside' #1 test case for all four passes. A comparison of all measures of geometry can be viewed in Appendix D.1

## 7.2 Unit testing

### 7.2.1 First layer

Initially the accuracy of the base-case model is assessed across all measures of geometry for varying resolutions. Table 12 suggests that the Euclidean volumetric data has a lower accuracy than all measures of geometry across all resolutions. This disparity reduces for a higher resolution, however it is still the least accurate form. The data shows most peaks to occur around 100 epochs, suggesting that the model is not tuned for maximum performance. This will be explored when the model parameters are tuned.

Table 12: Comparison of peak validation accuracy before overfitting of the base case model across all measures of geometry for a varying resolution. See Appendix D.2.1 for plots used to form this data.

Size (pixels)	Measure of geometry									
	Original		K		H		P <sub>1</sub>		P <sub>2</sub>	
	Acc. (%)	Epoch no.	Acc. (%)	Epoch no.	Acc. (%)	Epoch no.	Acc. (%)	Epoch no.	Acc. (%)	Epoch no.
10	31.6	100	68.4	100	81	63	81.6	100	73.7	100
40	73.9	95	87.0	77	85.7	75	84.2	100	86.8	100
80	78.4	94	82.1	62	86.8	73	86.8	85	86.8	100
125	79.5	80	82.1	79	88.4	33	85.3	68	89.5	100

Generally, Table 12 shows that for all measures of geometry a greater image resolution results in a higher accuracy. There are some instances where this does not apply, however this is due to the random nature of ML classification. It is assumed subsequent analysis in this study will use by default an input image size of 125 pixels.

Figure 43 shows a linear correlation between computation time and the size of image. This is similar for all measures of geometry. As this study is a proof-of-concept, computational efficiency is not critical. This may become more critical when applied practically when controlling the process in-the-loop.

The next stage in the testing framework assesses the effect of different data augmentation strategies on the classification accuracy for each measure of geometry for the base-case model. Figure 44 shows the validation accuracy for the model using a dataset with over 3 passes to be 'stepped' in shape for mean curvature. This is due to the small sample set, and results in large variations in validation accuracy. A smaller validation set

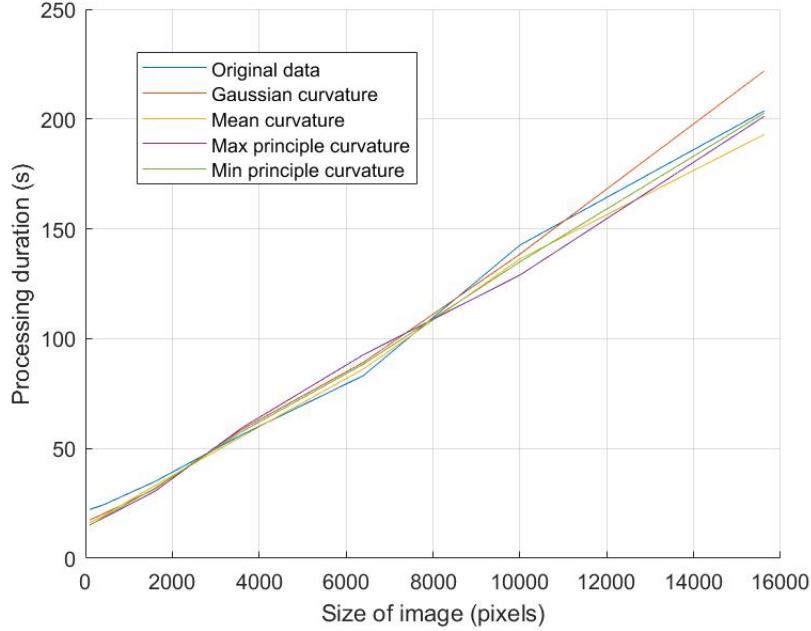


Figure 43: Effect of image resolution on computation duration for all geometrical forms

results in overfitting and accuracy misrepresentation. Therefore, this strategy is disregarded from subsequent analysis.

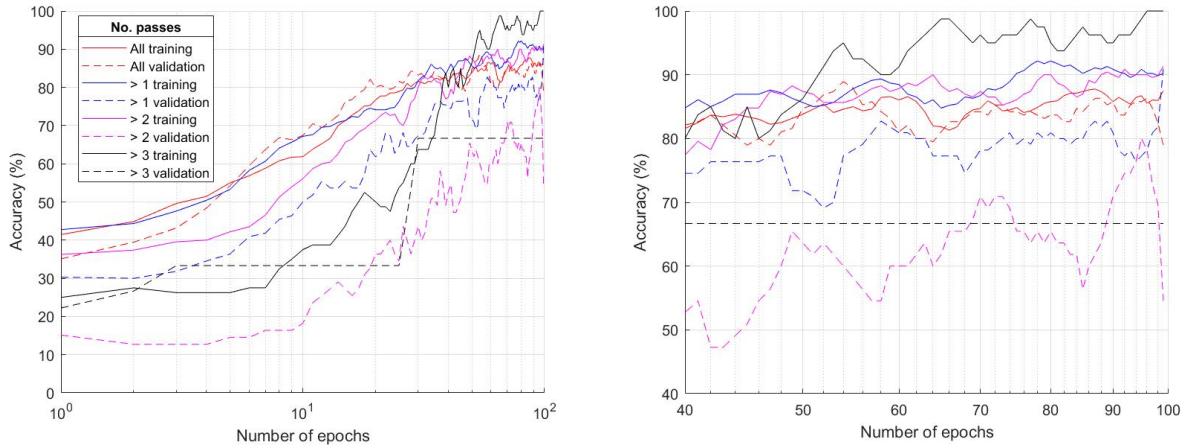


Figure 44: (a) Assessment of the training and validation accuracy of the base-case layer 1 model using mean curvature as the measure of geometry. This is across 100 epochs with varying augmentation strategies implemented. (b) Zoomed in view of Figure (a)

Figure 45 suggests that for a decreasing size of dataset, there is an increasing deviation between the training and validation accuracy for the model. This is consistent with all measures of geometry. A greater deviation suggests data is being over or underfitted, which is not useful. This deviation is accepted if the two accuracies converge at a higher number of epochs, however the plots in Appendix D.2.1 show this is not the case. A comparison of the accuracy of each augmentation strategy cannot be made due to bias caused by the variation in the size of dataset. Therefore, it is assumed subsequent analysis in this study will use by default the maximum size of dataset.

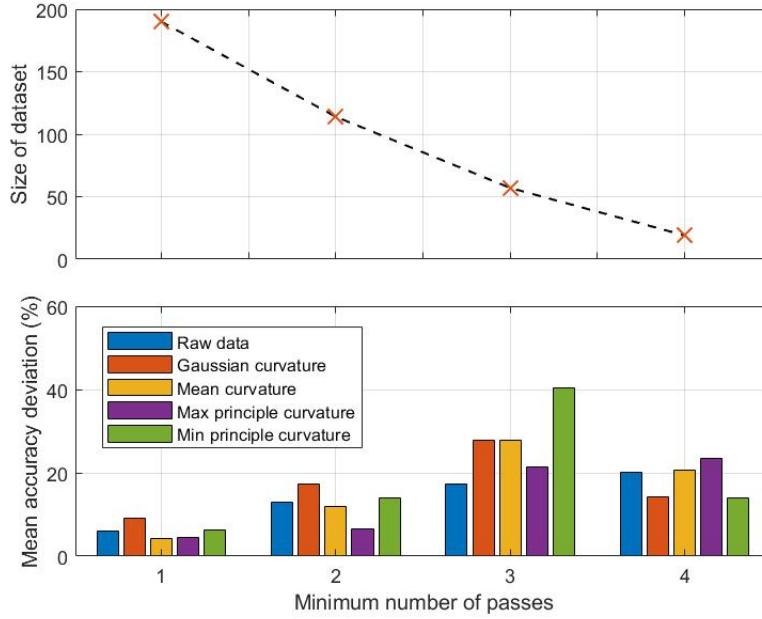


Figure 45: Assessing overfitting and underfitting of data of the layer 1 model when applying different augmentation strategies

Figure 46 suggests the accuracy of the model for each measure of geometry is similar. As this study is a proof-of-concept, the model will be tuned with  $P_1$  as the validation data fits the training data well.

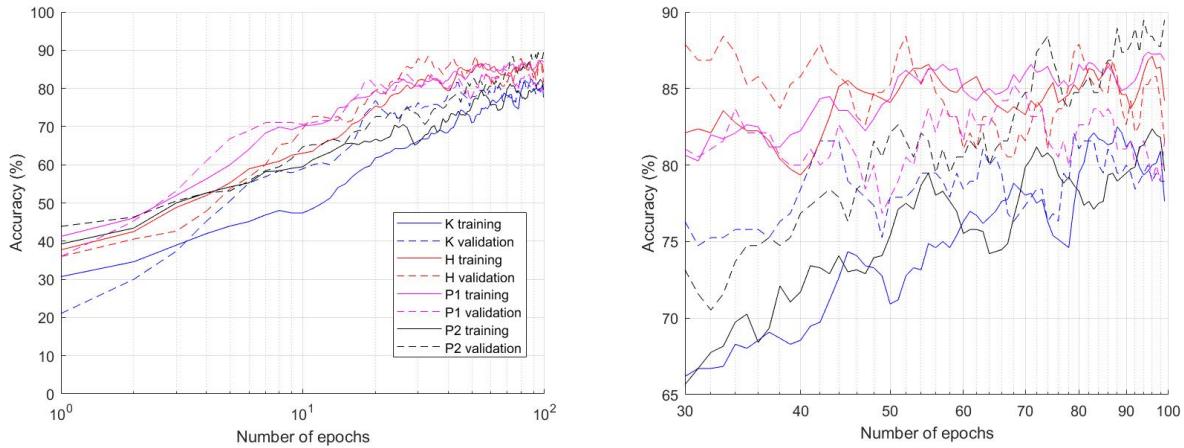


Figure 46: (a) Assessment of the training and validation accuracy of the base-case layer 1 model for each measure of geometry for images of size 125 pixels and use of entire dataset. This is across 100 epochs. (b) Zoomed in view of Figure (a)

The model was tuned by evaluating different sized networks outlined in Section 6.5.2. Figure 47 shows that the networks are overfitting at a varying number of epochs. The ‘tiny’ network performs the best, overfitting at 92%. To confirm this, the categorical cross-entropy is compared in Figure 48. It displays similar a trend, with the ‘tiny’ network chosen for the tuned model with 130 epochs.

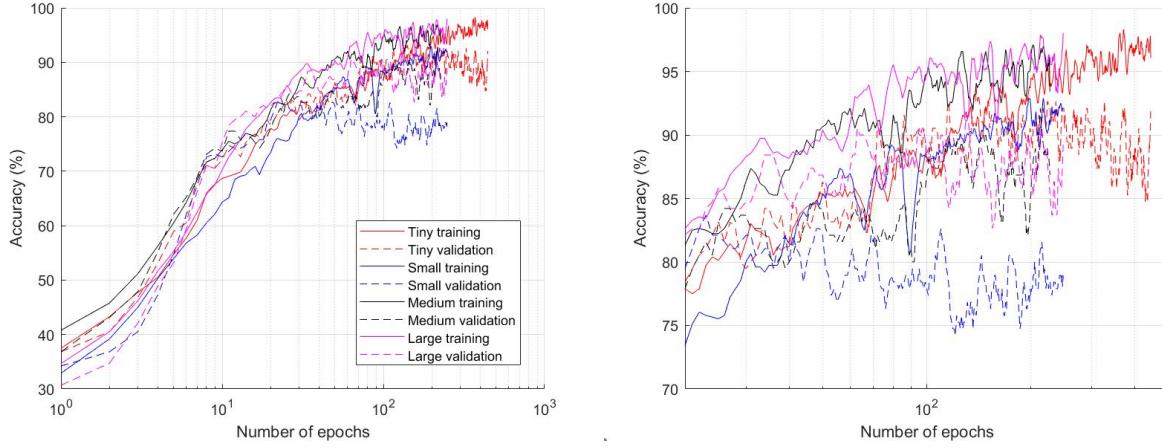


Figure 47: (a) Assessment of the training and validation accuracy of the layer 1 model using  $P_1$  for varying neural network size. The entire dataset is used, with images of size 125 pixels. The model is trained across 100 epochs. (b) Zoomed in view of Figure (a)

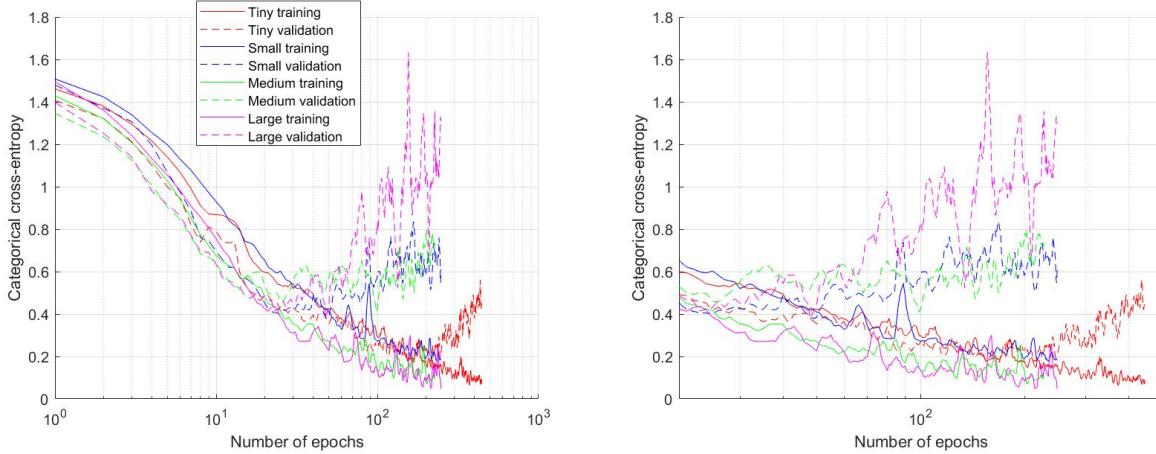


Figure 48: (a) Assessment of the training and validation categorical cross-entropy of the layer 1 model using  $P_1$  for varying neural network size. The entire dataset is used, with images of size 125 pixels. The model is trained across 100 epochs. (b) Zoomed in view of Figure (a)

### 7.2.2 Second layer

It is assumed that the analysis of the resolution and data augmentation within the first layer applies to the second layer. Therefore, the subsequent analysis will be based on a image size of 125 pixels and encompass the entire dataset.

The MAE is evaluated for a varying size of network across all strategies for each parameter. This is accumulated per strategy as the size of network can only be varied between strategies. These results are plotted in Appendix D.2.2. The network size and epoch for which the cumulative MAE is at a minimum is recorded in Table 13. For each strategy, it was ensured that the data was not being overfitted within the range of epochs listed.

Table 13: Selection of neural network size and epoch for layer 2 with respective mean MAE per strategy for all strategies. MAE is of normalised parameters

Strategy	Network size	Sum of MAE across parameters	Epoch number	Number of parameters	Average MAE across parameters
Centre-to-outside	Medium	0.3	100	6	0.05
Overlayed	Small	0.15	100	5	0.03
Vertical expansion	Small	0.55	100	4	0.14
Horizontal expansion	Medium	0.1	160	4	0.025
Triangular	Large	0.12	100	6	0.02

The ‘vertical expansion’ strategy has the greatest comparative error, being at least 3 times greater on average per parameter. Figure 49 suggests the main source of this error is due to the prediction of the central x position.

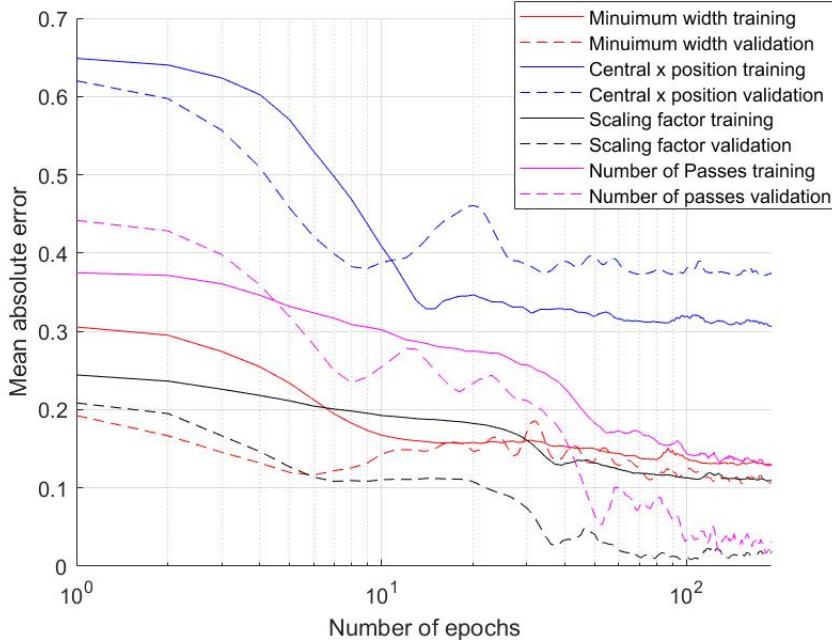


Figure 49: Comparison of the MAE in the layer 2 model for parameters defining the ‘Vertical expansion’ tracking strategy using the ‘small’ sized network. MAE is of normalised parameters

### 7.3 Integration testing

The two layers were combined, configuring the model using the tuned hyperparameters defined in the previous section. This model was then trained and the performance assessed using the test set. Figure 50 shows a first layer classification accuracy of 83%.

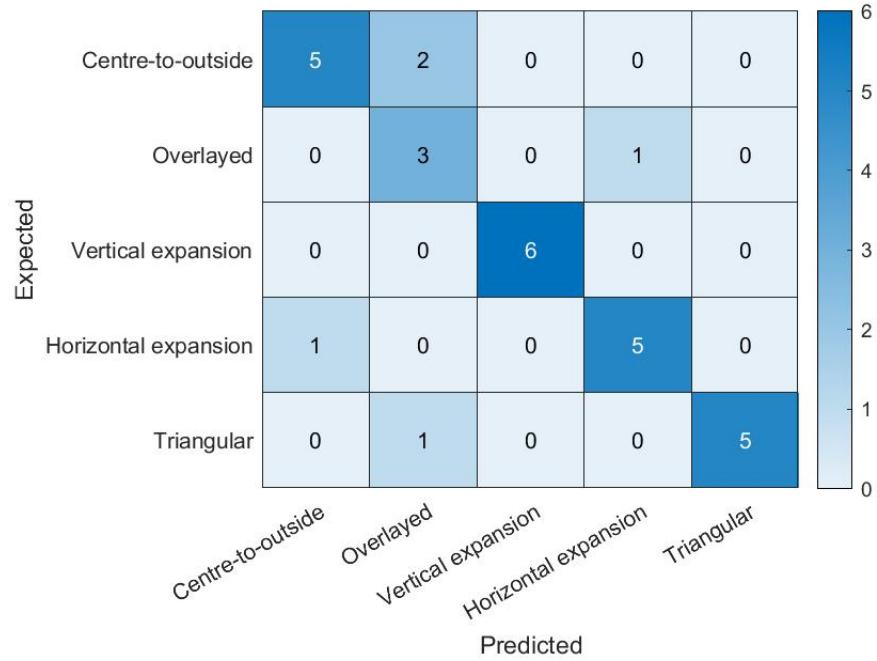


Figure 50: Confusion matrix for the first layer in the CNN architecture using the test set

Figure 51 shows the mean percentage error of the second layer for each parameter of each strategy.

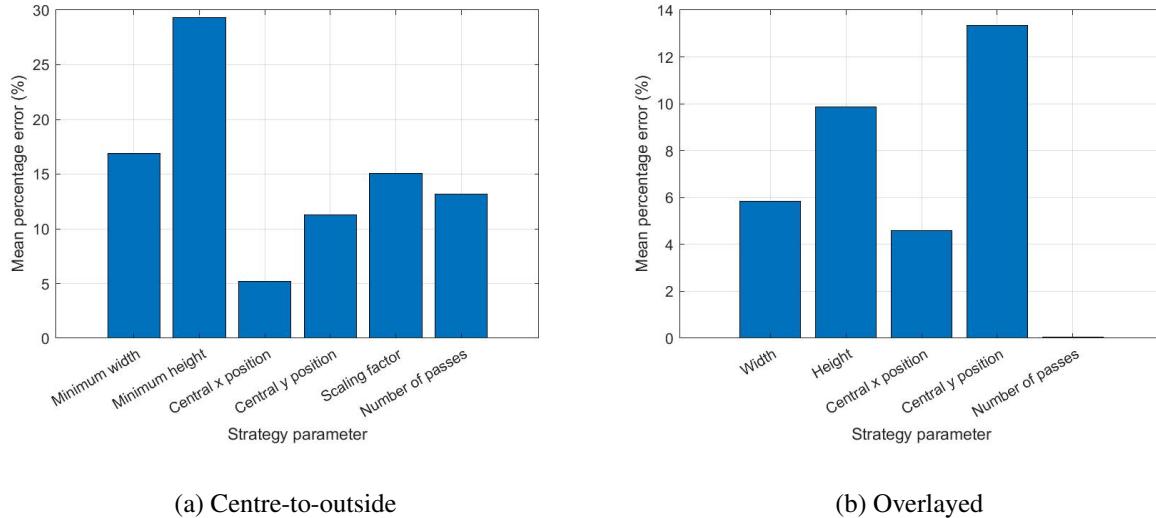
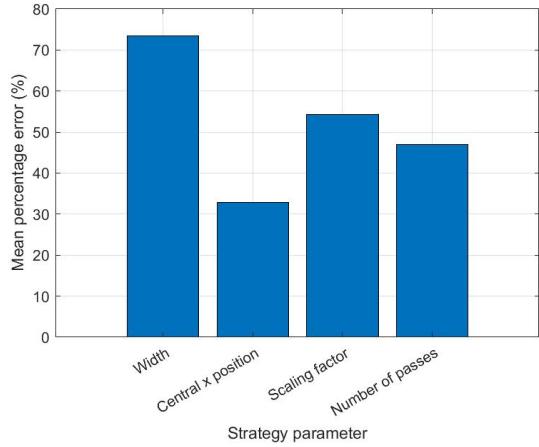
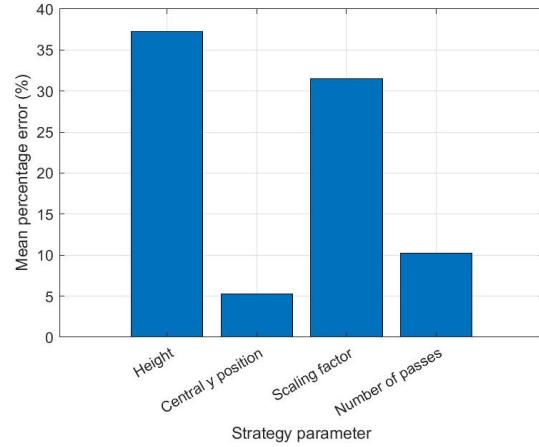


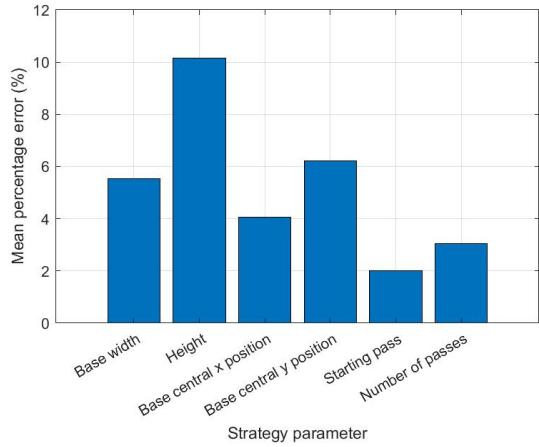
Figure 51: Mean percentage error of layer 2 model predictions for all strategies and all strategy parameters on the test set. This is with  $P_1$  used as input data, and all passes used in the dataset.



(c) Vertical expansion



(d) Horizontal expansion



(e) Triangular

Figure 51: Mean percentage error of layer 2 model predictions for all strategies and all strategy parameters on the test set. This is with  $P_1$  used as input data, and all passes used in the dataset.

## 8 Discussion and future work

### 8.1 Analysis of the final architecture

The optimisation of the first layer within the ML architecture yields an unbiased accuracy of 83%, which deviates from the validation accuracy of 92%. It is difficult to explain this deviation, as it could be due to the stochastic nature of ML architecture, which is made more significant due to the small sample size. More likely is that the model is overfitting the validation data giving an over-optimistic accuracy. A meaningful comparison cannot be made of classification accuracy for each strategy. This is because of the small number of samples and the unrepresentative sample population as the division of samples was created randomly. Despite this, an accuracy above 70% is consistent across all strategies.

The ‘overlaid’ strategy has the greatest amount of false predictions. These false predictions occurred for the centre-to-outside strategy for a one and two pass sample, and for the ‘triangular’ strategy for a two pass sample. ‘Overlaid’ is the only strategy that does not vary its tracking area between passes. Therefore, the mechanics of the process of wheeling a lower number of passes is most closely associated with this strategy. This may explain the higher number of false predictions. This is inferred from data with a small population, and requires a greater number of samples for verification.

Alternatively, false predictions may arise due to multiple tracking strategies being able to form the same sheet geometry. The classifier makes predictions based upon the most probable outcome, and there may be two or more strategies with similar predicted probabilities. To better understand the capability of the heuristic strategies, an investigation should be conducted on the feasibility of these tracking strategies in producing a set of standard geometries.

Within the second layer of the architecture, the ‘overlaid’ and ‘triangular’ strategies display the smallest percentage errors, the maximum between them being a 14% mean deviation from the true value. The ‘vertical expansion’ tracking strategy displays the largest error, the maximum being 72% mean deviation from the true value. This disparity in error could be a result of the limitation of the data or the model. Most likely, it is because the ‘vertical expansion’ strategy has a sample size 25% smaller than the ‘overlaid’ and ‘triangular’ datasets. This means the model is more likely to overfit the data.

These results suggest that the error is more significant in the second layer of the architecture than the first. This is as expected, as the second layer involves a multi-output regression problem. In practice, this error would be reduced through implementation of the full non-conventional control architecture. It is proposed that reinforcement techniques are used to iteratively optimise the model. This is combined with real-time, closed-loop control techniques informed by data gathered from a system of sensors to further reduce error during the forming process.

### 8.2 Limitations of data

The synthetic data used in the model is of limited use assessing the practical application of automating the EW. This is because there is no comparative measure of accuracy between the synthetic data and the actual process. Despite this, the data is useful for assessing the architecture as a proof-of-concept when applied to a problem of this nature. At the very least, it shows that the procedures put in place can handle the format and quantity of data. It also suggests that the model is capable of detecting rules within similarly formatted data as the experimental dataset, however it must be noted that the rules derived are not the same.

The data has several practical limitations that result in issues in the model. The dataset included all possible pass configurations as it displayed good fitting properties. This is despite the dataset including samples with only one pass, which are not descriptive enough to be attributed to one strategy. Despite this, the model has an accuracy attributed to these one pass cases of over 70%. This is because the model is overfitting the dataset due to the limited size and diversity of sample population for each strategy. It is suggested that future experimental work is performed with a greater number of passes so the relationship between the number of passes and accuracy of the model can be investigated. From this, a lower limit for the number of passes can be defined.

Images in Figure 42 display very small changes of geometry that are not visible due to the range of normalisation being discrete values between 0 to 255. This means features that the CNN relies on for classification cannot be extracted. One method proposed is introducing a filter that increases contrast on individual images to make features more visible.

The study concluded that there was minimal difference in each measure of geometry when assessing the performance of the model.  $P_1$  was selected as it displayed marginally better fitting characteristics, however  $K$  is theoretically more applicable as it is intrinsic to the geometry of the surface. Further refinement of the model and data is required to compare the performance of the different measures of geometry.

### 8.3 Limitations of model

The small size of sample population is a significant limiting factor in the feasibility of applying the ML methods proposed in this study practically. Despite the sampling methods being designed to be representative, the model overfits to these specific examples due to the size of the sample population. A practical ML control system must be built using a large dataset that produces a better fitted model.

This study does not consider bias attributed to the selection of training data. Only one configuration is considered, which is not representative of the entire dataset. Instead, [Raschka \(2018\)](#) proposes a technique called ‘k-fold cross-validation’, where the dataset is iterated over  $k$  times. For each iteration, the dataset is split into  $k$  parts, with one part being the validation set, and the remaining parts merged into the training set. This results in a more representative value for the accuracy.

A fundamental reason for errors in predictions across both layers is sub-optimal tuning of the model. The testing framework overlooks several hyperparameters, such as those defining the dropout or learning rate of the model. This can be tuned using tools available in TensorFlow such as the ‘HParams Dashboard’, which allows users to visualise and tune parameters ([Google, 2022](#)).

### 8.4 Future work

The architecture proposed within the study was not fully implemented, with the non-parametric layers incomplete due to time limitations. The advantage of the non-parametric method is that it reduces the need to classify the tracking strategy. This model would also be useful for furthering the understanding of local mechanics of the wheeling process, as it would build a relationship between the local tracking density and curvature of the sheet. Despite this, the difficulty is that the model has a multi-dimensional input and output, and it is uncertain whether the methods suggested in this study would be effective. Instead, state-of-the-art models presented in Section 2.3 might be more useful. There is also an issue associated with the stochastic description of the tracking density. For it to be effectively applied, the number of passes on the sheet would need to be large enough for the PDF to be statistically significant.

As well as improving the architecture, future work must focus on resolving the limitations of the synthetic data. Experimental data should be derived using the test cases in Section 4.8 using the experimental procedure proposed in Section 5. A sample set of 25 sheets have been bought and are ready to be tested. The pre-processing and post-processing processing tools, as well as the model architecture, is available on Github to view at [https://github.com/adamkirchel/FYP\\_submission.git](https://github.com/adamkirchel/FYP_submission.git).

## 9 Conclusion

This study provides a partial proof-of-concept for the use of ML methods for automation of the EW using synthetic data. This was achieved by meeting the objectives stated in Section 1.2. Parametric and non-parametric methods have been proposed to represent the EW process, which have been used to characterise heuristic tracking strategies used by the traditional smith. This novel documentation can be used to scientifically quantify the mechanics of these techniques in further study.

Using this documentation, a ML architecture has been built that was optimised through an iterative testing framework. The results suggest that the architecture proposed can successfully predict tracking strategies to

an accuracy of 83%. There was more significant error in the prediction of strategy parameters, however this was expected considering the number of outputs of the model, the small sample population, and sub-optimal model design. It was acknowledged that these results are limited by the synthetic nature of the data, and this study instead serves as justification for performing a full proof-of-concept using experimental data.

For this purpose, a series of experiments have been proposed to be carried out beyond this paper using the parametric representation of the EW. These experiments are aided by a UI application introduced in this paper that gives an accessible method to define tracking strategies for use on the EW machine. The experiments have been determined using novel sampling methodologies to reduce bias, however it is accepted that this bias is an inherent problem given the small sample population. Amendments have been suggested for the experiments to include greater number of passes to ensure bias is minimised.

## References

- Abdolrasol, M. G. M., Hussain, S. M. S., Ustun, T. S., Sarker, M. R., Hannan, M. A., Mohamed, R., Ali, J. A., Mekhilef, S. and Milad, A. (2021), 'Artificial neural networks based optimization techniques: A review', *Electronics* **10**(21).
- URL:** <https://www.mdpi.com/2079-9292/10/21/2689>
- Ahmed, E., Saint, A., Abdelrahman, S., Cherenkova, K., Das, R., Gusev, G. and Aouada, D. (2020), Textual evidence for the perfunctoriness of independent medical reviews, Vol. 2657, CEUR-WS, pp. 1–9.
- Allwood, J. M., Duncan, S. R., Cao, J., Groche, P., Hirt, G., Kinsey, B., Kuboki, T., Liewald, M., Sterzing, A. and Tekkaya, A. E. (2016), 'Closed-loop control of product properties in metal forming', *CIRP Annals - Manufacturing Technology* **65**, 573–596.
- Aloysius, N. and Geetha, M. (2018), A review on deep convolutional neural networks, Vol. 2018-January, Institute of Electrical and Electronics Engineers Inc., pp. 588–592.
- Alzubi, J., Nayyar, A. and Kumar, A. (2018), Machine learning from theory to algorithms: An overview, Vol. 1142, Institute of Physics Publishing.
- Anderson, D. (1990), 'The english wheel book'.
- Bajaj, C. L. (1994), 'Polynomial surface patch representations'.
- URL:** <https://docs.lib.psu.edu/cstech/1138WWW> [www.cs.psu.edu](http://www.cs.psu.edu)
- Barr, E. (2013), 'Professional sheet metal fabrication'.
- Bell, C., Jump, E., Kerr, W., Corney, J., Zuelli, N. and Savings, D. (2016), 'Correlation between von mises strain and material thinning in a hydroformed sample of ti35a aerospace grade titanium'.
- Botev, Z. I., Grotowski, J. F. and Kroese, D. P. (2010), 'Kernel density estimation via diffusion', *Annals of Statistics* **38**, 2916–2957.
- Bowen, D. T., Music, O., Erdinc, A. A., Shokrani, A. and Loukaides, E. G. (2021), Numerical modelling and deformation mechanics of the english wheel process, Springer Science and Business Media Deutschland GmbH, pp. 457–466.
- Bowen, D. T., Russo, I. M., Cleaver, C. J., Allwood, J. M. and Loukaides, E. G. (2022), 'From art to part: Learning from the traditional smith in developing flexible sheet metal forming processes'.
- Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A. and Vandergheynst, P. (2017), 'Geometric deep learning: Going beyond euclidean data', *IEEE Signal Processing Magazine* **34**(4), 18–42.
- URL:** <https://doi.org/10.1109%2Fmsp.2017.2693418>
- Cho, H. and Leu, M. (1998), 'Artificial neural networks in manufacturing processes: Monitoring and control', *IFAC Proceedings Volumes* **31**(15), 529–537. 9th IFAC Symposium on Information Control in Manufacturing 1998 (INCOM '98), Nancy, France, 24–26 June.
- URL:** <https://www.sciencedirect.com/science/article/pii/S1474667017406070>
- Cooper, D. R., Rossie, K. E. and Gutowski, T. G. (2017), 'The energy requirements and environmental impacts of sheet metal forming: An analysis of five forming processes', *Journal of Materials Processing Technology* **244**, 116–135.

- Daehn, G. S. (2021), A vision of numerically controlled, autonomous manufacturing and metal forming, Springer Science and Business Media Deutschland GmbH, pp. 13–23.
- Dasgupta, A. and Nath, A. (2016), ‘Classification of machine learning algorithms’, *International Journal of Innovative Research in Advanced Engineering (IJIRAE)* **3**.
- URL:** [www.ijirae.com](http://www.ijirae.com)
- Eastwood (2013), ‘English wheel instructions’.
- URL:** <https://www.eastwood.com/metal-fabrication/english-wheel.html>
- Golle, M., Hoffmann, H., Lohmann, B., Lueth, T. C., Scherer, D. and Weber, S. (2007), Driving as a flexible manufacturing method for mass customization of individualized sheet metal products.
- URL:** <http://www.utg.de>,
- Google (2022), ‘Hyperparameter tuning with the hparams dashboard’. Accessed: 2022-05-03.
- URL:** [https://www.tensorflow.org/tensorboard/hyperparameter\\_tuning\\_with\\_hparams](https://www.tensorflow.org/tensorboard/hyperparameter_tuning_with_hparams)
- Groover, M. (2015), *Automation, Production Systems, and Computer-Integrated Manufacturing*, 4 edn, Pearson Higher Education.
- Hamouche, E. and Loukaides, E. G. (2018), ‘Classification and selection of sheet forming processes with machine learning’, *International Journal of Computer Integrated Manufacturing* **31**, 921–932.
- Hartmann, C., Opritescu, D. and Volk, W. (2016), ‘An artificial neural network approach for tool path generation in incremental sheet metal free-forming’, *Journal of Intelligent Manufacturing* **30**, 757–770.
- Hartmann, C. and Volk, W. (2019), Knowledge-based incremental sheet metal free-forming using probabilistic density functions and voronoi partitioning, Vol. 29, Elsevier B.V., pp. 4–11.
- Hou, Q., Leng, J., Ma, G., Liu, W. and Cheng, Y. (2019), ‘An adaptive hybrid model for short-term urban traffic flow prediction’, *Physica A: Statistical Mechanics and its Applications* **527**.
- Ilangovan, B., Monfared, R. P. and Jackson, M. (2016), ‘An automated solution for fixtureless sheet metal forming’, *International Journal of Advanced Manufacturing Technology* **82**, 315–326.
- Jia, Y.-B. (2020), ‘Gaussian and mean curvatures \*’.
- Kazmi, I., You, L. and Zhang, J. J. (2013), A survey of 2d and 3d shape descriptors, pp. 1–10.
- Liu, S., Shi, Z., Lin, J. and Li, Z. (2020), ‘Reinforcement learning in free-form stamping of sheet-metals’, *Procedia Manufacturing* **50**, 444–449. 18th International Conference on Metal Forming 2020.
- URL:** <https://www.sciencedirect.com/science/article/pii/S2351978920317790>
- Longyord, W. (2014), *Learning the English Wheel*.
- Lundstrom, M. (2003), ‘Moore’s law forever?’, *Science* **299**, 210–211.
- URL:** <https://www.jstor.org/stable/3833324>
- Marciniak, Z., Duncan, J. and Hu, S. (2002), *Mechanics of Sheet Metal Forming*, 2 edn, Elsevier.
- Minton, J. (2017), ‘Mathematical modelling of asymmetrical metal rolling processes’.
- Modes, C. D., Bhattacharya, K. and Warner, M. (2011), ‘Gaussian curvature from flat elastica sheets’, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **467**, 1121–1140.
- Mori, K., Yamamoto, M. and Osakada, K. (1996), ‘Determination of hammering sequence in incremental sheet metal forming using a genetic algorithm’, *Journal of Materials Processing Technology* **60**, 463–468.
- Music, O. and Allwood, J. M. (2012), ‘The use of spatial impulse responses to characterise flexible forming processes with mobile tools’, *Journal of Materials Processing Technology* **212**, 1139–1156.
- Opritescu, D., Sachnik, P., Yang, Z., Golle, R., Volk, W., Hoffmann, H., Schmiedl, F., Ritter, M. and Gritzmann, P. (2012), Automated driving by standardizing and scaling the manufacturing strategy, Vol. 3, Elsevier B.V., pp. 138–143.
- Opritescu, D. and Volk, W. (2015), ‘Automated driving for individualized sheet metal part production - a neural network approach’, *Robotics and Computer-Integrated Manufacturing* **35**, 144–150.
- Opritescu, D. and Volk, W. (2016), ‘Variation of components by automated driving: A knowledge-based approach for geometric variance’, *International Journal of Material Forming* **9**, 9–19.

- Phung and Rhee (2019), 'A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets', *Applied Sciences* **9**, 4500.
- Praveena, M. and Jaignaesh, V. (2017), 'A literature review on supervised machine learning algorithms and boosting process', *International Journal of Computer Applications* **169**, 975–8887.
- Raschka, S. (2018), 'Model evaluation, model selection, and algorithm selection in machine learning'.  
**URL:** <http://arxiv.org/abs/1811.12808>
- Rentsch, B. (2018), 'Virtual design and adaptive control of metal spinning processes'.
- Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, Cambridge University Press.
- Rossi, G. . and Nicholas, P. (2018), 'Modelling a complex fabrication system', *Proceedings of eCAADe 2018: Computing for a better tomorrow* pp. 811–820.
- Salvador, J., Oliveira, J. and Breternitz, M. (2020), 'Reinforcement learning: A literature review'.  
**URL:** <https://www.researchgate.net/publication/344930010>
- Schuh, G., Bergweiler, G., Fiedler, F., Bickendorf, P. and Colag, C. (2019), A review on flexible forming of sheet metal parts, IEEE Computer Society, pp. 1221–1225.
- Shi, B., Bai, S., Zhou, Z. and Bai, X. (2015), 'Deeppano: Deep panoramic representation for 3-d shape recognition', *IEEE Signal Processing Letters* **22**(12), 2339–2343.
- Shining3D (2022), 'Einscan pro 2x plus'. Accessed: 2022-05-03.  
**URL:** <https://www.einscan.com/handheld-3d-scanner/2x-plus/>
- Shorten, C. and Khoshgoftaar, T. M. (2019), 'A survey on image data augmentation for deep learning', *Journal of Big Data* **6**, 199–207.
- Sinha, A., Bai, J. and Ramani, K. (2016), Deep learning 3d shape surfaces using geometry images, in B. Leibe, J. Matas, N. Sebe and M. Welling, eds, 'Computer Vision – ECCV 2016', Springer International Publishing, Cham, pp. 223–240.
- Vazquez, R. E. and Coleman, J. (2017), 'Efficient fabrication of double curved surfaces'.  
**URL:** <https://www.azahner.com/blog/efficient-fabrication-of-double-curves-in-architecture/1/25>
- Witczak, M. (2007), *Lecture Notes in Control and Information Sciences* 354.
- Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J. (2014), '3d shapenets: A deep representation for volumetric shapes'.  
**URL:** <https://arxiv.org/abs/1406.5670>
- Xu, D., Shi, Y., Tsang, I. W., Ong, Y. S., Gong, C. and Shen, X. (2020), 'Survey on multi-output learning', *IEEE Transactions on Neural Networks and Learning Systems* **31**, 2409–2429.
- Yang, D. Y., Bambach, M., Cao, J., Duflou, J. R., Groche, P., Kuboki, T., Sterzing, A., Tekkaya, A. E. and Lee, C. W. (2018), 'Flexibility in metal forming', *CIRP Annals - Manufacturing Technology* **67**, 743–765.
- Yuanyai, C. and Nembhard, H. B. (2010), Design of experiments: A key to innovation in nanotechnology.
- Zhang, W. (2020), 'Geometry of curves and surfaces'.
- Zhi, S., Liu, Y., Li, X. and Guo, Y. (2018), 'Toward real-time 3d object recognition: A lightweight volumetric cnn framework using multitask learning', *Computers Graphics* **71**, 199–207.  
**URL:** <https://www.sciencedirect.com/science/article/pii/S0097849317301735>
- Zhong, R. Y., Xu, X., Klotz, E. and Newman, S. T. (2017), 'Intelligent manufacturing in the context of industry 4.0: A review', *Engineering* **3**, 616–630.
- Zou, J., Han, Y. and So, S.-S. (2009), *Overview of Artificial Neural Networks*, Humana Press, Totowa, NJ, pp. 14–22.  
**URL:** [https://doi.org/10.1007/978-1-60327-101-1\\_2](https://doi.org/10.1007/978-1-60327-101-1_2)

## A Design of experiments documentation

### A.1 Design specification

Table 14: Specification for the design of experiments

Ref.	Criteria	Description	Relevance to objective 1	Relevance to objective 2
A1	Accuracy of measurement	Data needs to be defined to a specific accuracy	Future researchers can be assured of its limitations	Classification will be more efficient if data is more accurate
A2	Reproducibility of experiment	Method needs to be presented to allow experiment to be reproduced	This allows researchers to be able to use this dataset in their own research	The dataset can be extended by other researchers to create a more efficient model
A3	Repeatability of experiment	Procedure repeatable for use on the same machine in the same conditions	Errors within the experiment are minimised to ensure the data is reliable for use in research	Errors within the experiment are minimised to ensure greater accuracy in classification
A4	Relevant results	Experiment relevant to industrial forming processes and traditional techniques used by artisans	This data is more appealing for research if it has an applicable context in industry	By making the data relevant, the results from the ML model are also relevant in industrial applications
A5	Amount of data	Strategy must be able to generate a large amount of data	A greater sample size provides greater statistical significance to the results, and greater insights to inform future research	ML models perform better when there is more data for it to learn. Despite this, the data needs to be representative of all processes otherwise there is the threat of bias
A6	Diversity of data	Data must cover numerous strategies in order to broaden understanding of global deformation	Results spanning a broader set of manufacturing strategies will give greater scope for future research in identifying what strategies to prioritise	Data spanning a greater range of strategies would mean that the results can be differentiated more easily, and therefore the algorithms defining classification are more effective
A7	Scalability of data	The method in which the data is represented must be scalable to represent a spectrum of patterns for one manufacturing strategy	Enabling scalability in the data allows for results to be interpolated and a broader data set to be formed	Classification is more effective if results are scalable and can be interpolated

Table 14: Specification for the design of experiments

Ref.	Criteria	Description	Relevance to objective 1	Relevance to objective 2
A8	Bias in data collected	Bias in the data must be accounted for and minimised	If bias is identified and accounted for then there is transparency in the data collected and it can be seen as a trusted academic source	The ML model may not be representative of the process due to do under or overfitting of data, or due to bias associated with the specific samples selected

## A.2 Experimental concepts

### A.1 - Tracking strategies from heuristic techniques

An example strategy is shown in Figure 52 where one side of the sheet is wheeled more than the other. [Longyard \(2014\)](#) states this pattern induces greater curvature on the side that is wheeled more. Figure 52 shows the potential variation of the tracking pattern for this strategy. The spacing between tracks would remain constant as this is not the focus of this experiment.

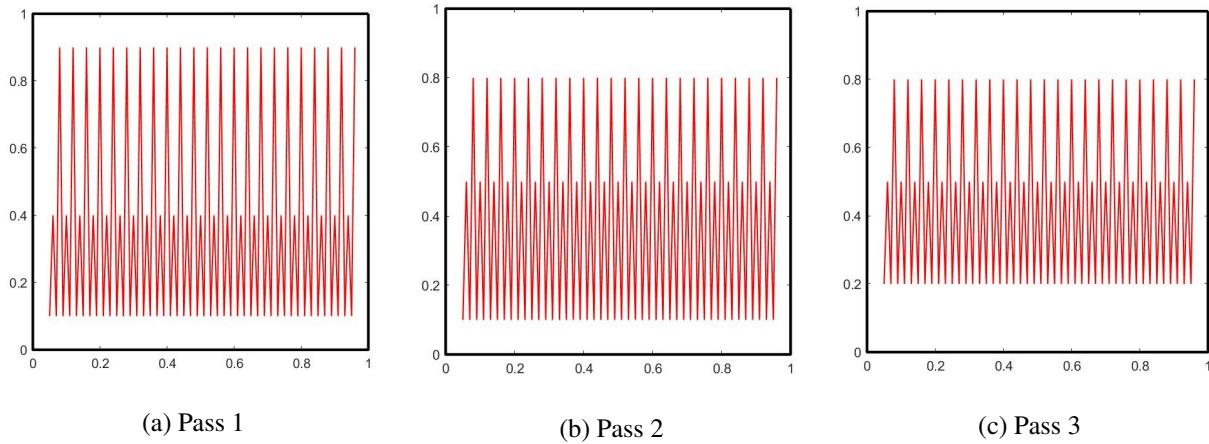


Figure 52: Concept A.1 - subsequent passes for one particular tracking strategy with (a) Lower amplitude of 0.3 upper amplitude of 0.8, (b) Lower amplitude of 0.4 upper amplitude of 0.7, and (c) Lower amplitude of 0.3 upper amplitude of 0.6 and starting position moved up by 0.1.

Other potential strategies are shown in Figure 53. Each strategy would be defined using a set of characteristic equations. This data set would allow analysis of the process mechanics of individual strategies from the variation between passes, and the comparison of strategies themselves. From this data set, strategies that have only previously been known heuristically can be scientifically defined and form a basis for automation techniques using the EW.

### A.2 - Tracking spacing

Figure 55 describes this concept across three different samples. This is similar to method deployed by ([Rossi and Nicholas, 2018](#)), who increased the tracking spacing across 6 samples from 5 to 35 mm. They recorded greater curvature in samples with a smaller spacing.

### B.1 - Area wheeled with reference to a point

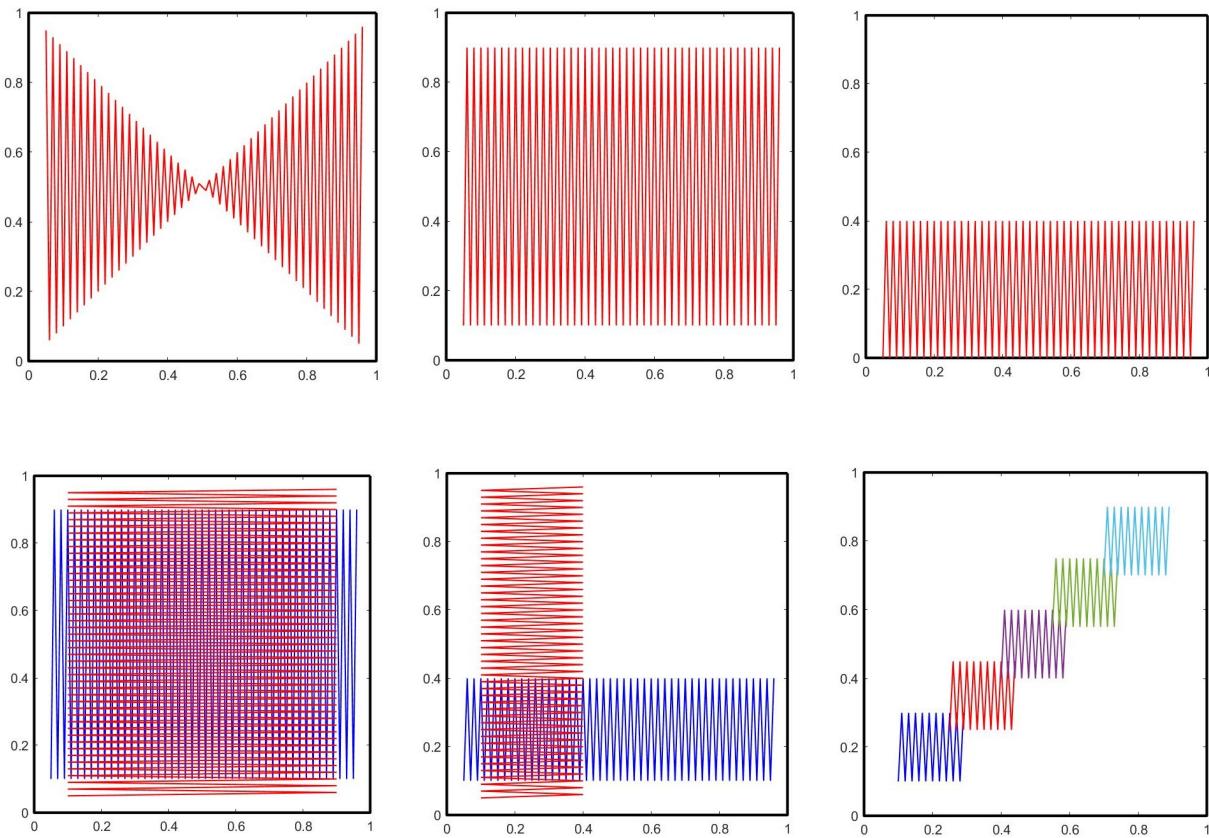


Figure 53: Concept A.1 - potential tracking strategies

Figure 55a shows one potential test case. For a square sheet, there are two planes of symmetry in a square sheet given that the tracking pattern cannot be rotated. Therefore these points are defined within a quarter of the sheet.

### B.2 - Area wheeled with reference to a line

Figure 55b shows one potential test case.

### C.1 - Distance from the edge of the sheet

Figure 56 visualises the procedure. Due to the planes of symmetry, only a quarter of the sheet needs to be considered.

### C.2 - Discrete cellular wheeling

An example is shown in Figure 57a with three different cells wheeled.

### C.3 - Discrete striped wheeling

An example is shown in Figure 57b with three different regions wheeled.

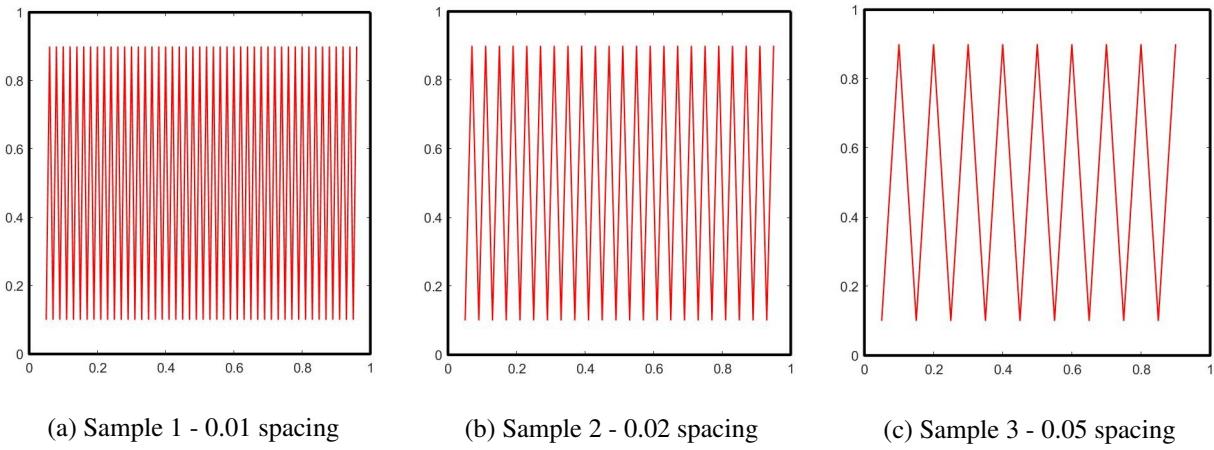


Figure 54: Concept A.2 - Increasing the spacing of tracking patterns

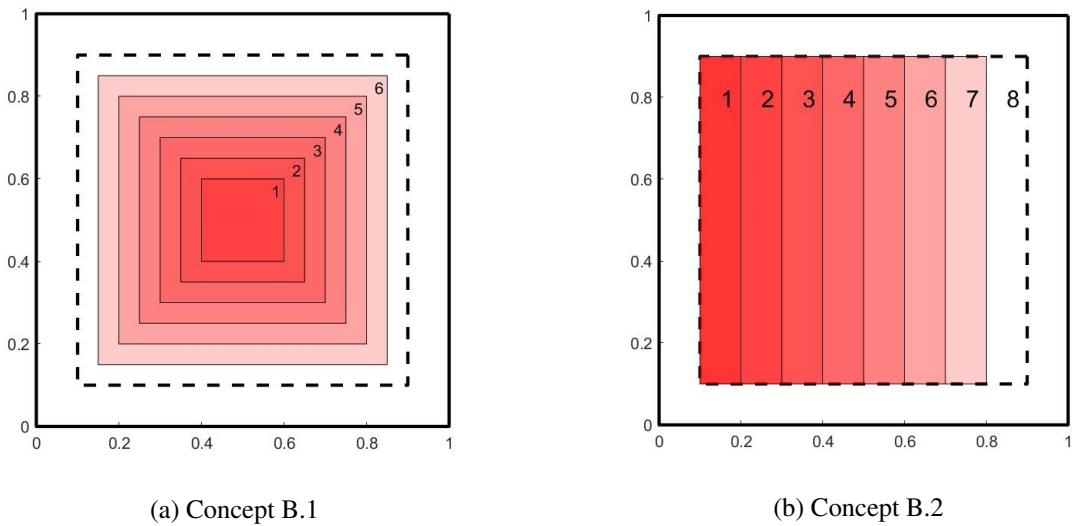


Figure 55: (a) Concept B.1 - An example strategy varying the area wheeled with reference to the central point, and (b) Concept B.2 - An example strategy varying the area wheeled with reference to a line coincident to the left-hand edge.

Concepts D.1 - G.1 do not have visual representations as they are self-explanatory.

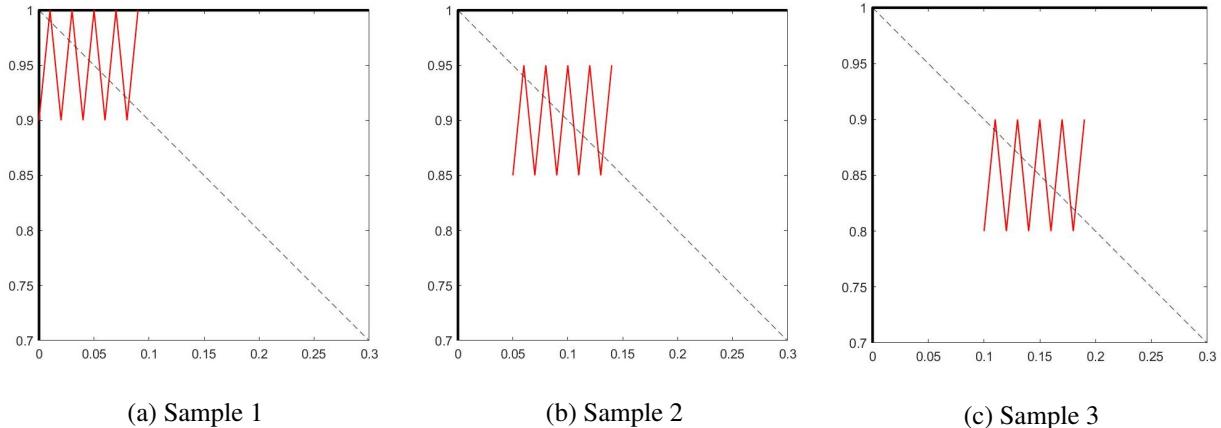


Figure 56: Concept C.1 - Example tracking patterns with distance between the centre of the pattern and the outer edge increasing. (a) 0.05 distance from outer edges, (b) 0.1 distance from outer edges, and (c) 0.15 distance from outer edges

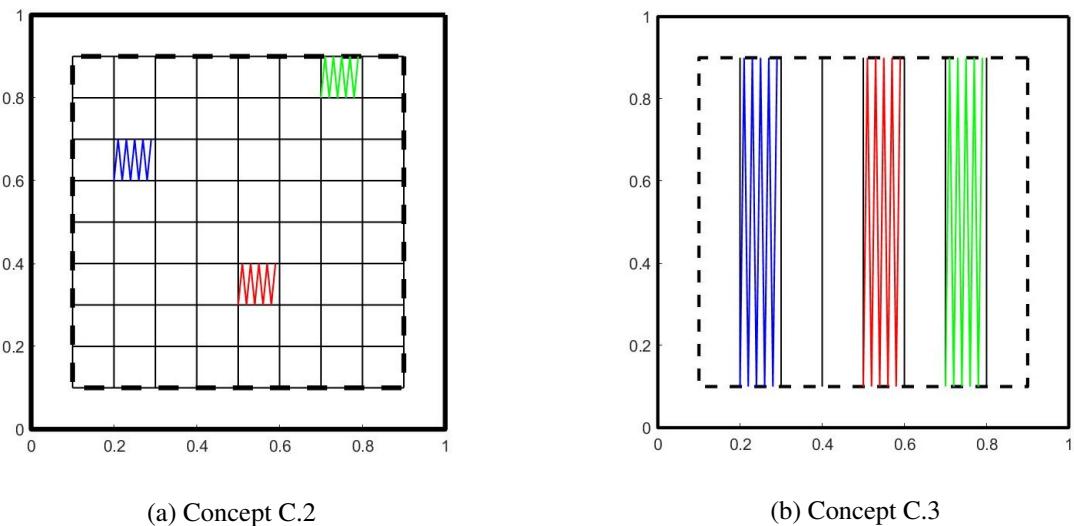


Figure 57: (a) Concept C.2 - Example sequence for the cellular wheeling concept. Order is red, blue then green. (b) Concept C.3 - Example sequence for the striped wheeling concept. Order is red, blue then green.

### A.3 Tracking strategies

Table 15 is used as reference for the characteristic equations for each tracking strategy.

Table 15: Parameters describing the characteristics of each tracking strategy

Strategy	Symbol	Definition
Centre-to-outside	$w_{min}$	Minimum width (mm)
	$h_{min}$	Minimum height (mm)
	$x_c$	Centre X (mm)
	$y_c$	Centre Y (mm)
	$F_{sc}$	Scaling factor
Overlaid	$w$	Width (mm)
	$h$	Height (mm)
	$x_c$	Centre X (mm)
	$y_c$	Centre Y (mm)
Vertical expansion	$w_{min}$	Minimum width (mm)
	$x_c$	Centre X (mm)
	$F_{sc}$	Scaling factor
Horizontal expansion	$h_{min}$	Minimum height (mm)
	$y_c$	Centre Y (mm)
	$F_{sc}$	Scaling factor
Triangular	$w_{tr}$	Base width (mm)
	$h_{tr}$	Height (mm)
	$x_{tr}$	Base centre X (mm)
	$y_{tr}$	Base centre Y (mm)

### 1. Centre to outside

An initial area is tracked in the centre. In this case it will be a square. This dimensions of this area is then expanded linearly for each subsequent pass.

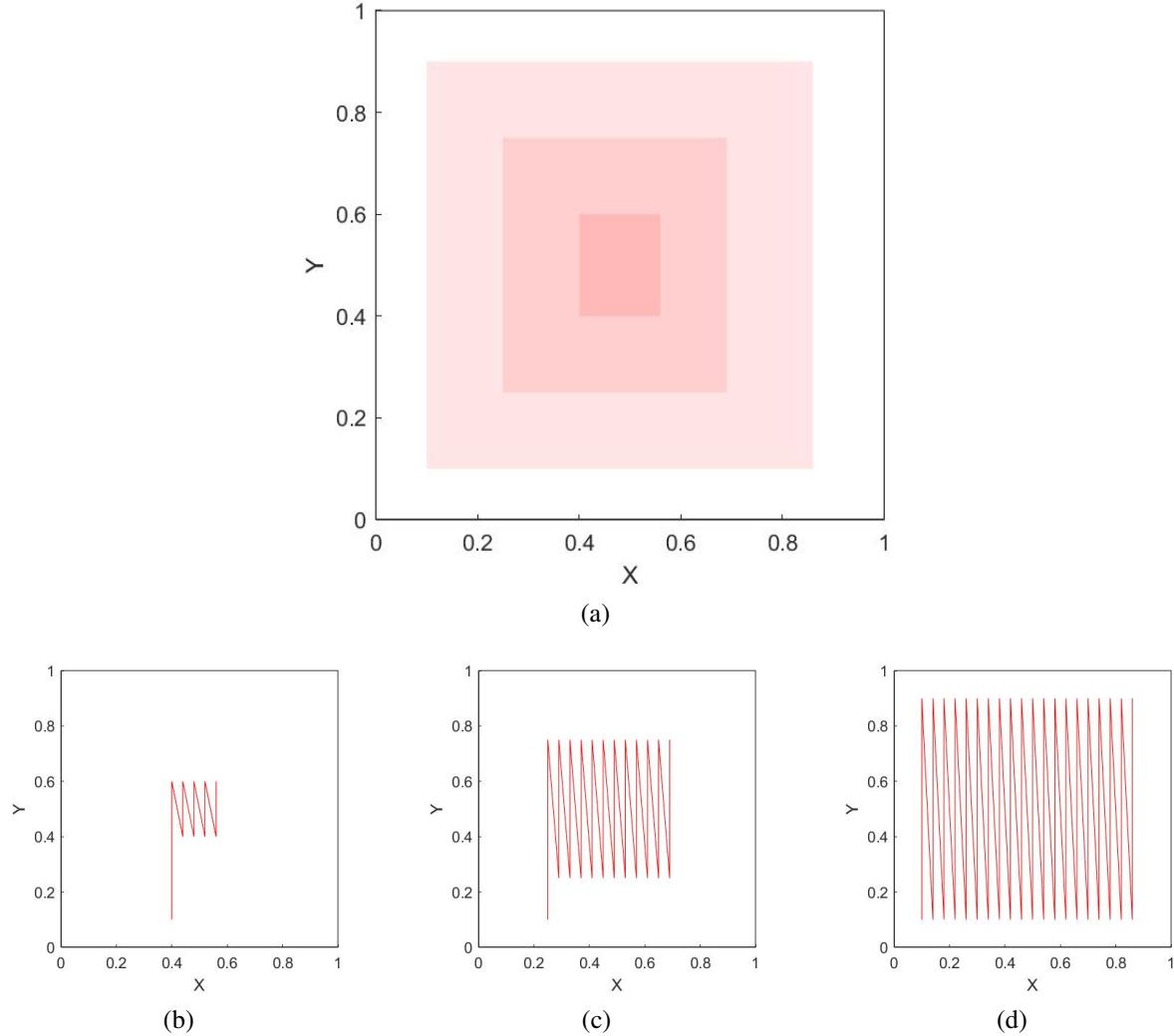


Figure 58: (a) Tracking density for the centre-to-outside tracking strategy, and (b), (c), and (d) showing the tracking pattern for each pass. Each plot is normalised.

Characteristic equations:

$$w = w_{min}(F_{scale} \times \frac{i}{N_{passes}} + 1) \quad (21)$$

$$h = h_{min}(F_{scale} \times \frac{i}{N_{passes}} + 1) \quad (22)$$

$$x = x_c \quad (23)$$

$$y = y_c \quad (24)$$

## 2. Diagonal

Two points are selected at equal distances but from the opposite edges of the sheet. These are the centre points of the first and last rectangular tracking patterns in the strategy, with a defined number of pattern linearly spaced in between.

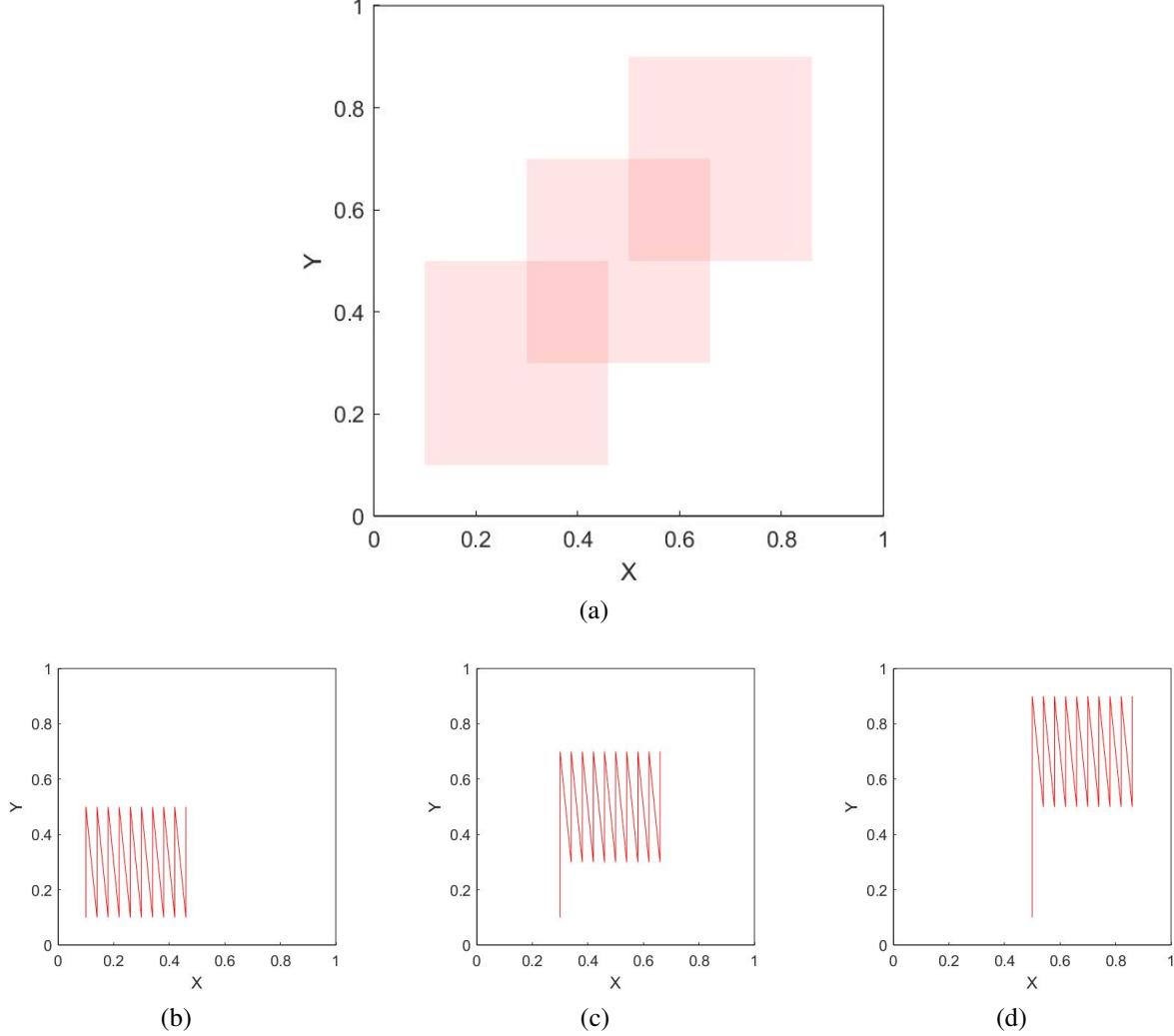


Figure 59: (a) Tracking density for the diagonal tracking strategy, and (b), (c), and (d) showing the tracking pattern for each pass. Each plot is normalised.

Characteristic equations:

$$w = w_{track} \quad (25)$$

$$h = h_{track} \quad (26)$$

$$x = ((1 - 2x_{st}) \times \frac{i - 1}{N_{passes}}) + x_{st} \quad (27)$$

$$y = ((1 - 2y_{st}) \times \frac{i - 1}{N_{passes}}) + y_{st} \quad (28)$$

### 3. Vertical expansion

A position is selected on the x axis that is the centre point of the initial tracking pattern. This has the smallest width, with the height defined as a constant. The consequent tracking pattern have the same central point but have a width determined by a scaling factor.

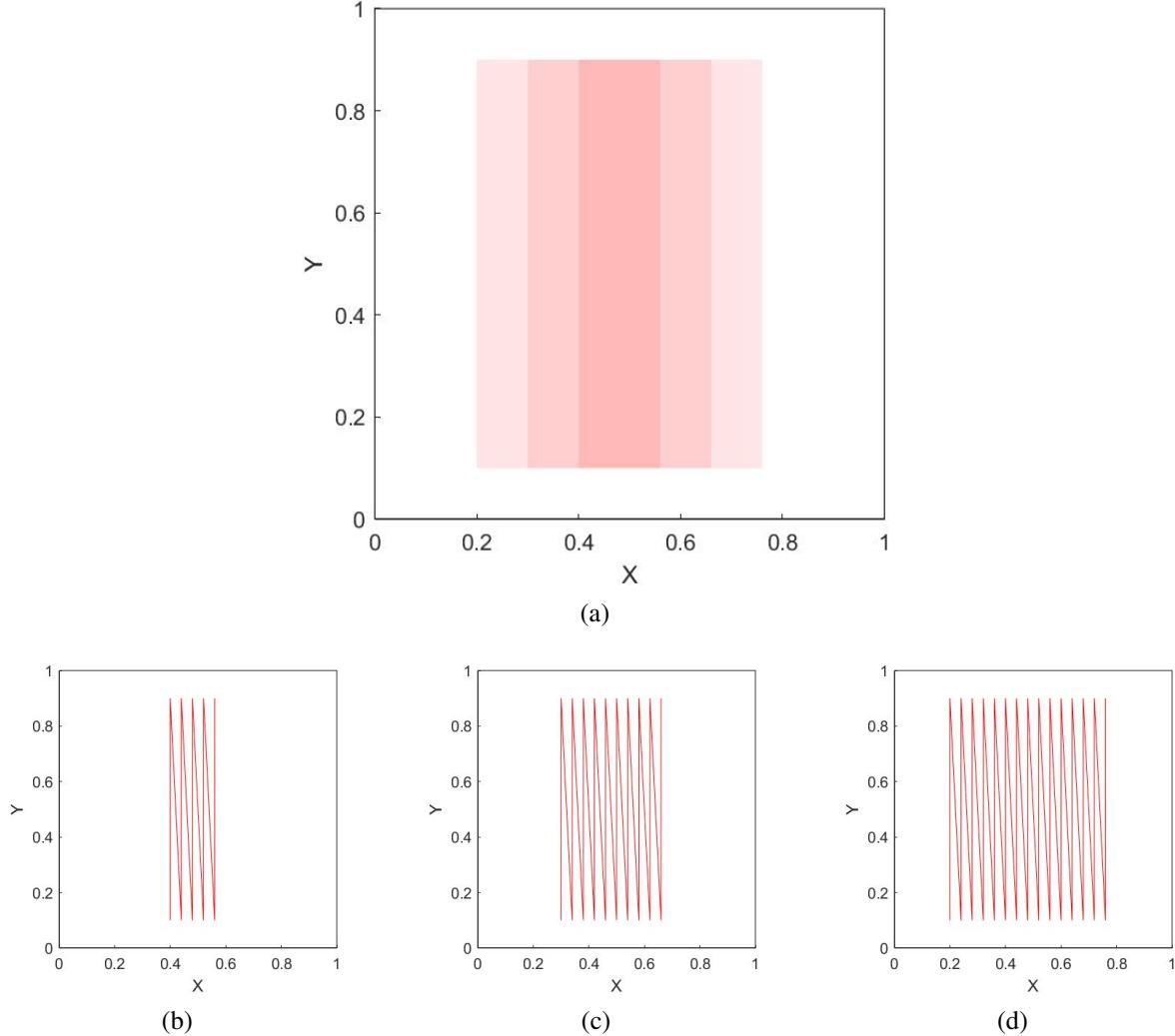


Figure 60: (a) Tracking density for the vertical expansion tracking strategy, and (b), (c), and (d) showing the tracking pattern for each pass. Each plot is normalised.

Characteristic equations:

$$w = w_{min}((F_{scale} \times \frac{i-1}{N_{passes}}) + 1) \quad (29)$$

$$h = 1 - 2\delta \quad (30)$$

$$x = x_c \quad (31)$$

$$y = l_{sheet}/2 \quad (32)$$

#### 4. Horizontal expansion

A position is selected on the y axis that is the centre point of the initial tracking pattern. This has the smallest height, with the width defined as a constant. The consequent tracking pattern have the same central point but have a height determined by a scaling factor.

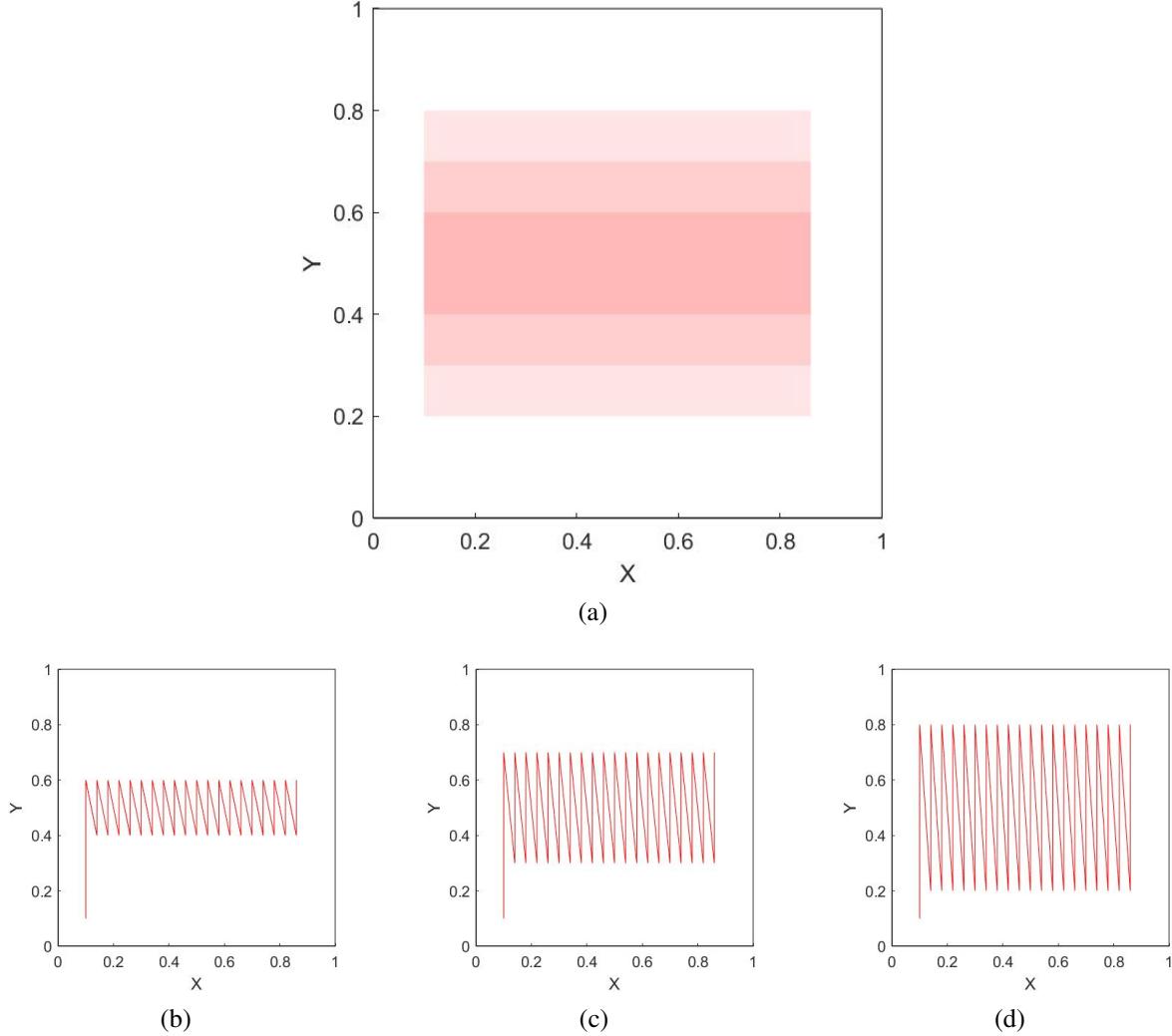


Figure 61: (a) Tracking density for the horizontal expansion tracking strategy, and (b), (c), and (d) showing the tracking pattern for each pass. Each plot is normalised.

Characteristic equations:

$$w = 1 - 2\delta \quad (33)$$

$$h = h_{min}((F_{scale} \times \frac{i-1}{N_{passes}}) + 1) \quad (34)$$

$$x = l_{sheet}/2 \quad (35)$$

$$y = y_c \quad (36)$$

## 5. Overlayed

Each tracking pattern has the same central point, height and width.

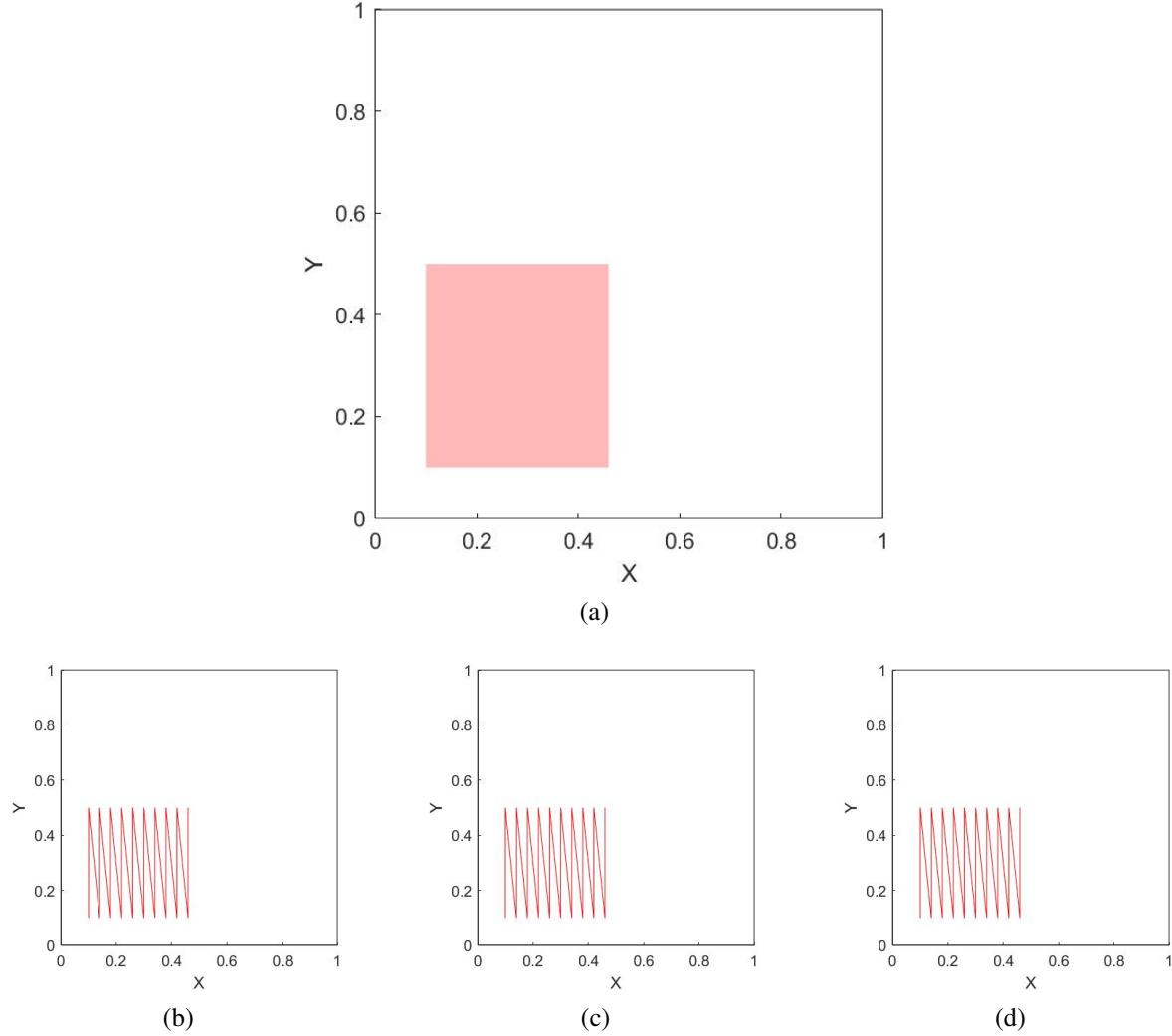


Figure 62: (a) Tracking density for the overlayed tracking strategy, and (b), (c), and (d) showing the tracking pattern for each pass. Each plot is normalised.

Characteristic equations:

$$w = w_{track} \quad (37)$$

$$h = h_{track} \quad (38)$$

$$x = x_c \quad (39)$$

$$y = y_c \quad (40)$$

## 6. Vertical tracking

A special case of horizontal expansion, but the tracking pattern only expands in one direction, not two.

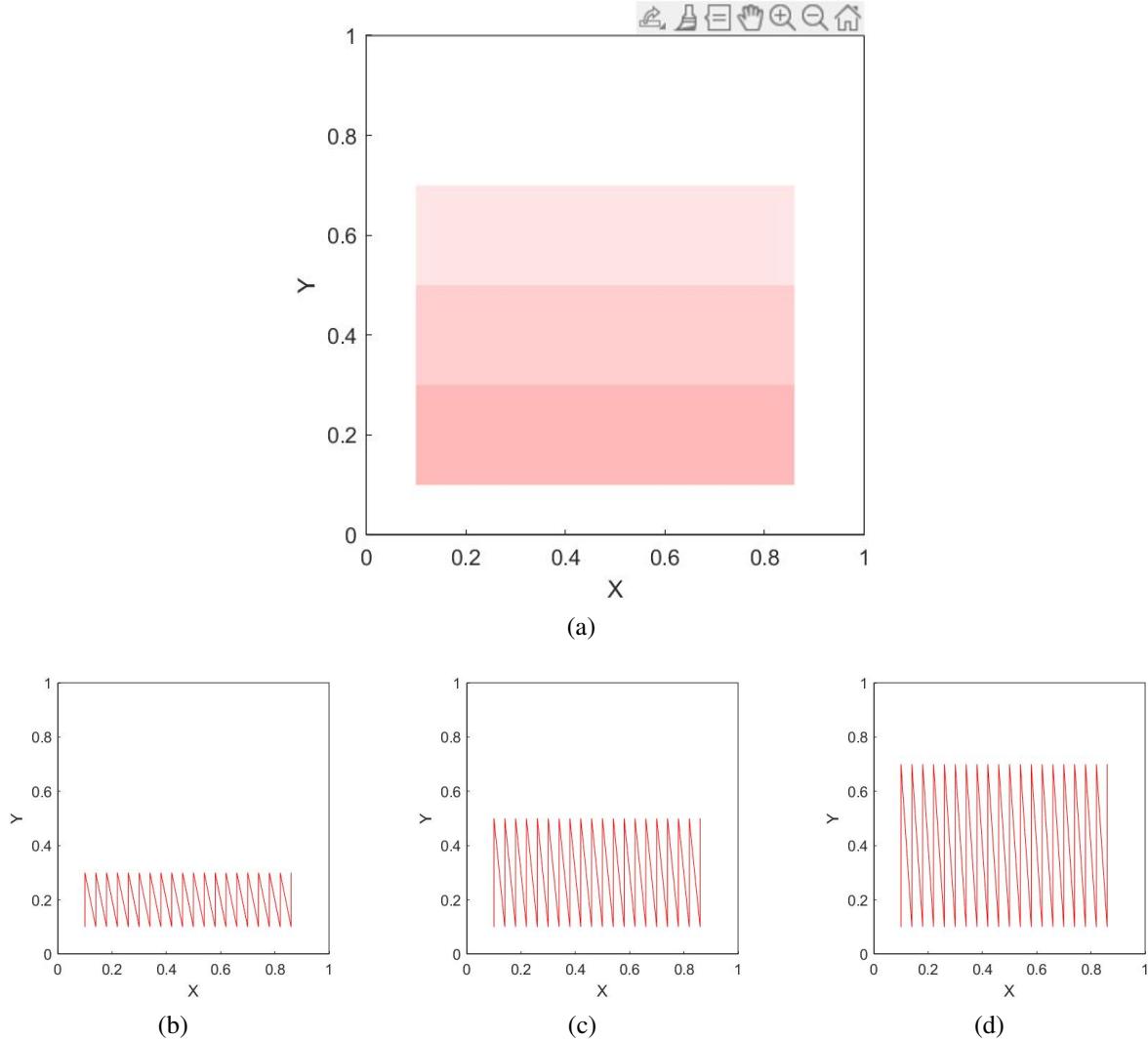


Figure 63: (a) Tracking density for the vertical tracking strategy, and (b), (c), and (d) showing the tracking pattern for each pass. Each plot is normalised.

Characteristic equations:

$$w = 1 - 2\delta; \quad (41)$$

$$h = h_{min} + (h_{max} - h_{min}) \times \frac{i - 1}{N_{passes}} \quad (42)$$

$$x = l_{sheet}/2 \quad (43)$$

$$y = \frac{h}{2} + \delta \quad (44)$$

## 7. Horizontal tracking

A special case of vertical expansion, but the tracking pattern only expands in one direction, not two.

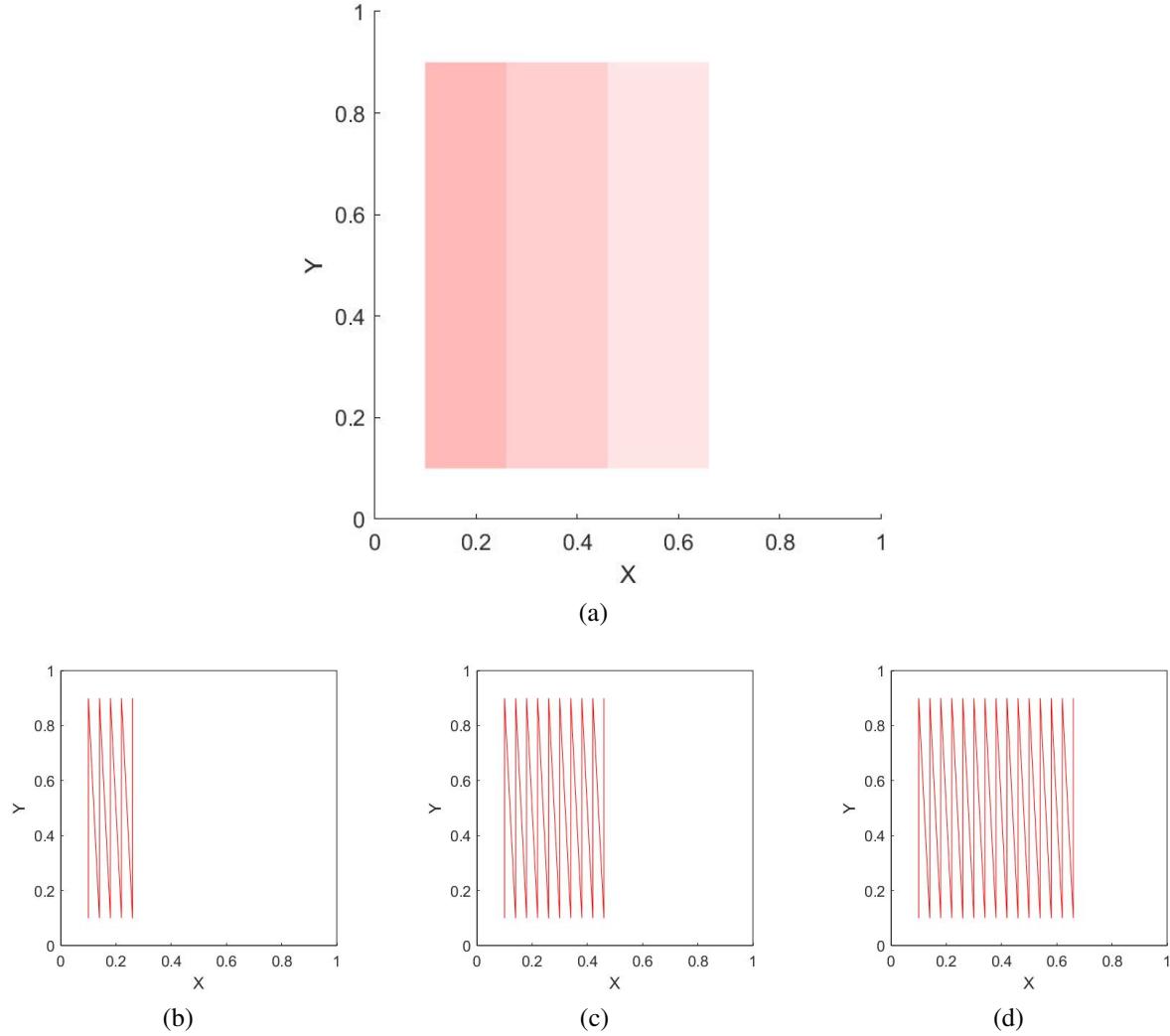


Figure 64: (a) Tracking density for the horizontal tracking strategy, and (b), (c), and (d) showing the tracking pattern for each pass. Each plot is normalised.

Characteristic equations:

$$w = w_{min} + (w_{max} - w_{min}) \times \frac{i - 1}{N_{passes}} \quad (45)$$

$$h = 1 - 2\delta \quad (46)$$

$$x = \frac{h}{2} + \delta \quad (47)$$

$$y = l_{sheet}/2 \quad (48)$$

### 8. Triangular tracking

The tracking patterns accumulate to form a triangular shape. The first tracking pattern is as wide as the base with the smallest height, with the last equal to the height of the triangle with the smallest width.

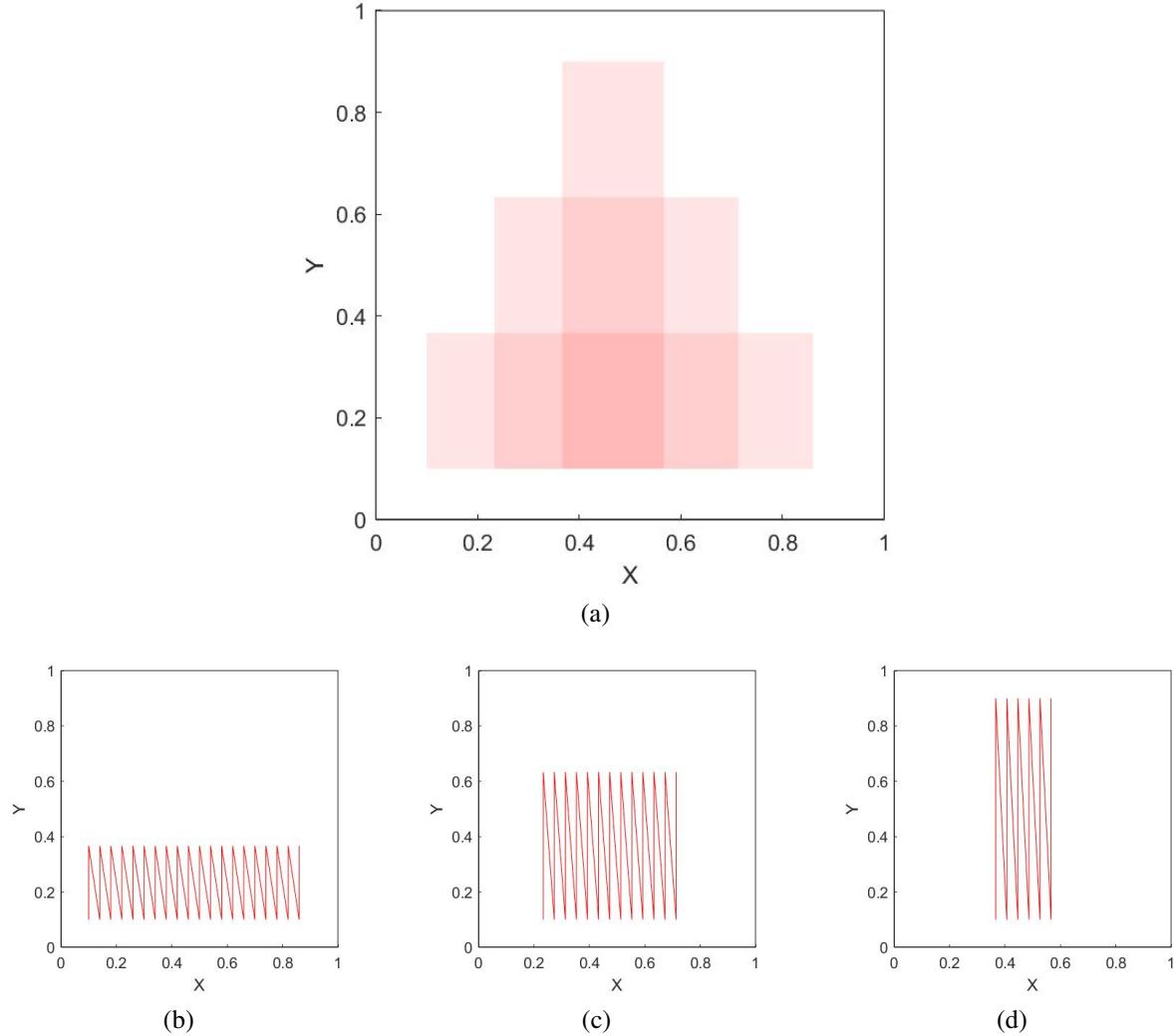


Figure 65: (a) Tracking density for the triangular tracking strategy, and (b), (c), and (d) showing the tracking pattern for each pass. Each plot is normalised.

Characteristic equations:

$$w = w_{tr} \left(1 - \frac{i-1}{N_{passes}}\right); \quad (49)$$

$$h = h_{tr} \frac{i}{N_{passes}} \quad (50)$$

$$x = x_c \quad (51)$$

$$y = y_c + \frac{h_{tr}}{2} \quad (52)$$

## B User-interface documentation

### B.1 Process parameter documentation

Table 16: Formal documentation for process parameters used in UI. All distances in mm unless stated otherwise.

Family	Ref.	Name	Description	User-defined?	Affects G-code?
Sheet parameters	1.1	Material	The material of the sheet. In this instance, only uni-material sheets are considered.	✓	
	1.2	Sheet size	The length of the sheet. Only square geometries are allowed.	✓	✓
	1.3	Sheet thickness	Average thickness of the sheet pre-deformation.	✓	
Machine parameters	2.1	Roll gap	The initial distance between the top tool and lower anvil without force applied.	✓	✓
	2.2	Feed rate	The length of sheet that passes through the wheel per second.	✓	✓
	2.3	Blow radius	The radius of the lower anvil. This EW comes with 5 variations.	✓	
	2.4	Crown radius		✓	
	2.5	Arc speed	The linear velocity of the end effectors during a 'pivot' operation of the sheet. Measured in mm/s.	✓	
	2.6	End effector boundary	The distance from the edge of the sheet that the vacuum pads of the end effector can be effectively attached. Default is 30mm.	✓	✓
Tool path parameters	3.1	Tracking spacing	The distance between each tracking motion made by the tool on the sheet.	✓	✓
	3.2	Drive-through distance	The distance from the edge that the sheet is fed through the wheel initially. This is so the wheel is in full contact with the sheet during tracking.	✓	✓
	3.3	Tracking style	Style of the tracking pattern. Default is 'N'.	✓	

Table 16: Formal documentation for process parameters used in UI. All distances in mm unless stated otherwise.

Family	Ref.	Name	Description	User-defined?	Affects G-code?
	3.4	Fixed wheeling boundary	Distance from the edge of the sheet that wheeling is not allowed. This marks a internal square on the sheet where tracking is allowed.	✓	✓
Tracking pattern parameters	4.1	Tracking pattern width	The width of the rectangular area.	✓	✓
	4.2	Tracking pattern height	The height of the rectangular area.	✓	✓
	4.3	Tracking pattern central position	The position of the centre of the rectangle relative to the bottom left corner of the sheet in the x and y axis.	✓	✓
Tracking strategy parameters	5.1	Number of passes	Number of individual tracking patterns making up one strategy.	✓	✓
	5.2	Centre to outside			
	5.2.1	Minimum width	Minimum width of the tracking pattern area.	✓	✓
	5.2.2	Minimum height	Minimum height of the tracking pattern area.	✓	✓
	5.2.3	Central position	Central position of the tracking pattern area in the x and y axes.	✓	✓
	5.2.5	Scaling factor	Multiplication factor applied to the tracking pattern pass with the smallest size to determine the tracking pattern of the final wheeled area.	✓	✓
	5.3	Overlaid			
	5.3.1	Width	Width of the tracking pattern area.	✓	✓
	5.3.2	Height	Height of the tracking pattern area.	✓	✓
	5.3.3	Central position	Central position of the tracking pattern area in the x and y axes.	✓	✓
	5.4	Vertical expansion			
	5.4.1	Minimum width	Minimum width of the tracking pattern area.	✓	✓
	5.4.2	Central x position	Central x axis position for all tracking patterns.	✓	✓

Table 16: Formal documentation for process parameters used in UI. All distances in mm unless stated otherwise.

Family	Ref.	Name	Description	User-defined?	Affects G-code?
	5.4.3	Scaling factor	Multiplication factor applied to the tracking pattern pass with the smallest size to determine the tracking pattern of the final wheeled area.	✓	✓
	5.5	Horizontal expansion			
	5.5.1	Minimum height	Minimum height of the tracking pattern area.	✓	✓
	5.5.2	Central y position	Central y axis position for all tracking patterns.	✓	✓
	5.5.3	Scaling factor	Multiplication factor applied to the tracking pattern pass with the smallest size to determine the tracking pattern of the final wheeled area.	✓	✓
	5.6	Triangular			
	5.6.1	Base width	Width of the base of the final triangular tracked area.	✓	✓
	5.6.2	Base height	Height of the final tracked triangular area.	✓	✓
	5.6.3	Central base position	Central position of the triangle base in the x and y axes.	✓	✓
	5.7	Track vertical			
	5.7.1	Minimum height	Minimum height of the tracking pattern area.	✓	✓
	5.7.2	Scaling factor	Multiplication factor applied to the tracking pattern pass with the smallest size to determine the tracking pattern of the final wheeled area.	✓	✓
	5.8	Track Horizontal			
	5.8.1	Minimum width	Minimum width of the tracking pattern area.	✓	✓
	5.8.2	Scaling factor	Multiplication factor applied to the tracking pattern pass with the smallest size to determine the tracking pattern of the final wheeled area.	✓	✓
	5.9	Diagonal			

Table 16: Formal documentation for process parameters used in UI. All distances in mm unless stated otherwise.

Family	Ref.	Name	Description	User-defined?	Affects G-code?
	5.9.1	Width	Width of the tracking pattern area.	✓	✓
	5.9.2	Height	Height of the tracking pattern area.	✓	✓
	5.9.3	Starting central position	Starting central position of the tracking pattern area in the x and y axes.	✓	✓
Processing parameters	6.1	Continuous or discrete	Boolean describing whether output will be discretised.	✓	✓
	6.2	Resolution	Resolution of discretisation performed on tracking pattern.	✓	✓

## B.2 Tracking strategy structure documentation

Table 17: Formal documentation for tracking strategy structure used in UI.

Ref.	Name	Name in structure	Type	Description
1	Corner positions	AreaCorners	5x2 double	Coordinates for the corner positions of the tracking pattern area
2	Tracking density matrix	DiscretePath	nx2 double	Matrix of binary values representing the wheeled section of the sheet
3	Node structure	node	1xn struct	Structure containing data for each node
3.1	Tool-specific node data	datum	1x1 struct	Structure containing data for actions and positions for the tool at each node. These are relative to the axes defined in Section 3.1
3.1.1	Global coordinates	global	1x1 struct	Global coordinates of the tool relative to the axes defined in Section 3.1
3.1.1.1	Global angle	angle	int	Absolute angle of tool relative to the axes defined in Section 3.1
3.1.1.2	Global position	position	3x1 double	Absolute position of tool relative to the axes defined in Section 3.1
3.1.2	Kinematic actions	global	1xn struct	Actions outlined in 3.3 such as translation and rotation operations
3.1.2.1	Rotation data	rotation	1xn cell	Rotation matrix and angle describing the operation

Table 17: Formal documentation for tracking strategy structure used in UI.

Ref.	Name	Name in structure	Type	Description
3.1.2.2	Translation data	translation	1xn cell	Translation vector and distance describing the operation
3.1.3	Homogeneous transformation matrix	global	4x4 double	Homogeneous transformation matrix calculated using theory presented in 3.3
3.2	Activity of vacuums	enabled	1x2 int	Array detailing which vacuum bellows are 'on' or 'off' at the end of each node via a boolean.
3.3	End-effector-specific node data	effectors	1x1 struct	Structure containing data for actions and positions for the tool at each node. These are relative to the axes defined in Section 3.2
3.3.1	Effector transfer	transfer	boolean	Boolean describing whether an end-effector transfer occurs on the current node
3.3.2	Effector positions	position	2x2 cell	Start and end position of both end effectors relative to the tool position
3.3.3	Effector actions	action	nx2 cell	A cellular array representative of the chronology of actions for each effector. These comprise of 'pivot', 'translation', and 'transfer' actions as outlined in ...
3.3.4	Machine axes	axes	1xn cell	A cellular array representing the machine axes at the start and end of each action.
3.4	Type of tool	tooltypes	1x1 char	Letter corresponding to type of tool used.
4	Setup parameters	setup	1x1 struct	Parameters required to setup machine for this test case. Descriptions of these can be found in Table 16
4.1	Sheet parameters	Sheet	1x1 struct	See Table 16 ref. 1
4.2	Machine parameters	Machine	1x1 struct	See Table 16 ref. 2
4.3	Tool path parameters	Path	1x1 struct	See Table 16 ref. 1
4.4	Fixed parameters	Fixed	1x1 struct	See Table 16 ref. ??
4.5	Processing parameters	Process	1x1 struct	See Table 16 ref. 6
4.6	Strategy parameters	Strategy	1x1 struct	See Table 16 ref. 5
5	G-code	code	nx2 cell	Cell array of G-code commands required to perform operation

## B.3 User-interface Quickstart Guide

### B.3.1 Defining the process parameters

There is a tab that allows users to specify the process parameters. For descriptions of these parameters see Table 16. All distances are given in mm. They are divided into categories, which can be seen in Figure 66:

- Sheet Parameters - Describe the characteristics of the sheet.
- Machine Parameters - Describe the characteristics of the machine.
- Processing Parameters - Used in post-processing of the data.
- Tool path parameters - Characteristics defined to constrain the tool path.

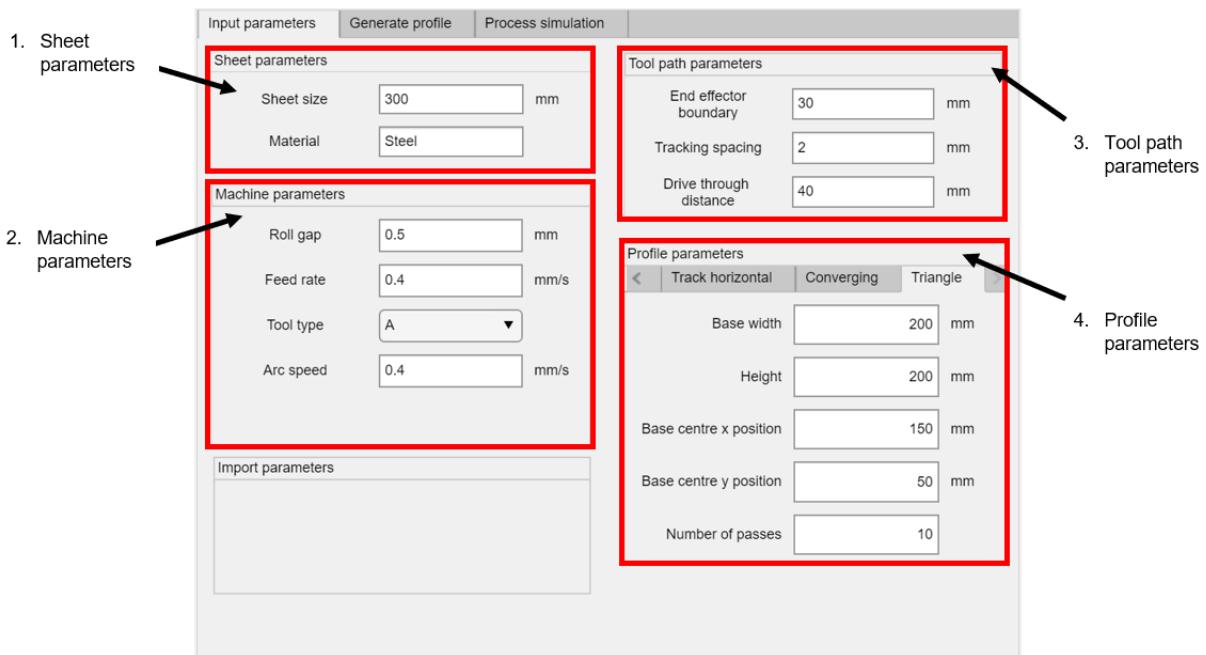


Figure 66: Define the process parameters

### B.3.2 Generate the tracking pattern

On the second tab 'Generate profile', the tracking pattern defined by the parameters you have just input can be visualised following these steps:

a. Select the strategy:

In the 'Select strategy' panel, you must define the strategy you want to create from the parameters input on the previous panel.

b. Define strategy parameters:

Within the 'Strategy parameters' tab, navigate to the selected strategy. This will list all parameters defining this strategy as documented in Table 16. Change these values as desired. Be aware a high number of passes will result in poor computational performance. Parameters do not have limits imposed, therefore ensure pattern does not exceed size of sheet.

c. Generate tracking strategy:

Next generate the pattern by clicking 'Create profile'. To clear the pattern click 'Clear profile'. These steps are displayed below in Figure 67.

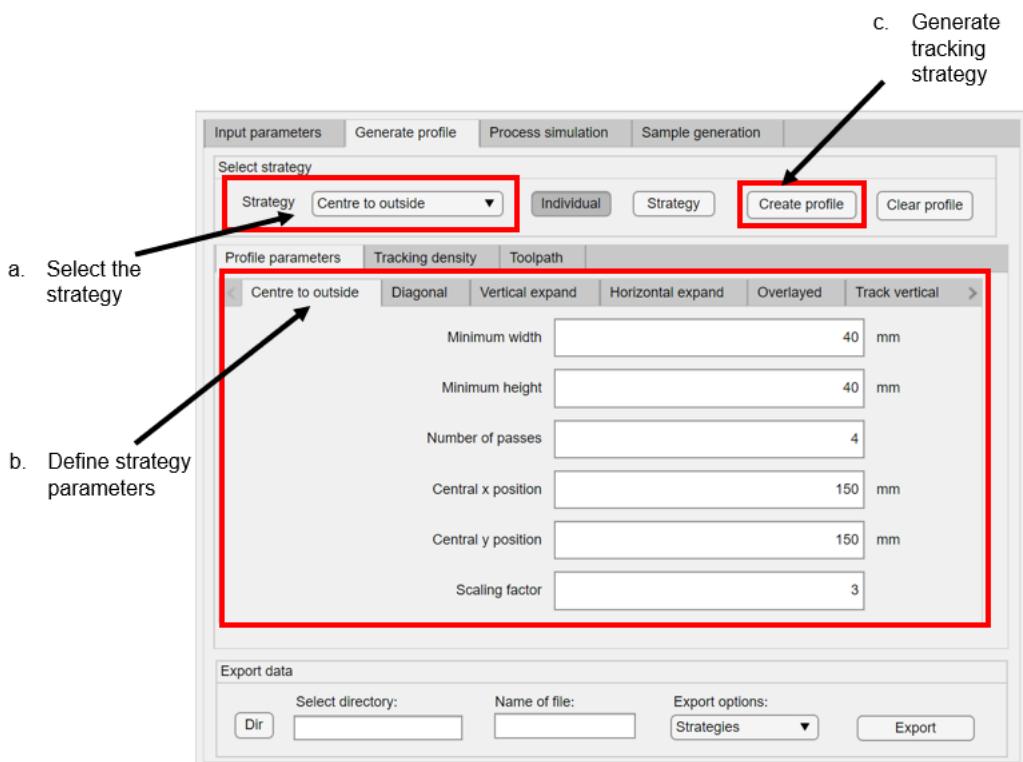


Figure 67: Selection of strategy, defining strategy parameters and generating tracking strategy

d. View tracking density:

The tracking density can be viewed in the tracking density tab below (Figure 68). This displays the area of each individual tracking pattern laying on top one another. A more intense red indicates a higher tracking density. The axes of the plot are normalised so the minimum and maximum are 0 and 1 respectively for both the x and y axes. This tool can be used to check the correct areas of the sheet

are being processed. Check boxes to the right allow the axes and sheet border to be toggled ‘on’ or ‘off’.

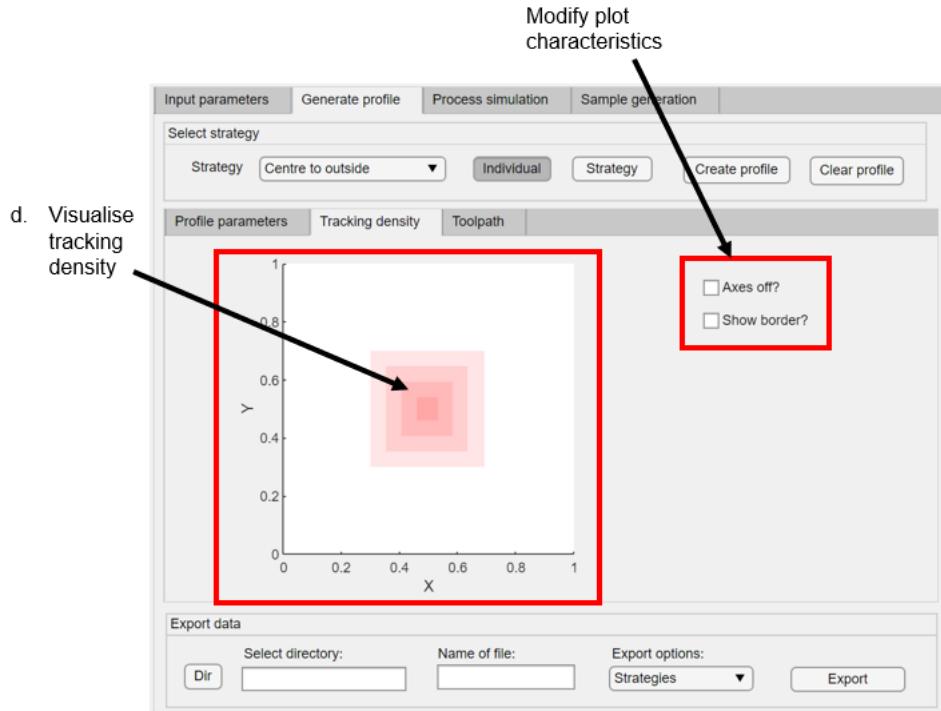


Figure 68: View the tracking density

e. View tracking patterns:

The individual tracking patterns can be viewed on the ‘Toolpath’ tab, shown in Figure 69. First, the pass number needs to be selected. Then, press ‘Generate’ to visualise the tracking pattern to the left. Toggles to the right allow you to customise the plot, including hiding nodes, the sheet border, and axes. To clear the plot press ‘Clear’.

Another feature allows the user to view all tracking patterns in an animation. Once the ‘Start animation’ button is pressed, the axes on the left cycles through each strategy chronologically. To stop the animation press ‘Stop’.

f. Export data:

In the export data panel, it allows you to export the tracking pattern data to a specified location in the directory. To specify the directory, either paste the path into the ‘Select directory’ text box, or press the ‘dir’ button, which opens up the directory UI. Enter the name you want to give the file into the ‘Name of file’ text box. There are many different options for the data that can be exported from the dropdown ‘Export options’:

- Strategies - .xml file of strategy setup data.
- Tracking pattern plot - .jpg file of selected tracking pattern.
- Tracking pattern plots - .jpg files of all tracking patterns for selected strategy.
- Tracking density plot - .jpg file of current tracking density plot.
- Tracking density - .csv file of tracking density, in format described in Section ??.
- G-code - .txt file of all G-code commands required to execute selected strategy.

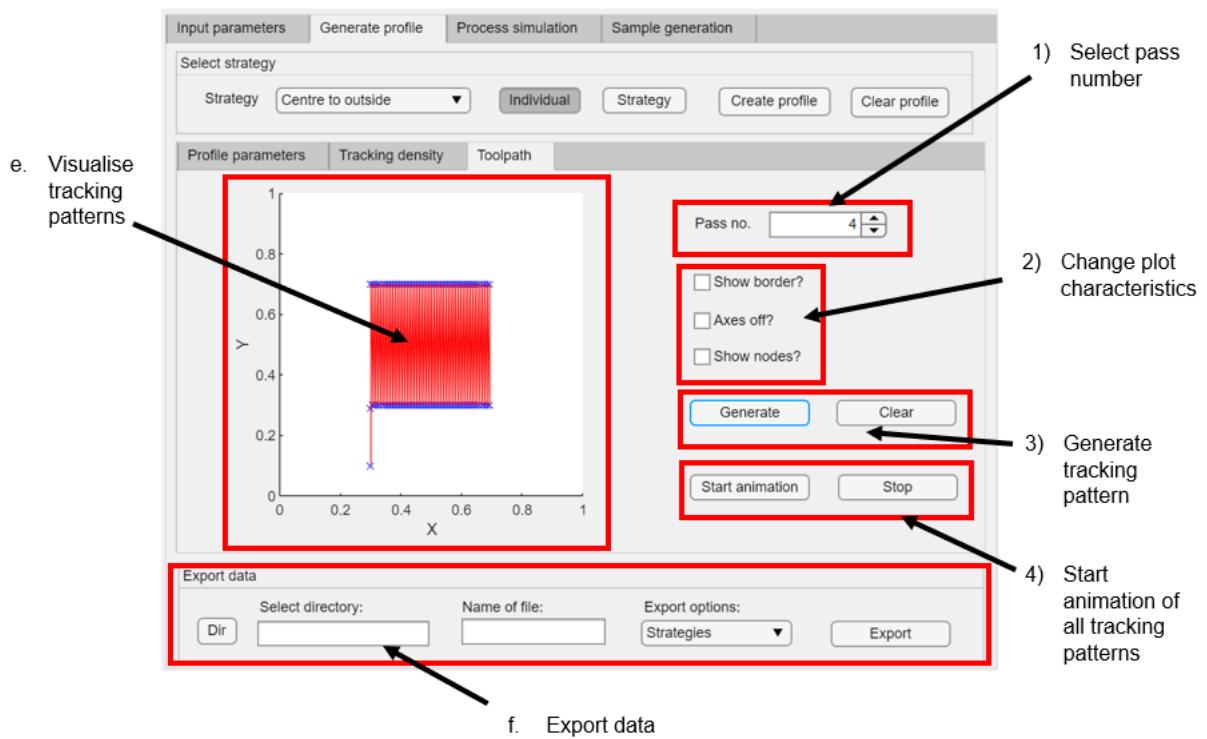


Figure 69: View the tracking pattern and export data

### B.3.3 Simulate the machine process

The tool in the 'Process simulation' tab allows you to simulate the machine process from the strategy you have just created. It is useful for debugging the issues with the G-code implemented on the machine, and testing if certain manufacturing strategies are feasible.

#### a. Starting the simulation:

To start the simulation first you need to select the 'Pass number' that you want to simulate from the numerical dial. Then press 'Start simulation' to start the simulation in the axes below.

#### b. During the simulation

The simulation has a timestep corresponding to each individual action defined in Section 5.1. You can press the 'Pause/Play' button to pause the animation on that action, and press that button again to resume. To stop the animation entirely press the 'Stop' button. The simulation has many features to allow users to analyse the process. These are described in Figure 70.

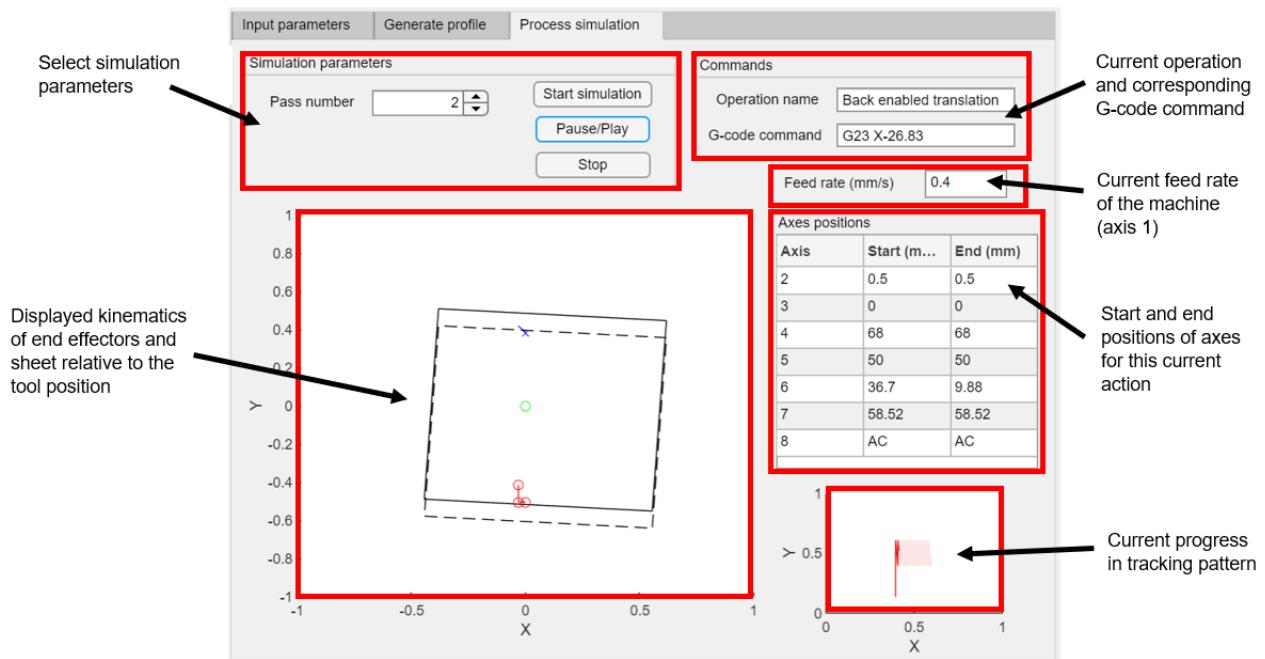


Figure 70: Features of the simulation

### B.3.4 Generate samples for experimentation

The ‘Sample generation’ tab allows users to create test strategies using statistical sampling techniques to avoid bias.

#### a. Select general sampling parameters:

First, the parameters defining the sampling method needs to be defined. Figure 71 shows the user inputs for these parameters can be found in the ‘Sampling parameters’ panel under the ‘General’ tab.

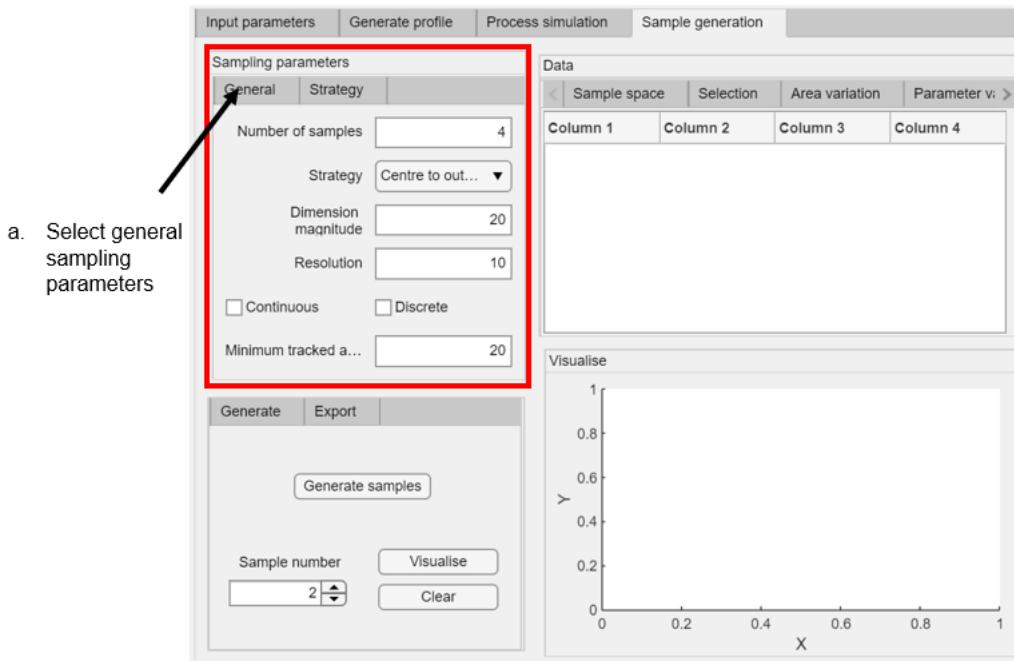


Figure 71: Select general sampling parameters

b. Select strategy sampling parameters:

Next, the maximum and minimum values for defining the sample space can be defined. Click the ‘Strategy’ tab and navigate to the strategy you selected in the previous step. Define these parameters here.

c. Generate samples:

Press the ‘Generate samples’ button to form the sample space and select samples according to theory presented in Section 4.7. This may take a few seconds.

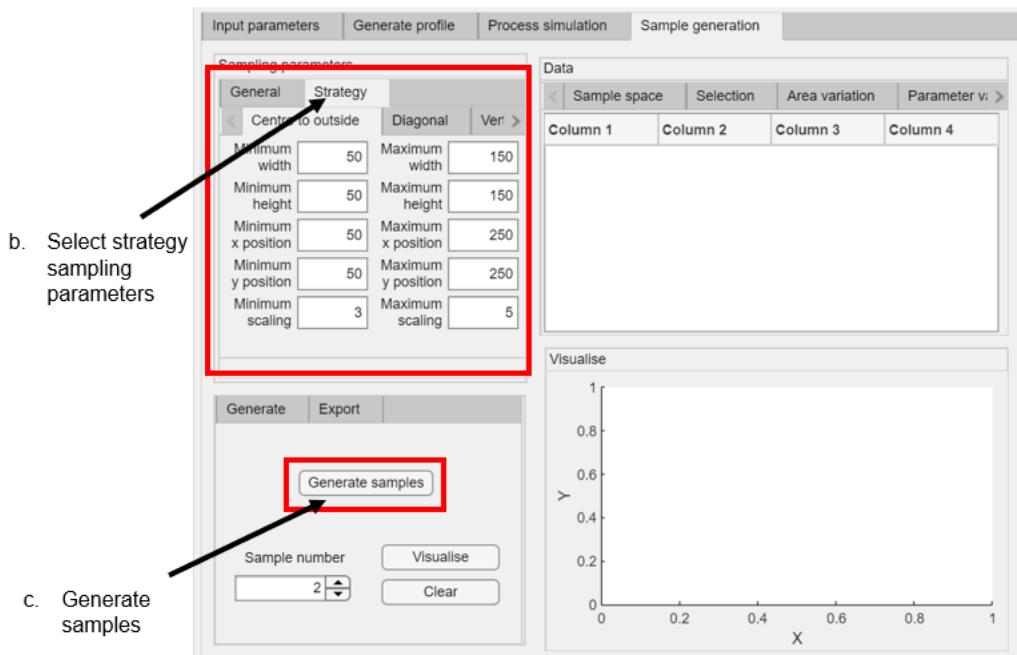


Figure 72: Select strategy sampling parameters and generate samples

d. Explore the sample space:

In the ‘Data’ panel, click the ‘Sample space’ tab to see a tabulated form of the entire sample space. The columns represent each strategy parameter, and each row in a new sample.

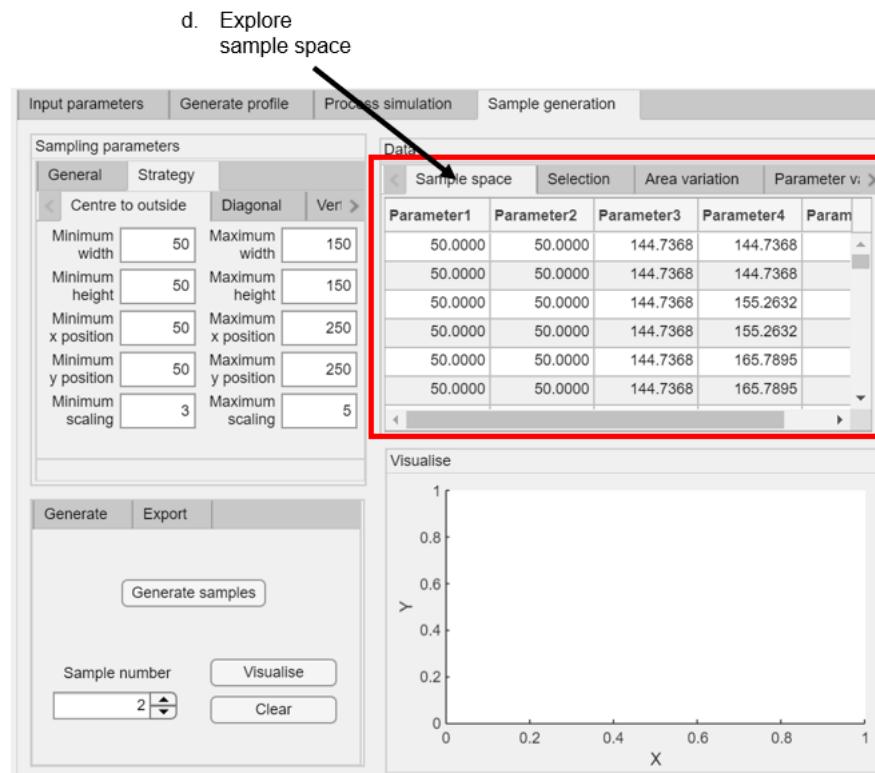


Figure 73: Explore the sample space

## e. Explore selected samples:

Click the ‘Selection’ tab. This will show the samples that have been selected using the sampling algorithm. Check that the parameters are as desired.

## f. Visualise strategies:

For further verification of the selected strategies, they can be visualised. Under the ‘Generate’ tab, select the sample number you want to visualise using the ‘Sample number’ numerical dial. Then click the ‘Visualise’ button to show the tracking density plot for that strategy in the axes on the right. To clear this axes press the ‘Clear’ button.

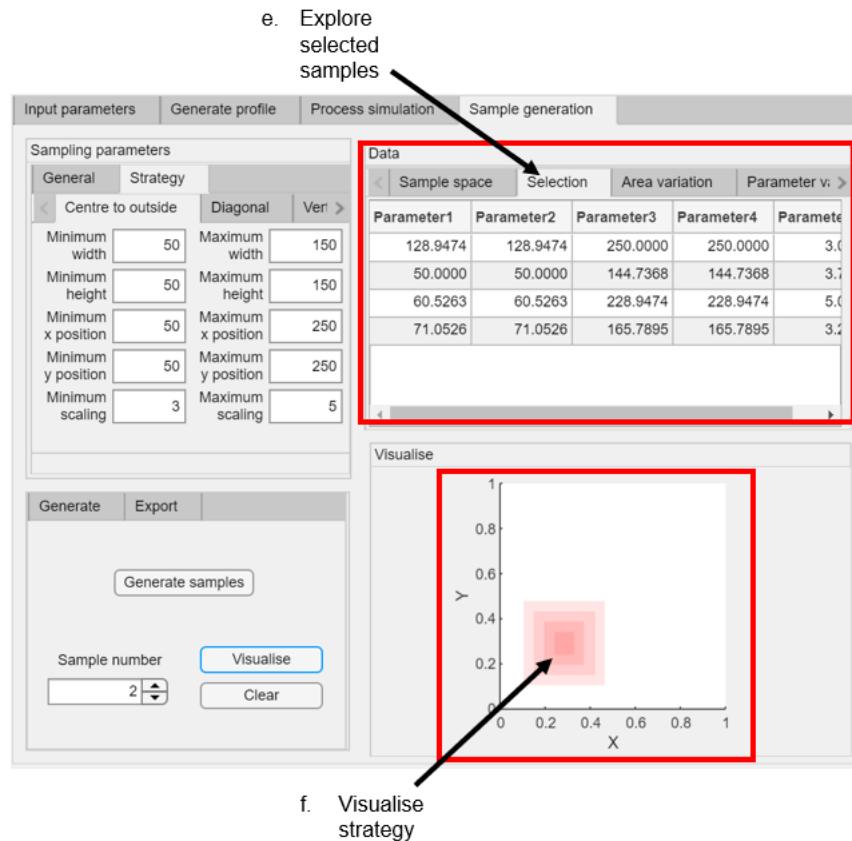


Figure 74: Explore selected samples and visualise strategies

## g. Analyse area variation:

In the ‘Data’ panel, click the ‘Area variation’ tab. Click the ‘Generate’ button. The axes in this tab shows the variation of the size of sample space with minimum area of tracking. This can be used to identify bias introduced by the minimum area defined.

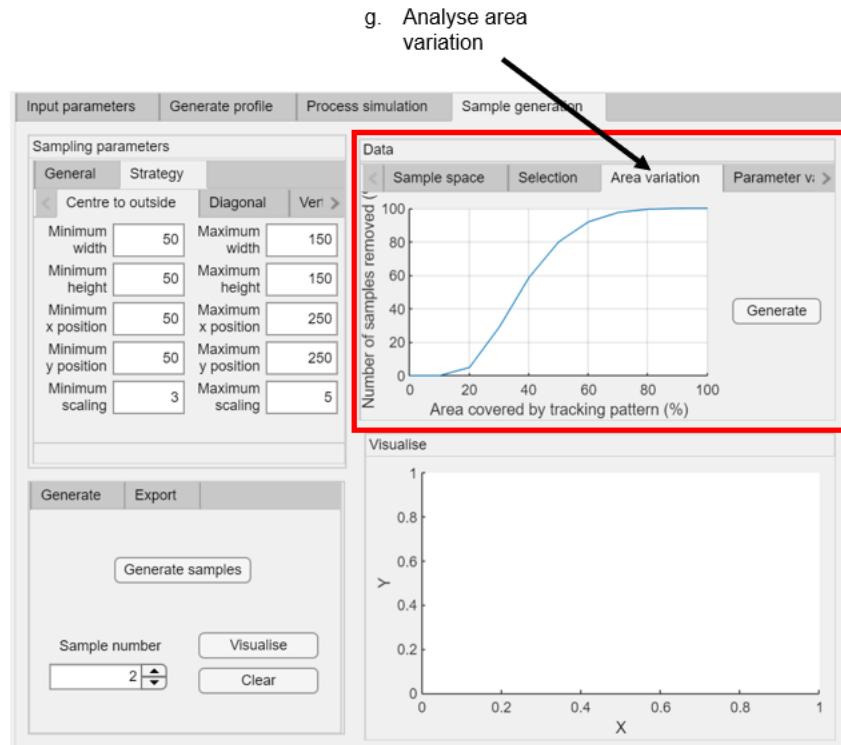


Figure 75: Analyse area variation

#### h. Analyse parameter variation:

In the ‘Data’ panel, click the ‘Parameter variation’ tab. Select the parameter number from the dropdown menu corresponding to the strategy parameter you would like to display. These are in the same order for the chosen strategy. Click the ‘Display’ button. The axes in this tab display box plots that illustrate the variation in the distribution of the parameter with minimum area. This can be used to identify bias introduced by the minimum area defined.

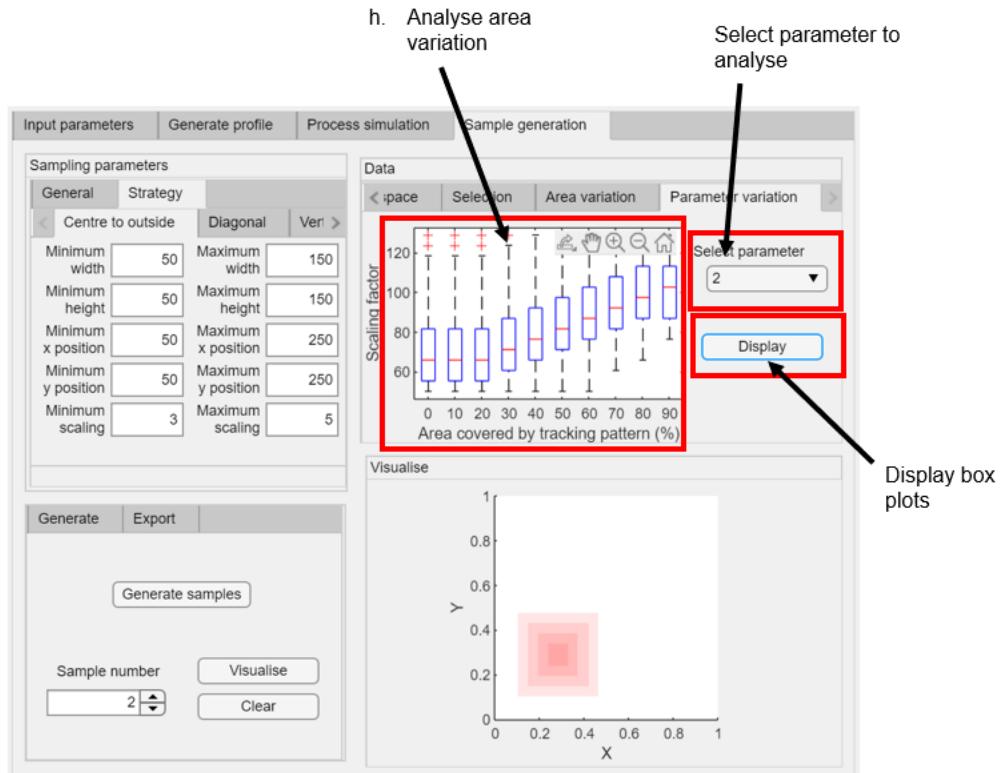


Figure 76: Analyse parameter variation

## C Model documentation

### C.1 Hyperparameters

Table 18: Description of model hyperparameters and base values for the tuning process

Hyperparameter	Definition	Base value		Tuned?
		Layer 1	Layer 2	
Pool size	Size of window to select maximum value	2	2	
Strides	Distance window moves each pooling step	2	2	
Number of convolution layers	Number of time convolution and pooling operations repeat until the end	3	3	
Layer-wise activation function	Activation function to increase non-linearity within the dataset during convolution	Rectifier	Rectifier	
Output activation function	Activation function that renders output	Softmax	Linear	
Loss function	Function that is used to tune weights of neural network	Categorical crossentropy	Mean absolute error	
Number of hidden layers	Layers between the input and output layers made up of neurons	1	1	✓
Number of neurons per layer	Number of neurons per hidden layers	128	128	✓
Optimiser	Algorithm used to update neural network weightings	Adam	Adam	
Number of epochs	Number of times algorithm iterates through entire dataset	100	100	✓

## C.2 Description of preliminary layer

The preliminary layer requires a three-dimensional prediction of the tracking density given a three-dimensional input describing the sheet geometry. There are no open-source models that can be applied to this problem. Instead, the data needs to be pre-processed to a form that can be applied to open-source packages. A concept is proposed that reduces the problem from a multi-output to a singular-output problem. The tracking density is discretised to a set of nodes, and classification of the tracking density of each node is made given the geometrical characteristics of the n-neighbouring nodes. This classification problem can be solved using two approaches:

1. Supervised model

A set of features are created for the n-nearest nodes to the selected node. These include the euclidean distance between each node and the selected node, and the geometrical descriptor at that node. The label indicates the tracking density of the selected node.

2. CNN

For each node, a surrounding  $n \times n$  grid of values that describe the geometry is converted into a grayscale image format. Each image has a label of the tracking density of the central node. This set of images used to train a CNN model.

This process makes several assumptions. It is reliant that the number of surrounding nodes chosen to the location of wheel can describe the deformation caused in its entirety. It also assumes that the geometrical descriptor used can describe local deformation accurately. The choice of the number of surrounding nodes will require investigation. Investigating this parameter can provide insight into the local mechanics of the wheeling process, notably how far from the position of wheeling is change in curvature still significant.

## D Model results

### D.1 Overview of input data



(a) Gaussian curvature



(b) Mean curvature



(c) Maximum principle curvature

Figure 77: Comparison of image representations of different geometries for the 'centre-to-outside' # 1 tracking strategy for all four passes

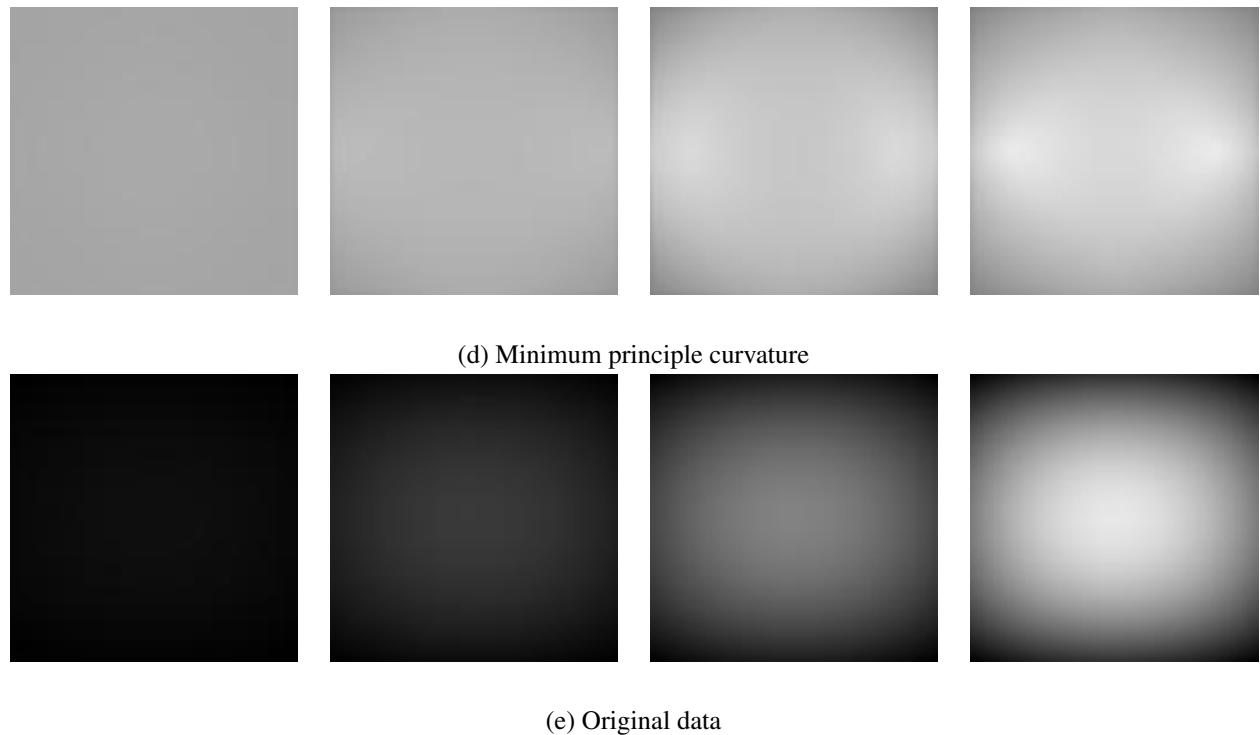


Figure 77: Comparison of image representations of different geometries for the 'centre-to-outside' # 1 tracking strategy for all four passes

## D.2 Unit testing

### D.2.1 First layer

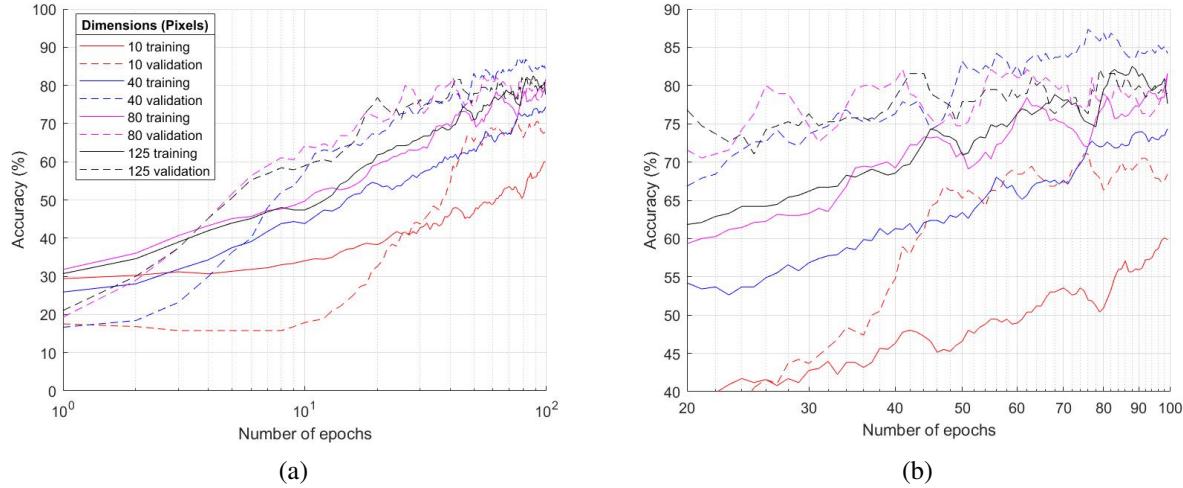


Figure 78: (a) Assessment of the training and validation accuracy of the base-case model using Gaussian curvature as the measure of geometry. This is across 100 epochs with varying image resolutions. (b) Zoomed in view of Figure (a)

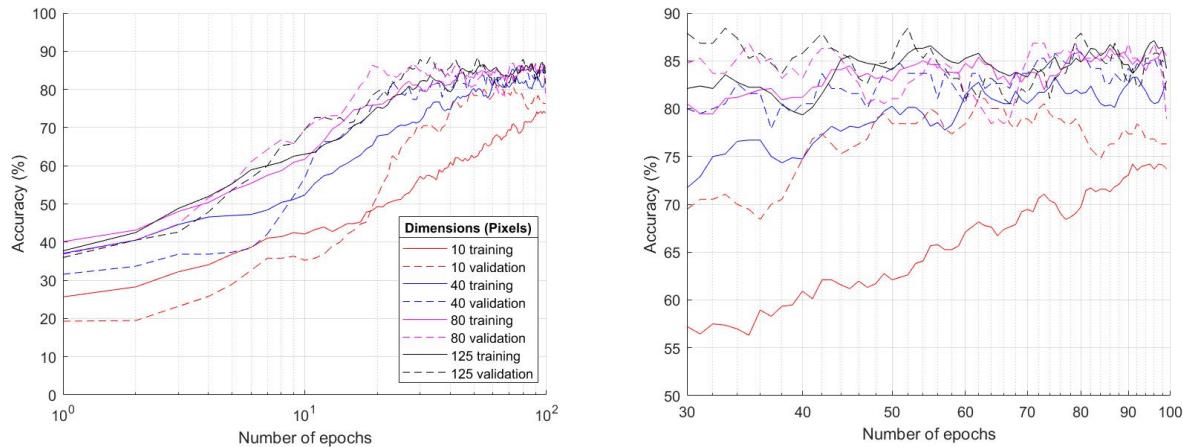


Figure 79: (a) Assessment of the training and validation accuracy of the base-case model using mean curvature as the measure of geometry. This is across 100 epochs with varying image resolutions. (b) Zoomed in view of Figure (a)

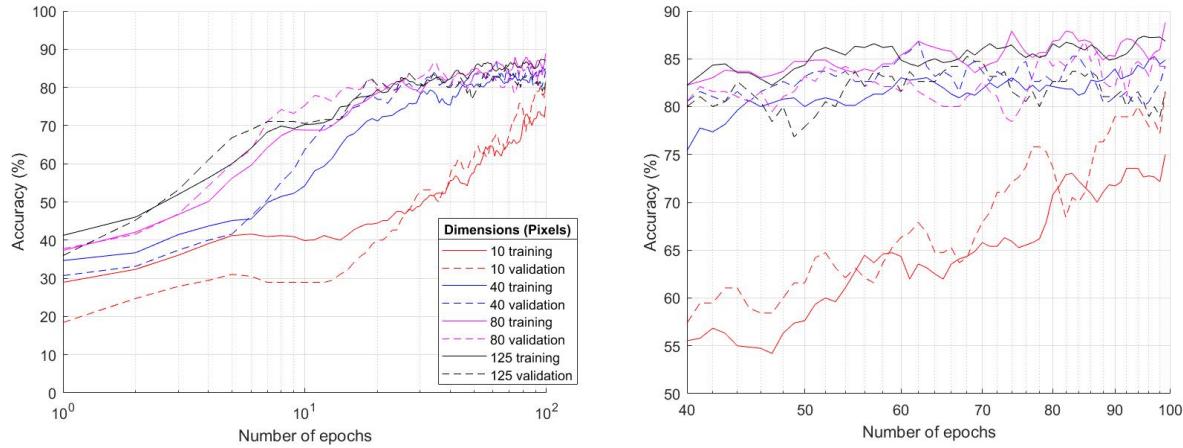


Figure 80: (a) Assessment of the training and validation accuracy of the base-case model using maximum principle curvature as the measure of geometry. This is across 100 epochs with varying image resolutions. (b) Zoomed in view of Figure (a)

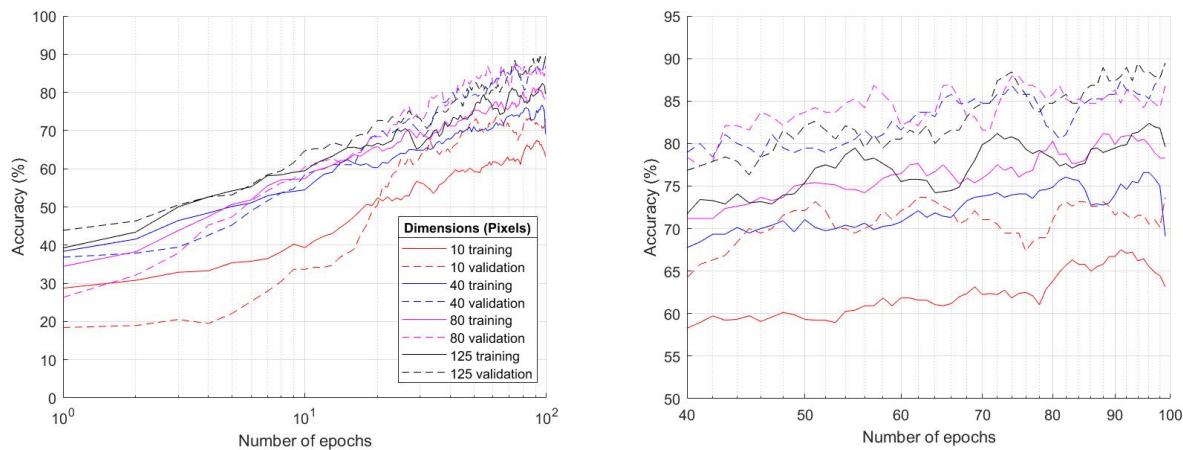


Figure 81: (a) Assessment of the training and validation accuracy of the base-case model using minimum principle curvature as the measure of geometry. This is across 100 epochs with varying image resolutions. (b) Zoomed in view of Figure (a)

### D.2.2 Second layer

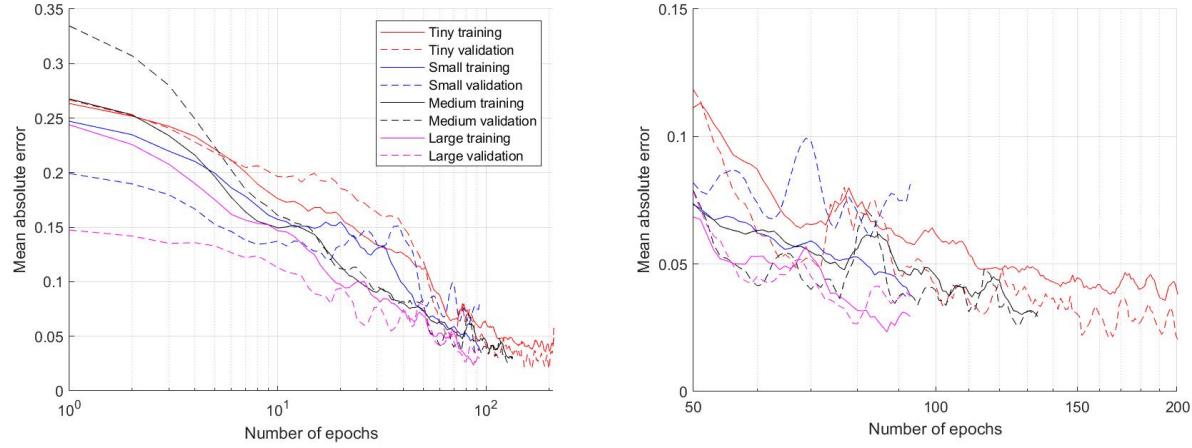


Figure 82: (a) Assessment of the training and validation MAE for the model predicting the minimum width parameter for the ‘centre-to-outside’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

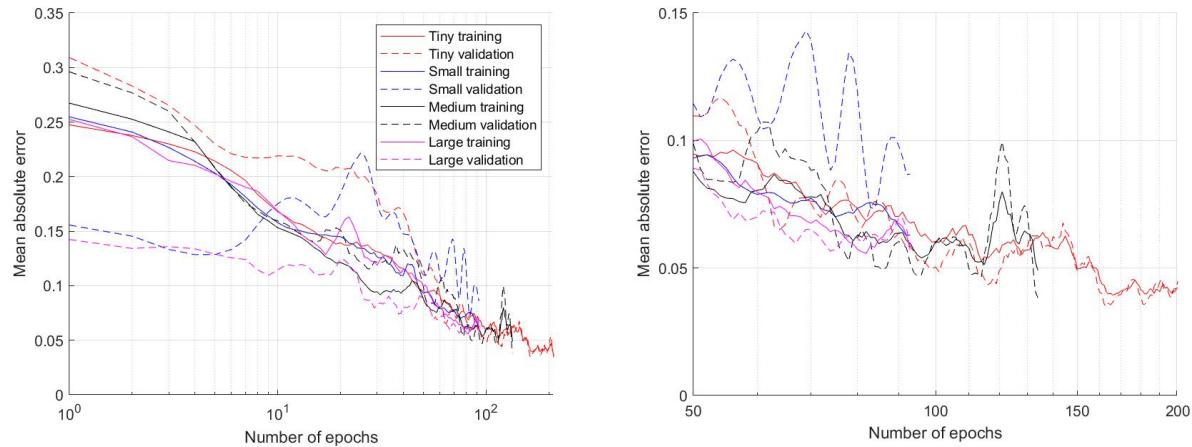


Figure 83: (a) Assessment of the training and validation MAE for the model predicting the minimum height parameter for the ‘centre-to-outside’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

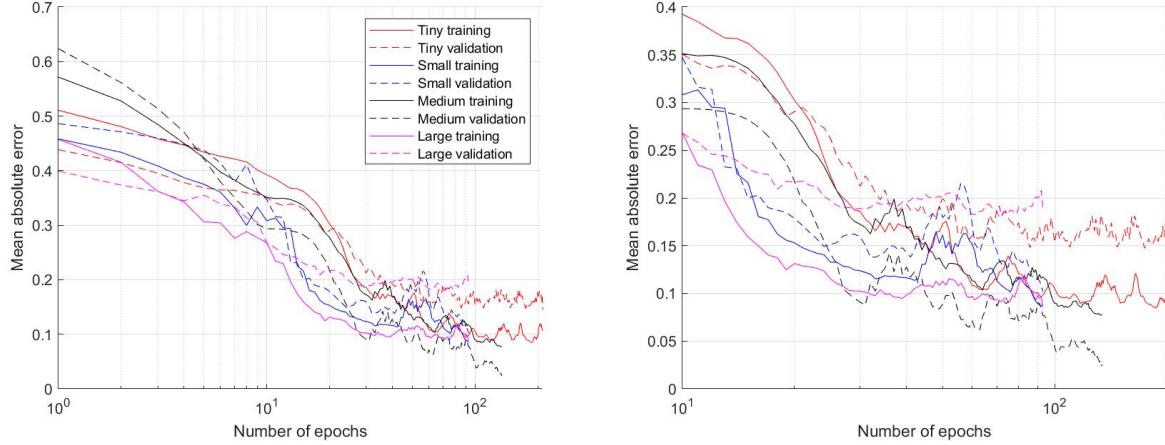


Figure 84: (a) Assessment of the training and validation MAE for the model predicting the central x position parameter for the ‘centre-to-outside’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

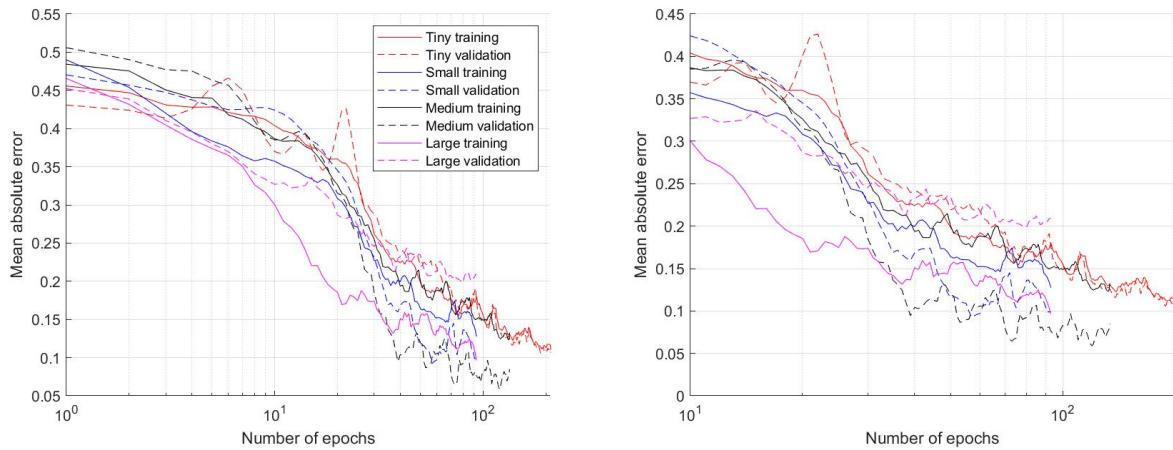


Figure 85: (a) Assessment of the training and validation MAE for the model predicting the central y position parameter for the ‘centre-to-outside’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

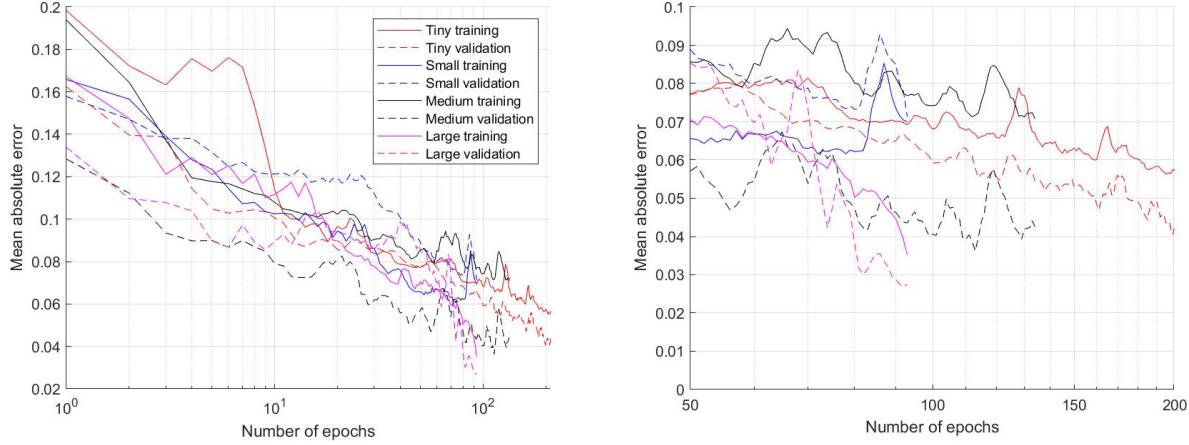


Figure 86: (a) Assessment of the training and validation MAE for the model predicting the scaling factor parameter for the ‘centre-to-outside’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

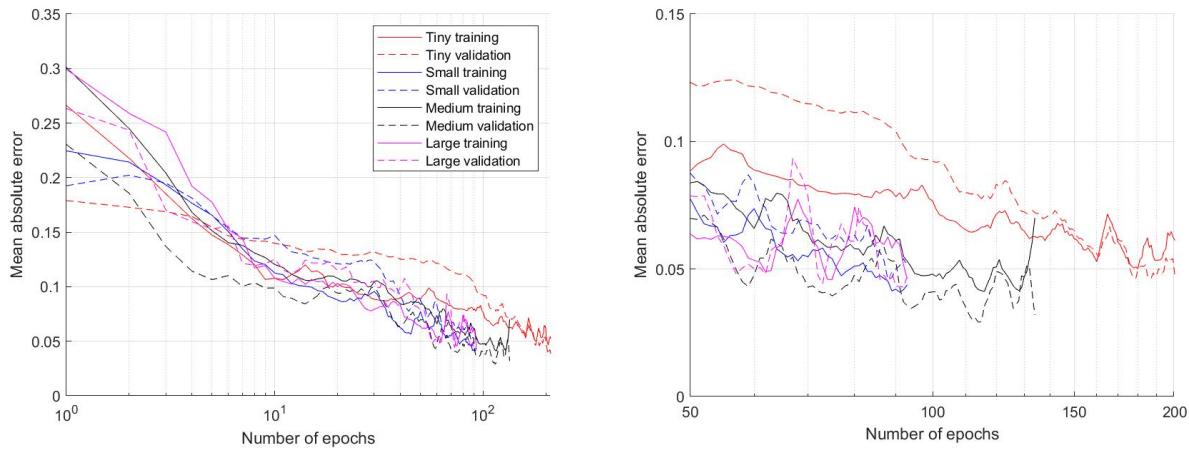


Figure 87: (a) Assessment of the training and validation MAE for the model predicting the number of passes parameter for the ‘centre-to-outside’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

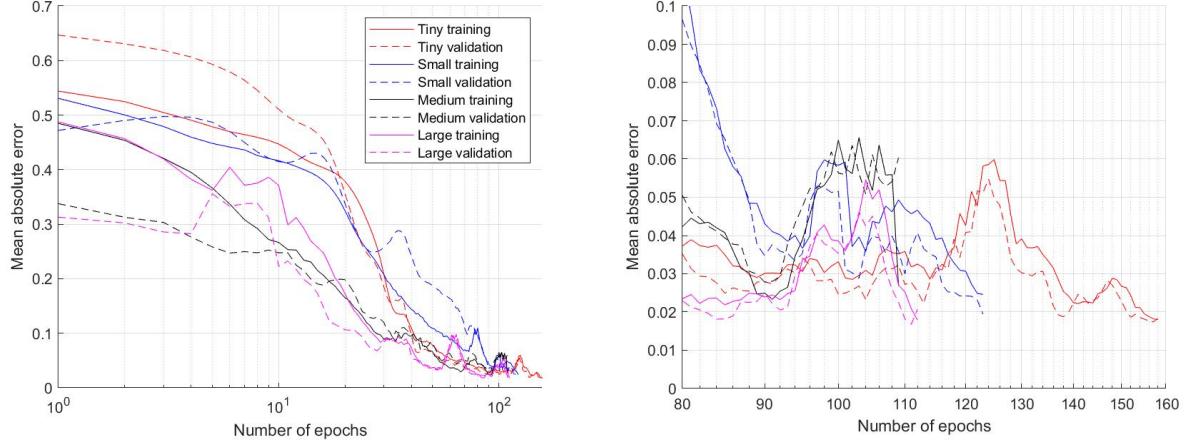


Figure 88: (a) Assessment of the training and validation MAE for the model predicting the width parameter for the ‘overlaid’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

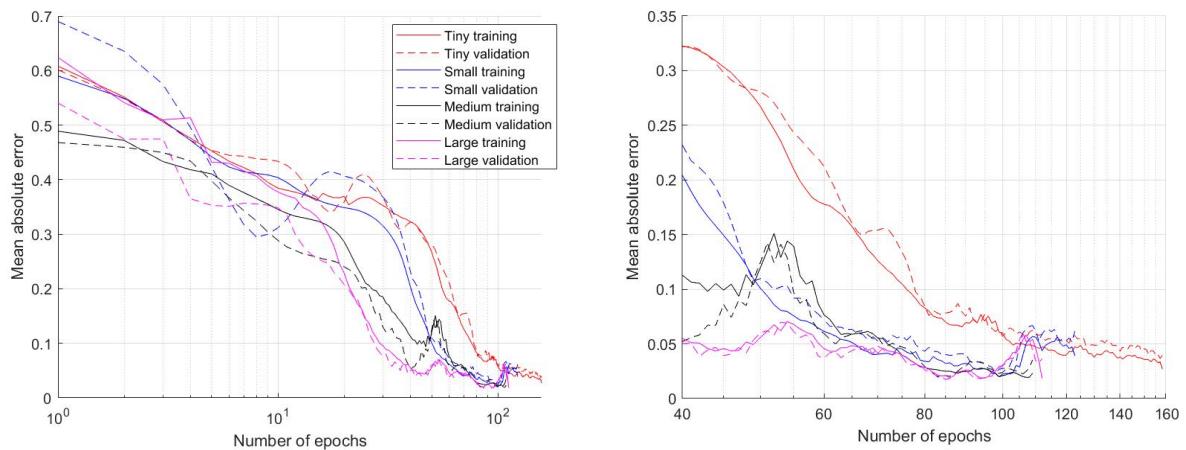


Figure 89: (a) Assessment of the training and validation MAE for the model predicting the height parameter for the ‘overlaid’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

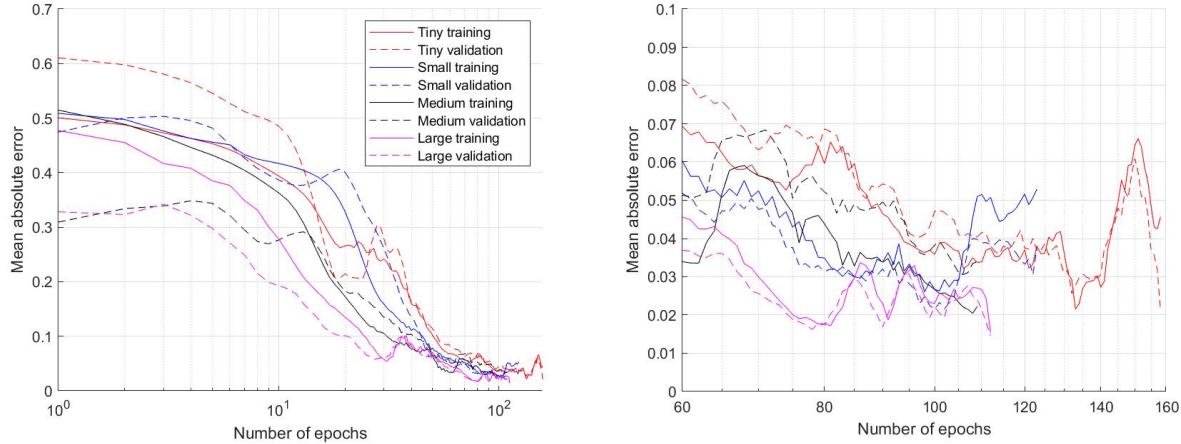


Figure 90: (a) Assessment of the training and validation MAE for the model predicting the central x position parameter for the ‘overlaid’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

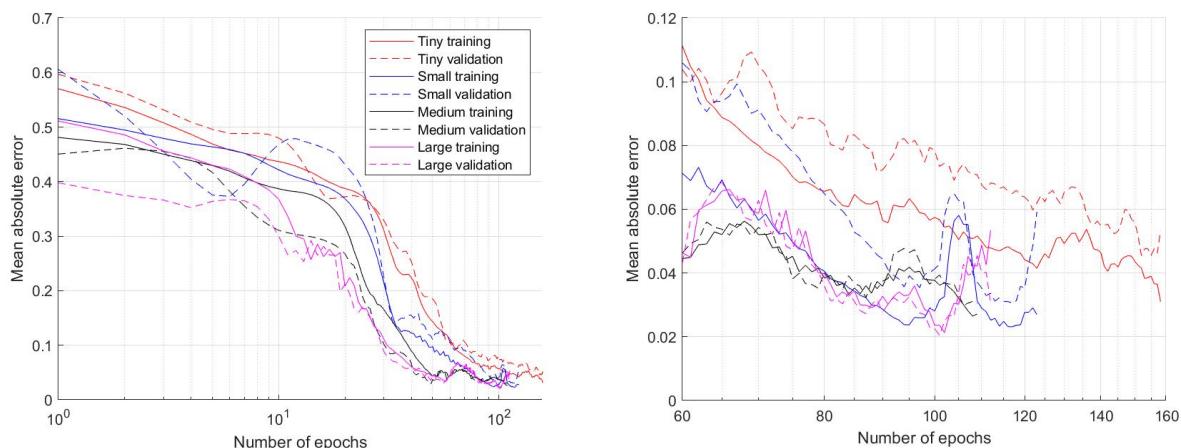


Figure 91: (a) Assessment of the training and validation MAE for the model predicting the central y position parameter for the ‘overlaid’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

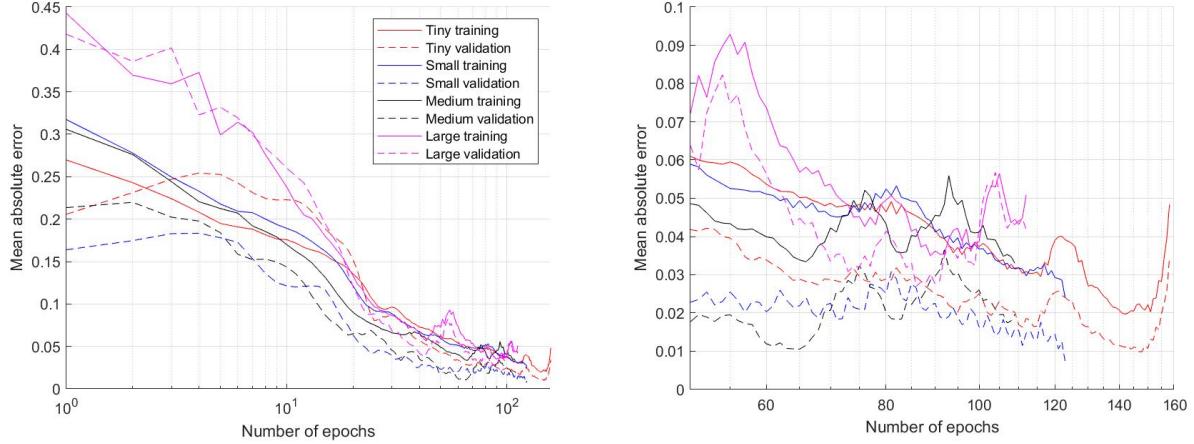


Figure 92: (a) Assessment of the training and validation MAE for the model predicting the number of passes parameter for the ‘overlaid’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

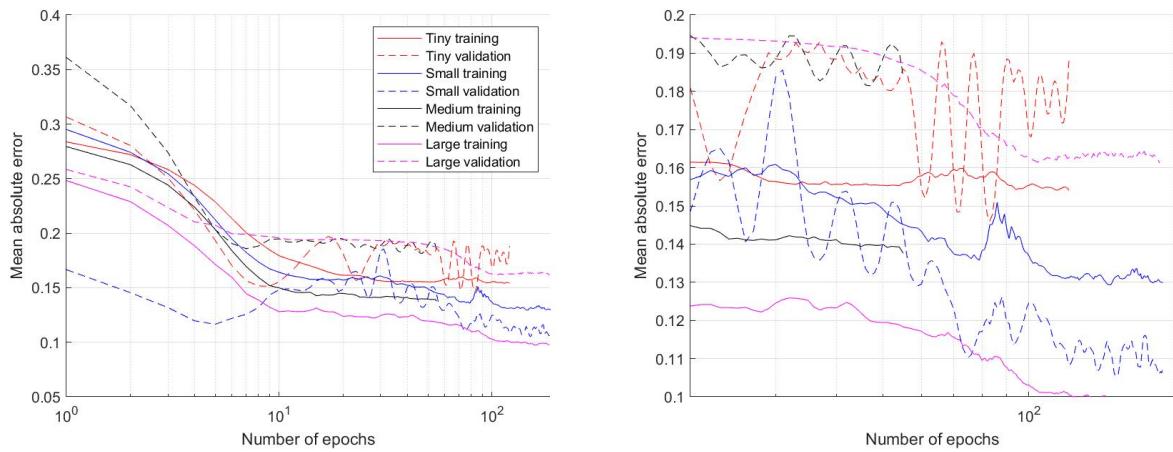


Figure 93: (a) Assessment of the training and validation MAE for the model predicting the minimum width parameter for the ‘vertical expansion’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

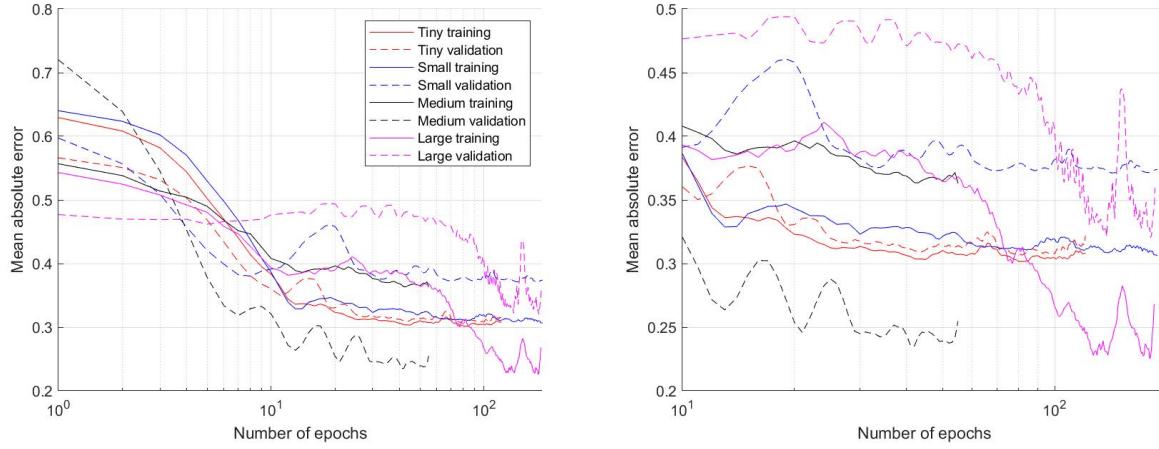


Figure 94: (a) Assessment of the training and validation MAE for the model predicting the central x position parameter for the ‘vertical expansion’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

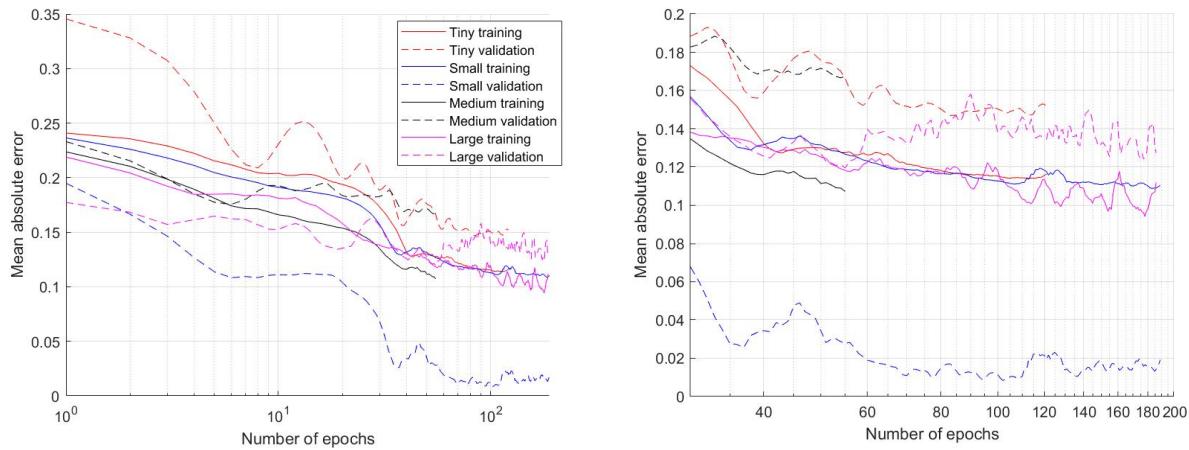


Figure 95: (a) Assessment of the training and validation MAE for the model predicting the scaling factor parameter for the ‘vertical expansion’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

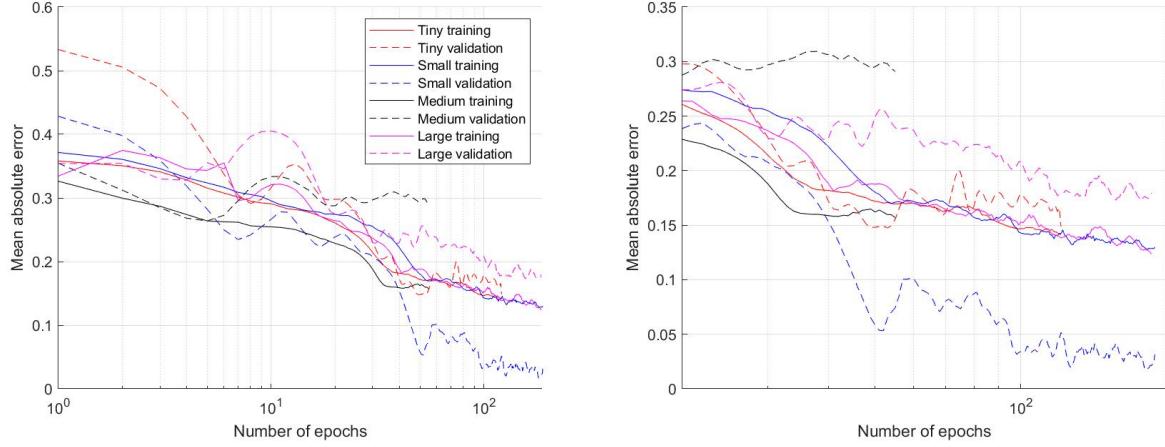


Figure 96: (a) Assessment of the training and validation MAE for the model predicting the number of passes parameter for the ‘vertical expansion’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

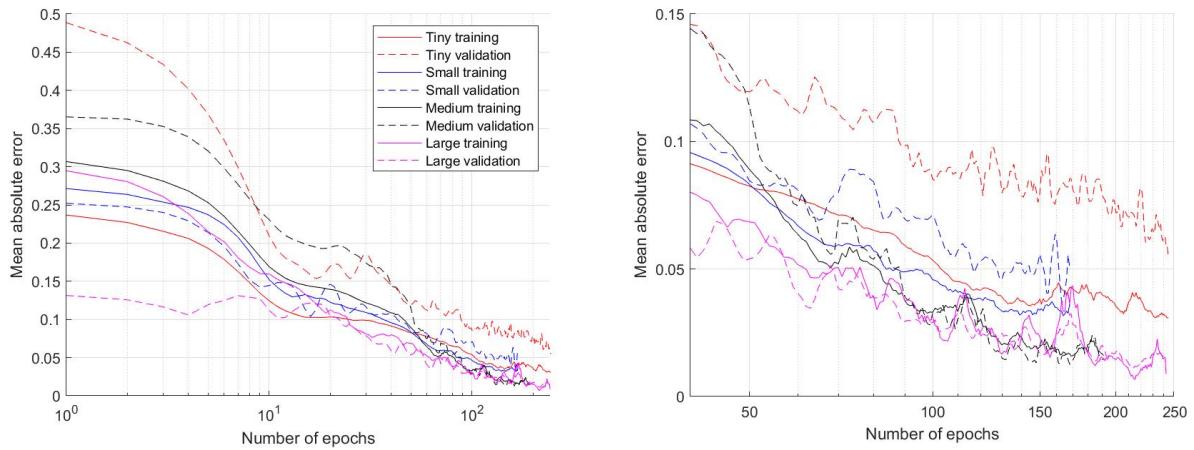


Figure 97: (a) Assessment of the training and validation MAE for the model predicting the minimum height parameter for the ‘horizontal expansion’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

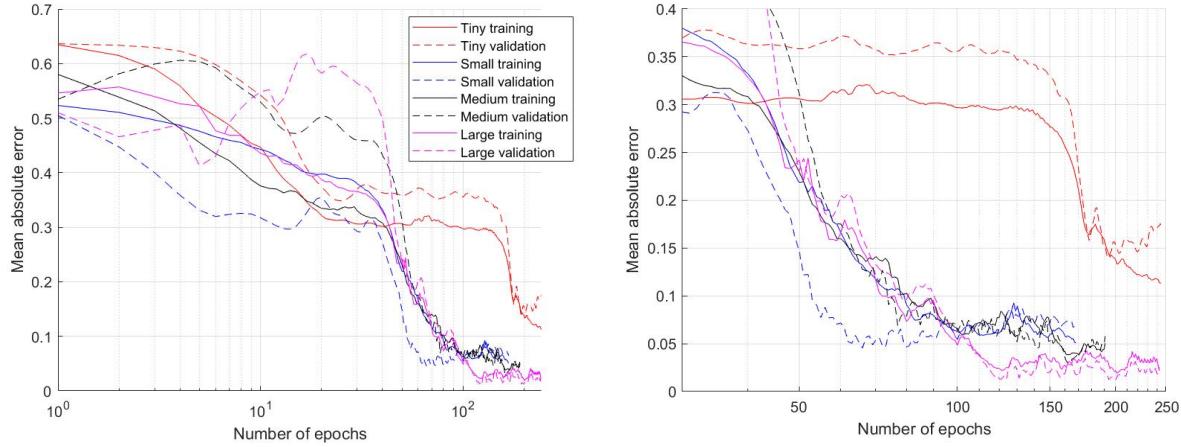


Figure 98: (a) Assessment of the training and validation MAE for the model predicting the central y position parameter for the ‘horizontal expansion’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

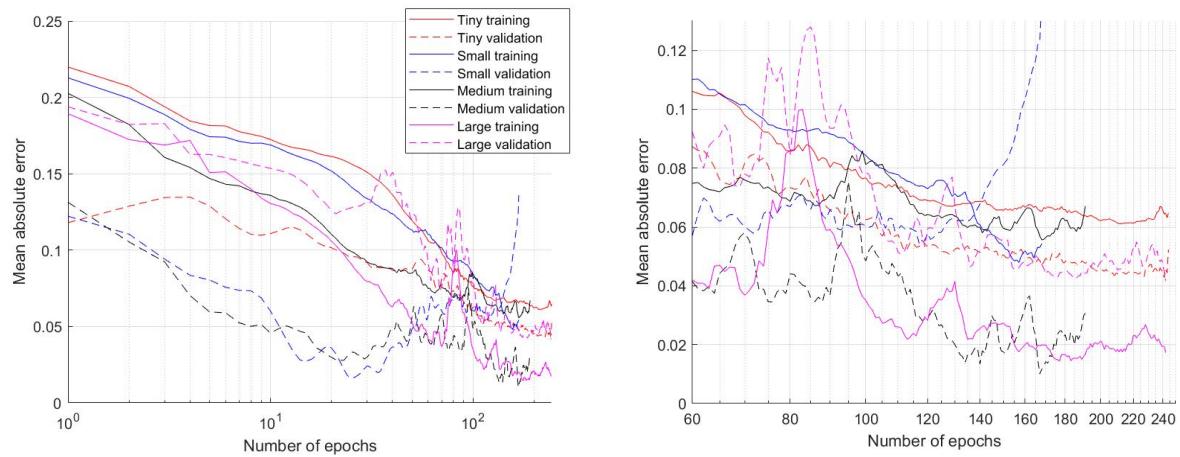


Figure 99: (a) Assessment of the training and validation MAE for the model predicting the scaling factor parameter for the ‘horizontal expansion’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

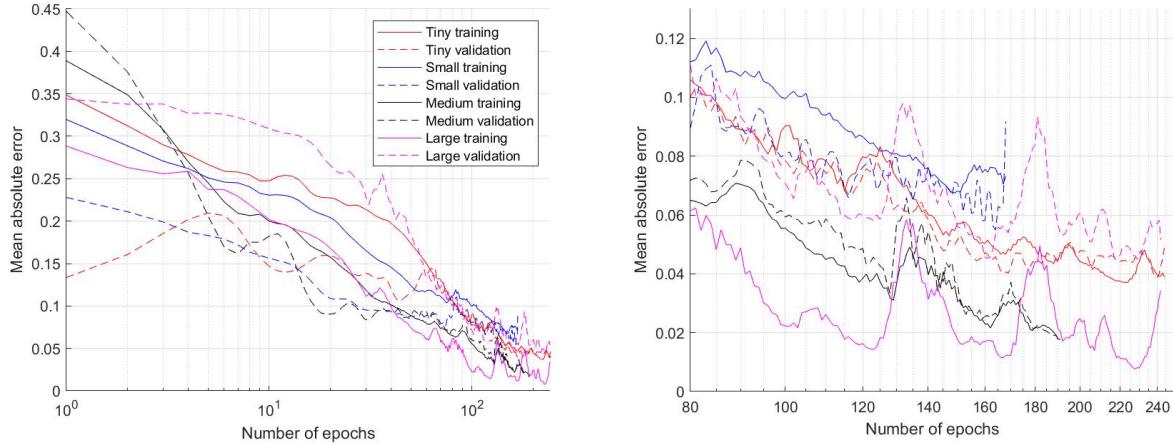


Figure 100: (a) Assessment of the training and validation MAE for the model predicting the number of passes parameter for the ‘horizontal expansion’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

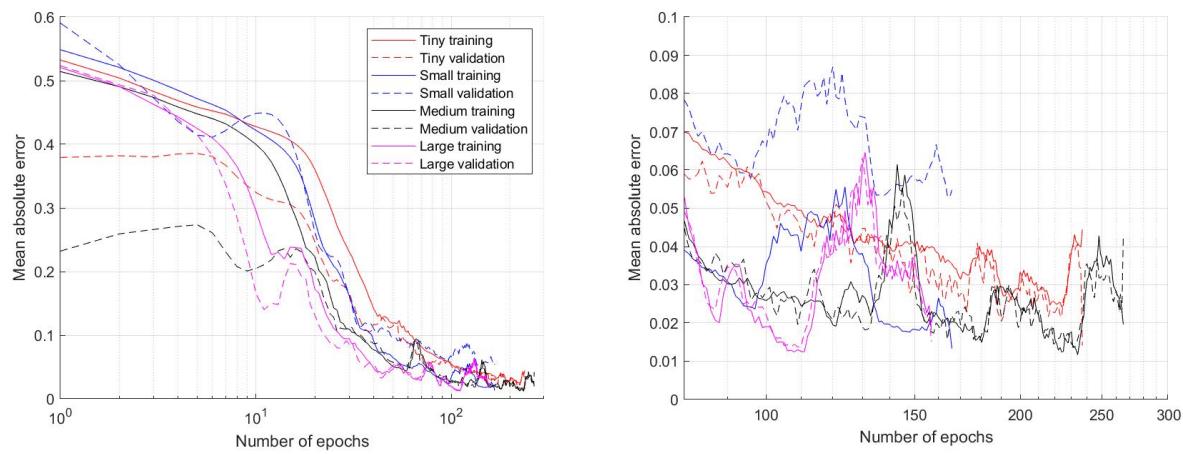


Figure 101: (a) Assessment of the training and validation MAE for the model predicting the base width parameter for the ‘triangular’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

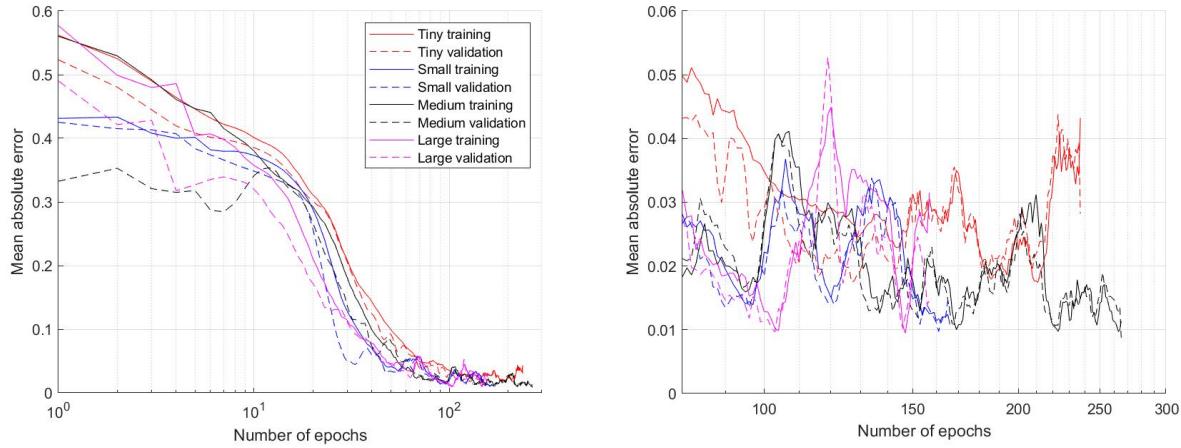


Figure 102: (a) Assessment of the training and validation MAE for the model predicting the height parameter for the ‘triangular’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

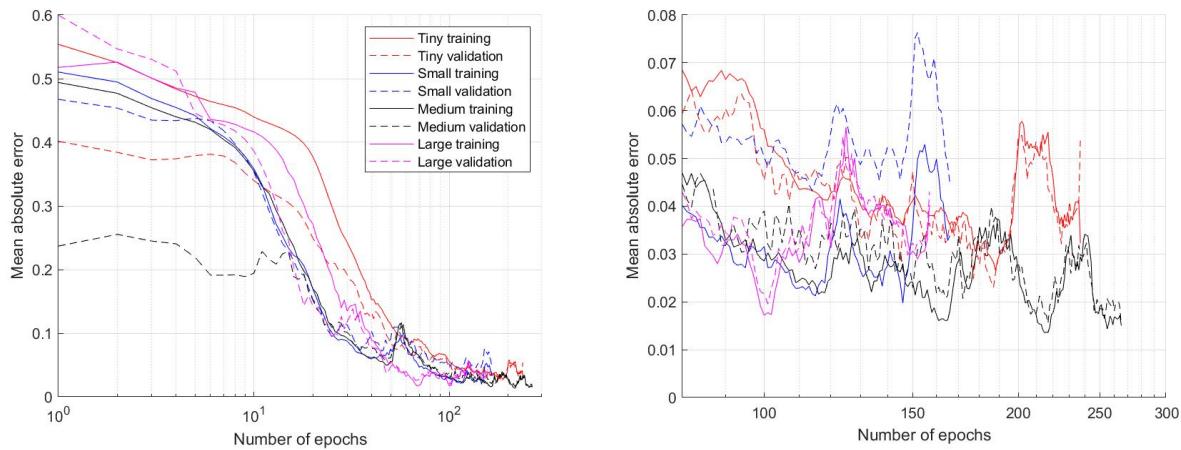


Figure 103: (a) Assessment of the training and validation MAE for the model predicting the base centre x position parameter for the ‘triangular’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

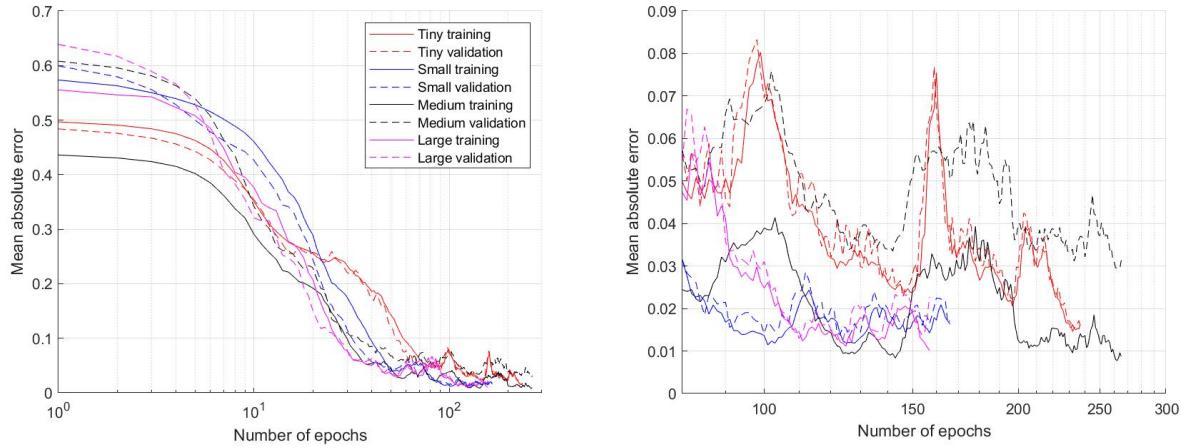


Figure 104: (a) Assessment of the training and validation MAE for the model predicting the base centre y position parameter for the ‘triangular’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

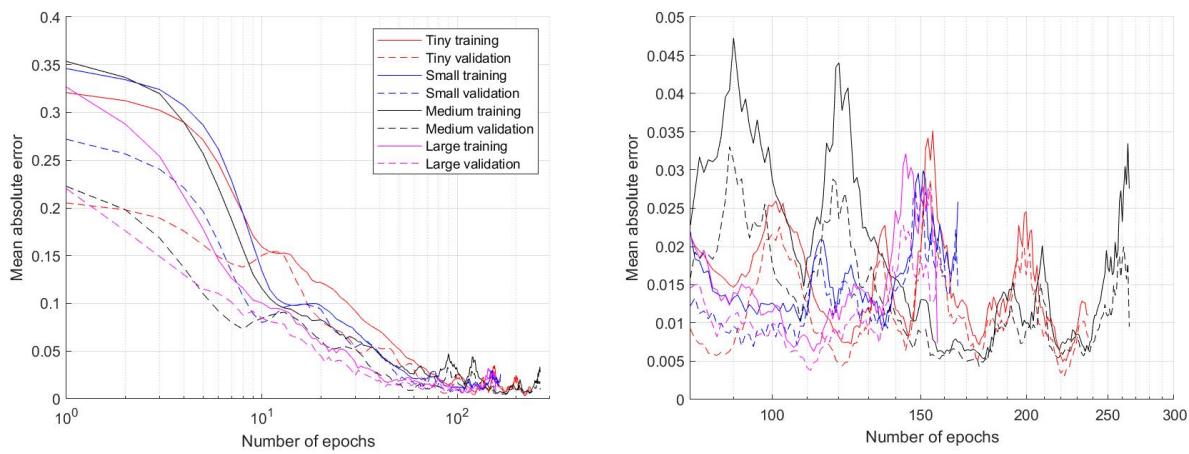


Figure 105: (a) Assessment of the training and validation MAE for the model predicting the start pass parameter for the ‘triangular’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

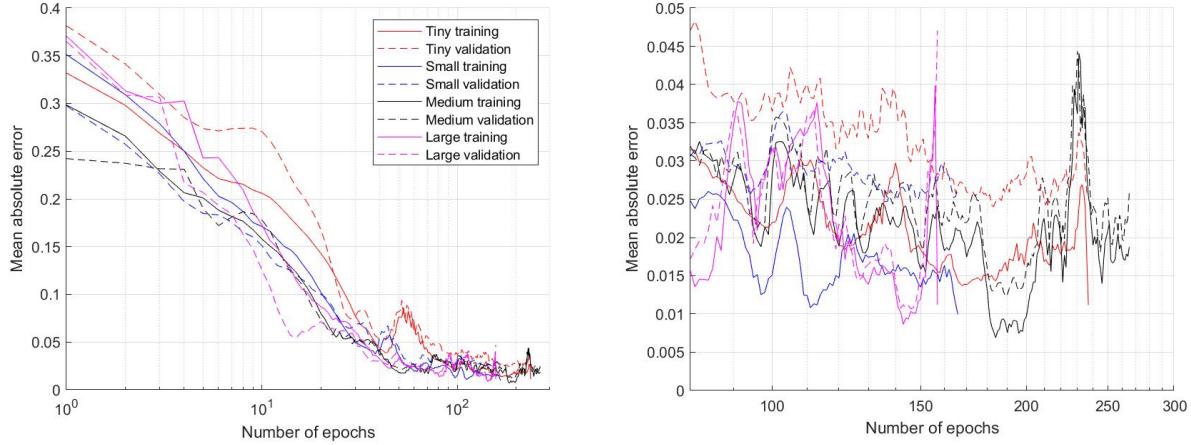


Figure 106: (a) Assessment of the training and validation MAE for the model predicting the number of passes parameter for the ‘triangular’ tracking strategy for varying neural network sizes. This is for the base-case model using maximum principle curvature as the measure of geometry. (b) Zoomed in view of Figure (a)

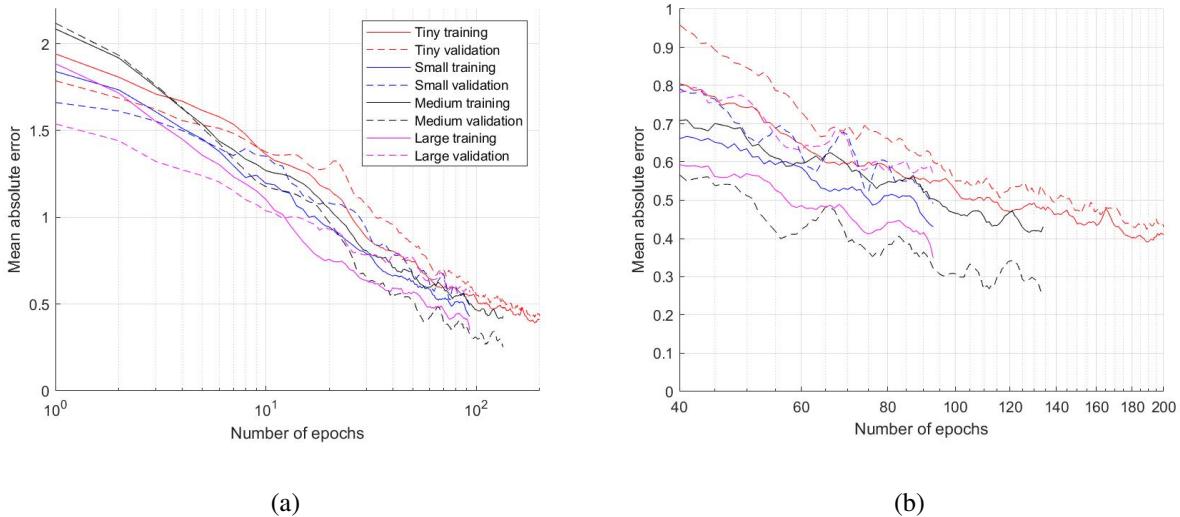


Figure 107: (a) Sum of MAE across all parameters for the ‘centre-to-outside’ tracking strategy for varying neural network size. This is with  $P_1$  used as input data, and all passes used in the dataset. (b) Zoomed in view of Figure (a)

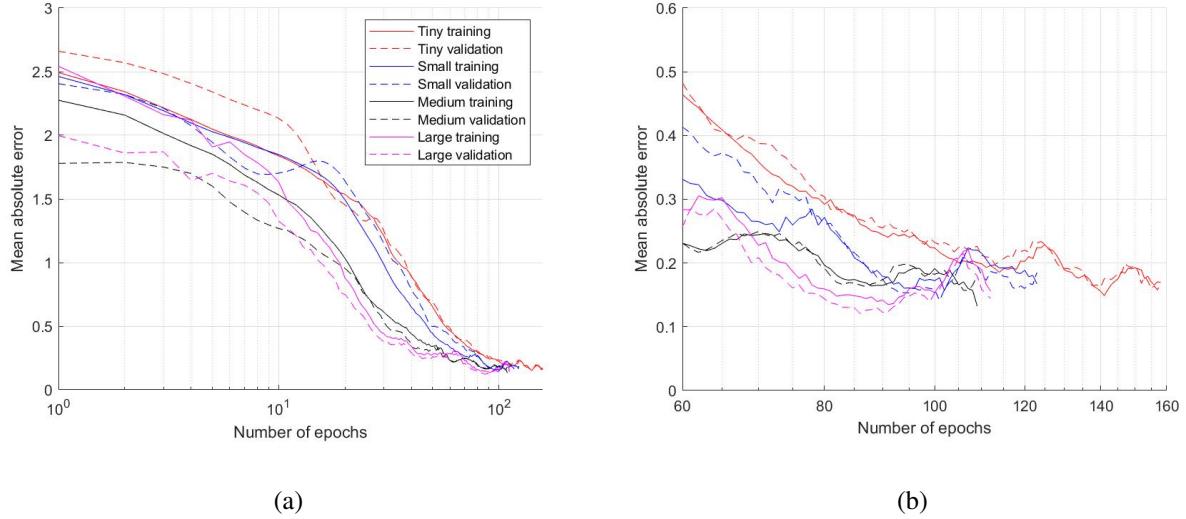


Figure 108: (a) Sum of MAE across all parameters for the ‘overlaid’ tracking strategy for varying neural network size. This is with  $P_1$  used as input data, and all passes used in the dataset. (b) Zoomed in view of Figure (a)

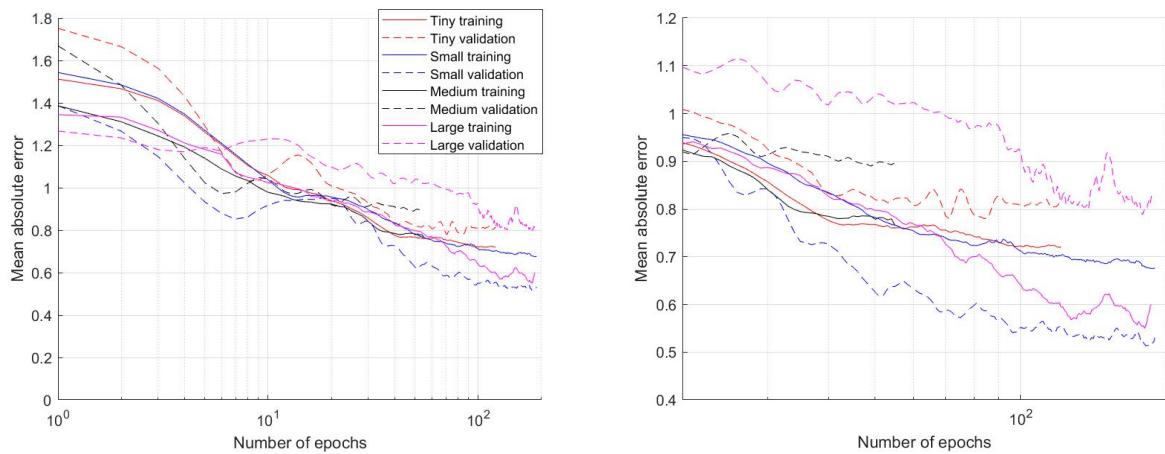


Figure 109: (a) Sum of MAE across all parameters for the ‘vertical expansion’ tracking strategy for varying neural network size. This is with  $P_1$  used as input data, and all passes used in the dataset. (b) Zoomed in view of Figure (a)

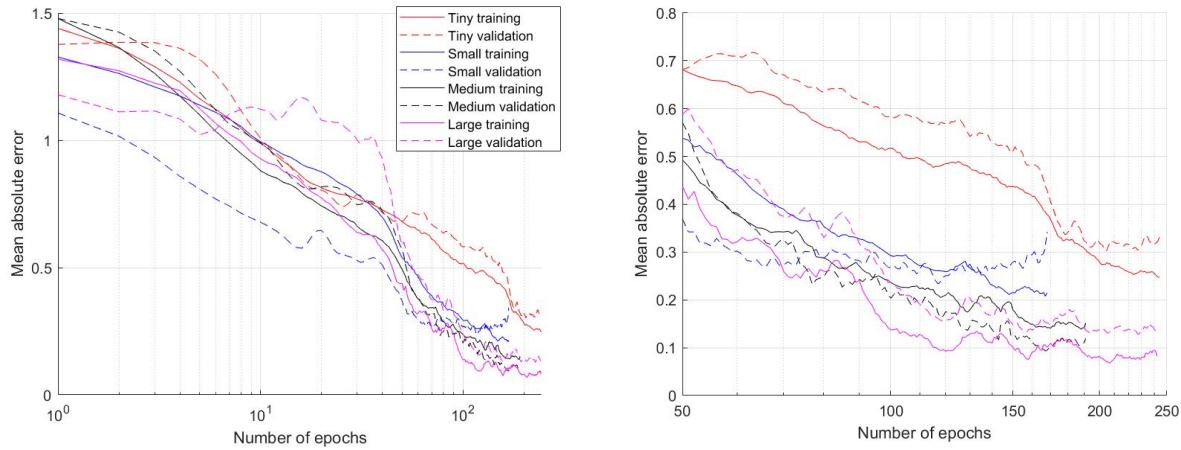


Figure 110: (a) Sum of MAE across all parameters for the ‘horizontal expansion’ tracking strategy for varying neural network size. This is with  $P_1$  used as input data, and all passes used in the dataset. (b) Zoomed in view of Figure (a)

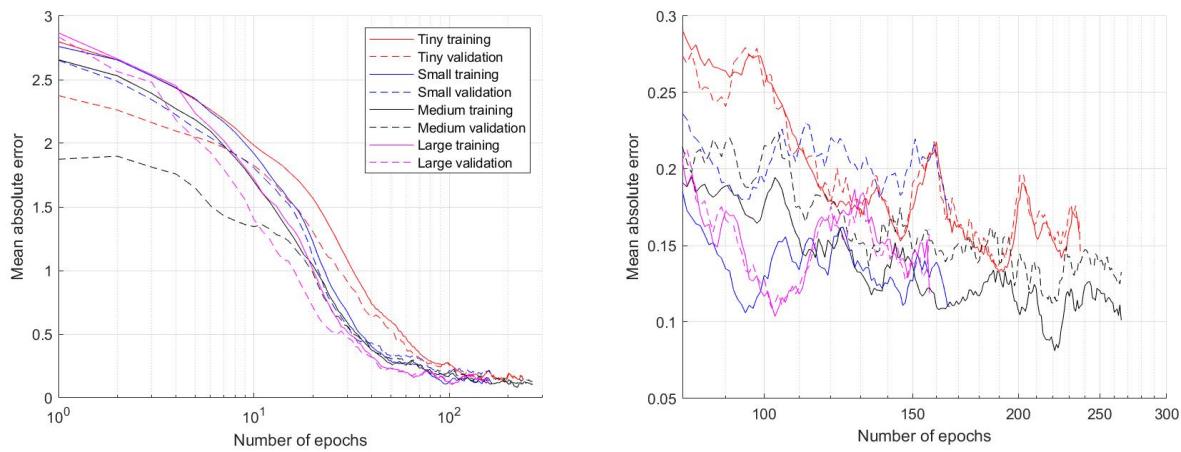


Figure 111: (a) Sum of MAE across all parameters for the ‘triangular’ tracking strategy for varying neural network size. This is with  $P_1$  used as input data, and all passes used in the dataset. (b) Zoomed in view of Figure (a)